

Mobility-Assisted, Energy-Efficient Algorithms for Distributed Inference in Wireless Sensor Networks

by

Thakshila Wimalajeewa Wewelwala-Hewage

B.Sc., University of Moratuwa, Sri Lanka, 2004

M.S., University of New Mexico, 2007

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
in
Electrical and Computer Engineering

The University of New Mexico

Albuquerque, New Mexico

October, 2009

©2009, Thakshila Wimalajeewa Wewelwala-Hewage

Dedication

To my parents, husband and sisters

*“Just as treasures are uncovered from the earth, so virtue appears from good deeds,
and wisdom appears from a pure and peaceful mind. To walk safely through the
maze of human life, one needs the light of wisdom and the guidance of virtue” –*

Lord Buddha

Acknowledgments

I would like to convey my heartfelt gratitude to my academic advisor, Professor Sudharman Jayaweera, for giving me the privilege of working in Information Sciences and Systems Lab at UNM. I would like to thank him for all his support throughout the course of graduate studies.

I am also thankful to Dr. Chaouki T. Abdallah, Dr. Rafael Fierro, Dr. Cristina Pereyra and Dr. Amy Galbraith to agreeing to serve as members in the dissertation committee.

I would also like to thank my friends at Information Sciences and Systems Lab for making a pleasant environment while working.

I take this opportunity to thank Administrative staff at ECE for their friendly support all the time for all the administrative work. My thanks go out to computer support group for their help in well maintaining machines and quick responses on installing necessary software.

Most importantly, this work would not be possible without love and the patience of my family. I am grateful to my loving parents W. H. Wimalajeewa and Karuna Wimalajeewa who have been a constant source of love and encouragement throughout. I would like to thank my loving husband Lalith, for his love, concern, understanding, and encouraging me to overcome the difficulties in life. I owe him a lot for giving a meaning to my life. I also would like to thank my loving sisters, Anuradha, Anushka, Shashikala and Priyanka for standing by my side always.

Mobility-Assisted, Energy-Efficient Algorithms for Distributed Inference in Wireless Sensor Networks

by

Thakshila Wimalajeewa Wewelwala-Hewage

ABSTRACT OF DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

The University of New Mexico

Albuquerque, New Mexico

November, 2009

Mobility-Assisted, Energy-Efficient Algorithms for Distributed Inference in Wireless Sensor Networks

by

Thakshila Wimalajeewa Wewelwala-Hewage

B.Sc., University of Moratuwa, Sri Lanka, 2004

M.S., University of New Mexico, 2007

PhD, Electrical and Computer Engineering, University of New Mexico, 2009

Abstract

Wireless sensor networks (WSNs) technology has been identified as one of the key innovations for the 21st century. WSNs consist of many up to thousands of small inexpensive lightweight distributed sensors which are capable of sensing and monitoring the physical world. With the recent developments of mobile sensor nodes either as mobile robots or unmanned autonomous vehicles, adding mobility to provide dynamic on-demand performance is becoming attractive in sensor networks. WSNs have unique advantages over other existing networks which make them suitable for many real world applications, due to certain factors such as ad hoc deployment, cost effectiveness and self-organization. However, efficient usage of WSNs in many applications is impeded by a plethora of challenges due to several limiting factors such

as finite node energy, low processing capabilities at nodes, and unreliable and imperfect wireless communication channels. In this dissertation, we investigate major problems in a resource limited wireless sensor network for detection and estimation, in signal processing and communication perspectives. Specific problems addressed in the dissertation can be categorized as follows:

1. Power management under correlated observations
2. Distributed node scheduling for sequential inference
3. Impact of node mobility on detection and dynamic state estimation in mobility assisted sensor networks

The dissertation is started by focusing on efficient power management in distributed detection and estimation with correlated local observations. In most potential applications, once deployed WSNs need to work unattended for a long time. Since sensor nodes are equipped with finite energy supplies, power management is considered to be a core issue in designing a WSN. Also, in a practical densely deployed sensor network, it is more likely that the observation noise is dependent across sensor nodes. The performance analysis under correlated observations is analytically complex compared to that with independent observations. We explore optimal power scheduling techniques with correlated observations under different practical correlation models.

Among all possible power consumption modes, the transmission power is the most dominant in a sensor node. A large communication power is consumed in a traditional centralized data/decision fusion since nodes have to communicate with a central fusion center. On the other hand, the reliability of centralized data fusion architecture depends on the robustness of the fusion center for failure. Hence, distributive approaches for data fusion are desirable in power constrained sensor networks. In the

second part of the dissertation, a distributed sequential methodology is proposed for parameter estimation in a stationary sensor network in which nodes are assumed to exchange information locally via noisy communication channels. Distributed algorithms for node selection in the sequential estimation process were developed taking the trade-off between information gains and the communication costs into account.

As far as a static sensor network is considered, the performance is basically determined by its initial configuration. Even if a all-static network meets the performance quality requirement at the initial deployment stage, it does not adapt to unpredictable dynamics in the network conditions, such as coverage holes caused by node failures or changing dynamics of the phenomenon being sensed over time. Recently mobile sensor nodes are proposed to be deployed in wireless sensor network applications to provide dynamic on-demand performance. However, deploying a large number of mobile nodes in a sensor network is expensive due to energy consumption of mobile nodes compared to that of static sensor nodes. In this dissertation, the cost of deploying mobile nodes is investigated in terms of the required node density to achieve desired performance measures. Several important performance measures in target detection in a mobility assisted sensor network were derived assuming random and independent node mobility models. In situations where random node mobility models are inefficient, it is required to navigate mobile nodes purposefully to compensate for the lack of performance resulted in a all-static network. To that end, a new interactive distributed protocol, collaborating among static and mobile nodes, for node mobility management is proposed to improve the coverage over time in an efficient manner. The worst case performance in detecting mobile targets is derived in terms of the *exposure*.

Contrast to continuous movements for target detection, due to energy constraints, it may be desired to keep mobile nodes stationary until a target is detected with a certain confidence level, or useful statistics regarding the target locations are available

under stationary configuration. We develop two decision fusion architectures for a hybrid sensor network when nodes are allowed to move only if necessary depending on the application requirements. Moreover, the cost of allowing nodes to be mobile is investigated analytically in terms of the minimum fraction of mobile nodes required to achieve desired performance gains under certain constraints.

Estimation of the state of a moving target is another important application of wireless sensor networks. In the last part of the dissertation, we address the problem of non-linear/non-Gaussian dynamic state estimation by hybrid sensor networks. The node mobility is exploited to improve the tracking quality dynamically as target moves.

Contents

List of Figures	xxi
List of Tables	xxix
Abbreviations	xxx
1 Overview of the Dissertation	1
1.1 Introduction	1
1.2 Main Contributions	2
1.2.1 Optimal power scheduling for distributed detection and estimation in stationary sensor networks with correlated observations:	2
1.2.2 Distributed node selection schemes for sequential estimation over noisy communication channels in stationary sensor networks:	3
1.2.3 Trade-off between mobile node density and detection performance measures in hybrid sensor networks with random mobility models:	4

1.2.4	Distributed protocol for mobile node navigation in a hybrid sensor network:	5
1.2.5	Worst-case detection performance in hybrid sensor networks:	6
1.2.6	Decision fusion models for hybrid sensor networks to detect randomly located targets:	6
1.2.7	Distributed tracking algorithm using particle filters with reactive mobility:	7
1.3	Dissertation Outline	7
2	Wireless Sensor Networks	9
2.1	Introduction	9
2.2	Static and Mobile Sensors	10
2.2.1	Static sensors	10
2.2.2	Mobile sensors	11
2.3	Data/Decision Fusion Architectures	12
2.3.1	Centralized data/decision fusion	12
2.3.2	Distributed data/decision fusion	12
2.4	Applications	13
2.5	Challenges	16
2.6	Signal Processing Aspects and Research Directions	17

3	Power Management for Detection with Correlated Observations	20
3.1	Introduction	20
3.2	Data Fusion Problem Formulation	23
3.3	Analysis of Optimal Fusion Performance	25
3.3.1	Independent local observations	26
3.3.2	Correlated local observations	27
3.4	Optimal Power Allocation with Independent Observations	29
3.5	Optimal Power Allocation with Correlated Observations	30
3.5.1	Penalty function approach for constrained optimization	31
3.5.2	Particle swarm optimization	32
3.5.3	Selection of parameter values for PSO	36
3.5.4	Power allocation based on the derived fusion error probability bound with correlated observations	37
3.6	Performance Results	38
3.7	Conclusions	50
3.8	Appendix 3A	52
4	Power Management for Estimation with Correlated Observations	54
4.1	Introduction	54
4.2	Sensor Network Model	56

4.3	MSE Performance with Non-Orthogonal Communication	57
4.4	Asymptotic MSE Performance with Non-Orthogonal Communication	58
4.4.1	$\rho=0$: Orthogonal communication	60
4.4.2	$\rho=1$: Perfect correlation between codes	60
4.5	Optimal Power Allocation in Fading Channels	61
4.5.1	Orthogonal communication and i.i.d observations	62
4.5.2	MAC and i.i.d. observations	63
4.5.3	MAC and correlated observations	66
4.6	Synchronization in Sensor Transmissions and the Effect of Synchro- nization Error on MSE Performance	68
4.7	Conclusions	71
5	Distributed Node Selection for Noisy Sequential Estimation	73
5.1	Introduction	73
5.2	Sensor Network Model	76
5.3	MMSE Performance	77
5.4	Sensor Node Selection	82
5.4.1	Optimal scheme	84
5.4.2	Distributed node selection: global approach	86

5.4.3	Distributed node selection: local approach	89
5.5	Performance Analysis	93
5.5.1	Exact target location is known at each node	93
5.5.2	Statistics of the target location is known at each node	98
5.6	Conclusions	100
6	Impact of Mobile Node Density on Detection Performance	102
6.1	Introduction	102
6.2	System Model	109
6.2.1	Target model	109
6.2.2	Node mobility models	109
6.2.3	Detection model	110
6.2.4	Preliminaries	111
6.3	Stationary Target Detection Performance with Mobility Model 1	112
6.3.1	Detection probability	112
6.3.2	Mean first contact distance for single-sensing detection	117
6.3.3	Detection latency	119
6.4	Detection Performance with Random Node Mobility Model 2 (Random Walk)	122
6.5	Probability of Node Connectivity	125

6.6	Simulation Results	128
6.6.1	With node mobility model 1	128
6.6.2	With node mobility model 2	134
6.6.3	Node connectivity	136
6.7	Conclusions	138
7	Interactive Distributive Mobility Protocol for Mobile Node Navigation	140
7.1	Introduction	140
7.2	Motivation and System Model	143
7.2.1	Motivation	143
7.2.2	Network model	144
7.3	Interactive, Distributed Mobility Protocol	144
7.3.1	Distributed protocol for node mobility management	146
7.3.2	Updating $\mathbf{g}_{s_k}(nT_m)$	149
7.3.3	Duplicate covering at a given time	151
7.3.4	Compensating for the lack of coverage in a static cell	152
7.4	Performance Evaluation	154
7.4.1	Presence probability at each cell	155
7.4.2	Average unvisited time of an arbitrary point	159
7.5	Conclusions	160

8	Worst-Case Detection Performance	163
8.1	Introduction	163
8.2	Network Model	165
8.2.1	Target model	165
8.3	Worst Case Detection Performance	166
8.3.1	Detection probability	166
8.3.2	Evaluating worst-case target exposure	166
8.4	Performance Results	171
8.5	Conclusions	172
9	Decision Fusion with Measurement Uncertainty under Reactive Mo-	
	bility	174
9.1	Introduction	174
9.2	Problem Formulation and System Model	176
9.2.1	Problem formulation	177
9.2.2	Node mobility model	178
9.2.3	Observation model	178
9.3	Detection Performance with Decision Fusion Architecture 1: Fusion Center Updating Decisions Over Time	181
9.3.1	Detection performance at k -th mobile node	182
9.3.2	Detection performance at k -th static node	186

9.3.3	Performance evaluation at fusion center with noisy communication	186
9.3.4	Mobility management for mobile nodes	188
9.4	Detection Performance with Decision Fusion Architecture 2: Nodes Updating Decisions Over Time	194
9.4.1	Detection performance at k -th mobile node	195
9.4.2	Detection performance at the k -th static node	197
9.4.3	Decision fusion performance with noisy communication	198
9.4.4	Mobility management for mobile nodes	199
9.5	Minimum Set of Mobile Nodes	200
9.6	Performance Results	203
9.6.1	Performance evaluation with decision fusion architecture 1	205
9.6.2	Performance evaluation with decision fusion architecture 2	208
9.7	Conclusions	212
9.8	Appendix 9A	213
9.9	Appendix 9B	215
10	Particle Filter Based Target Tracking with Reactive Mobility	218
10.1	Introduction	218
10.2	Problem Formulation and System Model	223

10.2.1	Sensor network model	223
10.2.2	State dynamics model	223
10.2.3	Observation model	224
10.2.4	Node mobility model	225
10.3	Cluster Based Target Tracking by Particle Filters	226
10.3.1	Centralized particle filter	226
10.3.2	Distributed cluster based target tracking for hybrid sensor network	229
10.4	Node mobility management	234
10.5	PCRLB Analysis	240
10.5.1	With raw observations	240
10.5.2	With binary observations	243
10.6	Performance Analysis	243
10.6.1	With raw observations	243
10.6.2	With binary observations	249
10.7	Conclusions	251
10.8	Appendix 10A	252
10.9	Appendix 10B	254
11	Summary of Results and Future Work	259
11.1	Summary of Results	259

Contents

11.2 Future Work	262
References	265

List of Figures

2.1	Sensing unit and transceiver/processor unit of a typical sensor node	11
2.2	Centralized data/decision fusion architecture with a fusion center . .	13
2.3	Distributed sequential data/decision fusion architecture	14
3.1	Total power Vs. probability of fusion error for independent observations: $\gamma_0 = 5dB$	38
3.2	Optimal power values of sensor nodes Vs. number of sensors for $n = 20$ and $n = 50$ when $\epsilon = 0.1$ and $\gamma_0 = 10dB$	39
3.3	Number of active sensors for independent observations.	40
3.4	Total power Vs. local SNR for independent observations	41
3.5	Comparison of the derived probability bound and the exact error probability for correlated observations; $\rho = 0.1$	43
3.6	Total power and the fusion error probability bound for correlated observations; $\gamma_0 = 5dB$, $n = 100$ and $\rho = 0.1$	44
3.7	Convergence of exterior penalty function based PSO: Best fitness returned for PSO iterations for a given penalty parameter. (Fusion error probability = 0.01)	45

3.8	Convergence of exterior penalty function based PSO: Convergence of penalty function to the original optimization problem. (Fusion error probability = 0.01)	46
3.9	PSO: Total power Vs. fusion error probability when observations are correlated; $n = 20, \rho = 0.1$	47
3.10	PSO: Total power Vs. fusion error probability when observations are correlated; $n = 20, \gamma_0 = 10dB$	48
3.11	PSO: Total power Vs. fusion error probability for different noise covariance models; in models 2 – 4, $\rho=0.1$	49
3.12	PSO based results Vs. results obtained assuming i.i.d. noise for noise models 1 and 3, $n = 20$ and $\gamma_0 = 10dB$	50
3.13	PSO: Total power Vs. fusion error probability for with the estimation error of the fading coefficients at the fusion center. $n = 20, \gamma_0 = 10dB, \rho = 0.1, \epsilon = 0.1$	51
4.1	MSE as a function of number of sensors n	61
4.2	The dependence of MSE on local observation correlation parameter ρ_d	62
4.3	Comparison of optimal power and uniform power for $\rho = 0$ and $\rho = 1$	65
4.4	The performance of the power allocation scheme based on MSE bound for $\rho = 1$ and correlated observations. Total power vs. MSE, $n = 20, \gamma_0 = 20dB$	67
4.5	The performance of the power allocation scheme based on MSE bound for $\rho = 1$ and correlated observations. Total power vs. number of sensor nodes, $\gamma_0 = 12dB, D_0 = 0.08$	68

4.6	The effect of the synchronization error for the MSE performance . . .	70
5.1	MMSE vs. number of sensors when observation noise is i.i.d.	80
5.2	MMSE vs number of sensors when observation noise is non-i.i.d. . . .	81
5.3	Illustration of the shortest path formulation of the optimization problem (5.9) for $n = 5$	83
5.4	Distributed sequential estimation process at node s_j for $j = 1, 2, \dots$	90
5.5	MMSE at the k -th processing node with the proposed node selection scheme based on local approach with both MMSE and mutual information as information utility measures: ($\sigma_c^2 = 0$, $\alpha' = 2$, $r_0 = 1$, $\sigma_0^2 = 0.1$).	95
5.6	MMSE at the k -th node with exact target location with MMSE as the information utility measure with no channel noise: $\sigma_{c(k-1,k)}^2 = \sigma_c^2 = 0$, $n = 40$, $\alpha' = 2$, $r_0 = 1$	96
5.7	MMSE at the k -th node with exact target location with MMSE as the information utility measure with channel noise: $\sigma_{c(k-1,k)}^2$'s are as given in (5.17), $n = 40$, $\sigma_c^2 = 0.001$, $\alpha' = 2$, $r_0 = 1$	97
5.8	MMSE at the k -th node when the statistics of target location are available with no channel noise: $\sigma_{c(k-1,k)}^2 = \sigma_c^2 = 0$, $\alpha' = 2$, $r_0 = 1$, $\sigma_0^2 = 0.1$	99
5.9	MMSE at the k -th node when the statistics of target location are available with channel noise: $\sigma_{c(k-1,k)}^2$'s are as given in (5.17), $\sigma_c^2 = 0.005$, $\alpha' = 2$, $r_0 = 1$, $\sigma_0^2 = 0.1$	100
6.1	Random mobility models of a mobile node	111

6.2	Hybrid network at times $t = 0$ and $t = nT_s$	113
6.3	Realization of random shapes at time nT_s	114
6.4	Minimum possible coverage area after completing 4 distinct steps, left: $r < \frac{\mu}{2}$, right: $\frac{\mu}{2} \leq r < \mu$	124
6.5	Detection probability with single-sensing detection Vs desired delay constraint with mobility model 1: $r = 20m$	129
6.6	Detection probability Vs fraction of mobile nodes in the network for single-sensing and 2-sensing detection models for mobility model 1; Desired detection delay is $t_D = 60s$	130
6.7	Minimum fraction of mobile nodes required to achieve a desired per- formance level within a desired delay constraint for mobility model 1	131
6.8	Mean value of the first contact distance for single-sensing detection with mobility model 1; $r = 20m$, $\bar{u} = 1m/s$	132
6.9	Average detection latency for single-sensing and 2-sensing detection models with mobility model 1; $r = 40m$, $\bar{u} = 1m/s$	133
6.10	Detection probability with single-sensing detection Vs desired delay constraint with mobility model 1: $r = 20m$, $b = 500m$, $N = 125$, $\lambda = 0.0005$	135
6.11	Detection probability lower bound with single-sensing detection Vs fraction of mobile nodes in the network with random walk mobility model after completing $n = 20$ steps: for $\mu = \sqrt{2}r$ and $\mu = 2r$: $r = 20m$	136
6.12	Probability of node connectivity for $r_m = 75m$ and $r_m = 100m$. . .	137

List of Figures

7.1	Hybrid sensor network consisting of both static and mobile nodes: solid circles-mobile nodes, checked circles-static nodes	141
7.2	Sensor network with only static nodes	145
7.3	A mobile node's candidate locations at a given time	147
7.4	Voronoi polygons for each static node: Solid square-static node loca- tions, solid circles-grid points (centers) corresponding to static nodes, void circles-grid points (centers) corresponding to grids not covered by static nodes	148
7.5	Duplicate covering at a given time	151
7.6	Compensating for the lack of coverage in <i>static</i> cells	153
7.7	Presence probability matrix with proposed mobility protocol, $N =$ 40 , $\lambda_m = 0.5$, $u_{max} = 10m/s$ (a). after moving steps $S_T = 100$ (b). after moving steps $S_T = 1000$ (c). after moving steps $S_T = 10,000$.	155
7.8	Presence probability matrix with bounced random walk model, $N =$ 40 , $\lambda_m = 0.5$, $u_{max} = 10m/s$ (a). after moving steps $S_T = 100$ (b). after moving steps $S_T = 1000$ (c). after moving steps $S_T = 10,000$.	156
7.9	Mean and the standard deviation of presence probabilities at <i>void</i> cells vs. the number of movement steps S_T (in log scale) for proposed protocol and the bounced random walk mobility model, $N = 40$, $\lambda_m = 0.5$, $u_{max} = 10m/s$ (a). Mean (b). Standard deviation	157
7.10	(a). Average time taken for an arbitrary point to be revisited for different network sizes N : $u_{max} = 10m/s$, $r = 10m$, $b \approx 200m$ (b). Average time taken for an arbitrary point to be revisited for different node speeds: $N = 60$, $r = 10m$, $b \approx 200m$	159

8.1	Vertex and edge assignment of the expansion graph from time nT_r to time $(n+1)T_r$ for 3×3 square grid; edge weights are not marked. Note that the vertex $(5, nT_r)$ at time nT_r corresponds to a non-boundary cell of the considered grid and it has 9 outgoing edges from time nT_r to $(n + 1)T_r$. All the other vertices at time nT_r correspond to boundary-cells. For vertices $(1, nT_r)$, $(3, nT_r)$, $(7, nT_r)$, $(9, nT_r)$ at time nT_r , they have 4 outgoing edges while for vertices $(2, nT_r)$, $(4, nT_r)$, $(6, nT_r)$, $(8, nT_r)$, they have 6 outgoing edges from time nT_r to $(n + 1)T_r$	168
8.2	Worst-case detection probability	172
9.1	Candidate locations for a mobile node at time t	179
9.2	Decision fusion architecture for the hybrid sensor network with fusion center updating decisions over time	181
9.3	Detector structure at the k -th mobile node for the decision making based on the observations during time interval $((n - 1)T_s, nT_s]$. . .	185
9.4	Shortest path representation for finding maximum detection probability at time $n_D T_s$ at k -th mobile node	192
9.5	Decision fusion architecture for the hybrid sensor network with nodes updating decisions over time	195
9.6	Detection probability at the fusion center with desired detection delay with perfect communication, with decision fusion model 1: $T_s = 1s$, $\bar{u} = 1m/s$, $\gamma_0 = 20dB$, $p = 0$	205

List of Figures

9.7	Comparing the detection performance with exact and random target locations, with decision fusion model 1 : $T_s = 1s$, $\bar{u} = 1m/s$, $\gamma_0 = 20dB$, $\alpha = 0.22$, $p = 0$	206
9.8	Detection performance with varying α and p , with decision fusion model 1: $T_s = 1s$, $\bar{u} = 1m/s$, $\gamma_0 = 20dB$, $q_x = q_y = 0.5$	207
9.9	Detection performance at the fusion center with decision fusion model 2: $\alpha = 0.22$, $N = 30$, $\bar{u} = 1m/s$, $p_k = p = 0$	209
9.10	Minimum number of mobile nodes required to achieve a desired performance level within a desired delay constraint, for $p = 0$, with decides fusion model 2	210
10.1	Cluster based target tracking with reactive mobility	222
10.2	Illustration of providing $\beta = 3$ coverage at predicted target locations	236
10.3	Estimated trace of the target trajectory with proposed mobility management scheme; $n_s = 36$, $n_m = 12$, $C = 4$, $u_{max} = 10ms^{-1}$, $r_D = 5m$	245
10.4	RMSE and PCRLB for the estimated target position when β is changing; $n_s = 36$, $n_m = 12$, $C = 4$, $u_{max} = 10ms^{-1}$, $u_{min} = 0$, $r_D = 5m$	246
10.5	RMSE and PCRLB for the estimated target position when u_{max} is varying; $n_s = 36$, $n_m = 12$, $C = 4$, $u_{min} = 0$, $r_D = 5m$, $\beta = 2$	247
10.6	RMSE and PCRLB for the estimated target position when the number of mobile nodes is increasing; fraction of mobile nodes is $5/12$; $n_s = 28$, $n_m = 20$, $C = 4$, $u_{min} = 0$, $r_D = 5m$	248

List of Figures

10.7	RMSE and PCRLB for the estimated target position with binary observations (constant threshold); fraction of mobile nodes is 5/12; $n_s = 28, n_m = 20, C = 4, u_{min} = 0, r_D = 10m, \tau = 2$	250
10.8	RMSE and PCRLB for the estimated target position with binary observations (dynamic threshold); fraction of mobile nodes is 5/12; $n_s = 28, n_m = 20, C = 4, u_{min} = 0, r_D = 10m$	251
10.9	Illustration of mobile sensor locations and predicted target position in a particular cluster	253

List of Tables

3.1	PSO terminology	33
3.2	Comparison of analytical and numerical results when $\rho = 0$, $\gamma_0 = 10dB$, $\epsilon = 0.01$, $n = 10$	47
10.1	Average distance (in m) a mobile node needs to move to provide β -coverage	249

Abbreviations

AF	Amplify-and-forward
BLUE	Best linear unbiased estimator
BSC	Binary symmetric channel
CSI	Channel-state information
DS-CDMA	Direct sequence code-division multiple-access
IR	Infra-red
MAC	Multiple-access channel
MMSE	Minimum mean squared error
MSE	Mean squared error
MVUE	Minimum variance unbiased estimator
N-P	Neyman-Pearson
PCRLB	Posterior Cramér-Rao lower bounds
PLL	Phase-locked loop
PSO	Particle Swarm Optimization

Abbreviations

SNR	Signal-to-noise ratio
TBMA	Type-based multiple-access
WSN	Wireless sensor network

Chapter 1

Overview of the Dissertation

“Not everything that counts can be counted, and not everything that can be counted counts”-Albert Einstein

1.1 Introduction

Recent advances in low-power micro sensors, integrated circuits and wireless communication technologies have enabled the development of wireless sensor networks (WSNs) [15, 28, 33, 85] for wide variety of applications. Sensor networks are capable of information gathering, processing and dissemination in diverse and hostile environments. WSNs have unique advantages over other existing networks which make them suitable for many real world applications, due to certain factors such as ad hoc deployment, cost effectiveness and self-organization. However, efficient usage of WSNs in many applications is impeded by a plethora of challenges. Basically these sensor nodes are to be deployed in hostile environments, for example in military applications or environment monitoring applications, with unattended for a long time. Thus power management is considered as a core issue in designing WSNs. On the

other hand, maintaining connectivity and networking functions among geographically dispersed sensor nodes is going to be another field of study in WSNs. Resource constraints of a typical WSN include limited node energy, slow processors, small memory sizes and limited bandwidth. When mobile nodes are embedded with static nodes, a set of problems has to be solved in terms of the cost and energy for locomotion in addition to that for sensing, processing and communication. These factors make the WSNs technology an active research area in the community in communication, signal processing and networking aspects.

Some active areas of recent research in WSNs include distributed detection/estimation, data/decision fusion, node scheduling, node connectivity, wireless communication via imperfect channels and resource management. In this dissertation we address several challenges in a mobility assisted sensor network for detection and estimation with respect to signal processing and communication aspects. Major contributions of the dissertation are summarized in the following.

1.2 Main Contributions

1.2.1 Optimal power scheduling for distributed detection and estimation in stationary sensor networks with correlated observations:

Previous research in detection and estimation by stationary sensor networks under resource constraints has mainly focused on independent local observations [6, 22, 24, 60, 98, 147, 148]. However, when sensor nodes are densely deployed, node observations are spatially correlated in a practical sensor network [25, 69, 138]. In [25], it was shown how dense the sensor network should be for the dependence among ob-

servations to be increased. With correlated observations, the performance analysis is not analytically tractable compared to that with independent observations. In the dissertation, we propose several approaches to find the optimal power allocation in a distributed sensor network under correlated observations at local nodes. To obtain the optimal power allocation analytically in terms of network parameters for distributed detection, we derive an easy-to-optimize upper bound for the fusion error probability under certain correlation models. The proposed optimal power scheduling scheme with the derived upper-bound can be implemented distributively with a small feedback from the fusion center. With arbitrary correlation models, we propose numerical-based algorithm for optimal power scheduling based on particle swarm optimization. We investigate the performance reduction when the decision rules are implemented considering conditional independent assumption when the observations are actually correlated. For the distributed estimation with correlated observations using non-orthogonal communication channels, we derive the large system performance measures at the fusion center. Moreover, the optimal power scheduling schemes with orthogonal as well as multiple access channels are derived to maintain a desired performance level at the fusion center in estimating static parameters.

1.2.2 Distributed node selection schemes for sequential estimation over noisy communication channels in stationary sensor networks:

In distributed sensor networks, due to energy limitations and failure tolerance properties, centralized data/decision fusion may not be desirable always. On the other hand, in centralized data/decision fusion, a large communication power is consumed at each node since nodes have to communicate with a central fusion center. In this dissertation, we investigate the problem of efficient data fusion with inter-node com-

munication. In particular, we propose a distributed sequential methodology for static parameter estimation in a stationary sensor network in which nodes update estimator sequentially based on their own observations and information received from previous nodes. In the sequential estimation process, we assume noisy communication channels among nodes in contrast to most existing work with perfect communication channels in sequential analysis. In such a sequential process not all nodes among all available sensor nodes provide useful information which improves the accuracy of the estimator. This motivates to find the set of nodes with proper ordering which improves the estimator performance as the sequential process continues. However, due to the distributed nature, the optimal set of nodes should be found without having explicit knowledge of measurements residing at each nodes since otherwise it needs a large communication burden to exchange measurements among nodes. Also it can be shown that finding such optimal set of nodes is computationally complex as the network size is large. To that end, we propose distributed algorithms for node selection in the sequential estimation process which can be implemented having only partial information on some parameters at neighboring nodes at each node, but still without much deviating from the optimal performance.

1.2.3 Trade-off between mobile node density and detection performance measures in hybrid sensor networks with random mobility models:

As far as a static sensor network is considered, the performance is limited by its initial network configuration. For example, to achieve a k -coverage in a random sensor network, with a network size of L , the sensor node density should be increased as $O(\log L + k \log \log L)$ [136] at the initial stage. On the other hand, even a all-static network meets the performance quality requirement at the initial deployment stage,

it does not adapt to unpredictable dynamics in network conditions such as coverage holes caused by node failures, or changing dynamics of the phenomenon being sensed over time. Recently mobile sensor nodes are deployed in wireless sensor network applications to provide dynamic performance. However, deploying a large number of mobile nodes in a sensor network is expensive due to energy consumption compared to that with static sensor nodes. We investigate the cost of deploying mobile nodes in terms of the node density required to achieve desired performance measures. For this discussion we assume random node mobility, which is reasonable assumption when it requires the minimum coordination among nodes and is justifiable in scenarios where mobile nodes do not have a priori information on the sensing field.

1.2.4 Distributed protocol for mobile node navigation in a hybrid sensor network:

Although random mobility models are desirable in many situations since they need minimum coordinations among nodes, they might not be ideal to provide an efficient coverage in a hybrid sensor network consisting of both static and mobile nodes. In a hybrid sensor network, it is cost effective if the node mobility is exploited only to compensate for the lack of performance resulted in all-static network. We propose a new interactive distributed protocol for node mobility management in a hybrid network such that the time that an arbitrary point in the sensing region is uncovered by at least one node, is minimized. The proposed protocol can be implemented distributively collaborating among static and mobile nodes in the local neighborhood.

1.2.5 Worst-case detection performance in hybrid sensor networks:

One of the fundamental problems in target detection is *exposure* which measures how well the region is covered by sensor nodes. This measure is important to diagnose the weaknesses of the coverage patterns of a sensor network in target detecting. We propose an efficient sequential technique to find the *exposure* to reflect the worst-case detection performance, using graph-theoretic techniques. The proposed algorithm is valid for arbitrary node mobility models.

1.2.6 Decision fusion models for hybrid sensor networks to detect randomly located targets:

Since using mobile nodes in a hybrid sensor network for continuous movements might be expensive due to energy constraints, it may be desired to keep mobile nodes stationary until a target is detected with certain confidence level, or useful statistics regarding the target locations are available under stationary configuration. In such scenarios, mobile nodes are allowed to move only if it is necessary to improve the detection performance. We develop a theoretical framework for decision fusion incorporating measurement uncertainties for such applications when the target location is random. Specifically two decision fusion models are proposed. The cost of allowing nodes to be mobile is explored in terms of the minimum set of mobile nodes required to move to reach at a desired performance level.

1.2.7 Distributed tracking algorithm using particle filters with reactive mobility:

Estimation of the state of moving targets is another important application of wireless sensor networks which has been attracted much attention by signal processing community. The Bayesian approach provides the general framework to solve dynamic state estimation problems, in which the key is to construct the probability density function (pdf) of the underlying state vector based on the available observations. When the state dynamics and observation models are linear and Gaussian, the optimal Bayesian estimator is given by well known Kalman filter [105]. However, in most real world applications, dynamic state estimation problems are non-linear and non-Gaussian. Under the Bayesian approach, for non-linear and non-Gaussian problems, obtaining the optimal solution in closed-form is not tractable. Use of particle filtering for solving non-linear/non-Gaussian dynamic state estimation problems is becoming an attractive in which the required pdfs are represented as a set of random sampling. Use of particle filters for target tracking applications is addressed by many authors in static sensor networks in different contexts [21, 35, 41, 54, 100, 101]. However, a less effort is made for target tracking in mobility assisted sensor networks. We develop a cluster-based distributed tracking algorithm based on particle filters in hybrid sensor networks with reactive mobility. The key feature in the proposed tracking algorithm is that the predicted target locations are covered dynamically to the desired *coverage level* exploiting node mobility.

1.3 Dissertation Outline

An introduction to wireless sensor networks focusing on signal processing and communication aspects is given in Chapter 2. Chapters 3 and 4 explore the optimal power

management techniques for detection and estimation in stationary sensor networks with correlated observations considering different communication architectures. In Chapter 5 we propose a distributed sequential methodology for estimation of a random parameter incorporating distributed node selection algorithms. In Chapter 6, we derive analytical formulas for different performance measures which are important in designing hybrid sensor networks consisting of both static and mobile nodes, under random node mobility models. Specifically, we investigate the trade-off between the mobile node density and the performance measures under certain constraints. Chapter 7 proposes a novel distributed protocol for node mobility collaborating with static and mobile nodes such that the uncovered area by static nodes will be covered in an efficient manner as time progresses. Chapter 8 develops an efficient sequential methodology to find the worst-case detection performance in a hybrid sensor network, in terms of the *exposure* when a target tries to escape the sensing region without being detected. In Chapter 9, we propose two decision fusion models for target detection incorporating measurement uncertainties for hybrid sensor networks to enhance the detection performance, if a target is detected by the stationary configuration with a certain confidence level. Chapter 10 provides a distributed cluster-based tracking algorithm in a hybrid sensor network with reactive node mobility. In Chapter 11, we conclude the dissertation providing some future work.

Chapter 2

Wireless Sensor Networks

2.1 Introduction

A wireless sensor network is a collection of densely deployed small, low-cost and low-power sensor nodes which are capable of collecting and disseminating environmental data. These sensor nodes are equipped with one or multiple sensing units, local processor with limited processing and wireless transceiver which enables wireless communication. Upon deployment, the sensor nodes organize themselves to form a connected network to perform the required tasks.

These small sensor nodes are capable of processing and intelligently analyzing local observations. To combine locally processed data at multiple sensor nodes, nodes communicate with neighboring nodes or with a separate fusion center over wireless channels to exchange information.

2.2 Static and Mobile Sensors

2.2.1 Static sensors

A typical sensor node used in wireless sensor networks is capable of performing (i). sensing, (ii). On-board processing, and (iii). Wireless communication. With current technological trends, sensing units, processors, and communication devices of sensor nodes are getting much smaller and cheaper compared to that with sensor networks in early ages. These sensor nodes can be deployed on ground, in the air, under water, on bodies of vehicles and inside buildings in either stationary or mobile configurations depending on the requirement and the possibilities. The initial deployment may be systematic (e.g. equi-space in a grid) or random depending on certain factors such as reachability and the cost of deployment. For example, a smart Dust optical mote, built by Dust Inc., Berkeley, CA, can be deployed by dropping to float to the ground [33].

Sensing

A sensing unit of a sensor node consists of one or multiple onboard sensors such as passive: acoustic, seismic, video, IR, magnetic or active: radar, ladar sensors [33]. Fig. 2.1 (a) illustrates an example sensing unit by *crossbow*, *MTS300*. *MTS300* sensor board has a variety of sensing modalities including light, temperature and acoustic and it can be used with the mote MICAz by *crossbow* (in Fig. 2.1 (b)) [1].

Processing and communication

A typical sensor node is associated with a programmable processor board with RF transceiver, which enables wireless communication. An example processor/radio



(a). MTS300 sensing board



(b). MICAz mote

Source: <http://www.xbow.com>

Figure 2.1: Sensing unit and transceiver/processor unit of a typical sensor node

board of MICAz mote by *crossbow* is shown in Fig. 2.1 (b). MICAz mote by *crossbow* is associated with IEEE 802.11.15.4 compliant RF transceiver and Atmel ATmega 128L low power micro-controller [1].

2.2.2 Mobile sensors

By mobile sensors we mean unmanned autonomous vehicles (UAV) or mobile robots equipped with various sensor nodes. With the technological advances in networking

and low cost high endurance electro-mechanical systems, it is now practical and relatively inexpensive to deploy robot vehicles for autonomous sensing and data collection in a broad area [155]. Example applications of mobile nodes are XYZ, [83], robomote, [116] and flip based sensors (which can jump only once) [27].

2.3 Data/Decsion Fusion Architectures

When spatially separated individual nodes collect observations on the phenomenon of interest (PoI) and process them locally, the processed local information should be efficiently combined to provide the best interpretation of the PoI. To perform the data/decison fusion task in sensor networks, several architectures are employed.

2.3.1 Centralized data/decison fusion

In a centralized fusion architecture, individual nodes transmit their locally processed data to a separate fusion center as shown in Fig. 2.2. The fusion center combines the summaries of observations received from local sensor nodes in an efficient way to provide a meaningful interpretation to the PoI. In this architecture, high level processing at local nodes may not necessary. However, nodes may require a large communication power.

2.3.2 Distributed data/decison fusion

Centralized decison/data fusion encounters several problems in a distributed sensor network under strict resource constraints. Since power consumption in wireless communication dominates the power consumption in processing and sensing [15] at a sensor node, the centralized architecture consumes a significant portion of the avail-



Figure 2.2: Centralized data/decision fusion architecture with a fusion center

able node energy. On the other hand, the reliability of such architectures depends on the robustness of the central fusion center. Distributed architectures, in which nodes communicate locally to exchange their information to make the final decision are more robust and efficient in terms of saving communication power. One such architecture is sequential communication as shown in Fig. 2.3, in which nodes communicate sequentially until a desired performance level is reached.

2.4 Applications

Wireless sensor networks are becoming a reality in many military and commercial applications. Current and potential applications of wireless sensor networks

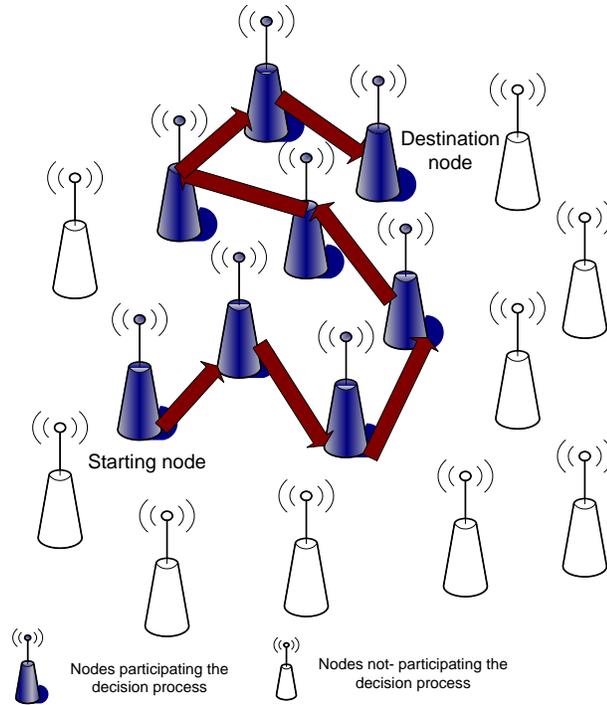


Figure 2.3: Distributed sequential data/decision fusion architecture

include, military sensing, environmental and habitat monitoring, infrastructure security, physical security, traffic surveillance, video surveillance, industrial and manufacturing automation, distributed robotics, environmental monitoring, building and structures monitoring, industrial sensing, traffic control, home automation and telemedicine [85,117,127,148]. In the following, we discuss some of these applications in detail.

- **Military applications:** Similar to advances in many other technologies, research and developments in sensor networks have been driven initially by military applications. For example, in early ages, sound surveillance systems consisting of acoustic arrays were deployed at strategic locations on the ocean bottom to detect and track submarines. More advanced techniques similar to the same

strategy were later used for monitoring events (e.g. earthquake and animal activity) in the ocean [33]. Sensor networks in various forms are used in military applications to monitor intruders or unauthorized activities. When sensor networks are used to monitor battlefields, adapting to dynamic environments, and network information processing under severe resource constraints are key challenges to be addressed.

- **Habitat/environmental monitoring:** Apart from military applications, sensor networks are becoming attractive in many commercial and civilian applications. In [85], researchers from UCB/Intel laboratory have shown that habitat monitoring is a driver application of wireless sensor networks. They have deployed a mote-based tiered sensor network on Great Duck Island, Maine to monitor the behavior of storm petrel. Another potential application of sensor networks is to monitor environmental conditions such as temperature, pressure and humidity. Sponsored by the government of Brazil, a very large scale sensor network consisting of different types of interconnected sensors including radar, image and environmental sensors, is deployed to provide environmental monitoring, drug trafficking monitoring and air traffic control for the Amazon Basin [64].
- **Industrial applications:** Industrial sensing applications of sensor networks include for example, monitoring machine conditions through determination of vibration or lubrication levels and deploying remote sensing nets to perform quality control tests on automate production and assembly lines.
- **Traffic control:** Another application area of sensor networks is vehicle traffic monitoring. When video cameras are used to monitor road segments with heavy traffic, due to cost of deploying them, traffic monitoring may be limited to few critical points. Cheap sensor nodes with embedded networking capabilities may be potential candidates that can be deployed at every road intersections

to count vehicle traffic. By communicating among neighboring nodes, a global traffic picture can be developed eventually, which can be queried by human operators or automatic controllers [33].

- Health applications: Sensor networks consisting of bio-medical sensors are potential candidates in health monitoring applications such as Glucose level monitoring, organ monitoring and cancer detectors. Other applications in this category include monitoring and tracking patients and doctors inside a hospital.

2.5 Challenges

In spite of the applicability of wireless sensor networks in a wide variety of applications, sensor networks pose a number of unique technical challenges which impede the deployment of many of the envisaged applications due to several factors [33, 47]:

- Energy limitations: Sensor nodes are equipped with a finite source of energy. When a sensor network is initially deployed, in most cases, it does not have human intervention. Replenishment of power sources in such networks may be impossible or difficult. In a sensor node, the main areas of power consumption are processing and communicating while the communication power dominates the processing power. It is a challenging problem to optimally utilize the available energy to perform the required task.
- Ad hoc deployment: In most applications, sensor nodes are to be deployed in regions where no infrastructure at all. For example, in military applications, sensor networks are deployed by dropping or throwing sensors into the sensor field. Under these scenarios, sensor nodes need to self organize to maintain optimal coverage and connectivity.

- **Dynamic changes:** Sensor networks need to be adaptable to changing network conditions over time. For example, network conditions are changing over time compared to that with at the initial stage of deployment. It is necessary to better cope with changing network conditions and dynamically varying on-demand requirements to facilitate required functions.
- **Reliability and fault tolerance:** It is important to maintain the reliability of the network after the initial deployment. For example, node failures may occur due to lack of power, physical damages or environmental interferences. The sensor network should be able to sustain network functionalities without any interruption due to node failures and other network faults.

2.6 Signal Processing Aspects and Research Directions

While sensor networks are used in a wide variety of different applications, they share a common set of technical issues in terms of signal processing and communication. According to signal processing perspectives, main tasks carried out by a wireless sensor network can be categorized as detection, classification, and estimation (static as well as dynamic states).

Some of the research areas in sensor networks can be listed as follows:

- **Local processing:** When distributed nodes make observations on the PoI, it is necessary to process them locally with the available limited processing capabilities of sensor nodes to produce a summary of the observations. One such approach is to quantize the observations to a finite number of messages prior to being transmitted to the fusion center [148]. For detection problems, it was

shown that [128], performing threshold tests on local likelihood ratios at sensor nodes is optimal with independent local observations but not true for correlated observations. Design of local processing schemes at sensor nodes under different network conditions is addressed by [48, 90].

- Collaborative signal processing: When spatially distributed individual nodes make observations on the PoI, they should be properly intergraded in such a manner that the maximum benefits are achieved utilizing scarce network resources efficiently. Collaborative signal processing in distributed detection, estimation and tracking is addressed in the literature in different contexts. Collaborative signal processing with a central fusion center under power and bandwidth constraints is addressed by [3, 6, 10, 22, 24, 60, 61, 69, 88, 90, 111, 119, 125, 143, 148, 152, 153] and [9, 39, 45, 58, 60, 61, 73, 81, 82, 87, 93, 94, 107, 108, 124, 141], while [34, 56, 63, 119, 123, 156, 157] addressed in terms of the distributed sequential processing. However, most of these existing work addressed collaborative signal processing in stationary sensor networks.
- Sensor scheduling and deployment: Initial deployment of nodes to maximize the network performance is important in designing sensor networks. In [62, 80, 135, 150, 158], initial node deployment strategies in resource constrained sensor networks are addressed in different contexts. Sensor scheduling is another important research direction, in which nodes switch between active and sleep modes periodically as desired, extending the node lifetime.
- Mobility assisted sensor networks: To cope with the challenges due to dynamically changing network conditions and for on-demand applications after initial deployment, an alternative solution is to add mobility into typical stationary sensor networks. Use of node mobility to reposition sensor nodes to provide a uniform coverage at the initial deployment stage was addressed by [27, 57, 135, 136, 146, 158]. Use of node mobility for continuous cov-

erage/capacity improvements is addressed by [51, 79, 86]. However, the performance gain achieved by adding node mobility depends on certain factors such as mobility patterns, cost of adding mobility and dynamics of the phenomenon being sensed. It is of interest to investigate the trade-off among these factors for efficient use of node mobility in mobility assisted sensor networks.

Chapter 3

Power Management for Detection with Correlated Observations

3.1 Introduction

Distributed detection is one of the fundamental tasks performed by a wireless sensor network (WSN). In distributed detection with a central fusion center, each node in the network sends a summary of its observation to the fusion center. The local processing to produce a summary of observations at distributed nodes can be a form of lossy compression or simple relaying. The fusion center makes use of partially processed data from local nodes to make the final decision. Since only a summary of observations is transmitted, decentralized detection has the potential to extend the lifetime of the sensor network at the expense of some performance reduction. The fusion performance of a decentralized detection system in a low power stationary WSN is limited by resource constraints, namely power and bandwidth. In a typical WSN, communication and computing capabilities of sensor nodes can be limited due to various design considerations such as finite energy and limited available band-

width. For example, it may be impractical to replace or recharge the batteries due to cost and operating environment considerations. Therefore, the power management is considered to be a core issue in designing a WSN.

Distributed detection and fusion in stationary sensor networks under resource constraints have been considered by many authors [3, 4, 6, 22–24, 26, 59–61, 95]. These have studied the fusion performance under given power or bandwidth constraints on the network. For example, in [22], it was shown that when the network is subjected to a joint power constraint, having identical sensor nodes (i.e. all nodes using the same transmission scheme) is asymptotically optimal for binary decentralized detection. When the whole system is subjected to a total average power constraint, [59] showed that it is better to combine as many as not-so-good local decisions as possible rather than relying on a few very good local decisions in the case of deterministic signal detection. The optimal power scheduling for distributed detection in a WSN with independent observations has recently been considered in [153], where an optimal power allocation scheme was developed with respect to the so-called J -divergence performance index. Neyman-Pearson detection of correlated random signals is addressed in [120] where the performance is analyzed via error exponent using the large deviations principle. Distributed detection using multiple access channels was considered in [78] with both synchronous and asynchronous transmissions. Distributed detection based on counting rules at the fusion center was addressed by [133] in which distributed nodes make likelihood ratio testing with a constant threshold among nodes.

However, most of the work on distributed detection under resource constraints discussed above considered independent observations at local sensor nodes. In practice, when nodes are densely deployed node observations can be spatially correlated [25, 44, 74, 132]. In this Chapter, we address the problem of how to optimally allocate the node power (we consider only transmit power) for detection of a constant

signal in a stationary sensor network with independent as well as correlated observations while keeping a certain performance level at the fusion center. We consider a WSN consisting of a fusion center and an n number of spatially distributed sensor nodes. The distributed nodes collect observations corrupted by Gaussian noise and perform amplify-and-forward (AF) local processing to compute a local message that is transmitted to the fusion center. The wireless channel between the nodes and the fusion center is assumed to undergo fading.

First we consider the case where the local observations are independent and derive the optimal power allocation scheme analytically. For the correlated observations, we derive the exact, as well as an upper bound for the fusion error probability that is easy to optimize when the local observation correlations are sufficiently small. Based on the derived upper bound, the optimal power allocation scheme is found analytically. Next, we use Particle Swarm Optimization (PSO), which is a numerical technique based on the movement and intelligence of particles of a swarm, to numerically find the optimal power allocation scheme for arbitrarily correlated Gaussian observations.

As we will show, according to the optimal power allocation scheme that conserves total power spent by the whole WSN, nodes with poor observation quality and/or bad channels are turned off while the other nodes transmit locally processed data to the fusion center. We will show that when local signal-to-noise ratio (SNR) is large, only a small number of nodes needs to be active to achieve the required fusion error performance while a relatively large number of nodes should be active when the local SNR is small. We also observe that the optimal power allocation scheme has considerably better performance over that with the uniform power allocation scheme specifically when the number of nodes in the network is large. It is also verified that the results obtained via PSO-based numerical method closely match with analytical results under the same network conditions when the observations are i.i.d.. We also investigate the performance of the analytical power allocation scheme derived

under the conditionally independent assumption in a network when observations are actually correlated. It can be seen that for large correlations, the conditionally independent assumption degrades the performance significantly compared to the performance of PSO-based method for correlated observations. It is also noted that the power allocation schemes, obtained analytically for independent observations and for correlated observations based on the derived probability bound, can be implemented distributively with a small feedback from the fusion center provided that the channel state information is available at nodes.

The remainder of this Chapter is organized as follows: Section 3.2 formulates the fusion problem. In Section 3.3 the optimal fusion performance is analyzed. The proposed optimal power allocation schemes with independent observations and correlated observations are discussed in Sections 3.4 and 3.5, respectively. Section 3.6 gives the performance results and finally concluding remarks are given in Section 3.7.

3.2 Data Fusion Problem Formulation

We consider a binary hypothesis testing problem in an n -node distributed wireless sensor network. The k -th sensor observation under the two hypotheses is given by,

$$\begin{aligned} H_0 : z_k &= v_k; \quad k = 1, 2, \dots, n \\ H_1 : z_k &= x_k + v_k; \quad k = 1, 2, \dots, n, \end{aligned} \tag{3.1}$$

where v_k is zero-mean Gaussian observation noise with variance σ_v^2 and x_k is the signal to be detected. In vector notation, (3.1) becomes $\mathbf{z} = \mathbf{x} + \mathbf{v}$ where \mathbf{v} is a zero mean Gaussian n -vector of noise samples with covariance matrix $\mathbf{\Sigma}_v$. In general we consider spatially correlated observations, so that $\mathbf{\Sigma}_v$ is not necessarily diagonal. We consider the detection of a constant signal so that $x_k = m$ for all k (the results hold

straightforwardly for any deterministic signal). Let us define the local signal-to-noise ratio $\gamma_0 = \frac{m^2}{\sigma_v^2}$. The prior probabilities of the two hypotheses H_1 and H_0 are denoted by $P(H_1) = \pi_1$ and $P(H_0) = \pi_0$, respectively.

We assume that amplify-and-forward (AF) local processing is used, according to which each node retransmits an amplified version of its own observation to the fusion center. Hence the local decisions sent to the fusion center are,

$$u_k = g_k z_k; \text{ for } k = 1, 2, \dots, n, \quad (3.2)$$

where g_k is the amplifier gain at node k . The received signal r_k at the fusion center under each hypothesis is given by;

$$H_0 : r_k = n_k; \quad (3.3)$$

and

$$H_1 : r_k = h_k g_k x_k + n_k; \quad (3.4)$$

for $k = 1, 2, \dots, n$ where $n_k = h_k g_k v_k + w_k$, h_k is the channel fading coefficient and w_k is the receiver noise that is assumed to be i.i.d. with mean zero and variance σ_w^2 . Defining $\mathbf{r} = [r_1, \dots, r_n]^T$ and $\mathbf{n} = [n_1, \dots, n_n]^T$, we have,

$$\mathbf{r} = \mathbf{A}\mathbf{x} + \mathbf{n} \quad (3.5)$$

where $\mathbf{A} = \text{diag}(h_1 g_1, h_2 g_2, \dots, h_n g_n)$. The detection problem at the fusion center can then be formulated as,

$$\begin{aligned} H_0 & : \mathbf{r} \sim p_0(\mathbf{r}) = \mathcal{N}(0, \mathbf{\Sigma}_n) \\ H_1 & : \mathbf{r} \sim p_1(\mathbf{r}) = \mathcal{N}(\mathbf{A}\mathbf{m}, \mathbf{\Sigma}_n) \end{aligned} \quad (3.6)$$

where $\Sigma_{\mathbf{n}} = \mathbf{A}\Sigma_{\mathbf{v}}\mathbf{A} + \sigma_{\mathbf{w}}^2\mathbf{I}$, $\mathbf{m} = m\mathbf{e}$, \mathbf{e} is the n -vector of all ones and \mathbf{I} is the $n \times n$ identity matrix. The log-likelihood ratio (LLR) for the detection problem (3.6) can be written as,

$$T(\mathbf{r}) = m\mathbf{e}^T \mathbf{A}\Sigma_{\mathbf{n}}^{-1}\mathbf{r} - \frac{1}{2}m^2\mathbf{e}^T \mathbf{A}\Sigma_{\mathbf{n}}^{-1}\mathbf{A}\mathbf{e}. \quad (3.7)$$

It is well known that optimal fusion tests should be threshold tests on the above LLR. Thus the optimal Bayesian decision rule at the fusion center is given by,

$$\delta(\mathbf{r}) = \begin{cases} 1 & \text{if } T(\mathbf{r}) \geq \ln \tau \\ 0 & \text{if } T(\mathbf{r}) < \ln \tau, \end{cases} \quad (3.8)$$

where τ is the threshold given by $\tau = \frac{\pi_1}{\pi_0}$ (assuming minimum probability of error Bayesian fusion).

3.3 Analysis of Optimal Fusion Performance

Note that the distribution of the decision statistic $T(\mathbf{r})$ is given by (under two hypotheses),

$$\begin{aligned} H_0 : T(\mathbf{r}) &\sim \mathcal{N}\left(-\frac{1}{2}m^2\mathbf{e}^T \mathbf{A}\Sigma_{\mathbf{n}}^{-1}\mathbf{A}\mathbf{e}, m^2\mathbf{e}^T \mathbf{A}\Sigma_{\mathbf{n}}^{-1}\mathbf{A}\mathbf{e}\right) \\ H_1 : T(\mathbf{r}) &\sim \mathcal{N}\left(\frac{1}{2}m^2\mathbf{e}^T \mathbf{A}\Sigma_{\mathbf{n}}^{-1}\mathbf{A}\mathbf{e}, m^2\mathbf{e}^T \mathbf{A}\Sigma_{\mathbf{n}}^{-1}\mathbf{A}\mathbf{e}\right). \end{aligned} \quad (3.9)$$

The false alarm probability of the optimal detector (3.8) at the fusion center is

$$P_f = P(T(\mathbf{r}) > \ln \tau | H_0) = Q\left(\frac{\ln \tau + \frac{1}{2}m^2\mathbf{e}^T \mathbf{A}\Sigma_{\mathbf{n}}^{-1}\mathbf{A}\mathbf{e}}{m\sqrt{\mathbf{e}^T \mathbf{A}\Sigma_{\mathbf{n}}^{-1}\mathbf{A}\mathbf{e}}}\right), \quad (3.10)$$

where Q -function is defined by $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{\zeta^2}{2}} d\zeta$. Similarly, the detection probability associated with the decision rule (3.8) is given by

$$P_D = P(T(\mathbf{r}) > \ln\tau | H_1) = Q\left(\frac{\ln\tau - \frac{1}{2}m^2 \mathbf{e}^T \mathbf{A} \Sigma_{\mathbf{n}}^{-1} \mathbf{A} \mathbf{e}}{m \sqrt{\mathbf{e}^T \mathbf{A} \Sigma_{\mathbf{n}}^{-1} \mathbf{A} \mathbf{e}}}\right). \quad (3.11)$$

Hence the probability of error at the fusion center for a Bayesian optimal detector can be shown as,

$$P(E) = P_f \pi_0 + (1 - P_D) \pi_1 = Q\left(\frac{1}{2} \sqrt{m^2 \mathbf{e}^T \mathbf{A} \Sigma_{\mathbf{n}}^{-1} \mathbf{A} \mathbf{e}}\right) \quad (3.12)$$

where the prior probabilities are assumed to be equal so that $\tau = 1$.

3.3.1 Independent local observations

When the node observations are uncorrelated the noise covariance matrix $\Sigma_{\mathbf{v}}$ is simply given by $\Sigma_{\mathbf{v}} = \sigma_v^2 \mathbf{I}$. Then the probability of fusion error in (3.12) is simplified to,

$$P(E) = Q\left(\frac{1}{2} m \sqrt{\sum_{k=1}^n \frac{h_k^2 g_k^2}{h_k^2 g_k^2 \sigma_v^2 + \sigma_w^2}}\right). \quad (3.13)$$

It is interesting to note that $\lim_{g_k^2 \rightarrow \infty, k=1, \dots, n} \sum_{k=1}^n \frac{h_k^2 g_k^2}{h_k^2 g_k^2 \sigma_v^2 + \sigma_w^2} = \frac{n}{\sigma_v^2}$ so that the probability of fusion error has a performance floor:

$$\lim_{g_k^2 \rightarrow \infty, k=1, \dots, n} P(E) \rightarrow Q\left(\frac{\sqrt{n\gamma_0}}{2}\right). \quad (3.14)$$

Therefore, when $g_k \rightarrow \infty$ for $k = 1, \dots, n$, for a fixed number of sensors, n , the probability of fusion error is ultimately limited by the observation quality at local sensor nodes regardless of the quality of the wireless channel.

3.3.2 Correlated local observations

It is not straightforward to evaluate $\Sigma_{\mathbf{n}}^{-1}$ in (3.12) analytically in closed form for a general $\Sigma_{\mathbf{v}}$ when the observations are correlated. In the following we consider a specific sensor network model and obtain an upper bound for $P(E)$ in (3.12) that is valid for small correlations. To that end let us assume a 1-D sensor network in which adjacent nodes are separated by an equal distance d and correlation between nodes i and j is proportional to $\rho_0^{|i-j|}$ where $|\rho_0| \leq 1$. Letting $\rho_0^d = \rho$, $\Sigma_{\mathbf{v}}$ can be written as

$$\Sigma_{\mathbf{v}} = \sigma_v^2 \begin{bmatrix} 1 & \rho & \cdot & \cdot & \cdot & \rho^{n-2} & \rho^{n-1} \\ \rho & 1 & \cdot & \cdot & \cdot & \rho^{n-3} & \rho^{n-2} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \rho^{n-1} & \rho^{n-2} & \cdot & \cdot & \cdot & \rho & 1 \end{bmatrix}. \quad (3.15)$$

Note that, when ρ is sufficiently small, we may approximate (3.15) by its tri-diagonal version by dropping second and higher order terms of ρ . Recall, from Bergstrom's inequality [2] that, for any two positive definite matrices \mathbf{P} and \mathbf{Q}

$$\mathbf{e}^T \mathbf{P}^{-1} \mathbf{e} \geq \frac{(\mathbf{e}^T (\mathbf{P} + \mathbf{Q})^{-1} \mathbf{e})(\mathbf{e}^T \mathbf{Q}^{-1} \mathbf{e})}{\mathbf{e}^T \mathbf{Q}^{-1} \mathbf{e} - \mathbf{e}^T (\mathbf{P} + \mathbf{Q})^{-1} \mathbf{e}}. \quad (3.16)$$

Since $m^2 \mathbf{e}^T \mathbf{A} \Sigma_{\mathbf{n}}^{-1} \mathbf{A} \mathbf{e} = m^2 \mathbf{e}^T (\Sigma_{\mathbf{v}} + \sigma_w^2 \mathbf{A}^{-2})^{-1} \mathbf{e}$, let $\mathbf{P} = (\Sigma_{\mathbf{v}} + \sigma_w^2 \mathbf{A}^{-2})$ and define the matrix \mathbf{Q} such that

$$\mathbf{Q} = \sigma_v^2 \begin{bmatrix} 1 & -\rho & \cdot & \cdot & -\rho^{n-2} & -\rho^{n-1} \\ -\rho & 1 & \cdot & \cdot & -\rho^{n-3} & -\rho^{n-2} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ -\rho^{n-1} & -\rho^{n-2} & \cdot & \cdot & -\rho & 1 \end{bmatrix}.$$

For small enough ρ it can be shown that $\mathbf{e}^T \mathbf{Q} \mathbf{e} > 0$. In fact, when $\Sigma_{\mathbf{v}}$ has the tri-diagonal structure (implying only the adjacent node observations are correlated),

it can be shown that for any $|\rho| < \frac{n}{2(n-1)}$, we will have $\mathbf{e}^T \mathbf{Q} \mathbf{e} > 0$. In general, if $\Sigma_{\mathbf{v}}$ is as in (3.15), this will be true for small enough ρ . Note that while noise covariance matrix (3.15) is an idealization, it can be used in many applications, such as traffic monitoring or in industrial monitoring, where the sensors are approximately equally spaced. The tri-diagonal version of (3.15) is a reasonable approximation when the correlation coefficient ρ is small, since then the second and higher order terms of ρ in (3.15) are negligible. From (3.16) it can be shown that,

$$\mathbf{e}^T (\Sigma_{\mathbf{v}} + \sigma_w^2 \mathbf{A}^{-2})^{-1} \mathbf{e} \geq \left(\frac{1}{\sum_{k=1}^n \frac{h_k^2 g_k^2}{2h_k^2 g_k^2 \sigma_v^2 + \sigma_w^2}} - \frac{1}{D} \right)^{-1}, \quad (3.17)$$

where $D = \mathbf{e}^T \mathbf{Q}^{-1} \mathbf{e}$. From (3.12) and (3.17), we then have the following upper bound for the fusion error probability when the observations are correlated and ρ is sufficiently small:

$$P(E) \leq Q \left(\frac{m}{2} \left(\frac{1}{\sum_{k=1}^n \frac{h_k^2 g_k^2}{2h_k^2 g_k^2 \sigma_v^2 + \sigma_w^2}} - \frac{1}{D} \right)^{-\frac{1}{2}} \right). \quad (3.18)$$

When $\rho = 0$ we have $D = n/\sigma_v^2$. Then $\lim_{g_k^2 \rightarrow \infty, k=1, \dots, n} \left(\frac{1}{\sum_{k=1}^n \frac{h_k^2 g_k^2}{2h_k^2 g_k^2 \sigma_v^2 + \sigma_w^2}} - \frac{1}{D} \right)^{-1} = \frac{n}{\sigma_v^2}$.

That is, the fusion error probability bound (3.18) also has a performance floor of $Q \left(\frac{\sqrt{n\gamma_0}}{2} \right)$ as in (3.14), when local amplifier gains are large. Thus both the exact fusion error probability and the proposed bound exhibit the same performance in the case of i.i.d. observations when the amplification gain is large.

3.4 Optimal Power Allocation with Independent Observations

In the following, we first derive the optimal power allocation scheme that minimizes the total power spent by the whole sensor network subjected to a threshold on the fusion error probability when local observations are i.i.d..

In general, the power allocation problem can be formulated as,

$$\begin{aligned} & \min_{g_k \geq 0, k=1, \dots, n} \sum_{k=1}^n g_k^2 \text{ such that} \\ P(E) &= Q\left(\frac{1}{2} \sqrt{m^2 \mathbf{e}^T \mathbf{A} \Sigma_{\mathbf{n}}^{-1} \mathbf{A} \mathbf{e}}\right) \leq \epsilon \text{ and} \\ & g_k \geq 0; \quad k = 1, 2, \dots, n, \end{aligned} \quad (3.19)$$

where ϵ is the required fusion error probability at the fusion center.

When local observations are i.i.d., the fusion error probability is given by (3.13). Hence, the first inequality in (3.19) becomes $\beta \leq \sqrt{\sum_{k=1}^n \frac{h_k^2 g_k^2}{h_k^2 g_k^2 \sigma_v^2 + \sigma_w^2}}$ where we have defined $\beta = \frac{2}{m} Q^{-1}(\epsilon)$. Since β is positive, the optimal power allocation problem can thus be rewritten as,

$$\begin{aligned} & \min_{g_k \geq 0, k=1, \dots, n} \sum_{k=1}^n g_k^2, \text{ such that} \\ & \beta^2 - \sum_{k=1}^n \frac{h_k^2 g_k^2}{h_k^2 g_k^2 \sigma_v^2 + \sigma_w^2} \leq 0, \text{ and} \\ & g_k \geq 0 \text{ for } k = 1, 2, \dots, n, \end{aligned} \quad (3.20)$$

The Lagrangian for the above problem is

$$G(L, \lambda_0, \mu_k) = \sum_{k=1}^n g_k^2 + \lambda_0 \left[\beta^2 - \sum_{k=1}^n \frac{h_k^2 g_k^2}{h_k^2 g_k^2 \sigma_v^2 + \sigma_w^2} \right] + \sum_{k=1}^n \mu_k (-g_k) \quad (3.21)$$

where $\lambda_0 \geq 0$ and $\mu_k \geq 0$ for $k = 1, 2, \dots, n$. Verifying KKT conditions, it can be shown that the optimal solution for (3.20) is given by,

$$g_k^2 = \begin{cases} \frac{\sigma_w^2}{h_k^2 \sigma_v^2} \left[\frac{h_k \sum_{j=1}^{K_1} \frac{1}{h_j}}{(K_1 - \beta^2 \sigma_v^2)} - 1 \right] & ; \text{ if } k < K_1 \text{ and } n > \beta^2 \sigma_v^2 \\ 0 & ; \text{ if } k > K_1 \text{ and } n > \beta^2 \sigma_v^2 \\ \text{infeasible} & ; \text{ if } n < \beta^2 \sigma_v^2 \end{cases} \quad (3.22)$$

where K_1 is found such that $f(K_1) < 1$ and $f(K_1 + 1) \geq 1$ for $1 \leq K_1 \leq n$ assuming, without loss of generality, $h_1 \geq h_2 \geq \dots \geq h_n$ where

$$f(k) = \frac{(k - \beta^2 \sigma_v^2)}{h_k \sum_{j=1}^k \frac{1}{h_j}}, 1 \leq k \leq n. \quad (3.23)$$

The proof of the uniqueness of such a K_1 and the global optimality of the solution (3.22) for the optimization problem (3.20) are shown in the Appendix 3A.

Since there is a feasible optimal solution only when $n > \beta^2 \sigma_v^2$, i.e.

$\gamma_0 > \frac{4}{n}(Q^{-1}(P_e))^2$, this implies that we can not achieve probability of errors below $Q\left(\frac{\sqrt{n\gamma_0}}{2}\right)$. Note that this is consistent with (3.14). The optimal solution for g_k^2 when $f(k) - 1 < 0$ and $n > \beta^2 \sigma_v^2$ can be rewritten as $g_k^2 = \frac{\sigma_w^2}{h_k^2 \sigma_v^2} \left(\frac{\sqrt{\lambda_0} h_k}{\sigma_w} - 1 \right)$, where $\sqrt{\lambda_0} = \frac{\sigma_w \sum_{k=1}^{K_1} \frac{1}{h_k}}{K_1 - \beta^2 \sigma_v^2}$. Hence, once the fusion center calculates λ_0 and broadcasts it, each node can determine its power distributively using λ_0 as side information provided that the channel state information is available at nodes.

3.5 Optimal Power Allocation with Correlated Observations

Since it is not possible to find a closed form optimal solution for g_k 's in (3.19) when observations are correlated, in the following we solve it numerically. For that,

we develop a stochastic evolutionary computation technique based on PSO [65, 66, 109]. Since PSO is not directly applicable for constrained optimization problems, we first transform our constrained optimization problem in (3.19) into an unconstrained optimization problem using the exterior penalty function approach [106, 151]. Note that the choice of PSO technique for our problem is motivated by several factors. In the constrained PSO method, only the fitness function is needed to evaluate at each iteration in contrast to evaluating gradients in other numerical techniques. Further, it was observed that (as explained in subsection 3.5.3) the PSO-based algorithm is robust against the changes in network parameters.

3.5.1 Penalty function approach for constrained optimization

Suppose that the optimization problem of interest is

$$\min f(\mathbf{X}) \text{ such that } h_j(\mathbf{X}) \leq 0; \quad j = 1, \dots, m. \quad (3.24)$$

Then the exterior penalty function for the above minimization problem can be formulated as [106, 151],

$$\phi(\mathbf{X}, r_k) = f(\mathbf{X}) + r_k \sum_{j=1}^m (\max[0, h_j(\mathbf{X})])^q, \quad (3.25)$$

where r_k is a positive penalty parameter and q is a non-negative constant. Usually, the value of q is chosen to be 2 in practice [106]. The exterior penalty function algorithm that finds the optimal solution for the problem (3.24) can be stated as below: (Note that subscript of \mathbf{X} denotes the index corresponding to penalty parameter while the superscript of \mathbf{X} denotes the iteration number of the minimization algorithm for a particular penalty parameter).

- **step 1:** Set $k = 1$. Start from any initial solution \mathbf{X}_k^1 and a suitable value of $r_k = r_1$.
- **step 2:** Find the vector \mathbf{X}_k^* that minimizes the function given in (3.25).
- **step 3:** Test whether the point \mathbf{X}_k^* satisfies all the constraints. If \mathbf{X}_k^* is feasible, it is the desired optimum and hence terminate the procedure. Otherwise go to next step.
- **step 4:** Choose the next value of the penalty parameter according to the relation $\frac{r_{k+1}}{r_k} = c$ where c is a constant greater than one and set $\mathbf{X}_{k+1}^1 = \mathbf{X}_k^*$ and $k = k + 1$. Go to step 2.

Assuming that $f(\mathbf{X})$ and $h_j(\mathbf{X})$, $j = 1, 2, \dots, m$ are continuous and that an optimal solution exists for (3.24), the unconstrained minima \mathbf{X}_k^* of (3.25) converge to the optimal solution of the original problem $f(\mathbf{X})$ as $k \rightarrow \infty$ and $r_k \rightarrow \infty$ [106]. In order to ensure the existence of a global minimum of $\phi(\mathbf{X}, r_k)$ in (3.25) for every positive value r_k , $\phi(\cdot)$ has to be a strictly convex function of \mathbf{X} . The following theorem, the proof of which can be found in [106], gives the sufficient conditions for $\phi(\mathbf{X}, r_k)$ to be strictly convex:

Theorem 1

If $f(\mathbf{X})$ and $h_j(\mathbf{X})$, for $j = 1, 2, \dots, m$ are convex and at least $f(\mathbf{X})$ or either one of $\{h_j(\mathbf{X})\}_{j=1}^m$ is strictly convex, then the function $\phi(\mathbf{X}, r_k)$ defined by (3.25) will be a strictly convex function of \mathbf{X} .

3.5.2 Particle swarm optimization

To evaluate optimal \mathbf{X}_k^* for each penalty parameter r_k as required in the *step 2* above, we use the particle swarm optimization technique. A brief overview of the particle swarm language is given in Table 3.1 and more details can be found in [109].

Table 3.1: PSO terminology

Particle/Agent	A Single individual in the swarm
Location/Position	An agent's n -dimensional coordinates which represent a solution to the problem
Swarm	The entire collection of agents
Fitness	A single number representing the goodness of a given solution
pbest	The location in parameter space of the best fitness returned for a specific agent
gbest	The location in parameter space of the best fitness returned in the entire swarm
V_{\max}	The maximum allowed velocity in a given direction

In the following we give the algorithmic steps needed to implement the PSO for a given problem:

- (I). Define the solution space and the fitness function: Pick the parameters that need to be optimized and give them a reasonable range in which to search for the optimal solution. The fitness function should exhibit a functional dependence that is relative to the importance of each characteristic being optimized.

We denote the swarm size by M . For each k in (3.25), we perform a PSO optimization algorithm to find \mathbf{X}_k^* . For each k , let us define, $\mathbf{X}_{k,m}$ as the position vector of the m -th particle; $\mathbf{P}_{k,m}$ as the *pbest* of the m -th particle; $\mathbf{P}_{k,gbest}$ as the *gbest* of the swarm; $\phi(\mathbf{X}_{k,m}, r_k)$ as the fitness value corresponding to the location $\mathbf{X}_{k,m}$ of the m -th particle; $\phi(\mathbf{P}_{k,m}, r_k)$ as the fitness value corresponding to the *pbest* $\mathbf{P}_{k,m}$ of the m -th particle; $\phi(\mathbf{P}_{k,gbest}, r_k)$ as the fitness value corresponding to the *gbest* of the swarm and $\mathbf{V}_{k,m}$ as velocity of the m -th particle. The maximum number of iterations of PSO for each k is set to S .

- (II). If $k = 1$ (i.e. the penalty parameter is r_1) initialize the swarm locations randomly. Otherwise set the initial positions of each particle to be the best *pbest* values for $k = k - 1$.

- Initializing position: For $k = 1$ and for each particle m , $m = 1, \dots, M$, $\mathbf{X}_{k,m}^1$ is chosen randomly. If $k > 1$, then $\mathbf{X}_{k,m}^1 = \mathbf{P}_{k-1,m}^S$ where $\mathbf{P}_{k-1,m}^S$ is the *pbest* of the m -th particle for $k = k - 1$ at the S -th iteration of PSO.
- Initializing *pbest*: Since its initial position is the only location encountered by each particle at the run's start, this position becomes each particle's initial *pbest*. i.e. $\mathbf{P}_{k,m}^1 = \mathbf{X}_{k,m}^1$.
- Initializing *gbest*: The first *gbest* is selected as the initial *pbest* which gives the best fitness value: $\mathbf{P}_{k,gbest}^1 = \mathbf{P}_{k,m_1}^1$ where $m_1 = \arg \min_{1 \leq m \leq M} \{\phi(\mathbf{P}_{k,m}^1, r_k)\}$.
- Initializing velocities: Initialize $\mathbf{V}_{k,m}^1$ as zeros for each particle m .

(III). Fly the particles through the solution space:

Each particle is then moved through the solution space. The following steps are performed on each particle individually.

- Evaluate the particle's fitness value and compare it with that of *pbest* and *gbest*. For each particle, if its fitness value is better than that of the respective *pbest* for that particle or the global *gbest*, then the appropriate locations are replaced with the current location. i.e., in the s -th iteration of the PSO, for each particle m , for $m = 1, \dots, M$, if $\phi(\mathbf{X}_{k,m}^s, r_k) < \phi(\mathbf{P}_{k,m}^s, r_k)$ then set $\mathbf{P}_{k,m}^s = \mathbf{X}_{k,m}^s$. Set $\mathbf{P}_{k,gbest}^s = \mathbf{P}_{k,m_s}^s$ where $m_s = \arg \min_{1 \leq m \leq M} \{\phi(\mathbf{P}_{k,m}^s, r_k)\}$.
- Update the particle's velocity: The velocity of the particle is changed according to the relative locations of *pbest* and *gbest*. The particles are "accelerated" in the directions of the locations of best fitness value according to the following equation [102, 109]:

$$\begin{aligned} \mathbf{V}_{k,m}^{s+1} = \mathcal{X} \{ & (w\mathbf{V}_{k,m}^s + c_1rand()(\mathbf{P}_{k,m}^s - \mathbf{X}_{k,m}^s) \\ & + c_2rand()(\mathbf{P}_{k,gbest}^s - \mathbf{X}_{k,m}^s)) \}, \end{aligned} \quad (3.26)$$

where \mathcal{X} is the constriction factor that is used to control and constrict

velocities; w is the inertia weight that determines to what extent the particle remains along its original course unaffected by the pull of $pbest$ and $gbest$, c_1 and c_2 are positive constants that determine the relative "pull" of $pbest$ and $gbest$ (in fact c_1 determines how much the particle is influenced by the memory of its best location and c_2 determines how much the particle is influenced by the rest of the swarm) and the random number function $\text{rand}()$ returns a number between 0 and 1.

- Move the particle: Once the velocity has been determined as in (3.26), move the particle to its next location as $\mathbf{X}_{k,m}^{s+1} = \mathbf{X}_{k,m}^s + \Delta t \mathbf{V}_{k,m}^{s+1}$. The velocity is applied for a given time step Δt .

(IV). Repetition: After the velocity and the position are updated the process is repeated starting at step (III) until the termination criteria are met. The termination criteria can be a user-defined maximum iteration number or a target fitness termination condition. In the latter case, the PSO is run for the user-defined number of iterations, but at any time if a solution is found that is greater than or equal to the target fitness value, then PSO is stopped at that point. In our work we set the maximum iteration number (S) for PSO as defined before. Once the termination criteria are met, the optimal solution \mathbf{X}_k^* for the unconstrained minimization problem (3.25) for given k is $\mathbf{P}_{k,gbest}^S$.

To solve the optimization problem in (3.19) when the observations are correlated we define the exterior penalty function as,

$$\phi(\mathbf{g}, r_k) = f(\mathbf{g}) + r_k \{(\max[h_1(\mathbf{g}), 0])^2 + \sum_{j=2}^m (\max[h_j(\mathbf{g}), 0])^2\}, \quad (3.27)$$

where $f(\mathbf{g}) = \sum_{i=1}^n g_i^2$, $h_1(\mathbf{g}) = \beta^2 - \mathbf{e}^T \mathbf{A} \Sigma_{\mathbf{n}}^{-1} \mathbf{A} \mathbf{e}$ and $h_{i+1}(\mathbf{g}) = -g_i$ for $i = 1, 2, \dots, n$ and $\mathbf{g} = [g_1, \dots, g_n]^T$. Here we have $m = n + 1$. When the observation noise is i.i.d, it can be shown that $\phi(\mathbf{g}, r_k)$ is a strictly convex function for

$g_i \geq \frac{\sigma_w^2}{3h_i^2\sigma_v^2}$ for $i = 1, 2, \dots, n$ and also it can be seen that when h_i 's are small enough the convexity of $\phi(\mathbf{g}, r_k)$ holds for $g_i \geq 0$, ensuring a global minimum for $\phi(\mathbf{g}, r_k)$. We will assume that $\phi(\mathbf{g}, r_k)$ has a global minimum for each r_k even when the observation noise is correlated under above conditions. Assuming that an optimal solution for (3.19) exists and since $f(\mathbf{g})$ and $h_j(\mathbf{g})$ for $j = 1, 2, \dots, m$, are continuous, as $k \rightarrow \infty$ and $r_k \rightarrow \infty$ the unconstrained minima \mathbf{g}_k^* of $\phi(\mathbf{g}, r_k)$ converge to the optimal solution of the original problem (3.19).

3.5.3 Selection of parameter values for PSO

The parameter set to be optimized is $\mathbf{g} = [g_1, \dots, g_n]^T$ and we define the solution space as $[0, \infty)$ for each parameter. To run the PSO the population size was selected as 30 which has been shown to be sufficient for many engineering problems [19]. Various values for inertia weight w have been suggested in the literature. Since larger weights tend to encourage global exploration and conversely smaller initial weights encourage local exploitations, [46] has suggested to vary w linearly from 0.9 to 0.4 over the course of the run. On the other hand, [102] suggested to gradually decrease w from 1.2 towards 0.1 over the run of a PSO. We allowed w to vary between 0.9 to 0.4 linearly since it gave a fast convergence over 100 iterations. c_1 and c_2 were both set to 2.0 [109], [102]. The constriction factor \mathcal{X} was set to 0.73 [102].

One of the main advantage of the PSO based method is that once the algorithm parameters are chosen as above, the algorithm seems to work over a large range of variations in problem parameters such as fading coefficients, n, ρ and ϵ . On the other hand, the choice of step size and the initial values for a conventional method such as Newton's was observed to depend heavily on the problem parameters. The designer has to change the step sizes and the initial values every time when the system parameters change. This becomes especially problematic since fading coefficients are

random. Hence, although once proper choices have been made, the Newton's and the proposed PSO-based methods show almost similar convergence properties, the PSO based method seems much easier to use.

3.5.4 Power allocation based on the derived fusion error probability bound with correlated observations

When observations are correlated we may use the bound (3.18) to obtain an approximate analytical solution to the power allocation problem via

$$\begin{aligned} & \min_{g_k \geq 0, k=1, \dots, n} \sum_{k=1}^n g_k^2 \text{ such that} \\ & q - \sum_{k=1}^n \frac{h_k^2 g_k^2}{2h_k^2 g_k^2 \sigma_v^2 + \sigma_w^2} \leq 0 \quad \text{and} \\ & g_k \geq 0; \quad k = 1, 2, \dots, n, \end{aligned} \tag{3.28}$$

where $q = (\frac{1}{\beta^2} + \frac{1}{D})^{-1}$ and, as before, $\beta = \frac{2Q^{-1}(\epsilon)}{m}$ (Note that, $q > 0$ since $D > 0$). We can use the same method as in Section 3.4 to find the optimal solution for (3.28). Defining a function $\tilde{f}(k) = \frac{(k-2\sigma_v^2 q)}{h_k \sum_{j=1}^k \frac{1}{h_j}}$ and assuming again, $h_1 \geq h_2 \geq \dots \geq h_n$ it can be shown that (steps of deriving (3.29) are similar to that in Section 3.4 and are omitted here), we can find a unique L_1 such that $\tilde{f}(L_1) < 1$ and $\tilde{f}(L_1 + 1) \geq 1$ for $1 \leq L_1 \leq n$. Then the solution to the problem (3.28) is given by,

$$g_k^2 = \begin{cases} \frac{\sigma_w^2}{2h_k^2 \sigma_v^2} \left[\frac{h_k \sum_{j=1}^{L_1} \frac{1}{h_j}}{(L_1 - 2\sigma_v^2 q)} - 1 \right] & ; \text{ if } k < L_1 \text{ \& } n > 2\sigma_v^2 q \\ 0 & ; \text{ if } k > L_1 \text{ \& } n > 2\sigma_v^2 q \\ \text{infeasible} & ; \text{ if } n < 2\sigma_v^2 q \end{cases} \tag{3.29}$$

Note from (3.29) that to achieve the required fusion error probability at the fusion center the total number of active sensors should be greater than $2\sigma_v^2 q$ in the optimal

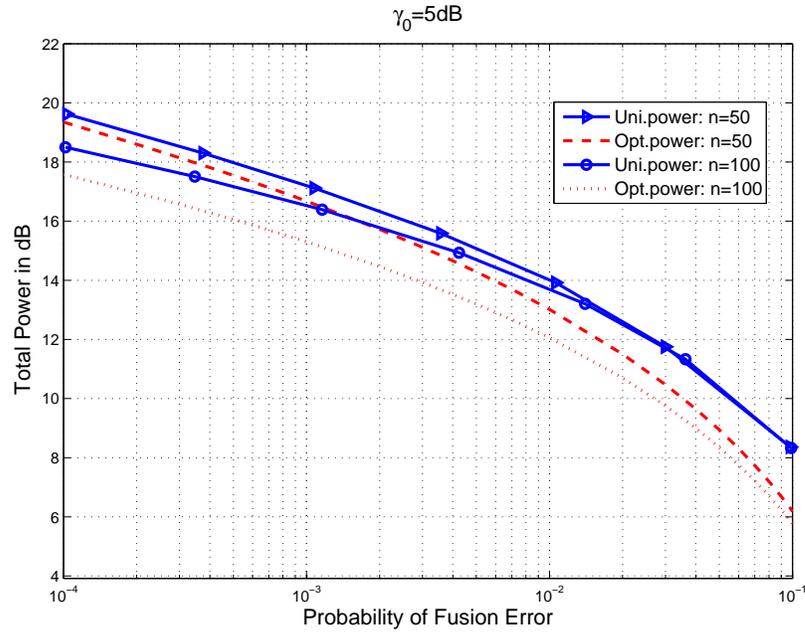


Figure 3.1: Total power Vs. probability of fusion error for independent observations: $\gamma_0 = 5dB$

solution where now is dependent on the correlation coefficient as well (since q depends on ρ).

When $\tilde{f}(k) - 1 < 0$ and $n > 2\sigma_v^2 q$, similar to Section 3.4, the optimal solution for g_k^2 in (3.29) can be rewritten as $g_k^2 = \frac{\sigma_w^2}{2h_k^2 \sigma_v^2} \left(\frac{\sqrt{\lambda'_0} h_k}{\sigma_w} - 1 \right)$, where $\sqrt{\lambda'_0} = \frac{\sigma_w \sum_{j=1}^{L_1} \frac{1}{h_j}}{L_1 - 2\sigma_v^2 q}$. Assuming channel state information is available at sensor nodes, each node can determine its power distributively using λ'_0 as side information, if the fusion center computes λ'_0 and broadcasts it to the sensor nodes.

3.6 Performance Results

In this section we illustrate performance gains possible with the derived optimal power allocation schemes. We assume that fading coefficients h_k 's of the channel

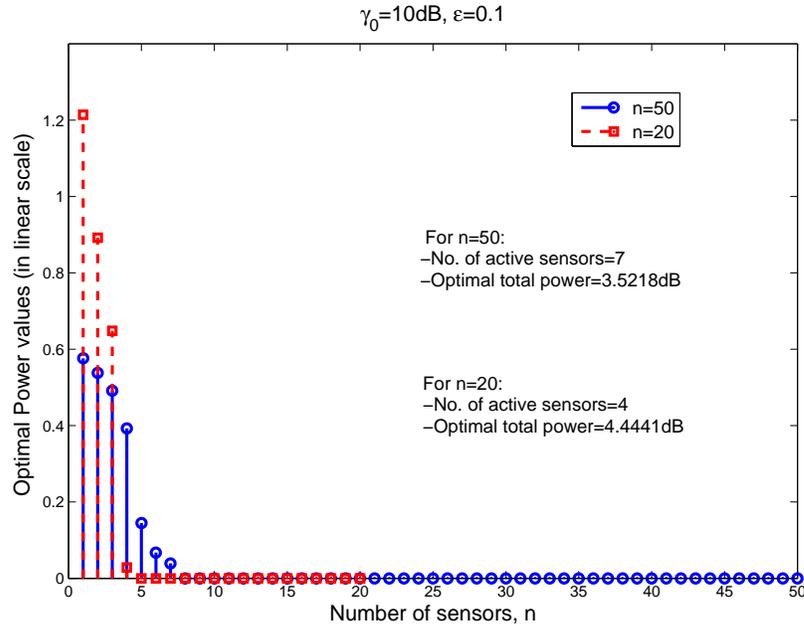


Figure 3.2: Optimal power values of sensor nodes Vs. number of sensors for $n = 20$ and $n = 50$ when $\epsilon = 0.1$ and $\gamma_0 = 10\text{dB}$

between sensors and the fusion center are Rayleigh distributed with a unit mean.

The results on Figs. 3.1 to 3.4 correspond to the optimal power allocation for i.i.d. observations. When observations are i.i.d. the optimal total power is given by $P_{Opt.} = \sum_{k=1}^{K_1} g_k^2$ where g_k^2 's are given in (3.22). The performance of the optimal scheme is compared with that of the uniform power allocation scheme in which each node uses same amplification gain such that $g_k = g$ for all k . To achieve a fusion error probability ϵ , with independent observations it can be shown that using (3.13) g is given as the solution to the following equation:

$$\epsilon = Q \left(\frac{mg^2}{2} \sum_{k=1}^n \frac{h_k^2}{g^2 h_k^2 \sigma_v^2 + \sigma_w^2} \right). \quad (3.30)$$

Figure 3.1 shows the total network power versus the desired fusion error proba-

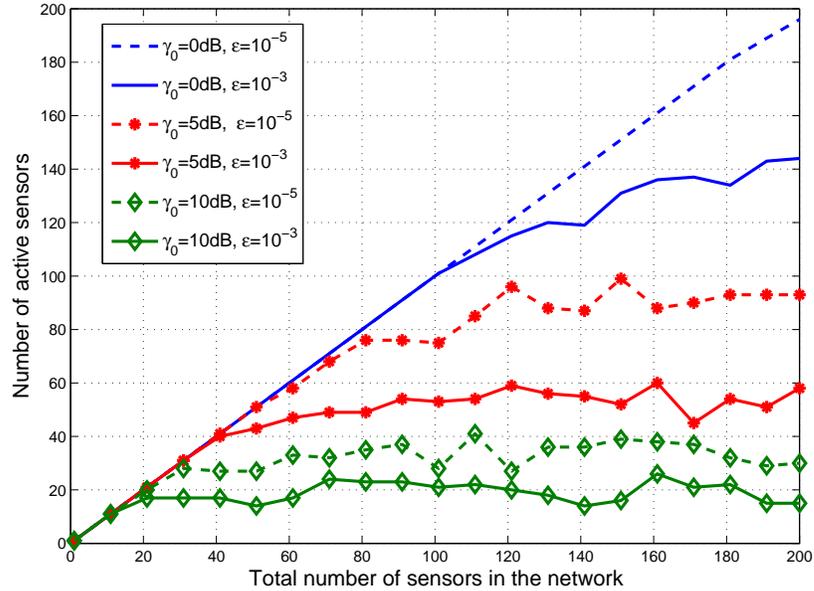


Figure 3.3: Number of active sensors for independent observations.

bility ϵ , for different values of n . It can be seen that when the number of sensors is increased then the energy saving due to proposed optimal scheme is more significant compared to that with uniform power allocation scheme. This is due to the fact that it is more likely that there will be more channels with good channel fading coefficients, when the number of sensors is large. By using those channels the network can spend a smaller total power, while still ensuring the required performance at the fusion center. From Fig. 3.1 it can also be seen that when the required fusion error probability is not significantly low, the gain of the optimal power allocation scheme over the uniform power allocation scheme is high.

An illustration of the power values needed by active nodes to meet a performance level (we let $\epsilon = 0.1$) with different n is shown in Fig. 3.2. Figure 3.2 essentially depicts how the required total power, to achieve the desired performance level, is divided among the active nodes.

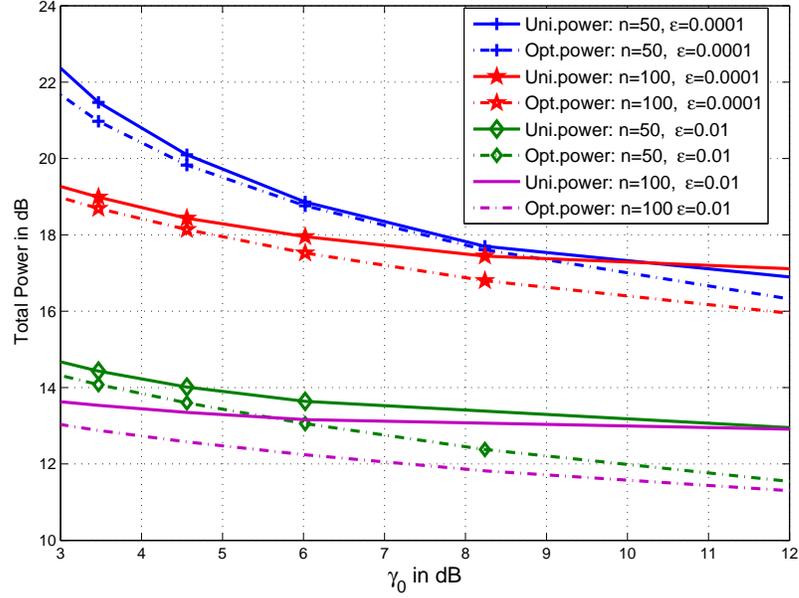


Figure 3.4: Total power Vs. local SNR for independent observations

The number of active sensors versus total number of sensors in the network to achieve a fusion performance levels of $\epsilon = 10^{-3}$ and $\epsilon = 10^{-5}$ with different γ_0 values is shown in Fig. 3.3. To achieve a given fusion error probability, it can be seen that only a small number of active sensors is needed when the local SNR is high. Fig. 3.3 also shows that a relatively large number of active sensors are needed to achieve lower fusion error probabilities compared to that of higher fusion error probabilities. This explains the high performance gain achieved at relatively higher fusion error probabilities as shown in Fig. 3.1. It is also noted from Fig. 3.3 that for relatively large γ_0 values, the number of active nodes needed to achieve the desired performance levels is nearly constant over the number of total nodes. This implies that when γ_0 is relatively large, it is optimal to divide the total power, required to achieve the desired fusion performance level, among a small number of nodes and switch the rest of the nodes off. After a certain number of total nodes, this required number of

active nodes is remaining almost same for relatively higher γ_0 values. However, it can be seen from Fig. 3.3 that for relatively low γ_0 values, the number of active sensors needed is nearly-linearly increasing with the number of total nodes. It implies that for low γ_0 region, it is optimal to divide the required total power among a relatively a large number of nodes, in contrast to what is observed in large γ_0 region.

In Figure 3.4 the total power versus the observation SNR γ_0 is shown for $n = 50$ and $n = 100$ parameterized by different fusion error probabilities. As expected, it can be seen that as γ_0 is increasing the total power required to achieve desired fusion performance levels is decreasing. Also, it is noted from Fig. 3.4 that the rate of decreasing of the total power (as γ_0 is increasing) is higher for low network sizes (i.e. for relatively small n) compared to that with larger network sizes. Moreover it is seen that effect of the optimal power scheduling scheme over the uniform power scheduling scheme is more significant in the relatively high SNR region. This phenomenon is understood from the results observed in Fig. 3.3, where at lower γ_0 region a relatively a large number of sensors has to be active to meet the desired performance levels. Then the optimal scheme suggests for a relatively large number of nodes to transmit their observations. Since in the uniform scheme, all nodes transmit their observations, the two schemes may consume approximately closer total power values in obtaining the desired performance level at lower γ_0 range.

In next two figures, the performance evaluation with the derived fusion error probability bound is presented. Before comparing the performance with optimal power allocation scheme, in Fig. 3.5 it shown that how tight the bound is for different network parameters, n and γ_0 assuming uniform power at each node. In Fig. 3.5, we have let $\rho = 0.2$. It can be seen from Fig. 3.5 that the derived bound (3.18) is a tight bound to the exact error probability (3.12) irrespective of the change of network parameters.

In Fig. 3.6 the fusion error performance with the optimal power allocation scheme

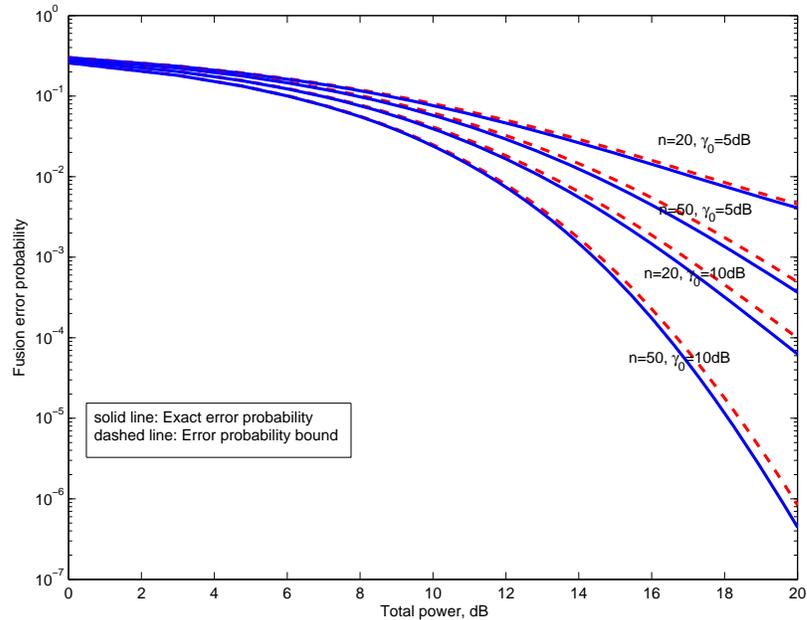


Figure 3.5: Comparison of the derived probability bound and the exact error probability for correlated observations; $\rho = 0.1$

for correlated observations based on the fusion error probability bound (3.18) is shown. The results in Fig. 3.6 are obtained assuming the observation noise covariance matrix has the tri-diagonal structure of (3.15) and we let $\rho = 0.1$. It can be seen from Fig. 3.6 that, in the optimal scheme, the total power required to keep the error probability bound under a certain threshold ϵ , is significantly less (in the region of moderate values of ϵ) than the total power needed with the uniform power allocation scheme to keep the exact fusion error probability under the same threshold. Performance gain achieved by the optimal scheme compared to that with the uniform scheme with the fusion error probability bound shows similar characteristics as with the i.i.d. observations when the network parameters n and γ_0 are changing (figures are not included for brevity).

Next we consider the performance results based on the constrained-PSO algo-

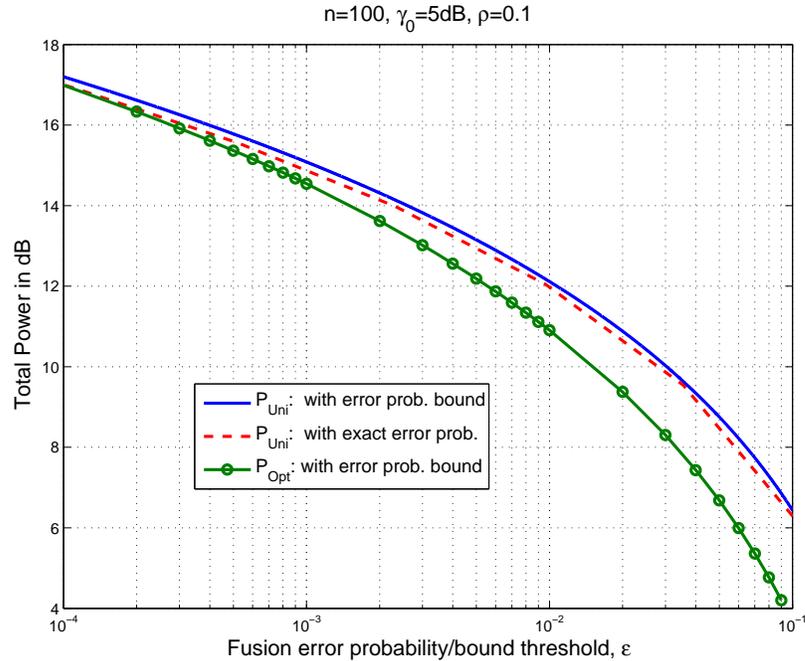


Figure 3.6: Total power and the fusion error probability bound for correlated observations; $\gamma_0 = 5\text{dB}$, $n = 100$ and $\rho = 0.1$

rithm. Note that we employed the PSO-based method for each penalty parameter r_k of the unconstrained optimization problem (3.27) until $\phi(\mathbf{g}_k^*, r_k) \rightarrow f(\mathbf{g}_k^*)$ where $\mathbf{g}_k^* = \text{argmin}_{\mathbf{g}_k} \phi(\mathbf{g}_k, r_k)$. For a given r_k the convergence of PSO algorithm is shown in Fig. 3.7. The starting penalty parameter r_1 was set to 2, and was increased in such a way that $\frac{r_{k+1}}{r_k} = 2$. It was observed that for each r_k the PSO algorithm converges rapidly. The convergence of unconstrained minimum of $\phi(\mathbf{g}, r_k)$ to the constrained minimum of $f(\mathbf{g})$ is shown in Fig. 3.8 in which the error between the penalty function and the objective function at the convergent point is 0.0023 after 7 iterations of r_k . That is, with a relatively smaller number of iterations, the unconstrained minimum of the penalty function $\phi(\mathbf{g}, r_k)$ approaches to that of the objective function $f(\mathbf{g})$.

The comparison of \mathbf{g}^* obtained numerically (via PSO) and analytically (3.22) under the same network conditions is shown in first two rows of the Table 3.2 for

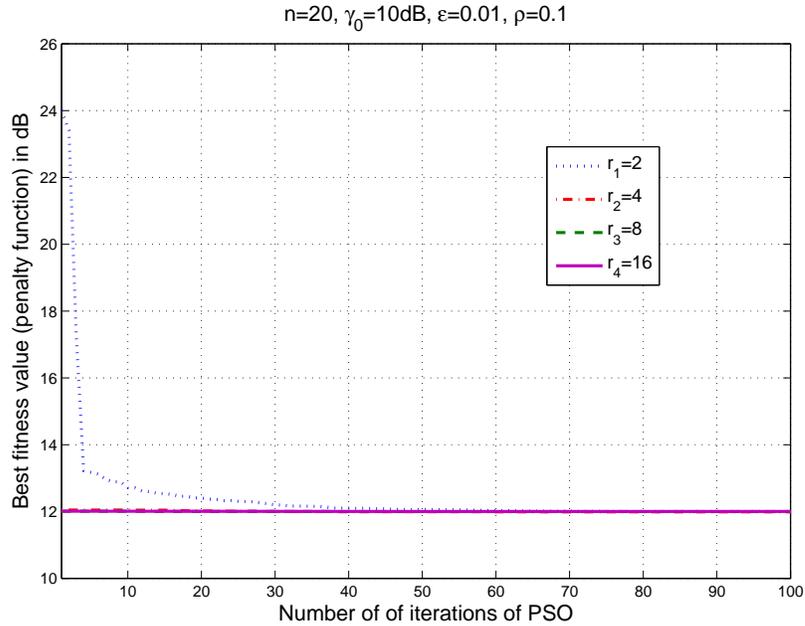


Figure 3.7: Convergence of exterior penalty function based PSO: Best fitness returned for PSO iterations for a given penalty parameter. (Fusion error probability = 0.01)

10 nodes when the observations are i.i.d.. It can be seen that the numerical results closely match with the analytical solution. The third row of Table 3.2 shows the optimal \mathbf{g}^* obtained numerically when $\rho = 0.1$, $n = 10$, $\gamma_0 = 10\text{dB}$ and $\epsilon = 0.01$. It shows that when the observations are correlated the optimal solution for (3.19) should turn off the sensors with poor channels similar to the analytical solution for i.i.d observations. But it is seen that then the sensors need more power when the observations are correlated for the same n , γ_0 and ϵ .

The dependance of the total network power (obtained via constrained-PSO) on the required fusion error probability when local observations are correlated is shown in Figs. 3.9 and 3.10 parameterized by ρ and γ_0 . Note that, the constrained-PSO method is applicable for any arbitrary observation noise correlation model. The results in Figs. 3.9 and 3.10 are based on the noise covariance matrix in (3.15).

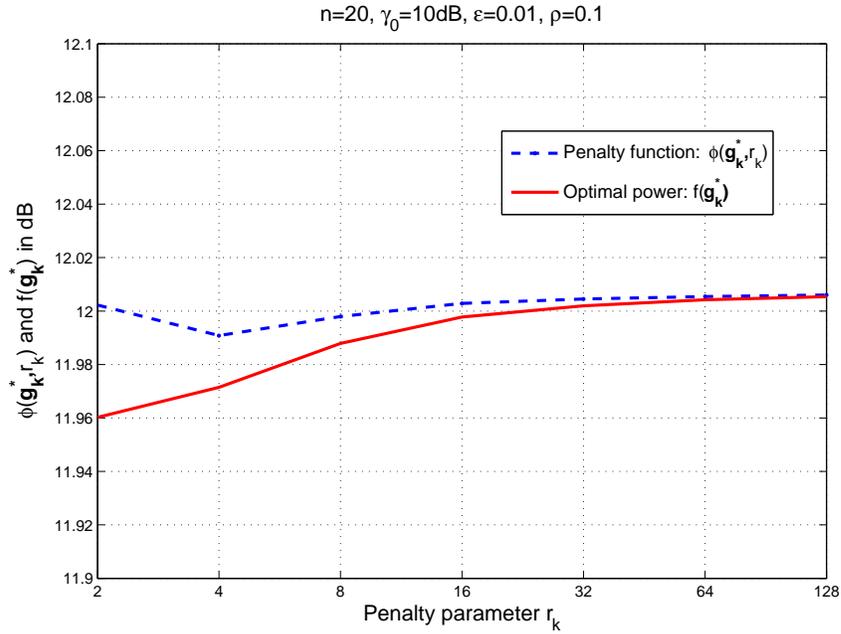


Figure 3.8: Convergence of exterior penalty function based PSO: Convergence of penalty function to the original optimization problem. (Fusion error probability = 0.01)

It can be seen that the fusion performance characteristics with respect to n and γ_0 for the correlated observations are similar to that with the i.i.d. observations. From Fig. 3.10, it can be seen that the network needs to spend more power to achieve the same performance level as the correlation coefficient of the observations increases. This is because, as the correlation between the observations is increased, the new information added by each additional sensor decreases (or the diversity on the observations is reduced) resulting in degraded fusion performance.

Figure 3.11 shows the results obtained from the constrained PSO algorithm for different noise covariance models. In the noise covariance matrix in model 1, the off-diagonal elements above the main diagonal (or below the main diagonal) are generated according to a uniform distribution on $[0,1]$. Model 2 refers to the noise covariance matrix $\Sigma_{\mathbf{v}}$ such that $(\Sigma_{\mathbf{v}})_{i,j} = \sigma_v^2 \rho$ for $i \neq j$ and $(\Sigma_{\mathbf{v}})_{i,j} = \sigma_v^2$ for $i = j$.

Table 3.2: Comparison of analytical and numerical results when $\rho = 0$, $\gamma_0 = 10dB$, $\epsilon = 0.01$, $n = 10$

\mathbf{g}^* : Analytical ($\rho = 0$)	[1.6172, 1.5888, 1.5555, 1.4666, 1.4616, 1.4107, 1.1231, 0, 0, 0]
\mathbf{g}^* : Numerical ($\rho = 0$)	[1.6163, 1.5696, 1.5548, 1.5014, 1.4501, 1.4099, 1.1212, 0.0013, 0.0066, 0.0008]
\mathbf{g}^* : Numerical ($\rho = 0.1$)	[1.6717, 1.5867, 1.6112, 1.5034, 1.5285, 1.4758, 1.3381, 0.3366, 0.0062, 0.0005]

Model 3 refers to (3.15) and Model 4 is its tri-diagonal version. $\rho = 0.1$ for models 2, 3 and 4. As observed earlier, for small ρ we may approximate model 3 by model 4. As in model 2, if the observation correlation is the same among all the sensors then the system needs more power to achieve the same performance level compared to models 3 and 4 in which the correlations decrease as separation between sensors increases. It can also be seen that when the correlation coefficients are randomly

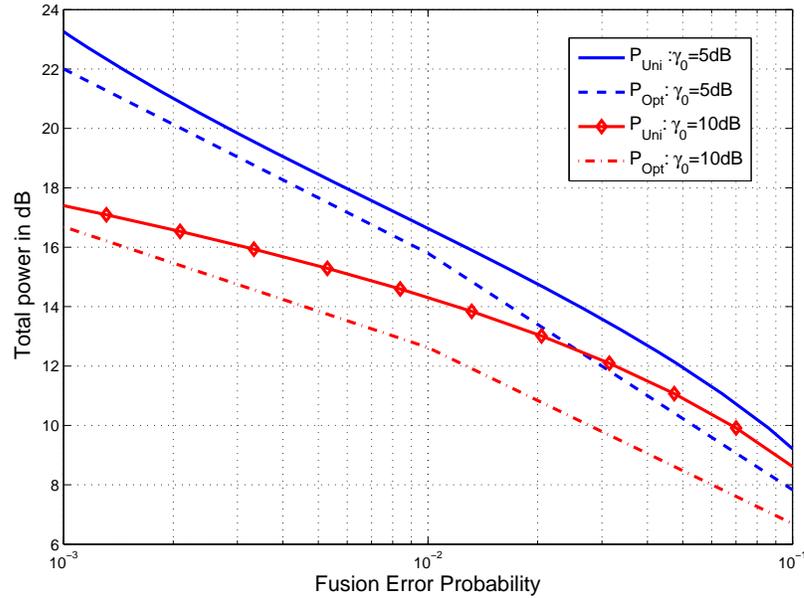


Figure 3.9: PSO: Total power Vs. fusion error probability when observations are correlated; $n = 20$, $\rho = 0.1$

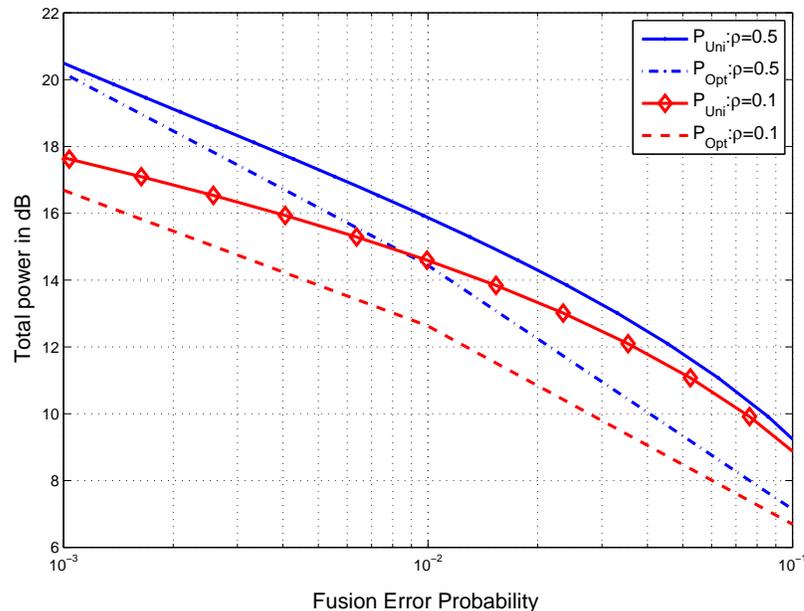


Figure 3.10: PSO: Total power Vs. fusion error probability when observations are correlated; $n = 20$, $\gamma_0 = 10dB$

selected between 0 and 1 as in model 1, the required power is significantly higher than that of other noise covariance models considered with small ρ values.

In the next experiment (Fig. 3.12) we analyze the performance loss caused by the independent assumption when the observations are actually correlated. For the results in Fig. 3.12, we assume two noise correlation models as defined by model 1 (random-uniform correlation model) and model 3 (Gauss-Markov model as given by (3.15)). In model 3, we evaluate the performance for different values of ρ . In Fig. 3.12 two dashed line plots are corresponding to the model 1 and all the solid line plots are corresponding to the noise model 3 with different ρ values. With correlation model 1, if the power allocation is done assuming independent observations to reach a desired fusion performance, it can be seen that a severe performance degradation will result. Thus, with random correlation models, it is more desirable to allocate

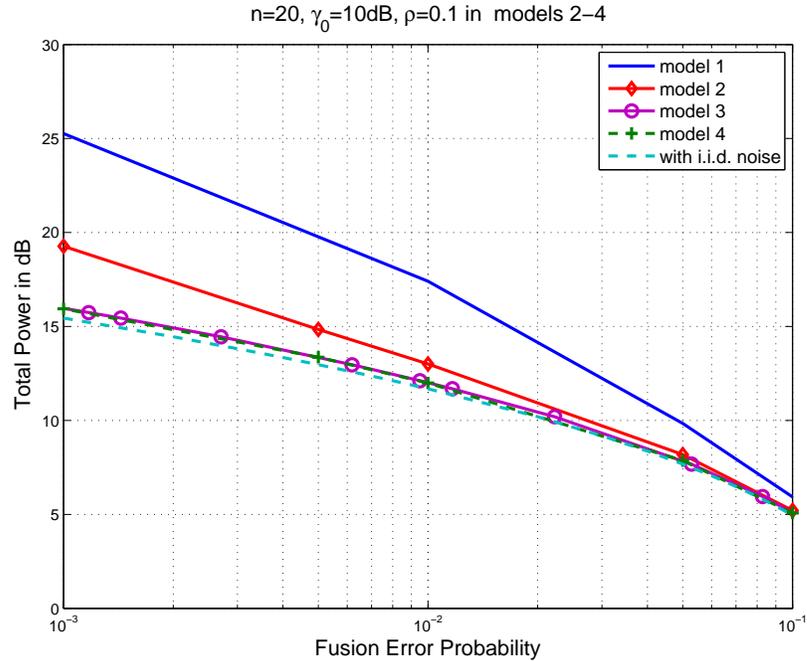


Figure 3.11: PSO: Total power Vs. fusion error probability for different noise covariance models; in models 2 – 4, $\rho=0.1$

power according to PSO-based optimal scheme rather than depending on the i.i.d. assumption for the local observations. With the noise Model 3, which may be more realistic in practice in many applications, it can be seen from Fig. 3.12 that for large ρ values the performance degradation on the independent assumption is severe compared to that with lower ρ values. Thus, for large correlation values (i.e. when the nodes are placed so densely) the independent assumption for local observations may will cause significant performance penalty. However, when ρ is small, as can be seen in Fig. 3.12 the assumption of conditional independence might not lead to severe performance penalties, although actual observations are correlated.

In the next experiment, we analyze the robustness of the proposed power allocation scheme when the estimation error of the channel fading coefficient is varied. So far we have assumed that transmitting nodes and the fusion center have the

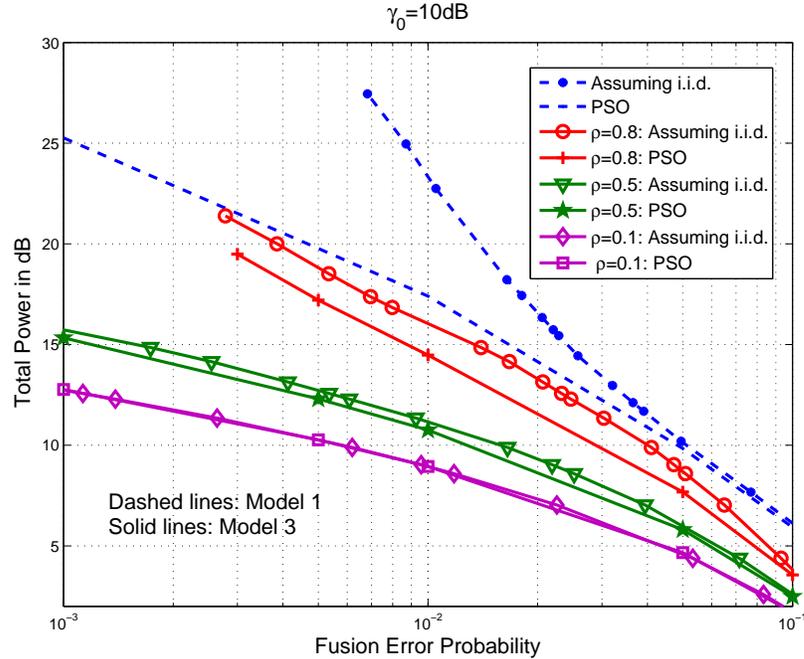


Figure 3.12: PSO based results Vs. results obtained assuming i.i.d. noise for noise models 1 and 3, $n = 20$ and $\gamma_0 = 10dB$

knowledge of exact channel fading coefficients. In practice, the fusion center has only estimates \hat{h}_k 's of channel coefficients. Let us assume that $\hat{h}_k = h_k + \delta_k$ where estimation error $\delta_k \sim \mathcal{N}(0, \sigma_\delta^2)$ and σ_δ^2 is the estimation error variance. The affect of the estimation error on the optimal power allocation scheme is shown in Fig. 3.13 with different σ_δ values. It can be seen that the proposed optimal power allocation scheme is robust for relatively small estimation errors.

3.7 Conclusions

In this Chapter we addressed the problem of optimal power scheduling for sensor nodes while meeting a desired fusion error probability, for data fusion in a wireless

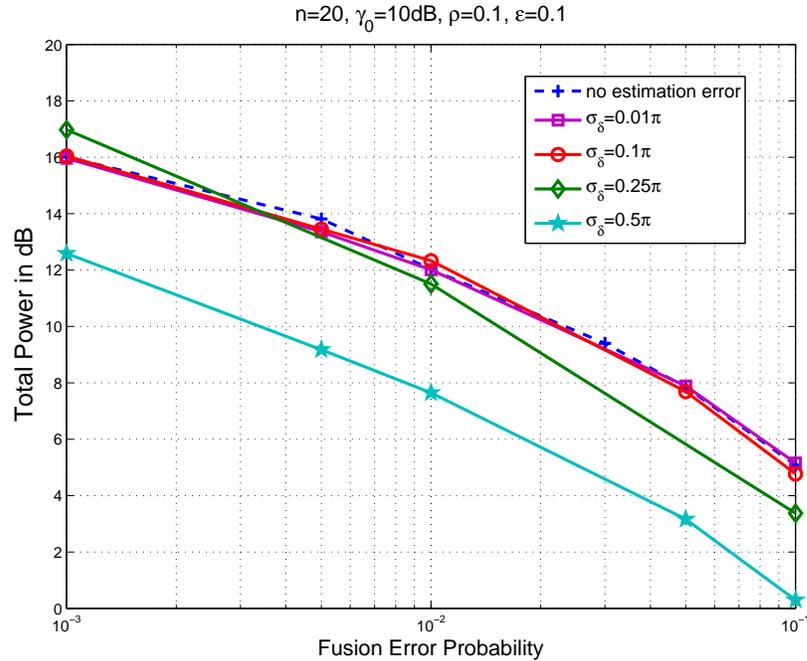


Figure 3.13: PSO: Total power Vs. fusion error probability for with the estimation error of the fading coefficients at the fusion center. $n = 20, \gamma_0 = 10\text{dB}, \rho = 0.1, \epsilon = 0.1$

sensor network with i.i.d. as well as correlated observations. When observations are i.i.d., we derived the optimal power allocation scheme analytically. For correlated observations, we derived an easy to optimize upper bound for the fusion error probability that is valid for sufficiently small observation correlations. When the observations are arbitrary correlated, we proposed an evolutionary computation technique based on PSO to evaluate the optimal power levels at sensor nodes. We showed that according to the optimal power allocation strategy the sensors with poor observations and/or bad channel qualities must be turned off to save the total power spent by the network. Moreover, when the local observation quality is very good it is sufficient to collect data from only a small number of sensors out of the total available nodes in the network (keeping others turned off). In the case of i.i.d. observations,

and with correlated observations with the derived upper bound, the derived optimal power scheduling schemes can be implemented distributively with only a small feedback from the fusion center. From numerical results based on constrained-PSO, we observed that the optimal power allocation scheme provides significant total energy savings over that of the uniform power allocation scheme especially when the number of nodes in the system is large or when the local observation quality is good. Also the PSO based method has significantly better performance compared to power allocation schemes assuming independent observations for relatively large observation correlations.

3.8 Appendix 3A

Uniqueness of K_1 : In the following, we show the existence of a unique K_1 , where $1 \leq K_1 \leq n$ such that $f(K_1) < 1$ and $f(K_1 + 1) \geq 1$ where $f(k) = \frac{(k - \beta^2 \sigma_v^2)}{h_k \sum_{j=1}^k \frac{1}{h_j}}$, $1 \leq k \leq n$ and we have assumed $h_1 \geq h_2 \geq \dots \geq h_n$. When $k = 1$, $f(1) = \frac{(1 - \beta^2 \sigma_v^2)}{h_1 \frac{1}{h_1}} < 1$. So, $f(k) > 1$ is not possible for all $k = 1, 2, \dots, n$. Therefore there are two possibilities: (I). $f(k) < 1$ for all $1 \leq k \leq n$: In this case we set $K_1 = n$. (II). There exists a unique K_1 such that $f(K_1) < 1$ and $f(K_1 + 1) \geq 1$, where $1 \leq K_1 \leq n$.

The uniqueness of K_1 implies that for any $k \geq K_1 + 1$, we should have $f(k) \geq 1$. This can be proved by showing that if $f(k) \geq 1$, then $f(k + 1) \geq 1$. When $f(k) \geq 1$, it implies that

$$f(k + 1) = \frac{(k - \beta^2 \sigma_v^2) + 1}{(h_k \sum_{j=1}^k \frac{1}{h_j} + 1) + (h_{k+1} - h_k) \sum_{j=1}^k \frac{1}{h_j}} \quad (3.31)$$

The second term of the denominator of (3.31) is negative or equal to zero since we have assumed that $h_{k+1} \leq h_k$. Hence $f(k + 1) \geq \frac{(k - \beta^2 \sigma_v^2) + 1}{h_k \sum_{j=1}^k \frac{1}{h_j} + 1} > 1$ as required.

Uniqueness of the minimum of (3.20): The uniqueness follows from the fact that, (3.22) is the only solution that satisfies the KKT conditions of the problem (3.20). Remaining is to show that the optimal solution (3.22) corresponds to a global minimum. To prove that, we will show that the Hessian matrix of the Lagrangian (3.21) is positive definite at the optimal solution. It can be seen that the Hessian matrix (H) of (3.21) is diagonal with $H_{k,k} = 2 + 2\lambda_0 h_k^2 \sigma_w^2 \frac{(3g_k^2 h_k^2 \sigma_v^2 - \sigma_w^2)}{(g_k^2 h_k^2 \sigma_v^2 + \sigma_w^2)^3}$ for $k = 1, 2, \dots, n$. As in (3.22), when $n > \beta^2 \sigma_v^2$ and $f(k) - 1 < 0$, the optimal power at k -th node is given by, $g_k^2 = \frac{\sigma_w^2}{h_k^2 \sigma_v^2} \left[\frac{h_k \sum_{j=1}^{K_1} \frac{1}{h_j}}{(K_1 - \beta^2 \sigma_v^2)} - 1 \right]$. Then $H_{k,k} = 2 \left[1 - \left(4 \frac{h_{K_1}}{h_k} f(K_1) - 3 \right) \right] > 0$, since $f(K_1) < 1$ and $h_{K_1} \leq h_k$. When $n > \beta^2 \sigma_v^2$ and $f(k) - 1 > 0$, optimal $g_k^2 = 0$ and then $H_{k,k} = 2 \left[1 - \left(\frac{h_k}{h_{K_1} f(K_1)} \right)^2 \right] > 0$, since then $f(K_1) > 1$ (that is $k \geq K_1$) and therefore $h_k \leq h_{K_1}$. Thus $H_{k,k} > 0$ for $k = 1, 2, \dots, n$ implying H is a positive definite matrix.

Chapter 4

Power Management for Estimation with Correlated Observations

4.1 Introduction

In Chapter 3, distributed detection by a WSN in the presence of correlated observations was considered. Distributed estimation of static as well as dynamic parameters by a set of distributed sensor nodes and a fusion center is another important topic in signal processing for sensor networks. In most distributed estimation approaches, it is assumed that the sensor nodes transmit their observations to the fusion center over a set of orthogonal channels. However, the use of bandlimited channels has been attracted considerable attention in the context of wireless sensor networks since the available bandwidth of the system is fixed regardless of the number of nodes.

Use of bandlimited channels in WSNs has been considered by recent research. In [94] the estimation over Type-Based Multiple Access (TBMA) was considered where each node transmits its observations using certain signaling in a shared channel. They have shown that TBMA is asymptotically optimal in the limit of large number

of nodes if the channel gains are identical. Power efficient distributed estimation of a random parameter over bandlimited channels was considered in [147]. The asymptotic performance analysis based on non-orthogonal channels for distributed detection was addressed in [60], [3].

In addition to limited bandwidth, as discussed in Chapter 3 for distributed detection, an important issue to be considered in WSNs is node power, since sensor nodes are usually equipped with small size batteries that can be expensive and/or difficult to replace. A considerable work has been done on power constrained WSNs for distributed estimation in the literature, to name a few [69, 72, 147, 148]. In [69] the minimum energy decentralized estimation with correlated data was addressed. They have exploited the knowledge of noise covariance matrix to select quantization levels at nodes and minimum power was derived accordingly to meet a target MSE. In [148], the optimal power scheduling scheme meeting a required target MSE at the fusion center (with independent observations) was considered assuming quantized decisions at local nodes. It was also shown that optimal power scheduling scheme improves the mean squared error performance by a large margin compared to that achieved by an uniform power allocation scheme. In [147], the same problem was addressed with AF processing at local nodes. Energy constrained distributed estimation is addressed in [72].

However, again the effect of the correlated observations is a crucial issue in practical sensor networks for distributed estimation. Effect of dependent noise in the estimation accuracy is presented in [118], for two different noise covariance models. The minimum energy decentralized estimation with correlated data was addressed in [69]. They exploited the knowledge of the noise covariance matrix to select quantization levels at nodes that minimized the power, while meeting a target mean-squared error.

In this Chapter, our contribution is on the estimation of a non-random param-

eter over bandlimited channels with correlated observations. Each node is assigned a signaling waveform (or code) which corresponds to DS-CDMA. We consider the cases where signaling waveforms are orthogonal, equi-correlated and perfectly correlated (multiple access channel-MAC). Assuming perfect synchronization in sensor transmissions, first we analyze the asymptotic MSE performance for correlated observations with equal power at nodes and identical channel gains. Next, we derive the optimal power allocation schemes for the communication with orthogonal and MAC to achieve a required MSE performance at the fusion center. It is shown that the optimal power scheduling scheme for MAC has a better performance over that of the orthogonal channels. We also discuss the effect of the synchronization errors on the MAC estimation performance.

The remainder of this Chapter is organized as follows. Section 4.2 presents the sensor network model and formulates the estimation problem. MSE performance at the fusion center is presented in Section 4.3. In Section 4.4, the asymptotic MSE performance is analyzed for correlated observations assuming equal power at sensor nodes and identical channel gains. Assuming channels undergo fading, the optimal power allocation schemes for orthogonal and MAC communication are presented in Section 4.5. In Section 4.6, the effect of the synchronization errors in sensor transmissions on estimation performance is discussed. The conclusions of this work are given in Section 4.7.

4.2 Sensor Network Model

Consider a WSN with n spatially separated sensor nodes. Each sensor has a measurement z_k of a non-random parameter θ :

$$z_k = \theta + v_k; \quad k = 1, 2, \dots, n$$

where v_k 's are assumed to be zero mean correlated additive noise with covariance matrix Σ_v . We assume that θ has a finite range so that its average energy is finite. Let us define the local signal-to-noise ratio $\gamma_0 = \frac{P_s}{\sigma_v^2}$ where P_s is the average power of the parameter to be estimated and σ_v^2 is the noise variance of each v_k . Each node performs AF processing on its observation with a gain of g_k .

4.3 MSE Performance with Non-Orthogonal Communication

In general, for non-orthogonal communication between nodes and the fusion center, k -th node is assigned a signaling waveform \mathbf{s}_k normalized such that $\mathbf{s}_k^T \mathbf{s}_k = 1$, for $k = 1, \dots, n$. The number of degrees of freedom in the signaling waveform is assumed to be N so that \mathbf{s}_k is a length N vector for $k = 1, \dots, n$. Then the transmitted signal u_k at each sensor node is given by $u_k = g_k z_k \mathbf{s}_k$ where g_k is the amplifier gain at the k -th node. Note that, we assume that each amplify-and forward (AF) local processing at each node (as in Chapter 3). A sufficient statistic for the estimation of θ at the fusion center is given by the output of a bank of n filters matched to the signalling waveforms \mathbf{s}_k 's. Assuming perfect synchronization in sensor transmissions, the matched filter output is given by [130],

$$\mathbf{y} = \mathbf{R}\mathbf{A}\mathbf{z} + \mathbf{w} \tag{4.1}$$

where \mathbf{R} is the code cross correlation matrix, $\mathbf{A} = \text{diag}(h_1 g_1, \dots, h_n g_n)$ where h_k 's are the channel fading coefficients and \mathbf{w} is the filtered Gaussian noise vector distributed as $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma_w^2 \mathbf{R})$ where σ_w^2 is the receiver noise power at the fusion center. In this Chapter we assume that \mathbf{R} has the following form which is a common as-

sumption in practice: $\mathbf{R} = \begin{bmatrix} 1 & \rho & \cdot & \cdot & \cdot & \rho & \rho \\ \rho & 1 & \cdot & \cdot & \cdot & \rho & \rho \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \rho & \rho & \cdot & \cdot & \cdot & \rho & 1 \end{bmatrix}$ where $|\rho| \leq 1$. The Best Linear

Unbiased Estimator (BLUE) at the fusion center based on matched filter output \mathbf{y} can be shown to be (which is the same as MVUE when the noise is Gaussian),

$$\hat{\theta}(\mathbf{y}) = \frac{\mathbf{e}^T \mathbf{A} \mathbf{R} \Sigma_{\mathbf{n}}^{-1} \mathbf{y}}{\mathbf{e}^T \mathbf{A} \mathbf{R} \Sigma_{\mathbf{n}}^{-1} \mathbf{R} \mathbf{A} \mathbf{e}},$$

where $\Sigma_{\mathbf{n}} = \mathbf{R} \mathbf{A} \Sigma_{\mathbf{v}} \mathbf{A} \mathbf{R} + \sigma_w^2 \mathbf{R}$ and \mathbf{e} is the n -length vector with all ones. The resulting MSE is given by,

$$\text{MSE}(\hat{\theta}) = (\mathbf{e}^T \mathbf{A} \mathbf{R} \Sigma_{\mathbf{n}}^{-1} \mathbf{R} \mathbf{A} \mathbf{e})^{-1}. \quad (4.2)$$

4.4 Asymptotic MSE Performance with Non-Orthogonal Communication

For asymptotic analysis, we assume that each node has the same amplification factor g and identical channel gains, $h_k = 1$ for $k = 1, \dots, n$. Then MSE in (4.2), using matrix inversion lemma, can be shown to be

$$\text{MSE}(\hat{\theta}) = \frac{1}{\frac{g^2}{\sigma_w^2} \left(\mathbf{e}^T \mathbf{R} \mathbf{e} - \frac{g^2}{\sigma_w^2} \mathbf{e}^T \left[(\mathbf{R} \Sigma_{\mathbf{v}} \mathbf{R})^{-1} + \frac{g^2}{\sigma_w^2} \mathbf{R}^{-1} \right]^{-1} \mathbf{e} \right)}.$$

Let us denote, $\mathbf{Z}_n = \left((\mathbf{R}\boldsymbol{\Sigma}_v\mathbf{R})^{-1} + \frac{g^2}{\sigma_w^2}\mathbf{R}^{-1} \right)$. Further, let us assume the noise covariance matrix $\boldsymbol{\Sigma}_v$ has the Gauss-Markov model, so that

$$\boldsymbol{\Sigma}_v = \sigma_v^2 \begin{bmatrix} 1 & \rho_d & \cdot & \cdot & \cdot & \rho_d^{n-2} & \rho_d^{n-1} \\ \rho_d & 1 & \cdot & \cdot & \cdot & \rho_d^{n-3} & \rho_d^{n-2} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \rho_d^{n-1} & \rho_d^{n-2} & \cdot & \cdot & \cdot & \rho_d & 1 \end{bmatrix} \quad (4.3)$$

where $|\rho_d| \leq 1$. It is easy to see that with noise covariance matrix (4.3), \mathbf{Z}_n becomes a circulant matrix for sufficiently large n . Since the inverse of a circulant matrix is also circulant, \mathbf{Z}_n^{-1} is a circulant matrix. It can be shown that for large n [50],

$$\mathbf{e}^T \mathbf{Z}_n^{-1} \mathbf{e} = n \lambda_{\mathbf{Z}^{-1}, M} \quad (4.4)$$

where $\lambda_{\mathbf{Z}^{-1}, M}$ is the largest eigenvalue of \mathbf{Z}_n^{-1} . By using eigenvalue decomposition (EVD) and exploiting the fact that all circulant matrices have same eigenvectors, we have $\mathbf{Z}_n^{-1} = U \left[\Lambda_{\mathbf{R}}^{-1} \Lambda_v^{-1} \Lambda_{\mathbf{R}}^{-1} + \frac{g^2}{\sigma_w^2} \Lambda_{\mathbf{R}}^{-1} \right]^{-1} U^*$ where $\Lambda_{\mathbf{R}}$ and Λ_v are diagonal matrices of eigenvalues of \mathbf{R} and $\boldsymbol{\Sigma}_v$ respectively. U is a unitary matrix where columns of U contain eigenvectors of an $n \times n$ circulant matrix. The m -th eigenvalue of \mathbf{Z}_n^{-1} is given by $\lambda_{\mathbf{Z}_n^{-1}, m} = \frac{\sigma_w^2 \lambda_{\mathbf{R}, m}^2 \lambda_{v, m}}{\sigma_w^2 + g^2 \lambda_{\mathbf{R}, m} \lambda_{v, m}}$. Now, (4.4) becomes,

$$\mathbf{e}^T \mathbf{Z}_n^{-1} \mathbf{e} = n \frac{\sigma_w^2 \lambda_{\mathbf{R}, M}^2 \lambda_{v, M}}{\sigma_w^2 + g^2 \lambda_{\mathbf{R}, M} \lambda_{v, M}}$$

where $\lambda_{\mathbf{R}, M}$ and $\lambda_{v, M}$ are maximum eigenvalues of \mathbf{R} and $\boldsymbol{\Sigma}_v$ respectively. It can be shown that [3, 50], for large n , $\lambda_{\mathbf{R}, M}$ equals to $(1 + \rho(n - 1))$ for $0 \leq \rho \leq 1$ and $1 - \rho$ for $-1 \leq \rho < 0$ respectively and $\lambda_{v, M} = \frac{\sigma_v^2(1+|\rho_d|)}{(1-|\rho_d|)}$ for $|\rho_d| < 1$. Then MSE asymptotically is given by,

$$\text{MSE}(\hat{\theta}) = \frac{(1 - \rho_d)\sigma_w^2 + g^2\sigma_v^2(1 + \rho_d)(1 - \rho + (n\rho)^+)}{ng^2(1 - \rho_d)(1 - \rho + (n\rho)^+)}$$

where $(x)^+$ equals 0 for $x < 0$, and otherwise equals to x .

4.4.1 $\rho=0$: Orthogonal communication

For orthogonal channels, the cross correlation between codes is $\rho = 0$. Then the asymptotic MSE is given by,

$$\text{MSE}(\hat{\theta}) = \frac{(1 - \rho_d)\sigma_w^2 + g^2\sigma_v^2(1 + \rho_d)}{ng^2(1 - \rho_d)}. \quad (4.5)$$

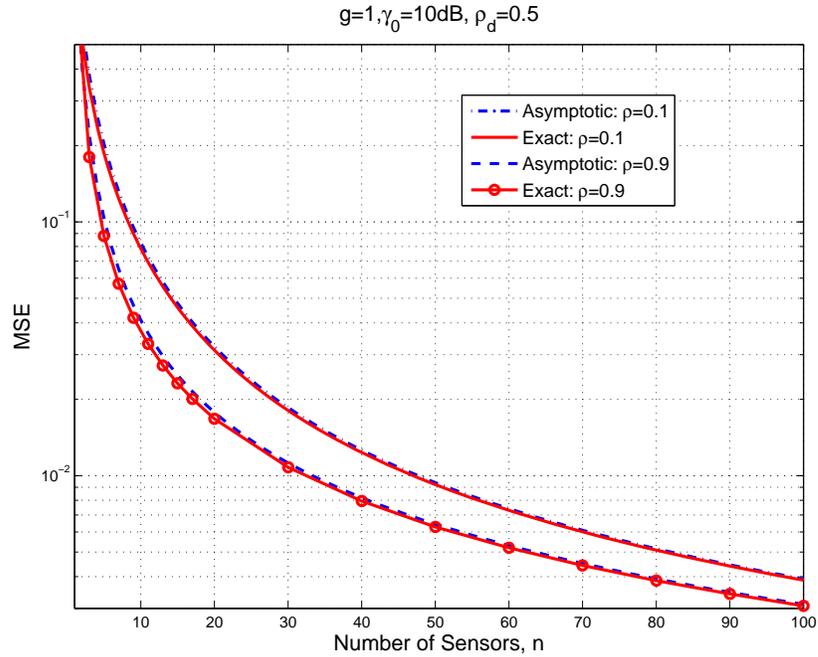
4.4.2 $\rho=1$: Perfect correlation between codes

When $\rho = 1$, we call the communication is over MAC. In this case, each node uses same signalling code.

Then the MSE asymptotically is given by,

$$\text{MSE}(\hat{\theta}) = \frac{(1 - \rho_d)\sigma_w^2 + ng^2\sigma_v^2(1 + \rho_d)}{n^2g^2(1 - \rho_d)}. \quad (4.6)$$

It is clear from (4.5) and (4.6), that the use of non-orthogonal channels improves the MSE performance. Figure 4.1 shows the derived asymptotic MSE performance and the exact MSE as a function of n for a given ρ_d . It can be seen that the derived asymptotic expression for MSE is a good approximation for the exact MSE even with relatively small n . The figure also shows that the MSE performance is improved by increasing code cross correlation. This is because, with AF local processing and non-orthogonal channels, the distributed sensor system tends to act as a cooperative beam-former. For $\rho = 1$, the system has a perfectly directed beam towards the fusion center that exploits the full coherent gain. In contrast, when $\rho = 0$, a set of orthogonal channels are used for sending information regarding the same estimator and does not have the cooperative beam-forming gain. In Fig. 4.2 the dependence of MSE on local observation correlation ρ_d is shown. It is observed that when observation correlation is larger, the MSE performance is degraded. This is because

Figure 4.1: MSE as a function of number of sensors n

of the fact that the new information added by the additional sensor nodes decreases as the correlation increases. However, it is shown that by increasing the code cross correlation ρ , a better performance can be achieved even when the observations are highly correlated.

4.5 Optimal Power Allocation in Fading Channels

In the following we assume the channels between sensor nodes and the fusion center undergo fading. The objective is to allocate the node power in an optimal way such that the minimum power is spent by the network to achieve a desired MSE

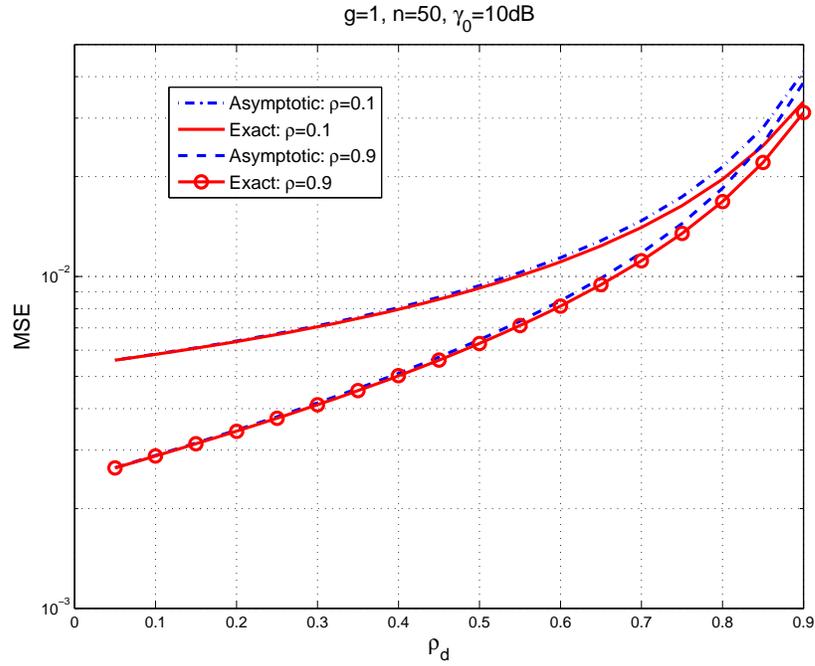


Figure 4.2: The dependence of MSE on local observation correlation parameter ρ_d

performance at the fusion center. The optimization problem can be formulated as

$$\min_{g_k \geq 0, k=1, \dots, n} \sum_{k=1}^n g_k^2 \quad \text{such that } \text{MSE}(\hat{\theta}) \leq D_0 \quad (4.7)$$

where D_0 is the required MSE threshold at the fusion center.

4.5.1 Orthogonal communication and i.i.d observations

When the observations are i.i.d., $\Sigma_v = \sigma_v^2 \mathbf{I}$. Since $\rho = 0$, $\mathbf{R} = \mathbf{I}$. Then MSE in (4.2) becomes $\text{MSE}(\hat{\theta}) = \left(\sum_{k=1}^n \frac{h_k^2 g_k^2}{\sigma_v^2 h_k^2 g_k^2 + \sigma_w^2} \right)^{-1}$. Letting $D = \frac{1}{D_0}$, the optimization problem

(4.7) becomes

$$\min_{g_k \geq 0, k=1, \dots, n} \sum_{k=1}^n g_k^2 \text{ such that}$$

$$D - \sum_{k=1}^n \frac{h_k^2 g_k^2}{h_k^2 g_k^2 \sigma_v^2 + \sigma_w^2} \leq 0 \quad (4.8)$$

The optimal solution g_k^{*2} for (4.8) can be shown to be,

$$g_k^{*2} = \begin{cases} \frac{\sigma_w^2}{h_k^2 \sigma_v^2} \left[\frac{h_k \sum_{j=1}^{K_1} \frac{1}{h_j}}{(K_1 - D\sigma_v^2)} - 1 \right] & ; \text{ if } f(k) - 1 < 0 \text{ and } n > D\sigma_v^2 \\ 0 & ; \text{ if } f(k) - 1 > 0 \text{ and } n > D\sigma_v^2 \\ \text{infeasible} & ; \text{ if } n < D\sigma_v^2 \end{cases} \quad (4.9)$$

where assuming, without loss of generality, $h_1 \geq h_2 \geq \dots \geq h_n$, $f(k) = \frac{(k - D\sigma_v^2)}{h_k \sum_{j=1}^k \frac{1}{h_j}}$, $1 \leq k \leq n$ and K_1 is found such that $f(K_1) < 1$ and $f(K_1 + 1) \geq 1$ for $1 \leq K_1 \leq n$. Note that letting $\sqrt{\delta_0} = \sigma_w \frac{\sum_{k=1}^{K_1} \frac{1}{h_k}}{K_1 - D\sigma_v^2}$, for $f(k) - 1 < 0$ and $n > D\sigma_v^2$, the optimal g_k^{*2} can be written as, $g_k^{*2} = \frac{\sigma_w^2}{h_k^2 \sigma_v^2} \left(\frac{h_k \sqrt{\delta_0}}{\sigma_w} - 1 \right)$. Hence, assuming channel state information (CSI) is available at sensor nodes, once the fusion center broadcasts $\sqrt{\delta_0}$, each node can determine its power using $\sqrt{\delta_0}$ as a side information.

4.5.2 MAC and i.i.d. observations

When $\rho = 1$ (MAC), the MAC output is simplified to,

$$y = \sum_{k=1}^n h_k g_k z_k + w, \quad (4.10)$$

where $w \sim \mathcal{N}(0, \sigma_w^2)$. The MSE estimator and the corresponding MSE are then given by,

$$\hat{\theta}_{BLUE}(y) = \frac{y}{\mathbf{e}^T \mathbf{A} \mathbf{e}} = \frac{y}{\sum_{k=1}^n h_k g_k} \quad (4.11)$$

and

$$\text{MSE}(\hat{\theta}) = \frac{\mathbf{e}^T \mathbf{A} \Sigma_v \mathbf{A} \mathbf{e} + \sigma_w^2}{(\mathbf{e}^T \mathbf{A} \mathbf{e})^2} \quad (4.12)$$

With i.i.d. observations, the MSE in (4.12) reduces to,

$$\text{MSE}(\hat{\theta}) = \frac{\sigma_v^2 \sum_{k=1}^n h_k^2 g_k^2 + \sigma_w^2}{(\sum_{k=1}^n h_k g_k)^2}. \quad (4.13)$$

Since $\text{MSE}(\hat{\theta})$ in (4.13) is not convex over g_k 's a variable transformation as in [147] is done to obtain a convex programming problem for (4.7). Let $q_k = h_k g_k$ for $k = 1, 2, \dots, n$ and $s = \sum_{k=1}^n q_k$. Then $g_k = \frac{q_k}{h_k}$ and the optimization problem becomes,

$$\begin{aligned} & \min_{q_1, \dots, q_n; s} \sum_{k=1}^n \frac{q_k^2}{h_k^2} \text{ such that} \\ & \sum_{k=1}^n q_k^2 + \frac{\sigma_w^2}{\sigma_v^2} \leq d s^2 \text{ and } s = q_1 + \dots + q_n, \end{aligned}$$

where $d = \frac{D_0}{\sigma_v^2}$.

By solving the above optimization problem, the optimal g_k^{*2} can be shown to be,

$$g_k^{*2} = \frac{\mu^2}{4} \frac{h_k^2}{(1 + \lambda_0 h_k^2)^2}, \quad k = 1, 2, \dots, n \quad (4.14)$$

where λ_0 can be found numerically by solving the equation $\sum_{k=1}^n \frac{\lambda_0 h_k^2}{(1 + \lambda_0 h_k^2)} = \frac{1}{d}$ and μ is given by, $\mu = 2 \frac{\sigma_w}{\sigma_v} \left(\frac{1}{\lambda_0^2 d} - \sum_{k=1}^n \frac{h_k^4}{(1 + \lambda_0 h_k^2)^2} \right)^{-\frac{1}{2}}$. The optimal total power spent is $P_{total} = \sum_{k=1}^n g_k^{*2} = \frac{\sigma_w^2}{\sigma_v^2} \lambda_0$. From (4.14), it can be seen that the optimal power

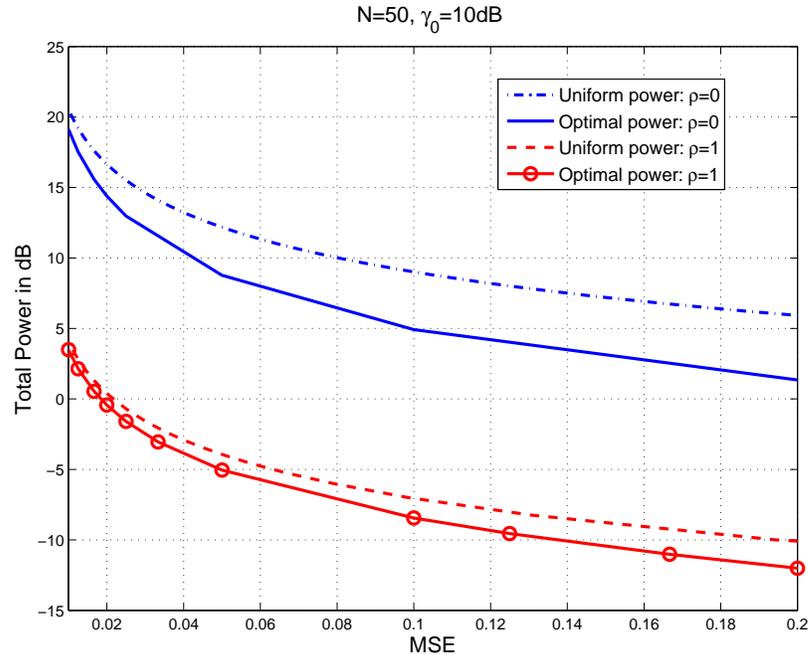


Figure 4.3: Comparison of optimal power and uniform power for $\rho = 0$ and $\rho = 1$

has a distributed structure with λ_0 and μ as side information from the fusion center assuming CSI is available at the transmitter.

However, assuming that the channel fading coefficients are Rayleigh distributed with a parameter σ , for large n , λ_0 and μ are given by, solutions to

$$\frac{n}{2\sigma^2} \left(\frac{1}{\lambda_0} e^{\frac{1}{2\sigma^2\lambda_0}} \text{Ei}\left(-\frac{1}{2\sigma^2\lambda_0}\right) + 2\sigma^2 \right) = \frac{1}{d} \quad (4.15)$$

and

$$\mu = 2 \frac{\sigma_w}{\sigma_v} \left(\frac{1}{\lambda_0^2 d} - \left[\frac{n}{\lambda_0^3} \left(\lambda_0 + \frac{1}{2\sigma^2} \right) + \frac{n}{2\sigma^2 \lambda_0^4} e^{\frac{1}{2\sigma^2\lambda_0}} \text{Ei}\left(-\frac{1}{2\sigma^2\lambda_0}\right) \left(2\lambda_0 + \frac{1}{2\sigma^2} \right) \right] \right) \quad (4.16)$$

where $\text{Ei}(\cdot)$ is the exponential integral defined as $\text{Ei}(x) = -\int_{-x}^{\infty} \frac{e^{-t}}{t} dt$. The performance of the optimal power allocation schemes derived in sections 4.5.1 and 4.5.2 are

shown in Fig. 4.3. As observed in Section 4.4, it is seen that the MSE performance is improved as ρ increases. Also it is observed that the derived optimal power allocation scheme has a better performance compared to the uniform power allocation scheme especially when the number of sensor nodes in the system is large and/or the required MSE is not significantly small.

It is noted from Sections 4.5.1 and 4.5.2 that for orthogonal communication ($\rho = 0$), it is optimal to activate the sensor nodes with good channel quality and high local SNR while turning off the sensor nodes with poor channel and local SNR quality. However, MAC ($\rho = 1$), it is optimal to combine all the observations irrespective of the channel and the local SNR quality. This is because, for $\rho = 1$, the system has a perfectly directed beam towards the fusion center that exploits a n factor of coherent gain when there are n sensor nodes in the network. Therefore, for $\rho = 1$, in the optimal power allocation scheme, all the sensor nodes are active to exploit the full coherent gain at the fusion center in contrast with $\rho = 0$ case where there is no cooperative beamforming gain.

4.5.3 MAC and correlated observations

With correlated observations, the MSE is given by (4.12). Since when the observations are correlated, it is difficult to obtain an analytical closed form solution for the optimal power allocation problem (4.7), using the fact that the Rayleigh quotient of a Hermitian matrix is upper bounded by its maximum eigenvalue, we find the following upper bound for the MSE (4.12),

$$\text{MSE}_B(\hat{\theta}) = \frac{\lambda_M \sum_{k=1}^n h_k^2 g_k^2 + \sigma_w^2}{(\sum_{k=1}^n h_k g_k)^2} \quad (4.17)$$

where λ_M is the maximum eigenvalue of $\Sigma_{\mathbf{v}}$. Now the optimal power allocation scheme is found to keep the MSE bound under a desired threshold D_0 . Following a

similar procedure as in Section 4.5.2, the optimal power can be shown to be

$$g_k^2 = \frac{\mu'^2}{4} \frac{h_k^2}{(1 + \lambda_0' \lambda_M h_k^2)^2}, \quad k = 1, \dots, n \quad (4.18)$$

where λ_0' is found by solving the expression $\lambda_0' \sum_{k=1}^n \frac{h_k^2}{(1 + \lambda_0' \lambda_M h_k^2)} = \frac{1}{D_0}$ and $\mu' = 2\sigma_w \left(\frac{1}{\lambda_0'^2 D_0} - \lambda_M \sum_{k=1}^n \frac{h_k^4}{(1 + \lambda_0' \lambda_M h_k^2)} \right)^{-1/2}$. The performance of the power allocation

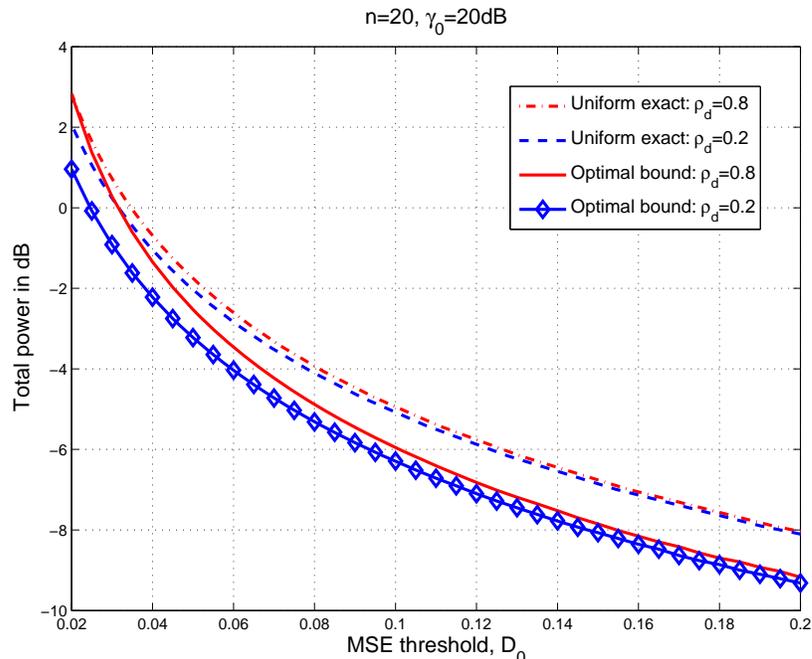


Figure 4.4: The performance of the power allocation scheme based on MSE bound for $\rho = 1$ and correlated observations. Total power vs. MSE, $n = 20$, $\gamma_0 = 20dB$

scheme based on the MSE bound is shown in Figs. 4.4 and 4.5. From Fig. 4.4 and 4.5 it can be seen that the optimal power allocation scheme based on the MSE bound significantly outperforms the uniform power allocation scheme based on exact MSE when the number of sensor nodes in the network n is large or the observation correlation coefficient ρ_d is relatively small or the local SNR quality γ_0 is moderate and high. However, it is seen from Fig. 4.5 that for large ρ_d , when n and γ_0 is small,

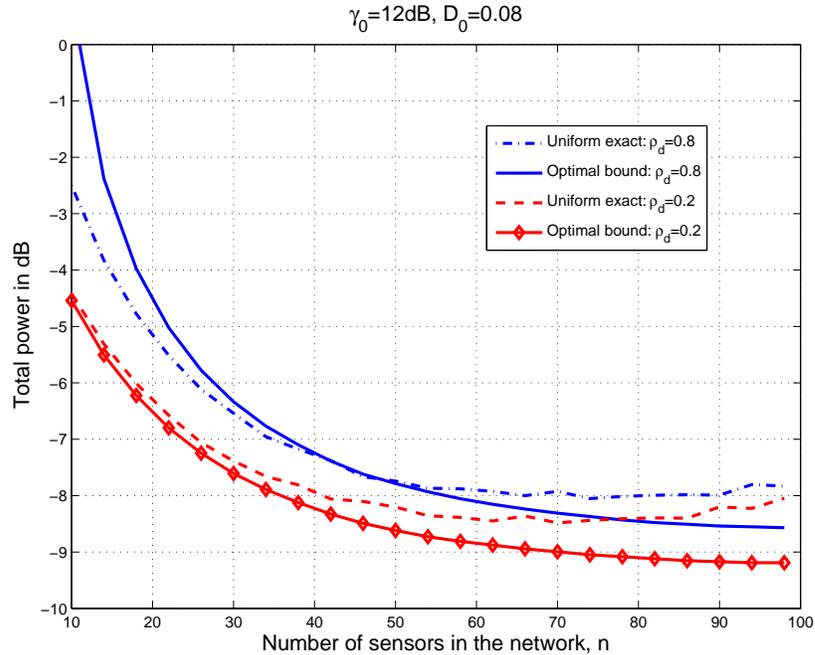


Figure 4.5: The performance of the power allocation scheme based on MSE bound for $\rho = 1$ and correlated observations. Total power vs. number of sensor nodes, $\gamma_0 = 12dB$, $D_0 = 0.08$

the power allocation scheme based on the MSE bound does not perform well. In those cases, the uniform power allocation scheme based on exact MSE provides less total power consumption.

4.6 Synchronization in Sensor Transmissions and the Effect of Synchronization Error on MSE Performance

An important assumption that has been made in the above analysis in MAC communication is perfect synchronization of sensor transmissions. In practice, achieving

perfect synchronization among nodes might be a difficult task. In this section we discuss a strategy for achieving synchronization and consider the impact of synchronization errors on the MSE performance. For the analysis given below we assume a network model with i.i.d. observations and $\rho = 1$.

We follow a similar strategy as described in [11] to achieve synchronization in the sensor network. We assume that there is a master-node which broadcasts the carrier and timing signals to the rest of the sensor nodes (slave nodes). Then there are $(n-1)$ slave nodes, each at distance $d_k + \delta_k$ from the master node for $k = 1, 2, \dots, n-1$ where d_k and δ_k are the nominal distance and the sensor placement error of the k -th node, respectively. The master node broadcasts a carrier signal $\cos(2\pi f_0 t)$ where f_0 is the carrier frequency. The received carrier signal at the k -th slave node is a noisy version of $\cos(2\pi f_0 t + \psi_k + \psi_{ek})$ where $\psi_k = \frac{2\pi f_0 d_k}{c}$ and $\psi_{ek} = \frac{2\pi f_0 \delta_k}{c}$. Each slave node employs a Phase Locked Loop (PLL) to lock onto the carrier. If each slave node precompensates for the difference in their nominal distances d_k , to the master node, by transmitting its modulated and locally processed observation with a proper delay and phase shift ψ_k , then the received signal at the fusion center is corrupted by the timing error and the phase error due to the sensor placement error δ_k . Considering only the phase error due to sensor placement error, the matched filter output at the fusion center is given by $y = \sum_{k=1}^n h_k g_k z_k \cos(\psi_{ek}) + w$. To analyze the effect of phase error due to sensor placement error, we assume that the placement error δ_k is distributed as Gaussian with zero mean and the variance σ_δ^2 which is much smaller than the wavelength λ_0 . Then the phase error $\psi_{ek} \sim \mathcal{N}(0, \sigma_\psi^2)$ and we assume that σ_ψ^2 is small. To obtain the BLUE estimator, we take the expectation of y with respect to both z_k and ψ_{ek} . i.e. $E(y) = \theta e^{-\frac{\sigma_\psi^2}{2}} \sum_{k=1}^n h_k g_k$ assuming the observation noise w_k is i.i.d.. Then the BLUE estimator is $\hat{\theta}_{BLUE}(y) = \frac{y}{e^{-\frac{\sigma_\psi^2}{2}} \sum_{k=1}^n h_k g_k}$ and the resulting

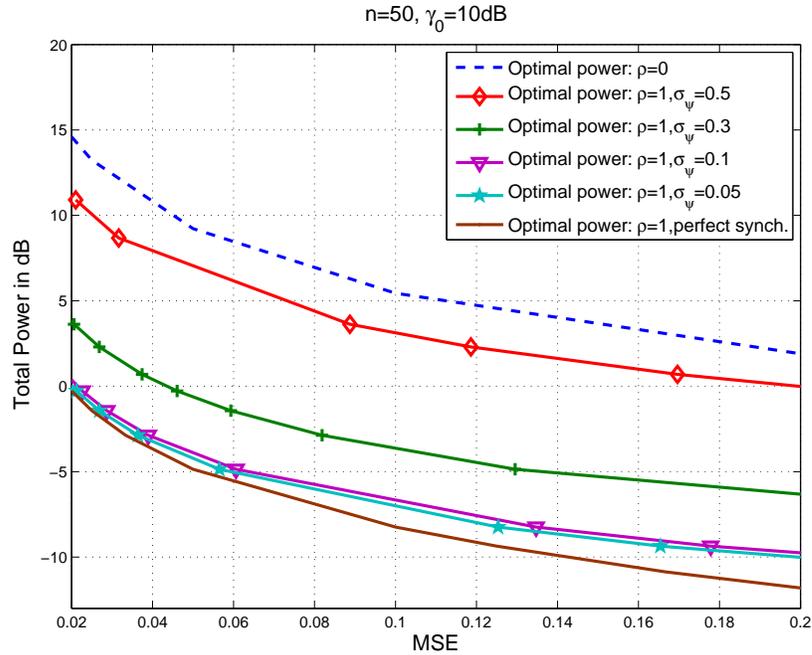


Figure 4.6: The effect of the synchronization error for the MSE performance

MSE with the phase error is given by,

$$\begin{aligned} \text{MSE}'(\hat{\theta}) &= \frac{e^{-\sigma_\psi^2} \sigma_v^2 \sum_{k=1}^n h_k^2 g_k^2 + \sigma_w^2}{e^{-\sigma_\psi^2} (\sum_{k=1}^n h_k g_k)^2} \\ &= \frac{\sigma_v^2 \sum_{k=1}^n h_k^2 g_k^2 + e^{\sigma_\psi^2} \sigma_w^2}{(\sum_{k=1}^n h_k g_k)^2} \end{aligned}$$

which is greater than the MSE with perfect synchronization in (4.13), showing that the synchronization error causes a degradation of MSE performance at the fusion center. Figure 4.6 shows the effect of the synchronization error on the MSE performance for i.i.d. observations. It can be seen that when the variance of the phase error σ_ψ^2 is significantly small, the affect of the synchronization error on the MSE performance with MAC ($\rho = 1$), is small. Even for relatively large σ_ψ^2 , the use of MAC gives significant performance compared to that of orthogonal channels ($\rho = 0$).

4.7 Conclusions

The distributed estimation of a non-random parameter in a bandlimited channel with AF processing at local nodes is addressed in this Chapter. We consider in general, correlated observations. First, assuming equal power and identical channel gains, asymptotic performance of MSE was analyzed for correlated observations. It was shown that the performance based on the derived asymptotic expression closely matches with the exact MSE performance even for relatively small network sizes. It was also shown that the use of non-orthogonal channels results significant performance over that of the orthogonal channels.

Next, assuming fading channels between sensor nodes and the fusion center, we derived the optimal power allocation schemes with both orthogonal and MAC channels while keeping the required MSE at the fusion center under a given threshold. In the case of i.i.d. observations, it was shown that the derived optimal power allocation scheme has a distributed implementation with a limited feedback from the fusion center. Also it was shown that the optimal power allocation schemes with both MAC and orthogonal channels have better performance over corresponding uniform power allocation schemes. For correlated observations with $\rho = 1$, the power allocation scheme was found analytically using the derived bound for the MSE. It was shown that the optimal power allocation scheme based on the MSE bound has a significant performance over the uniform power allocation scheme based on the exact MSE when n is large, γ_0 is high and for relatively small observation correlation coefficient ρ_d .

When the communication between the sensor nodes and the fusion center is non-orthogonal, the coherent gain achieved above is based on the assumption that the sensor transmissions are perfectly synchronized. We also discussed the synchronization of the sensor transmissions and the effect of synchronization errors on the MSE performance. It was shown that, for relatively small synchronization errors, the

performance of the power allocation scheme for MAC channels does not have a significant degradation and it is still better than that of using orthogonal channels. Also it gives an insight on deciding the level of tolerance of the sensor placement errors within which the multiple-access communication has better performance over the orthogonal communication.

Chapter 5

Distributed Node Selection for Noisy Sequential Estimation

5.1 Introduction

In typical sensor network applications considered in literature (as discussed in Chapters 3 and 4), it is assumed that the spatially separated sensor nodes send their locally processed information to a fusion center to obtain the final decision [7, 82, 94]. However, as discussed in Chapter 2, a major problem of such a centralized approach is the large power consumption for communication. The reliability of such architecture depends on the robustness of the fusion center. In some sensor network applications, it is required that any distributed node has the ability to form the final decision or the estimator by collaborating locally with other nodes in the network reducing the large communication burden as with centralized schemes. Distributed sequential estimation, in which nodes update the local estimators sequentially, is one way of achieving collaborative estimation without depending on a central fusion center. Such schemes are more robust against failures compared to centralized schemes. Since all nodes in

the network may not have useful information regarding the Phenomenon of Interest (PoI), it is required to select nodes which carry useful information for the estimation process. This ensures that only nodes which contribute to the final decision need to be in active in the decision process while others may remain idle preserving their transmit energy.

The distributed sequential estimation problem was formulated in [34, 154, 156]. According to [34, 154], a lead node sequentially queries the sensor nodes and updates its estimator (based on the posterior distribution of the state of the PoI) until a desired performance level is reached. In these schemes, the lead node has to keep track of all nodes which have been participated in the decision process at each processing step. In [156], the posterior distribution (belief) at the current node is transmitted to the next node where it updates the state of belief based on the current belief and the new measurement at that node. Note that in this scheme, if the belief (posterior distribution) cannot be represented by a parameterizable distribution, grid samples of the distribution should be transmitted to the next node leading to a considerable communication burden. However, when the belief cannot be represented by a standard parameterizable distribution, [156] proposed to approximate the belief by a parameterizable distribution and the corresponding parameters are transmitted to the next node. The communication complexity in transmitting belief then is determined by the number of parameters and their dimensions. To find the next best node in the sequential estimation process, several information utility measures based on entropy and the network geometry were proposed in [34]. In [161] a node selection algorithm for target tracking based on the posterior Cramer-Rao Lower Bound (CRLB) was presented. However, neither of these work considered the noise in inter-node communication links. In [63], the sequential estimation of a non-random parameter over noisy correlated channels was considered. However, it did not consider the best ordering of the nodes for the sequential processing.

In this Chapter, we consider the problem of distributed sequential estimation of a random parameter in which the updated estimate of a node is sent to the next node via a noisy channel. In the proposed schemes, each node in the decision process needs to transmit only two parameters, namely the updated estimator and the corresponding minimum mean squared error (MMSE). We propose two greedy algorithms to find the ordering of nodes for the estimation process based on a reward function that reflects the trade-off between an information utility measure and the communication cost between nodes. Two schemes are different from each other in terms of the search space; global search or a local search. We propose to use mutual information as the information utility measure and investigate the use of MMSE of the estimator as an alternative when it is difficult to compute the mutual information. Note that, the mutual information utility measure selects the node that provides the maximum amount of new information regarding the PoI as the next processing node, given the current estimate. In the global search based scheme, we assume that any two nodes in the network can communicate with each other and the next node is searched over all possible unvisited nodes to maximize the relevant objective function. In the scheme based on local search, on the other hand, we assume that each node has a set of neighbors that it can communicate at an affordable communication cost. Candidate next nodes at each node are allowed to be selected only from these neighbors. Information utility measures of the candidate nodes are computed according to the current node's information and the knowledge of sensor positions of neighbor nodes and target positions. In the proposed scheme based on local search, each node has to keep track of only its neighbors to determine which nodes have been participated in the decision process, while in the scheme with global search, each node has to keep track of all unvisited nodes in the whole network. From simulation results we see that the performance of the proposed scheme with local approach becomes closer to that with the global approach after processing a relatively small number of nodes. Also when the two proposed node selection

schemes are used an improved performance with a smaller number of processing nodes can be achieved compared to the nearest node selection method. However, it should be noted that, with proposed node selection schemes a global optimal MMSE solution is not guaranteed since they are greedy-type algorithms. We compare the two proposed schemes with the global optimal solution obtained via optimal shortest path algorithms (forward dynamic programming) and show that the performances of both proposed schemes are close to that with the optimal scheme with a relatively small number of processing nodes.

The remainder of this Chapter is organized as follows: Section 5.2 presents the sensor network model. Section 5.3 formulates the distributed sequential estimation problem over noisy communication channels and derives the estimator performance. The proposed distributed node selection schemes based on an information utility measure and the inter-node communication cost are discussed in Section 5.4. Section 5.5 discusses the performance results and concluding remarks are given in Section 5.6.

5.2 Sensor Network Model

Consider a spatially distributed, sensor network consisting of n number of nodes. Denote by s_k the k -th node, for $k = 1, \dots, n$. Note that, when there is no ambiguity, we use s_k and k to denote the k -th processing node interchangeably. The network is deployed to estimate the signal amplitude emitted by a possible target (e.g. a sound source) based on the following observation model at node s_k :

$$z'_k = \frac{\theta}{\|\mathbf{x}_k - \mathbf{x}_t\|^{\alpha/2}} + v'_k, \text{ for } k = 1, \dots, n, \quad (5.1)$$

where θ is the parameter to be estimated (target amplitude) that is assumed to be Gaussian with zero mean and variance σ_θ^2 , \mathbf{x}_k and \mathbf{x}_t denote the positions of sensor

node s_k and the target, respectively, v'_k is the measurement noise that is assumed to be white Gaussian with zero mean and variance σ_0^2 and α is the path loss exponent that is determined by the propagation environment. This model can be used, for example, in applications in which acoustic sensors are used to estimate the amplitude of sound signals emitted by a target [34, 96]. By rearranging (5.1), we can re-write the observation at node s_k in the equivalent form of

$$z_k = \theta + v_k, \text{ for } k = 1, \dots, n, \quad (5.2)$$

where now v_k is assumed to be independent but not identically distributed. In particular, v_k is Gaussian with mean zero and variance $\sigma_k^2 \propto \|\mathbf{x}_k - \mathbf{x}_t\|^\alpha$.

5.3 MMSE Performance

The idea is to estimate the parameter θ sequentially via inter node communication. Let s_1 be the starting node of the sequential estimation process. The starting node estimates the parameter based on its own observation, $z_1 = \theta + v_1$. Assuming that the parameter θ is independent of observation noise v_1 , the optimal Minimum Mean Squared Error (MMSE) estimate at node s_1 based on z_1 is given by

$$\hat{\theta}_1(z_1) = \frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma_1^2} z_1, \quad (5.3)$$

and the corresponding MMSE, denoted by M_1 , of the estimator (5.3) is

$$M_1 = \frac{\sigma_1^2 \sigma_\theta^2}{\sigma_1^2 + \sigma_\theta^2} = \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_\theta^2} \right)^{-1}. \quad (5.4)$$

Equivalently, (5.3) can be expressed as $\hat{\theta}_1(z_1) = \frac{M_1}{\sigma_1^2} z_1$.

The sequential estimation process is continued until either the desired performance level is reached or observations at all nodes are processed. When $k = 1$, if the MMSE M_1 does not meet the desired performance, the estimator $\hat{\theta}_1$ is transmitted to the next node, selected based on a certain criteria, over a noisy channel. The criteria for selection of next node is discussed in a later section. For $k > 1$, the k -th node estimates the parameter θ based on its own observation and the received estimator from the $(k - 1)$ -th node. The effective observation vector at node s_k (for $k > 1$) is

$$\mathbf{z}_k = \begin{bmatrix} z_k \\ q_k \end{bmatrix} = \begin{bmatrix} \theta + v_k \\ \hat{\theta}_{k-1} + n_k \end{bmatrix}, \text{ for } k = 2, \dots, n,$$

where q_k is the noise corrupted decision from node s_{k-1} . The channel noise n_k , from node s_{k-1} to node s_k is assumed to be independent Gaussian with mean zero and variance $\sigma_{c(k-1,k)}^2 \propto \|\mathbf{x}_k - \mathbf{x}_{k-1}\|^{\alpha'}$ for $k = 2, \dots, n$ where α' is the path loss index of communication channels between nodes. The MMSE estimator at node s_k can thus be shown as,

$$\hat{\theta}_k(z_k, q_k) = \frac{M_k}{\sigma_k^2} z_k + \frac{M_k(\sigma_\theta^2 - M_{k-1})}{M_{k-1}(\sigma_\theta^2 - M_{k-1}) + \sigma_\theta^2 \sigma_{c(k-1,k)}^2} q_k, \quad (5.5)$$

where M_k is the MMSE at the s_k -th node that can be shown to be

$$M_k = \frac{\sigma_\theta^2}{\sigma_\theta^2 d_k^2 + 1}, \quad (5.6)$$

where $d_k^2 = \frac{1}{\sigma_k^2} + \frac{(\sigma_\theta^2 - M_{k-1})^2}{\sigma_\theta^2 [M_{k-1}(\sigma_\theta^2 - M_{k-1}) + \sigma_\theta^2 \sigma_{c(k-1,k)}^2]}$ and M_{k-1} is the MMSE at the node s_{k-1} that is assumed to be available at node s_k . Note that the MMSE at the s_k -th node is determined only by statistics of observations and channel noise and it is reasonable to assume that they can be made available at neighbors [63]. From (5.5), it can be seen that the MMSE estimator at node s_k is determined by its own observation, information from the node s_{k-1} and the channel noise quality.

Since MMSE M_k in (5.6) depends on the channel noise of inter-node communication links, it is interesting to examine the behavior of M_k with respect to the corresponding channel quality. We consider following two extremes: Channel quality is good such that $\sigma_{c(k-1,k)}^2 \rightarrow 0$ and channel quality is poor such that $\sigma_{c(k-1,k)}^2 \rightarrow \infty$ for $k = 2, 3, \dots, n$. In the first case, we have

$$\lim_{\sigma_{c(k-1,k)}^2 \rightarrow 0} M_k = \frac{M_{k-1}}{1 + \frac{M_{k-1}}{\sigma_k^2}}, \text{ for } k = 2, 3, \dots, n.$$

Therefore, it is seen that when $\sigma_{c(k-1,k)}^2 \rightarrow 0$, $M_k \leq M_{k-1}$ for all k . That is, by sending the node s_{k-1} 's decision to the node s_k always improves the MMSE performance at node s_k . On the other hand, if inter-node communication channel quality is poor, we have

$$\lim_{\sigma_{c(k-1,k)}^2 \rightarrow \infty} M_k = \frac{\sigma_\theta^2 \sigma_k^2}{\sigma_\theta^2 + \sigma_k^2}, \text{ for } k = 2, \dots, n. \quad (5.7)$$

That is, when the quality of inter-node communication link is poor, the performance at node s_k does not depend on the decision at node s_{k-1} , but is entirely determined by the observation quality at node s_k . It implies that there will be a certain threshold value for channel quality of inter-node communication links which ensures that $M_k \leq M_{k-1}$ for $k = 2, 3, \dots, n$. Indeed, it can be shown that if $\sigma_{c(k-1,k)}^2$ satisfies the following inequality for $k = 2, 3, \dots, n$

$$\sigma_{c(k-1,k)}^2 \leq \frac{M_{k-1}^2 (\sigma_\theta^2 - M_{k-1})}{\sigma_k^2 (\sigma_\theta^2 - M_{k-1}) - M_{k-1} \sigma_\theta^2}, \quad (5.8)$$

then $M_k \leq M_{k-1}$; i.e. sending the decision at node s_{k-1} to node s_k improves the MMSE performance at s_k . This is further discussed in Section 5.5.

If we assume that the node observations are i.i.d and the inter-node communication is noiseless such that $\sigma_k^2 = \sigma_0^2$ and $\sigma_{c(k-1,k)}^2 = 0$ for $k = 2, 3, \dots, n$, it can be

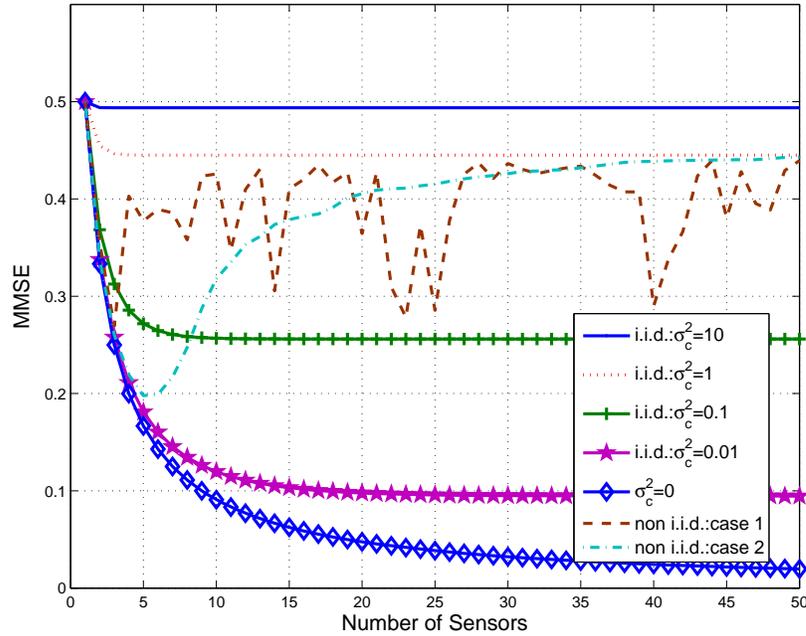


Figure 5.1: MMSE vs. number of sensors when observation noise is i.i.d.

shown that the MMSE at node s_k (5.6) reduces to $M_k = \frac{\sigma_\theta^2 \sigma_0^2}{\sigma_v^2 + k \sigma_\theta^2}$, which is a monotonically decreasing function of k . It is also interesting to see that in this case the minimum number of nodes n_{min} required to achieve a required MMSE performance level ϵ is given by, $n_{min} = \sigma_0^2 \left(\frac{1}{\epsilon} - \frac{1}{\sigma_\theta^2} \right)$.

In the case of i.i.d. observation noise such that $\sigma_k^2 = \sigma_0^2$ for all k , Fig. 5.1 shows the MMSE performance of the sequential estimation process with different channel noise qualities of inter-node communication links. In Fig. 5.1, we have let $\sigma_0^2 = 1$ and $\sigma_\theta^2 = 1$. In the special case when channel noise is also i.i.d. such that $\sigma_{c(k-1,k)}^2 = \sigma_c^2$ for all k , from Fig. 5.1 it can be seen that $M_k \leq M_{k-1}$ holds for all k . Moreover, as expected from (5.7) the MMSE performance converges to 0.5 as σ_c^2 increases. It is expected that when both observations and channel noise are i.i.d., the performance of the MMSE estimator is independent of the order of the processing nodes. Figure

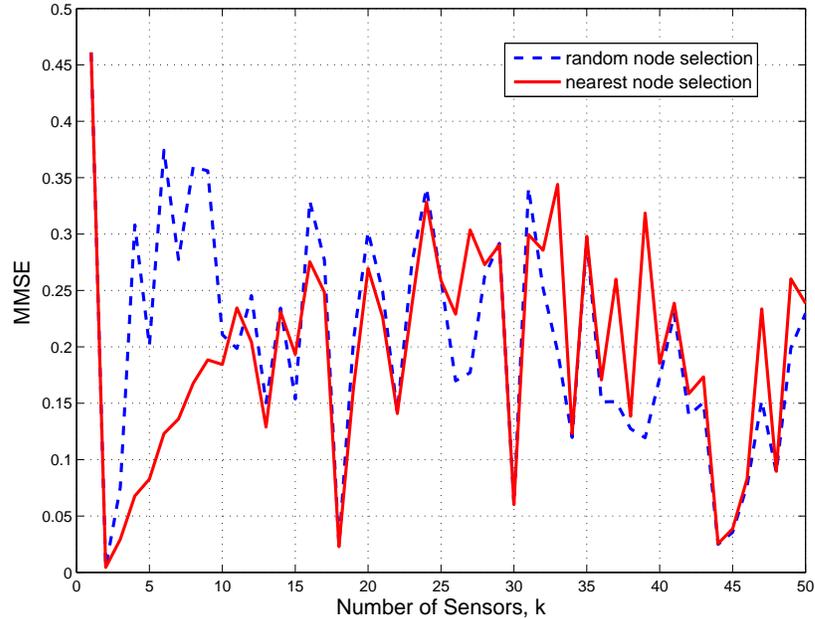


Figure 5.2: MMSE vs number of sensors when observation noise is non-i.i.d.

5.1 also shows the performance of the MMSE estimator when channel noise is not identical (still the observation noise is i.i.d.). We have considered two cases: In the first, $\sigma_{c(k-1,k)}^2$'s are drawn randomly from a uniform distribution in $[0, 1]$ without any order. In the second case, these random $\sigma_{c(k-1,k)}^2$'s are arranged in an ascending order. From Fig. 5.1 it can be seen that whenever the condition (5.8) is satisfied at node k , $M_k \leq M_{k-1}$. In this case, to find the node where the minimum MMSE is achieved, the process should be continued for all nodes. On the other hand, in case 2, where nodes are selected with minimum distance from the current node, we observe that after a certain node the MMSE starts to monotonically increase. Therefore, it is enough to continue the sequential estimation process only until this specific node, thereby, saving the network power.

Figure 5.2 shows the MMSE performance of the sequential distributed estimation

process with non-identical observations and channel noise. Dashed line corresponds to channel noise variance drawn from a uniform distribution without any order while the solid line corresponds to channel noise variance in ascending order with k . In both cases, the observation noise variances are drawn from a uniform distribution on $[0, 1]$. As can be observed from Fig. 5.2, when observations are not i.i.d., just selecting the nearest node as the next node does not always improve the performance. Therefore, when observations are not identical, it is required to have an information driven approach to select the nodes with higher information gain as well as lower communication cost.

5.4 Sensor Node Selection

When a sensor network is deployed to estimate a PoI as discussed above, all nodes in the network might not carry the same amount of useful information regarding the PoI. Thus it is of interest to find the optimal sequence of nodes which contains the minimum number of nodes required to reach a desired performance level. Denote s_k to be the k -th processing node, for $k = 1, 2, \dots, n$ and s_1 to be the initial node in the sequential estimation process, as before. Let ϵ be the desired MMSE performance level and $\hat{\mathcal{S}}_k = \{\hat{s}_1, \hat{s}_2, \dots, \hat{s}_k\} \subseteq \{s_1, s_2, \dots, s_n\}$ be a sequence of distinct nodes with $\hat{s}_1 = s_1$. Then the optimal node ordering problem can be formulated as,

$$\begin{aligned} & \min |\hat{\mathcal{S}}_k| \\ & \text{such that } M_k \leq \epsilon, \end{aligned} \tag{5.9}$$

where M_k is as defined in (5.6) and the minimization is over all possible distinct node sequences of length k (including ordering of the nodes), for $k = 1, \dots, n$, starting at node s_1 . If the relevant information regarding PoI (essentially the observation noise variance) and sensor positions at all nodes are available at node s_1 , to compute the

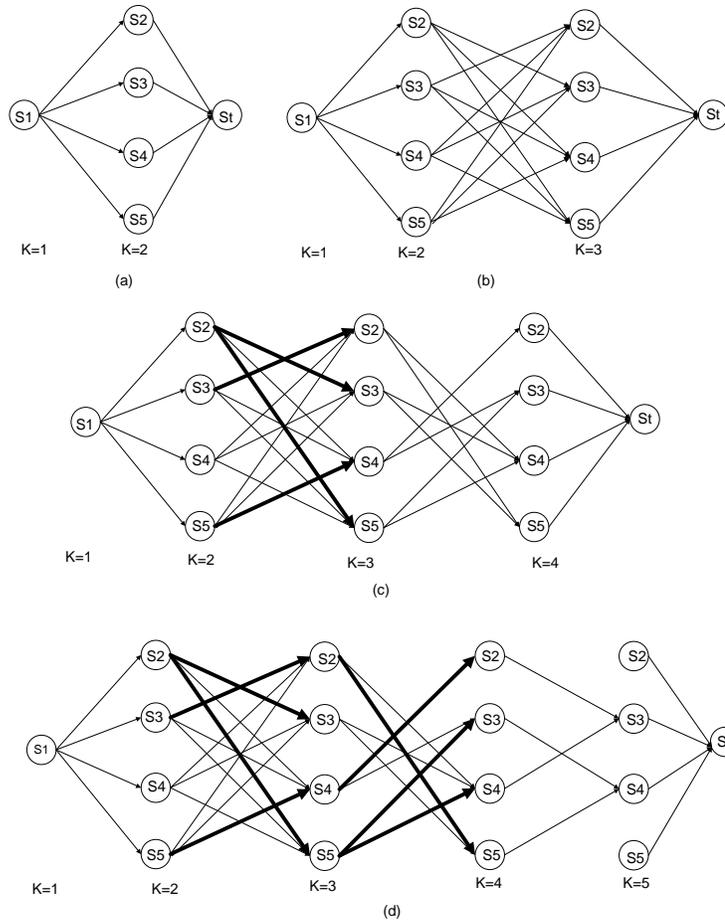


Figure 5.3: Illustration of the shortest path formulation of the optimization problem (5.9) for $n = 5$

optimal set $\hat{\mathcal{S}}_k$ that yields the global minimum, the optimization should be performed over all $\frac{n!}{(n-k)!}$ possible sequences for $k = 1, \dots, n$ at the worst case. It can be shown that this global optimal set $\hat{\mathcal{S}}_k$ can be found, at a worst case complexity order of $O(n^3)$ by converting the problem into an directed expansion graph, using shortest path algorithms.

5.4.1 Optimal scheme

For illustration, we assume that there is total of $n = 5$ sensor nodes, s_1, s_2, s_3, s_4, s_5 with s_1 being the starting node. Thus the sequential estimation process should be terminated by at most 5 steps and equivalently the graph needs to be expanded up to 5 steps. To find the optimal node sequence according to (5.9) based on shortest path algorithms, we construct the following expansion graph. We construct the trellis at each k for $k = 2, \dots, 5$, by concatenating of $k - 1$ copies of the state space where the state space consists of sensor nodes s_2, s_3, s_4, s_5 , as shown in Fig. 5.3. In Fig. 5.3, the dummy state s_1 represents the starting sensor node, at $k = 1$ while s_t is terminating dummy node. Each branch from node s_j to s_t for $j = 2, \dots, 5$, is assigned a metric 0 while branches s_1 to s_j for $j = 2, \dots, 5$ are assigned metric $M_{1,j}$ where $M_{1,j}$ is the MMSE at node s_j at time (step) $k = 2$, when the initial starting sensor node is s_1 . Metrics of the other branches when $k > 2$ in the trellis are assigned as described in the following and depends on the value of k . For $k = 2$, the goal is to find the node sequence of length 2, that would result the minimum MMSE at $k = 2$ (equivalently, by processing any two nodes). In this case, when the metric at each branch is assigned as just described, finding the node sequence which will result the minimum MMSE after processing any two nodes, is equivalent to finding the shortest path from node s_1 to s_t as shown in Fig. 5.3(a). If the minimum MMSE found at $k = 2$ (over all possible 2-length node sequences) does not meet the desired value, $k = 3$ is considered as shown in Fig. 5.3(b) where now the goal is to find 3-length node sequence which gives the minimum MMSE at $k = 3$. When computing the metrics of the branches from nodes from steps $k = 2$ to $k = 3$, it should be noted that each node can query only 3 $((n - 1) - 1)$ another nodes, since a node is not processed twice. Thus the branches between nodes from the step $k = 2$ to $k = 3$ are connected as shown in Fig. 5.3(b) where the metrics associated with branches are assigned as follows. A branch connecting s_j at step $k + 2$ to s_i at step $k + 3$ is assigned

the metric $M_{j,i}$ for $j, i = 2, \dots, 5, i \neq j$ where $M_{j,i}$ is the MMSE at node s_i at step $k + 3$ when the previous node is s_j at step $k + 2$ and is given by, (5.6). Then finding the 3-length node sequence which will result the minimum MMSE at step $k = 3$ is again equivalent to finding the shortest path between s_1 and s_t in Fig. 5.3(b). If the minimum MMSE at step $k = 3$ does not meet the desired value, the next step is started. Now the objective is to find the 4-length node sequence which will result the minimum MMSE at $k = 4$. Now, to define the metrics for branches between nodes from step $k = 3$ to $k = 4$, we consider all the shortest paths from s_1 to all nodes at step $k = 3$. For illustration purposes, let us assume that $\{s_1, s_3, s_2\}$, $\{s_1, s_2, s_3\}$, $\{s_1, s_5, s_4\}$, $\{s_1, s_2, s_5\}$ be the node sequences which result minimum MMSE if the third node at step $k = 3$ is s_2, s_3, s_4 and s_5 , respectively as shown in Fig.5.3(c) with dark lines. Now, when defining metrics for branches between nodes from step $k = 3$ to $k = 4$, these sequences which result the shortest path up to step $k = 3$ are taken in to account. For example, the node s_2 at $k = 3$ can connect to only nodes s_4 and s_5 at step $k = 4$ since, s_2 and s_3 can not be processed again. In that way, at step $k = 3$, any node can connect to only two $((n - 1) - 2)$ another nodes. By assigning the metrics for those branches as the relevant MMSE values, again the problem is equivalent to find the shortest path between nodes s_1 and s_t as shown in Fig. 5.3(c). By continuing the same process from step $k = 4$ to $k = 5$, it can be seen that any node at $k = 4$ can connect to only one $((n - 1) - 3)$ node at step $k = 5$. The shortest path problem at each step k that would lead the global optimum solution can be solved by efficient shortest path algorithms. By generalizing this scheme for n nodes, it can be shown that at the worst case (i.e. to consider all n -length sequences), the expansion graph (trellis) has at most $(n - 1)^2$ number of vertices and $\frac{1}{2}(n - 1)^2(n - 2)$ number of edges resulting the average worst case complexity of order $O(n^3)$, if the shortest path problem is solved based on Dijkstra's algorithm or forward dynamic programming to yield the optimal sequence of nodes that results the global minimum MMSE.

However, with large network size n , the above shortest path approach would be computationally complex. If the optimization were to be performed at the initial node, it requires also a high communication burden since all nodes have to forward their relevant information to the starting node s_1 . Thus, in the following we propose greedy sequential algorithms to find the best ordering of nodes to achieve a desired performance level where each node in the decision process determines its next best node as a trade-off between the information gain and the communication cost. Both proposed schemes can be implemented with reduced computational and communication complexities compared to the optimal scheme. We also show that the algorithms discussed below can lead to the exact or close to exact results to the optimal scheme (computed based on shortest path algorithms) under certain conditions. Details are given in Section 5.5.

5.4.2 Distributed node selection: global approach

In the following we determine the best ordering of nodes sequentially that would complete the estimation process by reaching at the desired performance level with a minimum number of processing nodes as a trade-off between the information gain and the communication cost. Let us denote by $\mathcal{V} = \{1, 2, \dots, n\}$ the set of nodes in the network. Let \mathcal{V}_j denote the set of nodes that have been participated in the sequential estimation process up to step j . Let $s_j \in \mathcal{V}$ be the selected processing node at step j . Then the next node s_{j+1} at step $(j + 1)$ is chosen as,

$$s_{j+1} = \operatorname{argmax}_{s_k \in \mathcal{V}_j^c} R(s_j, s_k) \quad (5.10)$$

where \mathcal{V}_j^c denotes the set complement of \mathcal{V}_j with respect to \mathcal{V} and the objective function $R(s_j, s_k)$ is defined as

$$R(s_j, s_k) = \beta R_I(\theta, z_k, q_{j,k}) - (1 - \beta) R_c(s_j, s_k), \quad (5.11)$$

where $R_I(\cdot)$ and R_c are the information utility function and a measure of communications cost, $\beta \in [0, 1]$ is a trade-off parameter that balances the contributions from the two terms in (5.11) and $q_{j,k} = \hat{\theta}_j + n_{j,k}$ is the received signal at node s_k if it is chosen to be the next node when the current node is s_j and $n_{j,k}$ is the channel noise between nodes s_j and s_k . The choice of β will depend on the required information gain and the tolerable communications cost. Note that in this scheme, when the current processing node is s_j , the next best node is selected from the set of unvisited nodes up to step j .

There are several possible information utility measures that can be used to quantify the information gain provided by a sensor measurement. For example, [34], [156] provided a detailed description of entropy- and geometry-based information utility measures. In this Chapter, we consider two measures for information utility: (1). the conditional mutual information $I(\theta; z_{j+1}|q_{j,j+1} = \hat{\theta}_j + n_{j,j+1})$ which provides the greatest amount of new information when the current estimate is $\hat{\theta}_j$ (2). the MMSE $M_{j+1|j}$ at the s_{j+1} -th node, when the current node is s_j , for $s_j, s_{j+1} \in \mathcal{V}$. We explore the use of MMSE as an alternate information measure in the cases where computation of mutual information is difficult. When mutual information is used as the information utility measure, the first term $R_I(\cdot)$ in (5.11) is given by

$$\begin{aligned}
 R_I(\theta, z_k, q_{j,k}) &= I(\theta; z_k | q_{j,k} = \hat{\theta}_j + n_{j,k}) \\
 &= h(\theta | \hat{\theta}_j + n_{j,k}) - h(\theta | \hat{\theta}_j + n_{j,k}, z_k) \\
 &= \frac{1}{2} \log \left[\frac{\sigma_k^2}{\sigma_k^2 - M_{j,k}} \right]
 \end{aligned} \tag{5.12}$$

where $h(\cdot)$ denotes the differential entropy [37] and $M_{j,k}$ is the MMSE at node s_k when current node is s_j . From (5.6), it can be shown that $M_{j,k} \leq \sigma_k^2$ for all $j, k \in \mathcal{V}$ so that (5.12) is valid for all $j, k \in \mathcal{V}$.

When MMSE is used as the information utility measure, we have

$$R_I(\theta, z_k, q_{j,k}) = -M_{j,k} = -\frac{\sigma_\theta^2}{\sigma_\theta^2 d_{j,k}^2 + 1} \quad (5.13)$$

where $d_{j,k}^2 = \frac{1}{\sigma_k^2} + \frac{(\sigma_\theta^2 - M_j)^2}{\sigma_\theta^2 [M_j(\sigma_\theta^2 - M_j) + \sigma_\theta^2 \sigma_{c(j,k)}^2]}$.

The communication cost function between current node s_j and the possible next node s_k is taken to be $R_c(s_j, s_k) = \frac{1}{d_{max}}(\mathbf{x}_j - \mathbf{x}_k)^T(\mathbf{x}_j - \mathbf{x}_k)$ where d_{max} is the maximum distance between any two nodes in the network. Then the composite objective function (5.11) can be written as,

$$R(s_j, s_k) = \beta R_I(\theta, z_k, q_{j,k}) - \frac{(1 - \beta)}{d_{max}}(\mathbf{x}_j - \mathbf{x}_k)^T(\mathbf{x}_j - \mathbf{x}_k), \quad (5.14)$$

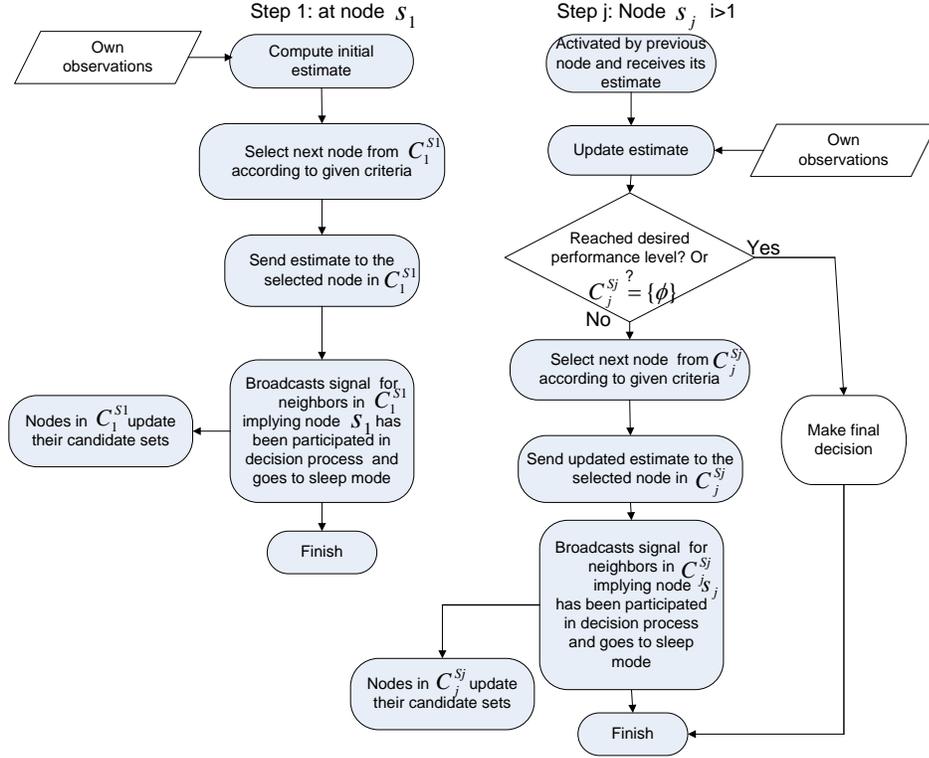
where $R_I(\theta, z_k, q_{j,k})$ is as given in (5.12) and (5.13), for the mutual information and the MMSE, respectively. To find the next best processing node, the node s_j has to compute the reward function (5.14) for all candidate nodes in \mathcal{V}_j^c . At the worst case, node s_j has to compute the reward function for $n - 1$ candidate nodes. On the other hand, decision process may consist of n nodes at the worst case, thus resulting in a worst case computational complexity of order $O(n^2)$ for the whole network. Note that in this scheme, in general the computational complexity is much more reduced (compared to the worst case complexity) since as the process continues, the number of nodes to be queried by the current processing node is decreased. However, in this scheme each node s_j has to keep track of nodes that have already been participated in the estimation process up to step j , which requires a large communication between nodes. Due to these, implementing this scheme distributively is difficult. Thus, in the following, we propose a distributed algorithm for sensor node selection with reduced computational and communication complexities in which each node only needs to keep track of its neighboring nodes to perform the sequential estimation.

5.4.3 Distributed node selection: local approach

Assume that k -th node in the network has a set of neighbors \mathcal{N}_k for $k = 1, \dots, n$ where the neighbors are determined based on a node's effective communication range and the affordable communication cost. We assume that each node has the same effective communication range, r_c , so that the criteria for selection of neighbors is the same for all nodes. In other words, each node s_k selects its neighbors as the nodes located inside a disk with an area of πr_c^2 centered at location \mathbf{x}_k of node s_k , for $k = 1, 2, \dots, n$. Moreover, if node s_k is a neighbor of node s_i , for $i \neq k$, then node s_i is also a neighbor of node s_k . Let s_j be the current processing node at step j . The node s_j selects the next node based on the objective function (5.13) from the set of candidate sensors $\mathcal{C}_j^{s_j}$ that is its neighbor nodes who have not been participated in the estimation process previously. Note that each node $s_j \in \mathcal{V}$ updates its set of candidate nodes $\mathcal{C}_j^{s_j}$ based on the information received from its neighbors. Each node has to keep track of the nodes participated in the estimation process only within its neighborhood. Thus, the next node s_{j+1} at step $(j + 1)$ is chosen as,

$$s_{j+1} = \operatorname{argmax}_{s_k \in \mathcal{C}_j^{s_j}} R(s_j, s_k). \quad (5.15)$$

To find the next best node according to (5.15), the node s_j has to compute the reward function (5.14) only for candidate nodes in the set $\mathcal{C}_j^{s_j}$. Denote $m = \max_{k \in \{1, 2, \dots, n\}} \{|\mathcal{N}_k|\}$ to be the maximum size of the set of neighbors for any node s_k in the network where $m \leq n - 1$. Thus each node s_j in the decision process has to compute the reward function only for a maximum of m nodes. Since there is a maximum of n nodes, this leads to a worst case computational complexity of order $O(mn)$ for the whole network. Since, in this scheme a node has to keep track of only its neighbors, the communication complexity is reduced compared to the scheme presented in subsec-


 Figure 5.4: Distributed sequential estimation process at node s_j for $j = 1, 2, \dots$

tion 5.4.2 whenever $m \ll n - 1$. In distributed sensor networks, network architectures where nodes only communicate with their neighbors to make local decisions are desirable due to network resource constraints. For example, in [159], each mobile node communicates with its one-hop neighbors at a given time to make a local estimate of the target state where the one-hop neighborhood at each node is dynamically changing. The proposed distributed sequential estimation process is summarized in Algorithm 5 and described in detail in Fig. 5.4.

Note that since node s_j selects the next node from the candidate set $C_j^{s_j}$, node s_j only needs to perform $|C_j^{s_j}|$ number of computations. Also, node s_j needs to keep track of the nodes which are not participated in the decision process in its

Algorithm 1 Sequential estimation process at step j at node s_j

```

1: while ( $j \geq 1$ ) do
2:   Compute estimate  $\hat{\theta}_j$ 
3:   Compute MMSE  $M_j$ 
4:   if ( $M_j < \text{Desired performance}$  or  $\mathcal{C}_j^{s_j} = \emptyset$ ) then
5:     Make final decision
6:     Go to sleep mode
7:   else
8:     Select next node from  $\mathcal{C}_j^{s_j}$ 
9:     Send estimate to the node selected
10:    Broadcast signal to nodes in  $\mathcal{C}_j^{s_j}$  implying node  $s_j$  has been participated in
        decision process
11:    Go to sleep mode
12:   end if
13: end while

```

neighborhood only. Once the final decision is made, a signal is broadcast implying the final decision is made. Then all unprocessed nodes go to sleep mode, until the next event occurs. Also it is to be noted that, when the effective communication range r_c is sufficiently large ($r_c \rightarrow \infty$), this scheme based on local approach converges to the scheme described in subsection 5.4.2. Thus the scheme described in subsection 5.4.2 can be considered as a special case of the proposed scheme in this subsection when $r_c \rightarrow \infty$.

Updating candidate set at the k -th processing node

Denote $\mathcal{C}_j^{s_k}$ to be the candidate set of node s_k at the step j . Algorithm for updating the candidate set at node s_k is explained in Algorithm 2.

Algorithm 2 Updating candidate set at k -th node

NOTATION

s_k : k -th node, s_j : processing node at step j , $\mathcal{C}_j^{s_j}$: candidate set of the processing node s_j at step j , $\mathcal{C}_j^{s_k}$: candidate set of the node s_k at step j

INITIALIZATION

$$\mathcal{C}_0^{s_k} = \mathcal{N}_k$$

UPDATING

- 1: **while** ($j \geq 1$) **do**
 - 2: $\mathcal{C}_j^{s_j} = \mathcal{C}_{j-1}^{s_j}$
 - 3: **if** $s_k = s_j$ (i.e. node s_k becomes the current processing node at step j) **then**
 - 4: $\mathcal{C}_j^{s_k} = \mathcal{C}_j^{s_j}$
 - 5: **else** $\{s_k \in \mathcal{C}_j^{s_j}$ (i.e. node s_k belongs to the candidate set of the current processing node at step j) $\}$
 - 6: $\mathcal{C}_j^{s_k} = \mathcal{C}_{j-1}^{s_k} \setminus s_j$
 - 7: **else**
 - 8: $\mathcal{C}_j^{s_k} = \mathcal{C}_{j-1}^{s_k}$
 - 9: **end if**
 - 10: **end while**
-

Note that, node s_j is not a neighboring node for any node in the network except for those that are in \mathcal{N}_j itself. Thus it is not necessary for nodes that are not in \mathcal{N}_j to keep track of node s_j . According to this scheme each node is required to communicate with and keep track of only its neighbors. However, this process will be terminated when the current node does not have any candidate neighboring nodes (i.e. $\mathcal{C}_j^{s_j} = \emptyset$), where \emptyset is the null set, irrespective of whether the desired performance level is reached or not, eventhough there might be remaining nodes in other neighborhoods of the network. However, as observed from simulations, this does not seem to cause a significant performance loss.

In both schemes discussed above in subsections 5.4.2 and 5.4.3, a global minimum is not guaranteed in general since in both schemes current node selects the next best node from all unvisited nodes in the network (in scheme discussed in 5.4.2), or in neighborhood (in scheme discussed in 5.4.3). As discussed at the beginning of this section, the optimal node ordering which yields the global minimum over all possible distinct node sequences can be computed, at a complexity of $O(n^3)$ via shortest path algorithms. We observe (see Section 5.5) that when there is no channel noise, the global scheme discussed in subsection 5.4.2 coincides with the optimal scheme that yields the global minimum (computed based on shortest path algorithm) and the local scheme proposed in subsection 5.4.3 performs close to the optimal scheme after processing relatively small number of nodes. Even when there is channel noise, we will see that both schemes perform fairly close to the optimal scheme. We refer to the node selection scheme presented in subsection 5.4.2 with global search as the scheme 1 and the proposed scheme with local search presented in subsection 5.4.3 as the scheme 2, in the rest of the Chapter.

5.5 Performance Analysis

Let us consider a 2D square sensor network of area A on $X \times Y$ plane. The locations of the k -th node and the target are denoted by $\mathbf{x}_k = (x_k, y_k)$, for $k = 1, \dots, n$, and $\mathbf{x}_t = (x_t, y_t)$, respectively. In the following we analyze the performance of a fixed 2D network when the target location is known exactly as well as statistically.

5.5.1 Exact target location is known at each node

First, we assume that the node s_j has knowledge of its own position, target location and the positions of its neighbors \mathcal{N}_j . Then the observation noise variance at the

k -th node according to the model (5.2) can be expressed as

$$\sigma_k^2 = \left(\frac{r_{kt}}{r_0} \right)^\alpha \sigma_0^2, \quad (5.16)$$

where $r_{kt} = \sqrt{(x_k - x_t)^2 + (y_k - y_t)^2}$ is the distance between the k -th node and the target, α is the path loss index and r_0 and σ_0^2 are constants. The channel noise variance between $(k - 1)$ -th node and the k -th node is given by

$$\sigma_{c(k-1,k)}^2 = \left(\frac{r_{k-1,k}}{r'_0} \right)^{\alpha'} \sigma_c^2, \quad (5.17)$$

where $r_{k-1,k} = \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}$ is the distance between the k -th node and the $(k - 1)$ -th node, α' is the path loss index and r'_0 and σ_c^2 are constants.

In the proposed schemes, node s_j computes the estimator and the MMSE according to (5.5) and (5.6). If the desired MMSE threshold is not met, node s_j sends its information to the node s_{j+1} , where the node s_{j+1} is selected from the candidate set \mathcal{V}_j^c according to (5.14) in the scheme 1 and from $\mathcal{C}_j^{s_j}$ according to (5.15) in the scheme 2. We assume that there is a total of 40 sensors deployed in a square region of 10×10 square units. The target is assumed to be at the origin and the initial node is selected randomly and assumed same for all plots. Neighbors at each node are selected as the set of nodes located within a disk of radius $r_c = 3$ units.

Figure 5.5 shows the performance of the sequential estimation process with the proposed node selection scheme 2 with both mutual information and MMSE as the information utility measures. It can be seen that MMSE acts as a good alternative for mutual information as the information utility measure for all β values considered. In cases where it is difficult to compute mutual information, we can use MMSE as the information utility measure (if it is easier to compute compared to mutual information). Thus, in the following results we use the MMSE as the information utility measure. Note that, in the following figures, we refer to the scheme which

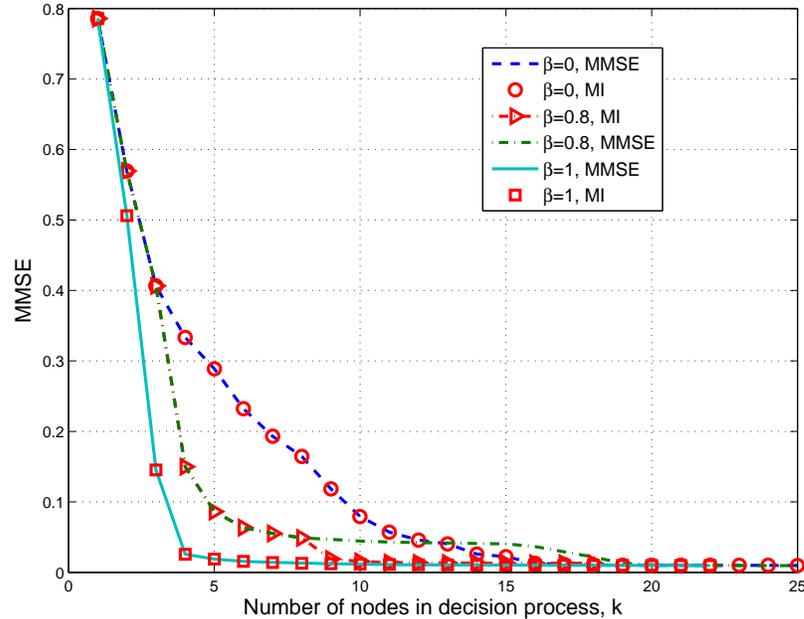


Figure 5.5: MMSE at the k -th processing node with the proposed node selection scheme based on local approach with both MMSE and mutual information as information utility measures: ($\sigma_c^2 = 0$, $\alpha' = 2$, $r_0 = 1$, $\sigma_0^2 = 0.1$).

results in the global minimum over all nodes, computed based on shortest path algorithms, as the optimal scheme.

Figure 5.6 shows the MMSE performance at the k -th node with no channel noise such that $\sigma_{c(k-1,k)}^2 = \sigma_c^2 = 0$ and Fig. 5.7 shows that with channel noise with $\sigma_{c(k-1,k)}^2$ as given in (5.17), with the MMSE as the information utility measure.

With no channel noise, it can be seen that when $\beta = 1$, the performance with node ordering based on proposed scheme 1 coincides with that of the optimal scheme which results in the global minimum. In that case, from Fig. 5.6 it can be seen that the performance of the proposed scheme 2 converges to that of scheme 1 (as well as to that with optimal scheme) after a relatively small number of processing nodes. For $\beta = 0.8$ and $\beta = 0$, it is seen that proposed scheme 1 and scheme 2 give similar

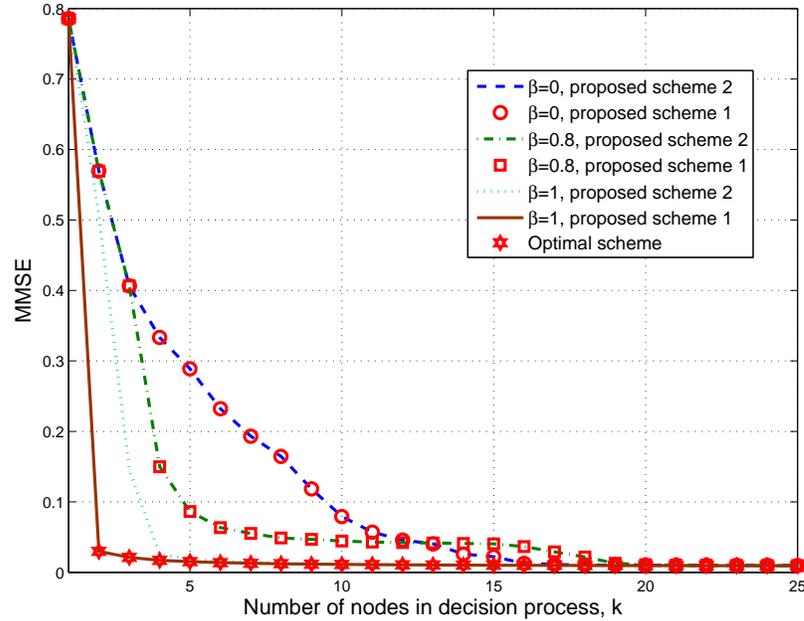


Figure 5.6: MMSE at the k -th node with exact target location with MMSE as the information utility measure with no channel noise: $\sigma_{c(k-1,k)}^2 = \sigma_c^2 = 0$, $n = 40$, $\alpha' = 2$, $r_0 = 1$

performance. For example, with $\beta = 1$, to achieve a required performance level of an MMSE of 0.05, scheme 1 requires 2 nodes, while scheme 2 requires 4 nodes. On the other hand, to achieve the same performance level, both scheme 1 and the scheme 2 require 8 and 12 nodes with $\beta = 0.8$ and $\beta = 0$, respectively. It is noted that the proposed scheme 2 is terminated at node 25, 32 and 35 with $\beta = 1$, $\beta = 0.8$ and $\beta = 0$, respectively, due to the reason discussed in subsection 5.4.3. However, it is seen that once such a number of nodes are processed node ordering does not affect the overall performance level. This implies that when the sequential estimation process is continued among a large number of sensors, the performance converges to the same value irrespective of how the nodes are selected, which of course is not desirable in many resource constrained sensor networks.

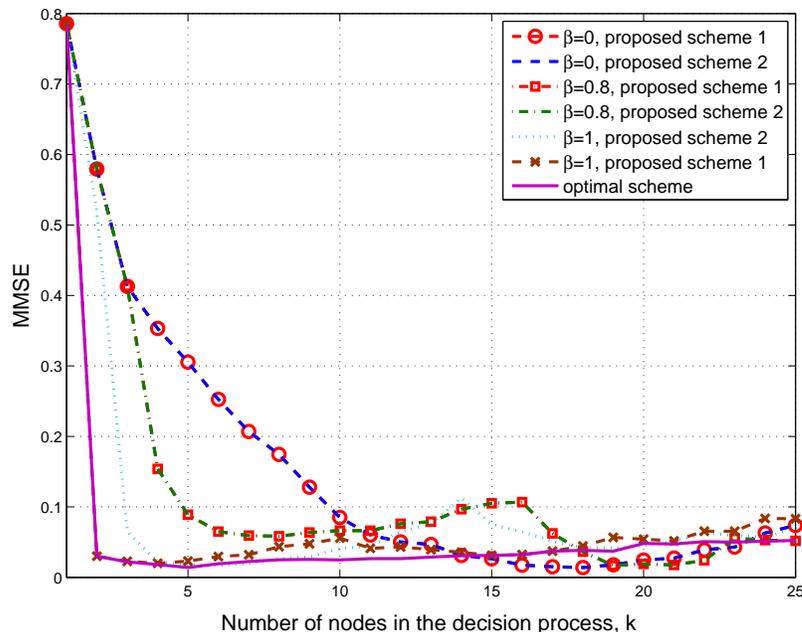


Figure 5.7: MMSE at the k -th node with exact target location with MMSE as the information utility measure with channel noise: $\sigma_{c(k-1,k)}^2$'s are as given in (5.17), $n = 40$, $\sigma_c^2 = 0.001$, $\alpha' = 2$, $r_0 = 1$

On the other hand, when there is channel noise, it is seen that continuing the sequential processing after some point does not yield improved performance as can be seen from Fig. 5.7. This essentially is due to the fact observed in (5.8). However, in this case, from Fig. 5.7 it can be seen that the proposed scheme 2 (local approach) with $\beta = 1$ gives closer performance to that of with the optimal scheme. Also we can see that the performance of proposed scheme 1 (global approach) and that of the scheme 2 is almost the same for $\beta = 0.8$ and $\beta = 0$. When $\beta = 1$, from Fig. 5.7 it can be seen that the proposed scheme 2 yields a lower MMSE compared to the scheme 1 after processing a certain number of nodes. This can be explained by noting the fact that both proposed schemes 1 and 2 are greedy-type algorithms. Thus they would not necessarily result in the same global minimum after completing the same number

of processing stages.

From these performance results, we can see that in the proposed sequential estimation process, the proposed greedy-type algorithms essentially results in a near-optimal solution in finding the best ordering of nodes compared to the optimal scheme which yields the global minimum at a high computational and communication cost.

5.5.2 Statistics of the target location is known at each node

In this section we assume that the two coordinates of target location x_t and y_t (with the origin at the center of the square) are distributed as marginal Gaussian with mean zero and the equal variance $\sigma_t^2 = 5$ units. Then it can be verified that the random variable $X = \frac{r_k^2}{\sigma_t^2}$ has a non-central chi-squared distribution with the pdf $f_X(x) = \frac{1}{2}e^{-(x+\lambda_k)/2}I_0(\sqrt{\lambda_k x})$ where $\lambda_k = \frac{x_k^2 + y_k^2}{\sigma_t^2}$ and $I_a(x)$ is the modified Bessel function of the first kind given by $I_a(x) := (x/2)^a \sum_{i=0}^{\infty} \frac{(x^2/4)^i}{i!\Gamma(a+i+1)}$ where $\Gamma(z) = \int_0^{\infty} t^{z-1}e^{-t}dt$ is the Gamma function. Using (5.16), the average MMSE at the k -th node is given by

$$\begin{aligned} \bar{M}_k &= \frac{\sigma_t^2 \sigma_0^2 Q_{k-1}}{r_0^2} \mathbb{E}_x \left\{ \frac{x}{x + Q_{k-1}} \right\} \\ &= \frac{\sigma_\theta^2 \sigma_0^2 Q_{k-1} e^{-\frac{\lambda_k}{2}}}{2r_0^2} \left[J_0(\sqrt{\lambda_k Q_{k-1}}) Q_{k-1} e^{\frac{Q_{k-1}}{2}} Ei \left(-\frac{Q_{k-1}}{2} \right) \right. \\ &\quad \left. + \sum_{i=0}^{\infty} \frac{(\frac{\lambda_k}{4})^i}{i!\Gamma(i+1)} \sum_{l=1}^{i+1} (l-1)! (-Q_{k-1})^{i+1-l} \left(\frac{1}{2}\right)^{-l} \right], \end{aligned} \quad (5.18)$$

where $Q_{k-1} = \frac{r_0^2 \sigma_\theta^2 B_{k-1}}{\sigma_t^2 \sigma_0^2 (B_{k-1} + \sigma_\theta^2 A_{k-1})}$, $A_{k-1} = (\sigma_\theta^2 - \bar{M}_{k-1})^2$, $B_{k-1} = \sigma_\theta^2 \left[\bar{M}_{k-1} (\sigma_\theta^2 - \bar{M}_{k-1}) + \sigma_\theta^2 \sigma_{c(k-1,k)}^2 \right]$ and $J_0(\cdot)$ is the zero-th order Bessel function of first kind.

Figures 5.8 and 5.9 show the MMSE performance of the 2D sensor network without and with channel noise, respectively, when the exact target location is not known. The network parameters are the same as that in Figs 5.6 and 5.7. It can be seen that

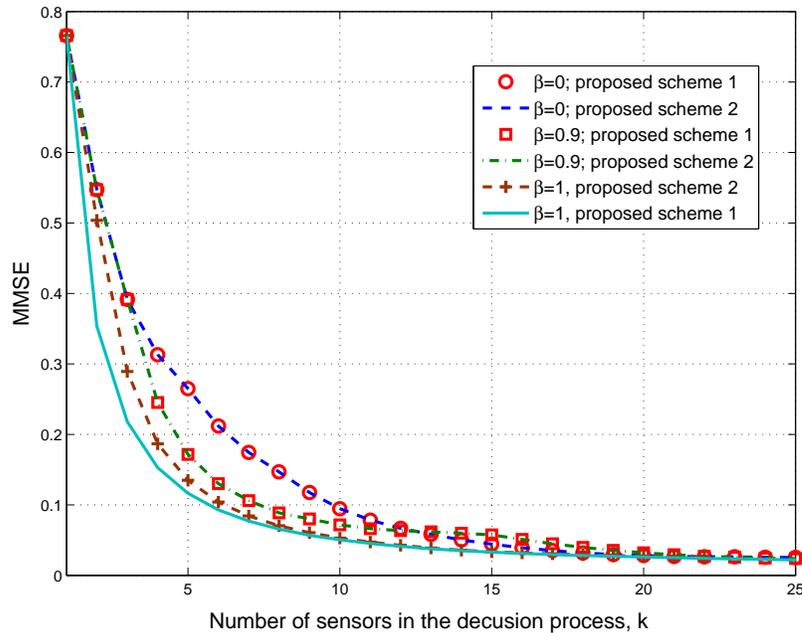


Figure 5.8: MMSE at the k -th node when the statistics of target location are available with no channel noise: $\sigma_{c(k-1,k)}^2 = \sigma_c^2 = 0$, $\alpha' = 2$, $r_0 = 1$, $\sigma_0^2 = 0.1$

the performance of the proposed scheme 2 (local approach) is closer to that with the proposed scheme 1 (global approach) with $\beta = 1$ both with perfect as well as noisy inter-node communication. For other values of β considered, the performance of the proposed scheme 2 almost coincides with that of the scheme 1. It is also noted that (although figures are not included) when the uncertainty σ_t^2 of the target location is high, the performance of the sequential estimation process does not depend much on the ordering of nodes.

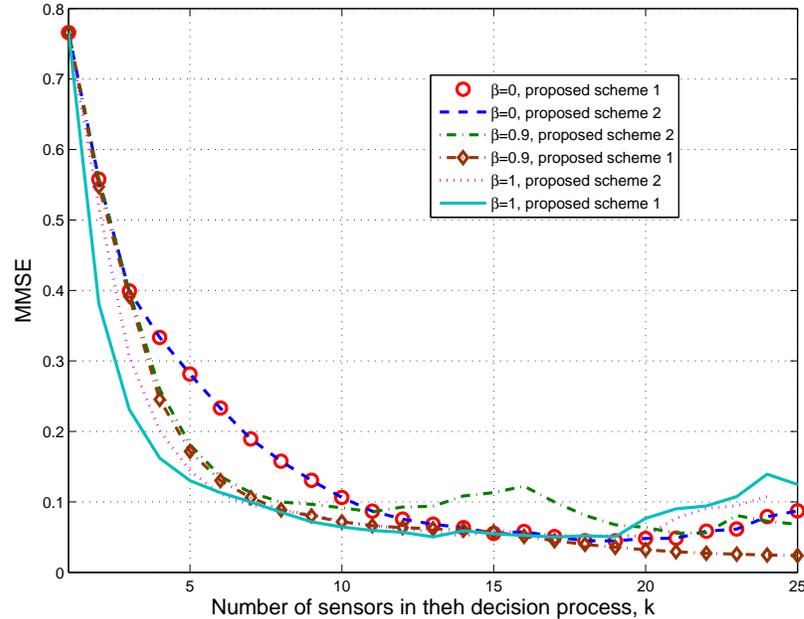


Figure 5.9: MMSE at the k -th node when the statistics of target location are available with channel noise: $\sigma_{c(k-1,k)}^2$'s are as given in (5.17), $\sigma_c^2 = 0.005$, $\alpha' = 2$, $r_0 = 1$, $\sigma_0^2 = 0.1$

5.6 Conclusions

We proposed a distributed sequential scheme for estimation of a Gaussian parameter over noisy communication links between nodes with distributed node selection algorithms. In the proposed scheme, each node makes a local estimate by combining its own observation and the estimator from the previous node. To update the estimator at the next node, the current node's decision is sent to the next node through a noisy communication channel. It was shown that such a sequential estimation scheme is useful only if the channel noise quality satisfies a certain threshold condition. We proposed two node selection schemes to select the best node ordering in the sequential estimation process based on a reward function which reflects the trade-off

between an information utility measure and the inter-node communication cost. To select the next best node, in scheme 1 the current node searches over all unvisited nodes in the network, while in the scheme 2 it searches over only its neighborhood. In the proposed scheme with local search, to perform the distributed sequential estimation process, each node has to only keep track of its neighboring nodes. We show that the performance of the scheme based on the local search gets closer to that with scheme with global search with a relatively small number of nodes. Since both proposed node selection schemes are greedy algorithms, they do not guarantee a global optimal solution. We compare the two proposed schemes with the optimal scheme computed based on optimal shortest path algorithms, which provides the global optimal solution and show that the performance of two proposed schemes gets very close to the optimal solution after processing relatively small number of nodes. We derived the MMSE performance for 2-D sensor network models when either the exact or only the statistics of the target position information are available at each node. The proposed sequential node selection scheme based on the local search can be performed distributively having only the information regarding neighbor nodes at each node.

Chapter 6

Impact of Mobile Node Density on Detection Performance

6.1 Introduction

When a large area needs to be monitored by a sensor network for target detection, estimation or tracking, it is necessary to deploy a large number of static nodes in the region of interest in order to achieve an acceptable performance. But to deploy a large number of sensors in a large network area or in hostile environments might not be practical or affordable. For example, to achieve a k -coverage in a random sensor network, with a network size of L , it needs to increase the sensor density as $O(\log L + k \log \log L)$ at initial deployment stage [136]. On the other hand, the coverage of a static sensor network will remain the same (or reduced due to node failures) after the initial deployment stage. This leads the sensor network to have coverage holes over time. In order to cope with the unreliability, and provide dynamic on demand coverage, static nodes can be integrated with mobile nodes. With the recent advances in deploying sensor nodes with mobile platforms, such as mobile robots or

unmanned autonomous vehicles, [83,99,116] integrated mobile-static sensor networks are becoming attractive research area. If only mobile nodes are deployed, a location is switched between covered and uncovered with the time as nodes move. On the other hand, deploying a large number of mobile nodes might be unaffordable due to the cost. By integrating static nodes with mobile nodes and efficiently collaborating between them, it is more likely that the network can provide a better sensing coverage on demand with less resource consumption, compared to that with in a all-static or all-mobile network. The goal of such a network is to exploit node mobility in an efficient manner to compensate for the lack of performance resulted in a all-static network.

Using node mobility to reposition nodes to provide uniform coverage at the initial deployment stage has been addressed by several authors, [27, 57, 134–136, 146, 158]. These work differ from each other based on algorithms they use. When nodes are deployed in a random fashion at initial deployment stage, an efficient coverage may not guarantee since sensors might be overly clustered or there might be small fractions of nodes in certain portions in the network. Random deployments may occur in cases when the sensor network is established by dropping or throwing sensors into the sensor field. Using mobile nodes to self organize at the deployment stage after initial random deployment is addressed in [158], where a force-directed algorithm based on attractive and repulsive forces is used to move nodes to enhance the coverage. In [135], three algorithms are presented to heal the coverage holes occurred at the initial deployment stage. Trade-off between mobile node density and the coverage (k -coverage) of hybrid sensor network at deployment stage is addressed in [136] with flip-based sensor nodes which can move only once and have limited mobility. However, these work do not address the dynamic coverage aspects that can be achieved by allowing node mobility over time.

In [79], the dynamic aspects of the coverage of sensor networks with mobile nodes

were addressed with random node mobility. They have investigated the coverage measure based on the Boolean sensing model and the detection time of a randomly located target in a mobile sensor network. They assumed a continuous random movement of sensor nodes where the movements are not controllable. However, they consider only all-mobile networks and do not provide the cost evaluations of mobile nodes in continuous movement. Analytical modeling of detection latency of a mobile sensor network was addressed in [31] where they have considered the detection of a stationary target and presented a performance comparison between mobile and static sensor networks based on latency. In [30], a cat-and-mouse game between targets and mobile nodes was presented based on the sensing capabilities of targets and mobile nodes in which mobile nodes try to detect the target as quickly as possible when the target is trying to evade the sensing region without being detected. [43] presented the delay of intrusion detection when the target is moving on a straight line and Brownian motion when there are disconnected clusters in the network. A game theoretic model for management of mobile sensors is presented in [110] where a game theoretical model was proposed to assign targets to mobile sensors in a multi-target tracking system. Event capturing using mobile sensor network is presented in [16] where the quality of coverage resulted by mobile sensor nodes is investigated. Brownian motion of mobile nodes to enhance the coverage in mobile sensor network is presented in [67], where they have derived the distribution of the time-until-detection of slowly moving targets. Distributed tracking by mobile sensor networks is addressed in recent research, for example in [99,159]. In [131], mobile node navigation towards a specific goal in a hybrid sensor network is addressed where static nodes are used to guide the mobile nodes. Distributed detection by hybrid sensor networks is addressed by recent work [122,149] when the sensor node and target positions are known. Target tracking performance of an integrated mobile-static sensor network was addressed in [68] where the mobile nodes are used to aid the data propagation when the communication ranges of static nodes are limited.

In addition to surveillance coverage improvement, node mobility in sensor networks is exploited in different contexts such as facilitating network connectivity and achieving better communication between sensor nodes and the fusion center. Impact of node mobility in sensor networks in terms of capacity/throughput improvement is addressed in [86], [51].

In this Chapter, our focus is to analyze several important performance measures and evaluate the cost of adding mobile nodes in target detection applications by a hybrid sensor network. We address the problem of detecting an arbitrary located stationary target using such a hybrid sensor network. We assume that the static nodes and the initial locations of mobile nodes are both independently and uniformly distributed in a two dimensional plane such that node locations follow a 2-D Poisson point process. In practice, random sensor deployment for sensor networks is desirable in many situations. For example, if a priori knowledge of the sensing field is not available at the deployment stage, it is more desirable to position sensor nodes randomly. Moreover, random node deployment is justifiable when it is more cost effective and practical to deploy nodes randomly in contrast to systematic deployment. Target detection in random stationary sensor networks has been studied by [18, 43, 70, 97, 137].

In network performance analysis, several random node mobility models which model real world node mobility are used in the literature. In the following, we discuss some of the widely used random mobility models for the performance evaluations in sensor/ad hoc networks.

- *Random straight line model*: In this model, a node selects the direction randomly and uniformly at the initial time and then moves on a straight line [79]. This is a very simple model used in sensor network simulations.
- *Brownian motion model*: Under this mobility model, at a given time a node

moves from its current location to a new location by selecting the direction and the speed randomly, both chosen from pre-defined ranges [17,67]. This is a memoryless mobility pattern since at each movement step, the direction and the speed for the next location, are chosen independently from the past. This model might lead to unrealistic movements sometimes such as sudden stops and sharp turns. Many derivatives of Brownian motion models in different dimensions are used for node mobility in sensor networks including random walk mobility model [89]. Random walk mobility model provides a steady state uniform distribution of node locations [89].

- *Random Waypoint model*: Random waypoint model is a commonly used model in the simulation of ad hoc networks [14,17]. In this model, at a given time, each node of the network chooses a destination point in the deployment area in uniform manner. The node moves to the selected destination point with a speed v chosen randomly and uniformly in the interval $[v_{min}, v_{max}]$. When it reaches the destination, it will remain stationary for a certain pause time and then starts moving again in the same manner. For a long running time of the movement process, the stochastic node distribution of random waypoint model converges to a non-uniform distribution although the initial node positions are uniform.
- *Gauss-Markov model*: In Gauss Markov mobility model, speed and direction at a given time are determined based on the corresponding values at the previous movement step, and a random variable [17]. In particular, the speed and the direction at time t , v_t and θ_t are given by,

$$\begin{aligned} v_t &= \alpha v_{t-1} + (1 - \alpha)\bar{v} + \sqrt{(1 - \alpha^2)}v'_{t-1} \\ \theta_t &= \alpha\theta_{t-1} + (1 - \alpha)\bar{\theta} + \sqrt{(1 - \alpha^2)}\theta'_{t-1} \end{aligned} \tag{6.1}$$

where $0 \leq \alpha \leq 1$ is the tuning parameter used to vary randomness, v'_{t-1} and θ'_{t-1} are Gaussian random variables and \bar{v} and $\bar{\theta}$ are constants representing the mean values of speed and the direction as $t \rightarrow \infty$. Note that the term α controls the randomness of the mobility model.

For the performance evaluations in this Chapter, we consider two specific random mobility models as discussed above which result approximately uniform steady state distribution of node locations: In the first model we assume that each mobile node initially selects a direction to move randomly and uniformly and then move on a straight line in the selected direction [79]. In the second model, the mobile nodes are assumed to follow 2-dimensional random walks. Such models for mobile nodes can be justified in situations where the network does not have any prior information regarding the sensing field and the target existence. Moreover, random and independent mobility model requires minimum coordination among mobile nodes.

In a mobility assisted sensor network, important performance measures that should be considered in addition to detection probability are (i). detection latency which accounts for the time that the target remains undetected, (ii). the mean contact distance at a given time which reflects the mean distance between the target and any point covered by the sensor network at a given time. Another important fact to be considered in a hybrid sensor network is the node connectivity. Since node mobility, the topology of the network is varying over time. However, to obtain the maximum benefit by the node mobility, it should be ensured that the nodes are connected at any give time.

For the target detection, in particular, we consider two detection models: single-sensing and k -sensing [70]. In single-sensing detection, the target is assumed to be detected if at least one sensor detects it providing the minimum guarantee on target detection [70]. In k -sensing detection, on the other hand, the target is assumed to be detected if at least k -sensors detects it where k is a design parameter. In this model,

the target is detected with lower false alarm probability than with single-sensing detection [70]. Under these detection and node mobility models, the main problems addressed in this Chapter can be categorized as,

1. Derive the detection probability analytically in stationary target detection by the hybrid sensor network for two specific random mobility models (as presented in subsection 6.2.2) for mobile nodes. We consider two detection models; single-sensing and k -sensing detection.
2. Derive the detection latency for both single-sensing and k -sensing schemes.
3. Analyze the trade-off between the mobile node density and the detection performance achieved within a desired delay constraint.
4. Derive the mean first contact distance between the target and the closest (to the target) point covered by the sensor network with at least one sensor.
5. Analyze the trade-off between mobile node density and the node communication ranges to maintain the node connectivity at a given time in the hybrid sensor network.

The Chapter is organized as follows: Section 6.2 explains the sensor network, target and detection models. Section 6.3 derives the detection performance measures, in terms of detection probability, latency and mean first contact distance with the mobility model 1 for single-sensing and k -sensing detection models and discusses their dependence on mobile node density. In Section 6.4, the detection performance with random walk mobility model is given. Probability of the node connectivity is addressed in Section 6.5. Performance results are shown in Section 6.6 and the concluding remarks are given in Section 6.7.

6.2 System Model

We consider a hybrid sensor network made of a large number of sensor nodes, N , deployed in a large region \mathcal{R} . When a large region is to be monitored by a sensor network, it is desirable to deploy a large number of inexpensive, low power sensor nodes to improve the expected performance. Specifically, we assume that there are N_s number of static nodes and N_m number of mobile nodes. Denote (x_{sk}, y_{sk}) to be the location of the k -th static node where x_{sk} and y_{sk} are assumed to be independently and uniformly distributed in $[-b/2, b/2]$ where $b \times b$ is the assumed dimension of the sensor network. Denote $\lambda = \frac{N}{b^2}$ to be spatial density of the nodes and $\lambda_m = \frac{N_m}{N}$ and $\lambda_s = \frac{N_s}{N}$ be the fractions of mobile and static nodes respectively. Note that we assume that the total number of sensor nodes, N and network dimension, $b \times b$ are large enough so that assumptions made in the rest of the Chapter are valid. Let \mathcal{V} be the set containing all node indices in the network and let \mathcal{V}_m and \mathcal{V}_s to be the sets containing mobile and static node indices, respectively.

6.2.1 Target model

We consider stationary target detection by the hybrid sensor network, where the target location is assumed to be an independently and uniformly distributed arbitrary point P_0 in the region \mathcal{R} .

6.2.2 Node mobility models

In this Chapter, we consider two random mobility models: In the first model (model 1), a mobile node moves independently in a direction θ selected randomly and uniformly where $\theta \sim \mathcal{U}[0, 2\pi)$, with an average speed of \bar{v} which is assumed to be the same for all mobile nodes. Note that we use $X \sim \mathcal{U}[a_1, a_2]$ to denote that X is

uniformly distributed in the interval $[a_1, a_2]$. Then at any time $t = nT_s$, a mobile node has moved on a straight line a distance of $n\bar{v}T_s$ where T_s is the length of each time step [79]. Second, in model 2, we consider that k -th mobile node follows a 2-dimensional random walk [77] of n steps at time nT_s with each of a length $\mu = \bar{v}T_s$. Random and independent mobility models are justifiable in scenarios where nodes do not have any prior knowledge of sensing field or target existence. Also random node mobility models are desirable when minimum node coordination is required. Model 1 assumed in the Chapter is the simplest mobility model which requires minimum control and coordination. Random walk mobility model can be justifiable when mobile nodes are characterized by uncontrolled dynamics, such as random ON-OFF transitions at each time step [113]. These two random models for a mobile node are illustrated in Fig. 6.1.

6.2.3 Detection model

We assume that each node has identical effective sensing range r with the sensing area of πr^2 . Although we assume homogeneous sensor nodes for simplicity, the results can easily be extended for heterogeneous sensor nodes having different sensing ranges.

We assume a binary detection model in which the point P_0 is considered to be detected with probability 1 by the sensor s_k at time $t = nT_s$ if it lies in sensor-coverage area $C_k(nT_s)$ [158], where $C_k(nT_s)$ is the coverage area of node s_k at time nT_s for $n = 0, 1, 2, \dots$. Formally, we can express the probability that the node s_k detects the target at time interval $[0, nT_s)$ as:

$$P_{d_k}(nT_s) = \begin{cases} 1 & \text{if } P_0 \in C_k(nT_s) \\ 0 & \text{otherwise} \end{cases}$$

Note that for a static node, the coverage area $C_k(nT_s)$ is constant over time. Thus if the target is not detected by a static node initially, it will never be detected. However,

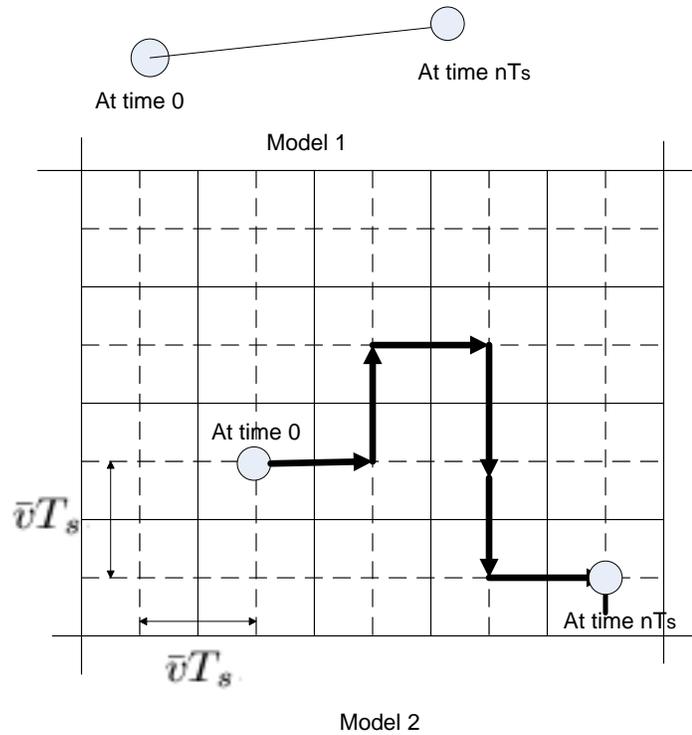


Figure 6.1: Random mobility models of a mobile node

with a mobile node, since the coverage is varied over time, there is a probability for the target to be detected as time progresses.

6.2.4 Preliminaries

Boolean model

Let $\mathcal{P} \equiv \{\alpha_i, i \geq 1\}$ in \mathbb{R}^k is a point process and $\{S_i, i \geq 1\}$ be a sequence of independently and identically distributed random sets, independent of \mathcal{P} . The collection of sets $\mathcal{C} = \{\alpha_i + S_i, i \geq 1\}$ is called a coverage process [55]. When \mathcal{C} is driven by a stationary Poisson point process (i.e. \mathcal{P} is a stationary Poisson point process),

the coverage process \mathcal{C} is called a Boolean model [55]. Since we assume that static node locations and initial mobile node locations are independently and identically distributed in a vast two dimensional area, the sensor locations can be modeled as a two-dimensional Poisson point process with intensity λ , when the total number of nodes and the sensing region are large. With the considered random mobility models, since mobile nodes make independent and identical random movements, at any time instance $t = nT_s$, sensor locations still form a 2-D Poisson point process with the same intensity [112] when the area $b^2 \lim_{\rightarrow} \infty$.

Notation

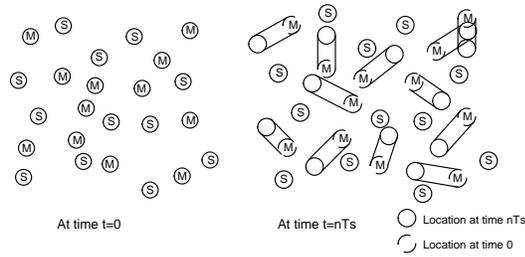
We use $\mathcal{A}(S)$ and $\mathcal{P}(S)$ to denote the area and perimeter of the set S . Denote by $P + S$ the set centered at P with a shape of S .

6.3 Stationary Target Detection Performance with Mobility Model 1

6.3.1 Detection probability

In the following, we consider two modes of detection: Single-sensing detection and k -sensing detection [70]. In Single-sensing detection, the target is considered as detected if it is captured by at least one sensor. In this case, target's presence is obtained with the minimum guarantee. On the other hand, detection by multiple sensors ensure lower false alarms. In k -sensing detection model, the target is considered as detected if it is detected by at least k sensors where k is a design parameter [70].

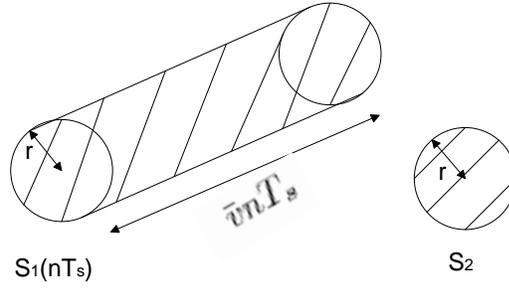
In this section, we analyze the detection performance with the random node mo-


 Figure 6.2: Hybrid network at times $t = 0$ and $t = nT_s$

bility model 1, where the mobile nodes move in a straight line after selecting the direction independently and uniformly from $[0, 2\pi)$. Figure 6.2 shows an illustration of the coverage area of the sensor network at time $t = 0$ and time $t = nT_s$ with mobility model 1. With the assumption that the initial node locations are independent and uniform, we can model the sensor network as a Boolean model at any given time in which the driving point process is the initial Poisson point process with intensity λ and the shape distribution is varied with the time. Further, denote T_0 to be the average time a mobile node takes to leave the sensing region \mathcal{R} . Since we assume that the sensing region is large enough and the speed of a mobile node is small (e.g. for example, Robomote [116] mobile nodes have speed of $0.5 \sim 2m/s$), T_0 is assumed to be large. Thus the main focus in this Chapter is to analyze the detection performance in the region where $nT_s \leq T_0$. The corresponding coverage area $S(nT_s)$ at time $t = nT_s \leq T_0$ is distributed as

$$S(nT_s) = \begin{cases} S_1(nT_s) & \text{with prob } \lambda_m \\ S_2 & \text{with prob } 1 - \lambda_m \end{cases}, \quad (6.2)$$

where $S_1(nT_s)$ and S_2 are as shown in Fig. 6.3. The coverage area of k -th static sensor at time $nT_s \leq T_0$ is given by, $C_k^s(nT_s) = C_k^s = \mathcal{A}(S_2) = \pi r^2$, and the coverage area of the k -th mobile node at time $t = nT_s$ is given by (corresponding to shape $S_1(nT_s)$) $C_k^m(nT_s) = \mathcal{A}(S_1(nT_s)) = \pi r^2 + 2rnT_s\bar{v}$. Note that for $nT_s \geq T_0$, we have


 Figure 6.3: Realization of random shapes at time nT_s

$$C_k^m(nT_s) = \mathcal{A}(S_1(nT_s)) = \pi r^2 + 2rT_0\bar{v} \text{ while } C_k^s(nT_s) = C_k^s = \pi r^2.$$

The probability that the target is detected at time $t = nT_s$ is given by the following theorem.

Theorem 1. (Detection probability) *The probabilities of detection with single-sensing and the k -sensing models ($k \geq 1$) at time $t = nT_s$ are given by,*

$$P_D^1(nT_s) = \begin{cases} 1 - e^{-\lambda(\pi r^2 + 2\lambda_m r \bar{v} n T_s)} & \text{if } nT_s \leq T_0 \\ 1 - e^{-\lambda(\pi r^2 + 2\lambda_m r \bar{v} T_0)} & \text{if } nT_s > T_0 \end{cases} \quad (6.3)$$

and

$$P_D^k(nT_s) = \begin{cases} 1 - \sum_{j=0}^{k-1} \frac{(\lambda(\pi r^2 + 2\lambda_m r \bar{v} n T_s))^j e^{-\lambda(\pi r^2 + 2\lambda_m r \bar{v} n T_s)}}{j!} & \text{if } nT_s \leq T_0 \\ 1 - \sum_{j=0}^{k-1} \frac{(\lambda(\pi r^2 + 2\lambda_m r \bar{v} T_0))^j e^{-\lambda(\pi r^2 + 2\lambda_m r \bar{v} T_0)}}{j!} & \text{if } nT_s > T_0 \end{cases}$$

respectively.

Proof. In single sensing detection, the target is considered as detected, if at least one sensor captures it. If $\mathcal{C} \equiv \{\alpha_i + S_i, i \geq 1\}$ is a Boolean model with shapes S_i are distributed as S , the number of sets (shapes) that intersects an arbitrary point (or the number of sets that covers an arbitrary point) in the Boolean model has a

Poisson distribution with mean $\lambda \mathbb{E}\{\mathcal{A}(S)\}$ [55]. Note that with the mobility model 1, the average area covered by a mobile node within the time interval $[0, nT_s]$ is given by $\bar{C}^m(nT_s) = \mathcal{A}(S_1(nT_s)) = \pi r^2 + 2rnT_s\bar{v}$ if $nT_s \leq T_0$ and $C_k^m(nT_s) = \mathcal{A}(S_1(T_0)) = \pi r^2 + 2rT_0\bar{v}$ if $nT_s > T_0$. Now, as can be seen from the right plot of Fig. 6.2, at time $t = nT_s$, the hybrid sensor network can be considered as a Boolean model in which the diving point process is the initial Poisson point process and the shape distribution is given by 6.2, in which the average coverage areas are determined depending on whether $nT_s \leq T_0$ or $nT_s > T_0$. Denote $P_{P_0}(m, nT_s)$ to be the probability that m number of sensors cover the point P_0 at time $t = nT_s$, which is given by [55]

$$P_{P_0}(m, nT_s) = \frac{(\lambda \bar{C}(nT_s))^m e^{-\lambda \bar{C}(nT_s)}}{m!},$$

where $\bar{C}(nT_s) = (\lambda_m \bar{C}^m(nT_s) + (1 - \lambda_m)C^s)$ is the average coverage area of the network at time nT_s . Then the probability that no sensor covers the point P_0 , $P_{P_0}(0, nT_s)$, at time nT_s is given by $P_{P_0}(0, nT_s) = e^{-\lambda \bar{C}(nT_s)}$. The probability of the single-sensing detection at time $t = nT_s \leq T_0$ is thus given by,

$$\begin{aligned} P_D^1(nT_s) &= 1 - P_{P_0}(0, nT_s) = 1 - e^{-\lambda \bar{C}(nT_s)} \\ &= 1 - e^{-\lambda(\pi r^2 + 2\lambda_m r \bar{v} nT_s)}. \end{aligned}$$

For $t = nT_s > T_0$, we will get, $P_D^1(nT_s) = 1 - e^{-\lambda(\pi r^2 + 2\lambda_m r \bar{v} T_0)}$. In k -sensing detection, the target is considered to be detected if at least k sensors detect it. Probability that the point P_0 is covered by at least k sensors at time nT_s is given by,

$$\begin{aligned} P_D^k(nT_s) &= 1 - Pr(P_0 \text{ is covered by } k-1 \text{ or less sensors}) \\ &= 1 - \sum_{j=0}^{k-1} P_{P_0}(j, nT_s) \\ &= \begin{cases} 1 - \sum_{j=0}^{k-1} \frac{(\lambda(\pi r^2 + 2\lambda_m r \bar{v} nT_s))^j e^{-\lambda(\pi r^2 + 2\lambda_m r \bar{v} nT_s)}}{j!} & \text{if } nT_s \leq T_0 \\ 1 - \sum_{j=0}^{k-1} \frac{(\lambda(\pi r^2 + 2\lambda_m r \bar{v} T_0))^j e^{-\lambda(\pi r^2 + 2\lambda_m r \bar{v} T_0)}}{j!} & \text{if } nT_s > T_0 \end{cases} \end{aligned}$$

□

Since allowing more nodes to be mobile is not desirable in many applications due to energy constraints, it is required to determine the minimum fraction of mobile nodes to be deployed in order to achieve the desired performance during a given time interval. The following theorem states the minimum fraction of mobile nodes required to achieve a desired probability level within a desired time interval for single sensing detection.

Theorem 2. (Minimum mobile node density required with single sensing detection) *Let η_D be the desired detection probability to be achieved by the hybrid sensor network at time $t_D \leq T_0$. The minimum fraction of mobile nodes to be used to achieve η_D at time $t_D (\leq T_0)$ with single-sensing detection model is given by,*

$$\lambda_m^{min} = \begin{cases} \frac{-\log(1-\eta_D) - \lambda\pi r^2}{2\lfloor \frac{t_D}{T_s} \rfloor \lambda r \bar{v} T_s}, & \text{if } \eta_s \leq \eta_D \leq \eta_t \\ \text{infeasible,} & \text{otherwise,} \end{cases} \quad (6.4)$$

where $\eta_s = 1 - e^{-\lambda\pi r^2}$ and $\eta_t = 1 - e^{-\lambda[\pi r^2 + 2\lfloor \frac{t_D}{T_s} \rfloor r \bar{v} T_s]}$.

Proof. If the tolerable detection delay is $t_D (\leq T_0)$, and the desired detection probability is η_D , the minimum λ_m is characterized by,

$$\begin{aligned} & \min \lambda_m \\ \text{s.t.} \quad & P_D^1 \left(\left\lfloor \frac{t_D}{T_s} \right\rfloor T_s \right) \geq \eta_D \quad , \end{aligned}$$

where $P_D^1 \left(\left\lfloor \frac{t_D}{T_s} \right\rfloor T_s \right)$ is given by (6.3). This leads to

$$\lambda_m \geq \frac{-\log(1 - \eta_D) - \lambda\pi r^2}{2\lfloor \frac{t_D}{T_s} \rfloor \lambda r \bar{v} T_s}. \quad (6.5)$$

Note that (6.5) holds for a desired delay constraint, only if the desired detection probability η_D satisfies the condition $\eta_s \leq \eta_D \leq \eta_t$ where $\eta_s = 1 - e^{-\lambda\pi r^2}$ and $\eta_t = 1 - e^{-\lambda[\pi r^2 + 2\lfloor \frac{t_D}{T_s} \rfloor r \bar{v} T_s]}$ are the detection probabilities achieved by the network if all nodes are stationary ($\lambda_m = 0$), and if all nodes are allowed to move ($\lambda_m = 1$), respectively. \square

In the case of k -sensing detection, the minimum fraction of mobile nodes can be found by finding the minimum λ_m which satisfies the following inequality:

$$1 - \sum_{j=0}^{k-1} \frac{(\lambda(\pi r^2 + 2\lambda_m r \lfloor \frac{t_D}{T_s} \rfloor \bar{u} T_s))^j e^{-\lambda(\pi r^2 + 2\lambda_m r \lfloor \frac{t_D}{T_s} \rfloor \bar{u} T_s)}}{j!} \geq \eta_D,$$

However, if the desired delay constraint is such that $\lfloor \frac{t_D}{T_s} \rfloor \leq \frac{\pi r}{2\bar{u} T_s}$, the minimum fraction of mobile nodes can be found by finding the minimum λ_m which satisfies the following inequality:

$$\lambda_m - \frac{\log(f_1(k-1) + \lambda_m f_2(k-1))}{2\lambda r \lfloor \frac{t_D}{T_s} \rfloor \bar{u} T_s} \geq \frac{-\log(1 - \eta_D) - \lambda\pi r^2}{2\lfloor \frac{t_D}{T_s} \rfloor \lambda r \bar{u} T_s}$$

where $f_1(k-1) = \sum_{j=0}^{k-1} \frac{(\lambda\pi r^2)^j}{j!}$ and $f_2(k-1) = \frac{2r \lfloor \frac{t_D}{T_s} \rfloor \bar{u} T_s}{\pi r^2} \sum_{j=1}^{k-1} \frac{(\lambda\pi r^2)^j}{(j-1)!}$.

6.3.2 Mean first contact distance for single-sensing detection

An important measure to evaluate the quality of the target detection is to analyze the mean distance between the target and the closest point (to the target) covered by at least one sensor by the sensor network at any time instant. This is called the first contact distance of the target with single-sensing. When there are mobile nodes in the network, this measure essentially reflects how fast each point in the sensor network is covered over time.

The following theorem states the mean length of the first contact distance for single-sensing detection.

Theorem 3. (Mean first contact distance) *Denote $X^1(nT_s)$ to be the distance between the target, located at any arbitrary point in region \mathcal{R} , and the closest point covered by the sensor network by at least one sensor at time $t = nT_s$. Denote $\bar{X}^1(nT_s) = \mathbb{E}\{X^1(nT_s)\}$ to be the corresponding mean distance. Then $\bar{X}^1(nT_s)$ is given by,*

$$\bar{X}^1(nT_s) = \begin{cases} \frac{1}{\sqrt{\lambda}} e^{\frac{1}{\pi} \lambda \lambda_m^2 \bar{v}^2 n^2 T_s^2} Q\left(\sqrt{\frac{\lambda}{2\pi}} (2\pi r + 2\lambda_m \bar{v} n T_s)\right), & \text{if } nT_s \leq T_0 \\ \frac{1}{\sqrt{\lambda}} e^{\frac{1}{\pi} \lambda \lambda_m^2 \bar{v}^2 T_0^2} Q\left(\sqrt{\frac{\lambda}{2\pi}} (2\pi r + 2\lambda_m \bar{v} T_0)\right), & \text{if } nT_s > T_0 \end{cases}$$

and is upper bounded by,

$$\bar{X}^1(nT_s) \leq \begin{cases} \frac{1}{2\sqrt{\lambda}} e^{-\lambda(\pi r^2 + 2\lambda_m r \bar{v} n T_s)}, & \text{if } nT_s \leq T_0 \\ \frac{1}{2\sqrt{\lambda}} e^{-\lambda(\pi r^2 + 2\lambda_m r \bar{v} T_0)}, & \text{if } nT_s > T_0 \end{cases} \quad (6.6)$$

where Q -function is defined as $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt$.

To prove this theorem we use the following theorem regarding set intersection whose proof can be found in [55]. An isotropic random set is a set in which the distribution is invariant under independent and uniform rotations.

Theorem 4. *Consider the Boolean model as defined in section 6.2.4 with the shapes S distributed as isotropic convex sets. Let S_0 be a fixed convex subset in \mathbb{R}^2 . Then the number of sets in the Boolean model that intersects S_0 is poisson distributed with mean $\lambda(\mathcal{A}(S_0) + \mathbb{E}\{\mathcal{A}(S)\}) + \frac{1}{2\pi} \mathcal{P}(S_0) \mathbb{E}\{\mathcal{P}(S)\}$.*

Proof. (Theorem 3) Let the stationary target be located at any arbitrary point $P_0 \in \mathcal{R}$. Let $P_0 + S_0(x)$ represents the disk centered at P_0 with a shape defined by $S_0(x)$ with a radius of x . Let the distance between P_0 and the closest (to P_0) point covered

by the sensor network at time nT_s be $X^1(nT_s)$. Then the probability of $X^1(nT_s) > x$ is equivalent to,

$$\begin{aligned} Pr(X^1(nT_s) > x) &= Pr(\text{no set intersects the disk } P_0 + S_0(x) \text{ at time } nT_s) \\ &= e^{-\lambda(\mathcal{A}(S_0(x)) + \mathbb{E}\{\mathcal{A}(S(nT_s))\}) + \frac{1}{2\pi}\mathcal{P}(S_0(x))\mathbb{E}\{\mathcal{P}(S(nT_s))\}} \end{aligned} \quad (6.7)$$

where last step is obtained by applying theorem 4. In our case, $\mathcal{A}(S_0(x)) = \pi x^2$, $\mathbb{E}\{\mathcal{A}(S(nT_s))\}$ equals to $\pi r^2 + 2\lambda_m r \bar{v} n T_s$ if $nT_s \leq T_0$ and $\pi r^2 + 2\lambda_m r \bar{v} T_0$ if $nT_s > T_0$, $\mathcal{P}(S_0(x)) = 2\pi x$ and $\mathbb{E}\{\mathcal{P}(S(nT_s))\}$ equals to $2\pi r + 2\lambda_m \bar{v} n T_s$ if $nT_s \leq T_0$ and $2\pi r + 2\lambda_m \bar{v} T_0$ if $nT_s > T_0$. Hence the mean distance $\bar{X}^1(nT_s)$ equals to

$$\begin{aligned} \bar{X}^1(nT_s) &= \mathbb{E}\{X^1(nT_s)\} = \int_0^\infty Pr(X^1(nT_s) > x) dx \\ &= \begin{cases} \frac{1}{\sqrt{\lambda}} e^{\frac{1}{\pi} \lambda \lambda_m^2 \bar{v}^2 n^2 T_s^2} Q\left(\sqrt{\frac{\lambda}{2\pi}} (2\pi r + 2\lambda_m \bar{v} n T_s)\right), & \text{if } nT_s \leq T_0 \\ \frac{1}{\sqrt{\lambda}} e^{\frac{1}{\pi} \lambda \lambda_m^2 \bar{v}^2 T_0^2} Q\left(\sqrt{\frac{\lambda}{2\pi}} (2\pi r + 2\lambda_m \bar{v} T_0)\right), & \text{if } nT_s > T_0 \end{cases} \end{aligned} \quad (6.8)$$

where last step results by using (6.7). The upper bounds in (6.6) for $\bar{X}^1(nT_s)$ (for $nT_s \leq T_0$ and $nT_s > T_0$) are obtained by applying the upper bound for the Q -function, $Q(x) \leq \frac{1}{2} e^{-\frac{x^2}{2}}$ in (6.8). \square

For $nT_s \leq T_0$, from (6.6) it can be seen that when λ_m or n is increased, the mean length of the first contact distance is decreased for fixed λ and r . On the other hand, if there is only a stationary sensor network, $\bar{X}^1(nT_s)$ can be decreased by only increasing either λ or r . Note that, (6.6) shows the proper trade-off between $\bar{X}^1(nT_s)$, λ_m and n when the total node density λ and r are fixed.

6.3.3 Detection latency

In a hybrid sensor network embedded with mobile nodes, it is important to analyze the time delay till the target is first detected after appearing in the sensor network,

which is called the *detection latency* [31]. This measure essentially reflects the monitoring capability and how fast the target can be detected by allowing nodes to be mobile. First, we explore the dependence of the detection latency on mobile node density with single-sensing detection. In this discussion we assume that the target needs to be detected before mobile nodes leave the sensing region and the average time that a mobile node requires to leave the region under mobility model 1, T_0 , is sufficiently large.

Theorem 5. (Average detection latency for single-sensing detection) *Define the random variable τ_1 to be the time until the target is first detected by the hybrid sensor network with single-sensing. Then the average detection latency $\bar{\tau}_1$ of the hybrid sensor network in single-sensing detection (when $T_0 \lim_{\rightarrow} \infty$) is given by,*

$$\bar{\tau}_1 = \frac{e^{-\lambda\pi r^2}}{2\lambda\lambda_m\bar{v}r}. \quad (6.9)$$

It can be seen from (6.9) that for a given total node density λ and sensing range r , the average detection latency can be reduced by increasing the fraction of mobile nodes λ_m or speed of mobile nodes, \bar{v} .

Proof. Let τ_1 be the random variable which represents the time until the target is first detected by the hybrid sensor network with single-sensing. Then we have,

$$\begin{aligned} Pr(\tau_1 > t) &= Pr(\text{target is not detected until time } t (\leq T_0)) \\ &= e^{-\lambda(\pi r^2 + 2r\lambda_m\bar{v}t)} \end{aligned} \quad (6.10)$$

Then the mean value of τ_1 is given by,

$$\begin{aligned} \bar{\tau}_1 &= \int_0^{T_0} Pr(\tau_1 > t) dt = \frac{e^{-\lambda\pi r^2}}{2\lambda\lambda_m\bar{v}r} (1 - e^{-2r\lambda\lambda_m\bar{v}T_0}), \\ T_0 \lim_{\rightarrow} \bar{\tau}_1 &= \frac{e^{-\lambda\pi r^2}}{2\lambda\lambda_m\bar{v}r} \end{aligned} \quad (6.11)$$

□

Detection latency with k -sensing detection for $T_0 \lim_{\rightarrow} \infty$ is given by the following theorem.

Theorem 6. (Average detection latency for k -sensing detection) *The average detection latency with k -sensing detection is given by*

$$\bar{\tau}_k = \frac{e^{-\lambda\pi r^2}}{2\lambda\lambda_m\bar{v}r} \sum_{j=0}^{k-1} (\lambda\pi r^2)^j \tilde{f}(j), \quad (6.12)$$

where $\tilde{f}(j) = \sum_{i=0}^j \frac{1}{(j-i)!} \left(\frac{1}{\pi r^2}\right)^i$

Proof. Let τ_k be the random variable which represents the time until the target is first detected by the hybrid sensor network with k -sensing. Then $Pr(\tau_k > t)$ is given by,

$$\begin{aligned} Pr(\tau_k > t) &= Pr(\text{the target is not detected by } k - \text{sensing until time } t) \\ &= \sum_{j=0}^{k-1} \frac{[\lambda(\pi r^2 + 2\lambda_m\bar{v}rt)]^j}{j!} e^{-\lambda(\pi r^2 + 2\lambda_m\bar{v}rt)}, \end{aligned} \quad (6.13)$$

Then we have,

$$\begin{aligned} \bar{\tau}_k &= \int_0^\infty \sum_{j=0}^{k-1} \frac{[\lambda(\pi r^2 + 2\lambda_m\bar{v}rt)]^j}{j!} e^{-\lambda(\pi r^2 + 2\lambda_m\bar{v}rt)} dt \\ &= \sum_{j=0}^{k-1} \frac{\lambda^j}{j!} e^{-\lambda\pi r^2} \int_0^\infty [\pi r^2 + 2\lambda_m\bar{v}rt]^j e^{-2\lambda\lambda_m\bar{v}rt} dt \\ &= \frac{e^{-\lambda\pi r^2}}{2\lambda\lambda_m\bar{v}r} \sum_{j=0}^{k-1} (\lambda\pi r^2)^j \sum_{i=0}^j \frac{1}{(j-i)!} \left(\frac{1}{\pi r^2}\right)^i, \end{aligned}$$

where we have used the integral identity $\int_0^\infty x^{i-1} e^{-x} dx = \Gamma(i) = (i-1)!$ for an integer i where $\Gamma(\cdot)$ is the Gamma function. □

6.4 Detection Performance with Random Node Mobility Model 2 (Random Walk)

Now we consider that the mobile nodes follow 2-D random walk mobility model at each time step nT_s which is more practical mobility model used in network performance analysis [89], as shown in Fig. 6.1. Let us assume that the sensing region can be viewed as a virtual square lattice having a total of $\approx \frac{b^2}{\mu^2}$ square sites where $\mu = \bar{u}T_s$ is the lattice side length. The k -th mobile node is assumed to be at the center of a site. If the mobile node starts to move at time $t = 0$, the expected number of distinct sites visited by time nT_s , $\mathbb{E}\{G(nT_s)\}$ can be approximated by [20, 77],

$$\mathbb{E}\{G(nT_s)\} \approx \frac{b^2}{\mu^2} \left(1 - \left(\frac{cb^2}{\mu^2} \right)^{-\frac{\pi nT_s}{\mu^2 \log^2 \left(\frac{cb^2}{\mu^2} \right)}} \right),$$

where $c = 1.8456\dots$

In this Chapter we consider only the case $r \leq \mu$, since if the step size μ is selected such that $\mu \ll r$, there are large overlaps in the sensing areas at consecutive steps [77]. Thus it is more desirable to select step size of the random walk such that $\mu \geq r$, which reduces the overlapping of the coverage areas at consecutive steps of random walk. Since each mobile node performs independent and identical random walks at each time step, and the sensing range of each mobile node is identical, it can be seen that, $\{C_k^m(nT_s)\}_{k \in \mathcal{V}_m}$ are a set of independently and identically distributed random sets where $C_k^m(nT_s)$ is the area covered by the k -th mobile node at time nT_s . Denote $\bar{C}_k^m(nT_s) = \bar{C}^m(nT_s)$ to be the average coverage area of the k -th mobile node at time nT_s . The average area covered by a mobile node at time nT_s , $\bar{C}^m(nT_s)$ is then given by the following theorem.

Theorem 7. (Minimum average coverage area of a mobile node) *Assuming that $\mu \geq r$, the minimum average area covered by any single mobile node at time nT_s is*

given by,

$$\begin{aligned}\bar{C}_{min}^m(nT_s) &= \pi r^2 + (\mathbb{E}\{G(nT_s)\} - 1)^+ 2r\mu \\ &\quad - (\mathbb{E}\{G(nT_s)\} - 2)^+ (1 - \frac{\pi}{4})r^2.\end{aligned}\tag{6.14}$$

Proof. Assuming $\mu \geq r$, when there is $\mathbb{E}\{G(nT_s)\}$ number of distinct sites visited at time nT_s , there should be at least $\mathbb{E}\{G(nT_s)\} - 1$ number of steps to ensure that each point is connected to at least one lattice point (see Fig. 6.4). Then the minimum coverage area results if these lattice points are located such that each transition is orthogonal to the previous transition (That is, then the maximum amount of overlapping will occur with the minimum number of transitions). Figure 6.4 shows the realization of random walk when 4 distinct sites are visited with minimum number of 3 transitions. Left plot in Fig. 6.4 is corresponding to $r \leq \frac{\mu}{2}$, where there is no overlapping of the sensing range while the right plot in Fig. 6.4 corresponds to $\frac{\mu}{2} \leq r < \mu$ where there is overlapping of sensing range, between two consecutive steps. Based on geometric simplifications, in both cases as shown in Fig. 6.4, the minimum coverage area can be shown as,

$$\begin{aligned}\bar{C}_{min}^m(nT_s) &= \pi r^2 + (\mathbb{E}\{G(nT_s)\} - 1)2r\bar{u}T_s \\ &\quad - (\mathbb{E}\{G(nT_s)\} - 2)(1 - \frac{\pi}{4})r^2,\end{aligned}$$

which completes the proof. □

Then lower bounds for the detection probability in single-sensor and k -sensor detections can be shown as,

$$P_D^1(nT_s) \geq 1 - e^{-\lambda \bar{C}_{min}^m(nT_s)},\tag{6.15}$$

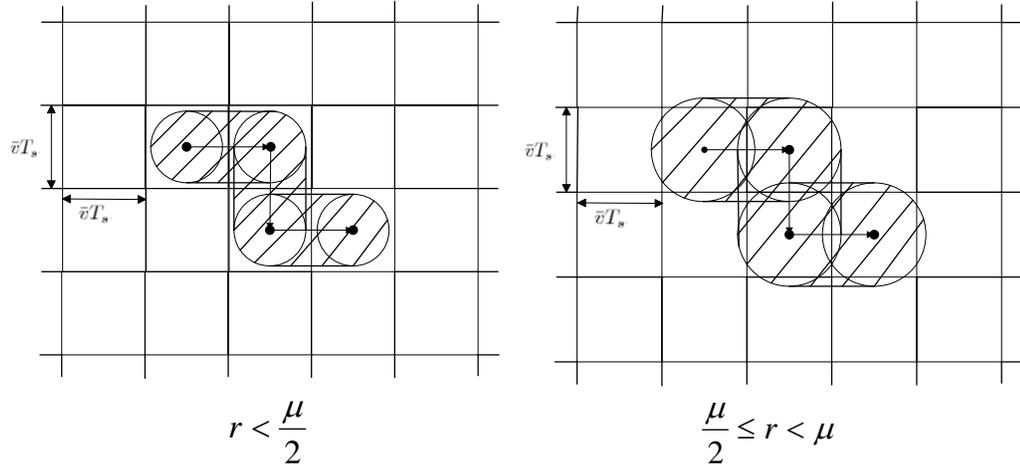


Figure 6.4: Minimum possible coverage area after completing 4 distinct steps, left: $r < \frac{\mu}{2}$, right: $\frac{\mu}{2} \leq r < \mu$

and $P_D^k(nT_s) \geq 1 - \sum_{j=0}^{k-1} \frac{(\lambda \bar{C}_{min}(nT_s))^j e^{-\lambda \bar{C}_{min}(nT_s)}}{j!}$, respectively, with $\bar{C}_{min}(nT_s) = \lambda_m \bar{C}_{min}^m(nT_s) + (1 - \lambda_m)\pi r^2$ where $\bar{C}_{min}^m(nT_s)$ is given by (6.14).

Let η_D be the desired detection probability lower bound to be achieved by the hybrid sensor network at time t_D . The minimum fraction of mobile nodes λ_m^{min} that should be used in order to achieve this probability bound, within the desired time is stated in the following theorem:

Theorem 8. (Minimum fraction of mobile nodes required to achieve a desired probability at a given time) *With single-sensing detection, if the desired detection probability lower bound, η_D , is to be achieved within a time interval t_D , the minimum fraction of mobile nodes that should be deployed in the hybrid network with single-sensing detection is given by*

$$\lambda_m^{min} = \frac{-\log(1 - \eta_D) - \lambda \pi r^2}{\lambda \left(\bar{G}_1(\lfloor \frac{t_D}{T_s} \rfloor T_s) 2r \bar{u} T_s - \bar{G}_2(\lfloor \frac{t_D}{T_s} \rfloor T_s) (1 - \frac{\pi}{4}) r^2 \right)}.$$

for $\mu \geq r$ where $\bar{G}_1(\lfloor \frac{t_D}{T_s} \rfloor T_s) = (\mathbb{E}\{G(\lfloor \frac{t_D}{T_s} \rfloor T_s)\} - 1)$ and $\bar{G}_2(\lfloor \frac{t_D}{T_s} \rfloor T_s) = (\mathbb{E}\{G(\lfloor \frac{t_D}{T_s} \rfloor T_s)\} - 2)$. ■

Proof. The proof follows directly from (6.15) and (6.14). □

6.5 Probability of Node Connectivity

When a mobile or static node detects a target, the decisions need to be combined to reach at a final decision at a separate fusion center or static node which acts as a base station. In a hybrid sensor network, to exploit the node mobility on the detection performance effectively, it is important that each mobile node is always connected to at least one static node or another mobile node. We assume that node x is connected to the node y (i.e., node x can communicate with node y , (may not be bi-directional necessarily)), if node y is within the communication range of node x . For bi-directional communication between x and y , node x also has to be within the communication range of node y . In the following we consider these scenarios separately. Since mobile nodes have to spend energy for mobility in addition to sensing and communication we assume that mobile nodes have relatively smaller communication range compared to that with static nodes. Let r_s and r_m be effective communication ranges of a static and a mobile node, respectively with $r_s \geq r_m$. In the following we consider the probability of network connectivity at any time instant under several communication architectures. In the following discussions, we assume that the sensing region is large enough so that T_0 (as defined before) is large. Then, under both random mobility models considered in the Chapter, it can be assumed that at any given time ($< T_0$), node locations follow a 2-dimensional Poisson point process with the same intensity as that with the initial PPP.

In the case where it is desirable for mobile nodes to communicate only with

static nodes at any given time, we find an approximation for the probability that each mobile node is connected to at least one static node at time $t \leq T_0$, $P_c^{m,s}(t)$. Consider an arbitrary mobile node m_k in the sensor network located at $\mathbf{r}_{m,k}(t)$ at time $t \leq T_0$. Then if there is at least one static node within its communication range at time $t \leq T_0$, (i.e. within the disk $D_k(\mathbf{r}_{m,k}(t), r_m) \equiv \mathbf{r}_{m,k}(t) + \pi r_m^2$, centered at $\mathbf{r}_{m,k}(t)$ with radius r_m) we say that the mobile node is connected to at least one static node at time t . As mentioned earlier, since node locations follow a 2-D PPP at any time $t \leq T_0$ with the same intensity as the initial point process, the probability that a mobile node is connected to at least one static node at any given time $t \leq T_0$ is given by $1 - e^{-\lambda(1-\lambda_m)\pi r_m^2}$. Since there is N_m number of mobile nodes located independently at any time t , probability that every mobile node is connected to at least one static node at time t is approximated by $P_c^{m,s}(t) \approx \left(1 - e^{-\lambda(1-\lambda_m)\pi r_m^2}\right)^{N_m}$. On the other hand, in some situations it might be required for a mobile node to send its decision to the closest node (either static or mobile) at a given time. A mobile node m_k is connected to at least one node at time $t \leq T_0$ if there is at least one static or mobile node within the disk $D_k(\mathbf{r}_{m,k}(t), r_m)$ at a given time instant $t \leq T_0$. Probability that there is at least one node (static or mobile) within the disk $D_k(\mathbf{r}_{m,k}(t), r_m)$ is given by, $Pr(\text{at least one node in the disk } D_k(\mathbf{r}_{m,k}(t), r_m)) \leq 1 - e^{-\lambda\pi r_m^2}$. Since there is N_m number of mobile nodes independently located at time $t \leq T_0$, the probability that any mobile node is connected to at least one node is given by $P_c^{m,any}(t) \approx \left(1 - e^{-\lambda\pi r_m^2}\right)^{N_m}$. When static nodes receive local decisions from mobile nodes, they need to combine them (with their own decisions) to make a final decision. Thus it is required to determine the transmission range of static nodes such that they can communicate with each other. Following a similar approach as before, it can be shown that the probability that each static node is connected to at least one another static node is approximated by $P_c^{s,s} \approx \left(1 - e^{-\lambda(1-\lambda_m)\pi r_s^2}\right)^{N_s}$. In cases where mobile nodes and static nodes may need to communicate in a bi-directional way to exchange their local information at time t , (that is a mobile node

may talk to any static node as well as static node may talk to a mobile node to exchange information at time t) it is necessary to maintain that no node is isolated in the network at any given time. The probability that any node is connected to at least one another node in the network at time $t \leq T_0$ can be approximated as $P_c(t) \approx \left(1 - e^{-\lambda\pi r_m^2}\right)^{N_m} \cdot \left(1 - e^{-\lambda\pi[(1-\lambda_m)r_s^2 + \lambda_m r_m^2]}\right)^{N_s}$.

It is important to analyze the trade-offs between the probability of node connectivity at a given time, mobile and static node densities and the communication ranges of nodes in the hybrid sensor network according to the application requirements. For example, let λ_m^D be the required fraction of mobile node density that should be deployed in the network to achieve desired performance level as described earlier in this Chapter according to specific mobility models. According to the architecture in which mobile nodes can communicate only with static nodes, assume that the network needs to maintain the connectivity between each mobile node and at least one static node with a probability of $\tilde{P}_c^{m,s}$ at a given time $t \leq T_0$. Then, as has been shown in [13], the trade-off between the transmission range of mobile nodes and the connectivity probability can be obtained using the probability of connectivity discussed in Section 6.5. On the other hand, for a given transmission range r_m for a mobile node, the mobile node density required such that a mobile node is connected to at least one static node with a probability of $\tilde{P}_c^{m,s}$ at a given time $t \leq T_0$ is given by

$$\lambda_m \approx 1 + \frac{\log(1 - (\tilde{P}_c^{m,s})^{1/N_m})}{\lambda\pi r_m^2}. \quad (6.16)$$

Similarly, the analysis on selecting required parameters can be performed to maintain other communication architectures discussed before.

6.6 Simulation Results

6.6.1 With node mobility model 1

We verify the analytical results obtained in this Chapter via extensive Monte-Carlo simulations. The dimension of the sensing area is assumed to be $b = 1000m$, such that area is $1000 \times 1000m^2$. Unless specified, for each figure in the following, 10^5 Monte-Carlo runs were performed. Mobile node velocity is set to $\bar{v} = 1m/s$. Initially a total of $N = 500$ sensor nodes are deployed independently and uniformly in the sensing field, such that the node density $\lambda = 0.0005$. A fraction λ_m of 500 total nodes, is directed to move according to the random mobility model 1 as described in subsection 6.2.2. With these assumed parameters, it can be shown that the average time a mobile node takes to leave the sensing region with the mobility model 1 is, $T_0 = 473.31655s$.

In the first experiment, the time varying detection probability is investigated when the fraction of mobile nodes is varying for a given sensing range for mobile and static nodes. Figure 6.5 shows the analytical and simulated results which reflect the time varying detection probability of the hybrid sensor network for single-sensing detection when the fraction of mobile nodes deployed is varied. In Fig. 6.5, we assume that $r = 20m$. From Fig. 6.5, we can see the derived analytical results almost exactly match with the simulation results for $nT_s \leq T_0$ and $nT_s > T_0$. It can be seen from Fig. 6.5, that after a certain time period, the detection probability reaches a steady state, which essentially means that the area is maximally covered by the mobile nodes (with static nodes) before they leave the sensing region. Interestingly, we see that when the fraction of mobile nodes is increasing, this steady state probability becomes 1 and it is reached well before the nodes leave the sensing region. This means that when λ_m increases, the network can be completely covered by the hybrid network within a shorter time (compared to T_0) with the mobility model 1. This

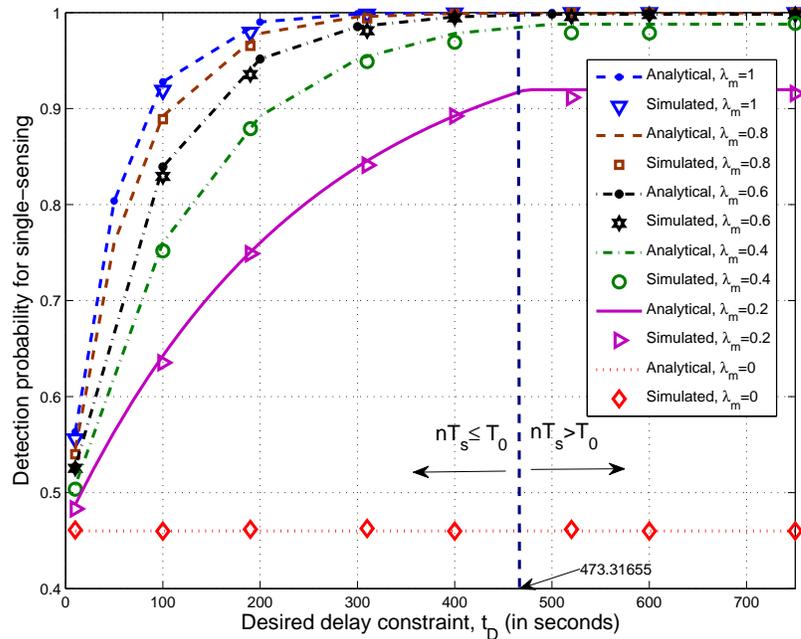


Figure 6.5: Detection probability with single-sensing detection Vs desired delay constraint with mobility model 1: $r = 20m$

phenomenon essentially reflects the trade-off between the fraction of mobile nodes and the probability of detecting the target before it disappears in the field. For example, if the target appearing time is less, to detect it before disappearing, it is desired that the total area is covered as quickly as possible, which needs a relatively larger fraction of mobile nodes. On the other hand, if the target appearing time is longer, then with a relatively small number of nodes is enough to cover the area with the desired quality. Also it is noted from Fig. 6.5 that, at earlier time intervals before the probability reaches steady state, the detection probability has rapid increment compared to the stationary configuration, and increases slowly as it approaches the steady state probability. Moreover, it is seen for Fig. 6.5 that by adding a small fraction of mobile nodes will boost the detection performance significantly compared to the stationary configuration, and the rate of performance improvement eventually

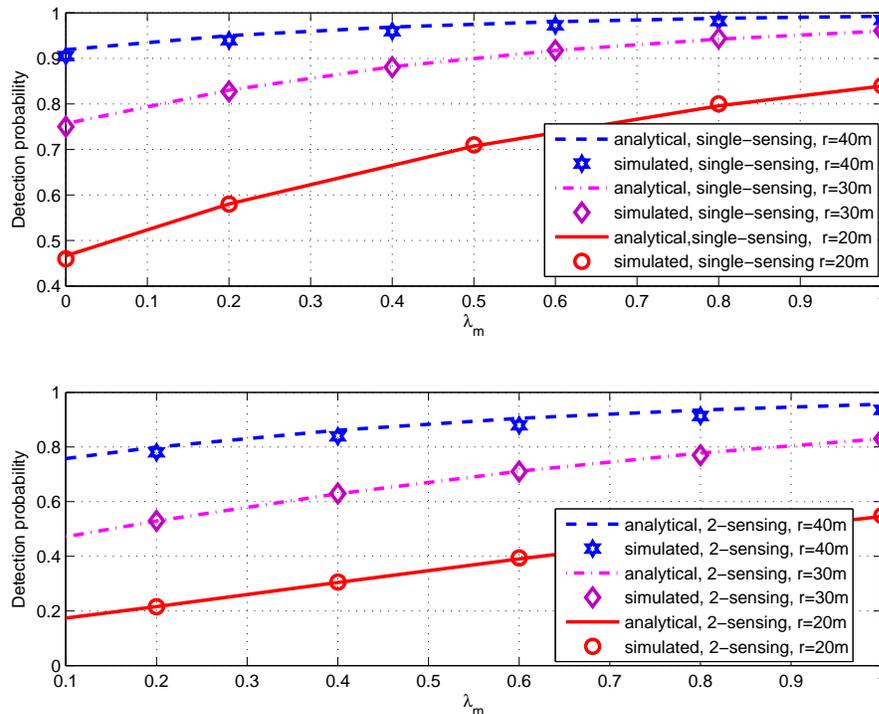


Figure 6.6: Detection probability Vs fraction of mobile nodes in the network for single-sensing and 2-sensing detection models for mobility model 1; Desired detection delay is $t_D = 60s$.

decreases as λ_m increases.

In the next experiment, the detection performance is evaluated with varying sensing ranges for single-sensing and 2-sensing detection models. Figure 6.6 shows the detection probabilities for single-sensing (top plot) and 2-sensing (bottom plot) detection models of the hybrid sensor network Vs the fraction of mobile nodes for a given desired delay constraint, when the sensing range is varied. In Fig. 6.6 we let the delay constraint $t_D = 60s < T_0$ in which the network has not reached the steady state performance. Note that, with mobility model 1, our interest is more on the dynamic performance results in the hybrid network before it reaches the steady

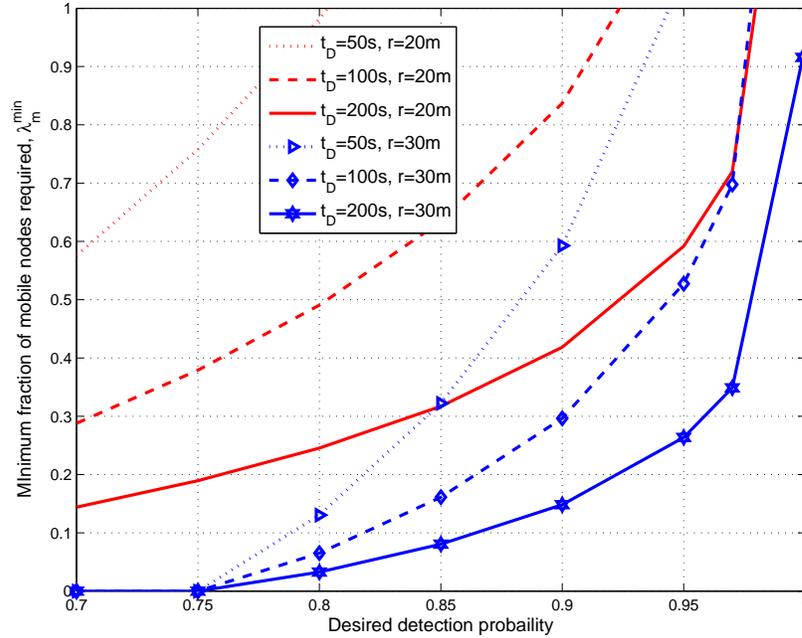


Figure 6.7: Minimum fraction of mobile nodes required to achieve a desired performance level within a desired delay constraint for mobility model 1

state (i.e. before the mobile nodes leave the sensing region). Different plots in Fig. 6.6 are corresponding to varying sensing ranges (for $r = 20m$, $r = 30m$ and $r = 40m$). From Fig 6.6, it can be seen that the derived analytical results perfectly match with the simulation results. It can also be seen that the detection probability is nearly-linearly increasing, when the fraction of mobile nodes is increasing, for a given sensing range around the considered delay constraint (i.e. around relatively lower delay constraints). Also, when the sensing range is increasing the increment in the detection probabilities over λ_m occurs at a lower rate for both single and 2-sensing detection models.

In Fig. 6.7, the minimum fraction of mobile nodes required to achieve a desired performance level within a desired delay constraint ($< T_0$) is shown for $r = 20m$

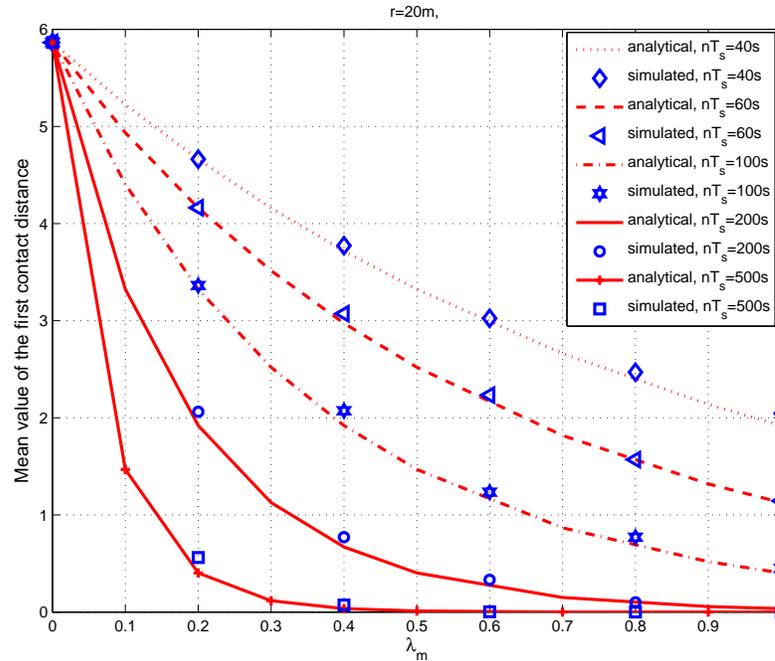


Figure 6.8: Mean value of the first contact distance for single-sensing detection with mobility model 1; $r = 20m$, $\bar{u} = 1m/s$

and $r = 30m$ with single sensing detection. It is seen that when the desired delay constraint is small, the minimum fraction of mobile nodes is increasing to achieve a desired performance level. Moreover, the effect of the mobile node density on the detection performance is more significant when the sensing range of the nodes is low, which is the most practical scenario in many sensor networks. In other words, it can be seen from Fig. 6.7 that when the sensing range is increasing, the variation of the required fractions of mobile nodes to achieve different detection thresholds, is less compared to that with lower sensing ranges.

The next experiment is performed to evaluate the performance of hybrid sensor network in terms of the mean first contact distance at a given time. Figure 6.8 shows the performance of the mean first contact distance derived in subsection 6.3.2, with

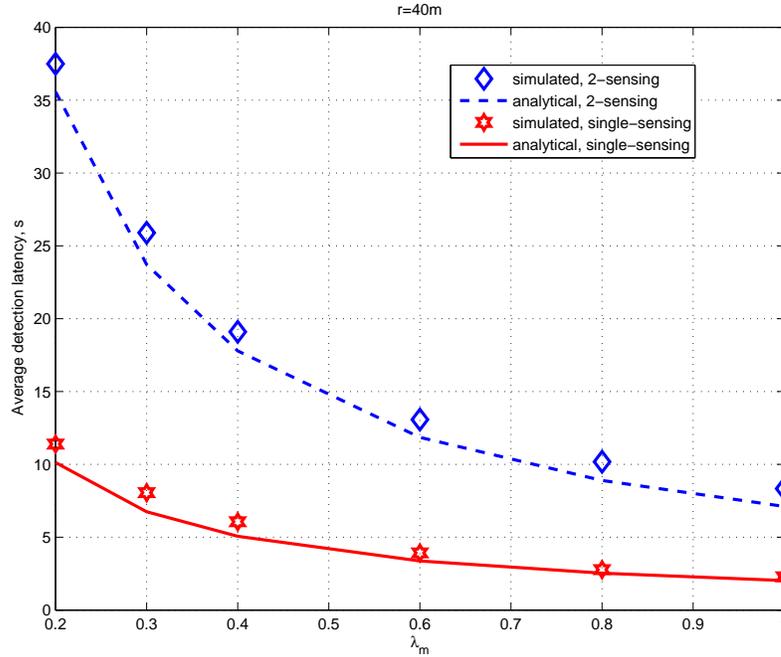


Figure 6.9: Average detection latency for single-sensing and 2-sensing detection models with mobility model 1; $r = 40m$, $\bar{u} = 1m/s$

the mobile node density. In Figure 6.8, we let $r = 20m$ and plots are corresponding to different delay constraints. From Fig. 6.8, it can be seen that the derived results for the mean first contact distance fairly match with the simulated results. Note that the mean first contact distance at a given time essentially means that how much, in average, a given arbitrary point is closer to any point in the network covered by at least one sensor at a given time. It can be seen from Fig. 6.8, as the time elapsed, any arbitrary point is getting closer to a point covered by the sensor network by at least one sensor much faster until a certain fraction of mobile nodes, and after that the mean distance reaches slowly to zero. This essentially reflects the proper trade-off between the fraction of mobile nodes required and the delay constraint in order to cover any arbitrary point in the network as time goes.

Figure 6.9 depicts the average detection latency for single-sensing and 2-sensing detection models with the fraction of mobile nodes. It can be seen that, for a given sensing range, with a smaller fraction of mobile nodes, the average delay of detection with 2-sensing model is significantly increased compared to that with single-sensing model. However, as λ_m is increasing, the difference of average detection delays of two sensing models becomes smaller. This essentially implies that to obtain the system performance with a higher confidence level (increasing k) with a smaller fraction of mobile nodes, it is required to wait a longer time compared to that with single-sensing model (lower or minimum possible performance level). Moreover, as λ_m increases, the average detection latency required to achieve a performance level with a higher confidence, is not significantly long compared to single-sensing detection model.

From the results in the Figures 6.5, 6.6, 6.7, 6.8, 6.9 it can be seen that the Boolean model is a good approximation for the hybrid sensor network considered in this Chapter when the number of nodes and the sensing area are relatively large. To further illustrate the suitability of Boolean/Poisson model with reduced number of nodes and network sizes, in Fig. 6.10 we plot the time varying detection probability for $b = 500m$ and $N = 125$ such that the node density is still $\lambda = 0.0005$. With these parameter values, it can be shown that the average time that a mobile node needs to leave the sensing region, $T_0 = 236.4925s$. From Fig. 6.10, it can be seen that the Boolean approximation does not give very accurate results when N and b^2 are relatively low.

6.6.2 With node mobility model 2

To see the performance of the derived detection probability lower bound, we perform Monte-Carlo simulations to obtain the exact detection probability with random walk mobility model. Figure 6.11 shows the analytical detection probability lower bound

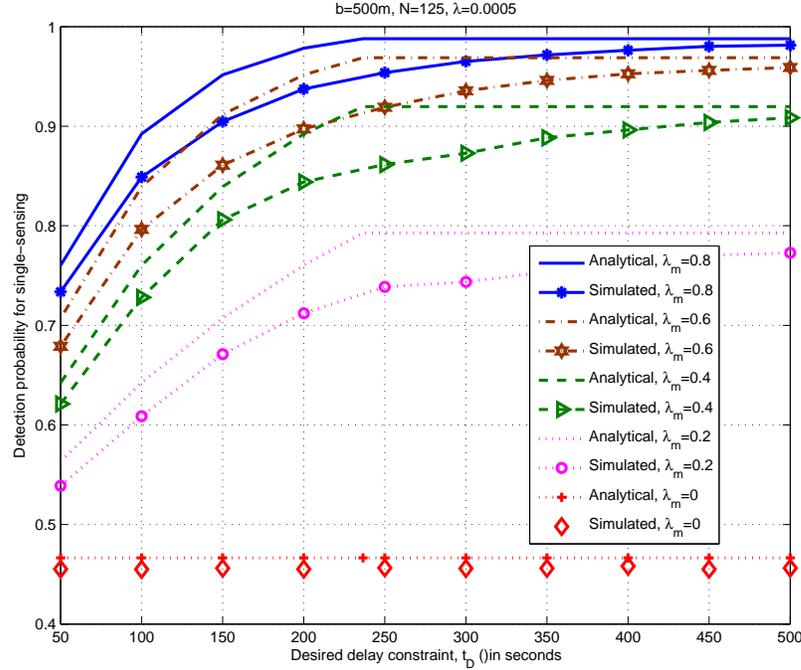


Figure 6.10: Detection probability with single-sensing detection Vs desired delay constraint with mobility model 1: $r = 20m$, $b = 500m$, $N = 125$, $\lambda = 0.0005$

and the exact detection probability vs the fraction of mobile nodes, with random walk mobility model after completing $n = 20$ steps. In Fig. 6.11, we let the step sizes of the random walk to be $\mu = \sqrt{2}r$ and $\mu = 2r$ where r is set to $r = 20m$. From Fig. 6.11, it can be seen that the derived lower bound is a good match for the exact detection probability. Moreover, when the step size of the random walk is selected relatively larger compared to the sensing radius of the node, it can be seen that the derived lower bound becomes much tighter for the exact detection probability. For a given sensing range, selecting a larger step size compared to the sensing range is more desirable in performing 2-D random walk, since then the overlapping of sensing coverage at consecutive steps is reduced.

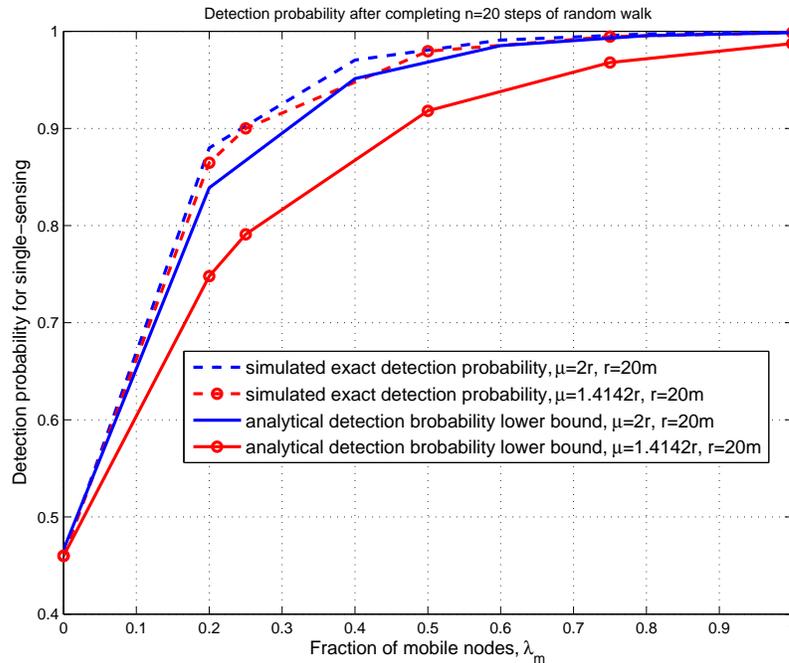
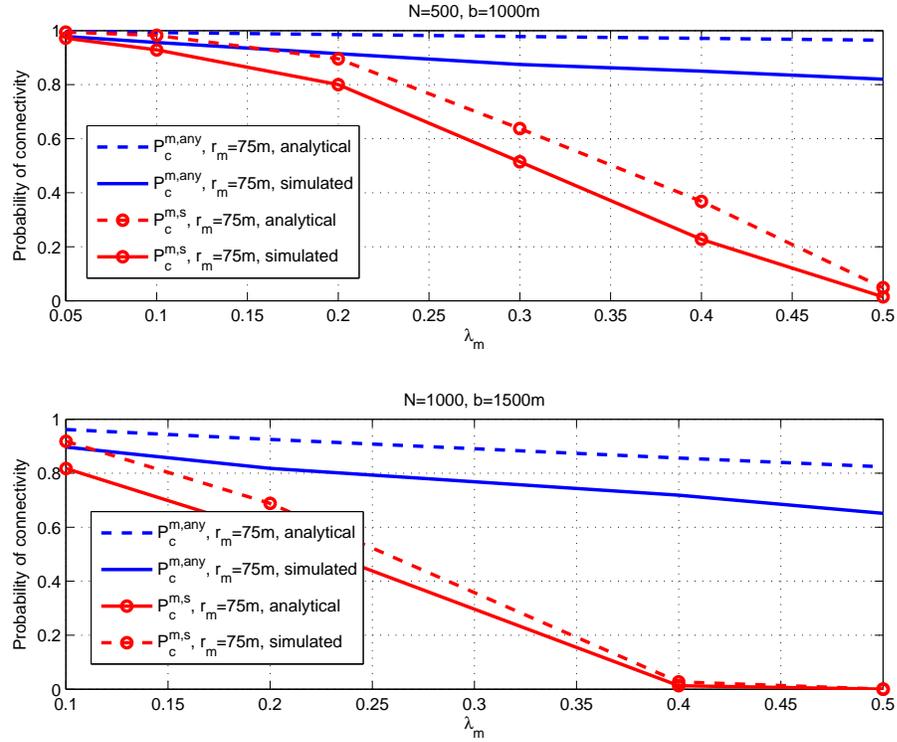


Figure 6.11: Detection probability lower bound with single-sensing detection Vs fraction of mobile nodes in the network with random walk mobility model after completing $n = 20$ steps: for $\mu = \sqrt{2}r$ and $\mu = 2r$: $r = 20m$

6.6.3 Node connectivity

Note that as mentioned earlier, with both random mobility models considered, at a given time instant node locations form a Poisson point process when the network size is sufficiently large. Thus the connectivity probabilities remain the same over time. To evaluate the performance of the derived approximations of the probability of node connectivity, we perform computer simulations for different communication architectures. In Figure 6.12, the simulated probability of connectivity and the analytical probability approximations for several architectures in Section 6.5 are shown; i.e. probability that each mobile node is connected to at least one static node $P_c^{m,s}$ and the probability that each mobile node is connected to at least one another node


 Figure 6.12: Probability of node connectivity for $r_m = 75m$ and $r_m = 100m$

(either static or mobile) $P_c^{m,any}$. In Figure 6.12, two subplots correspond to two different network sizes. In the top plot we assume that $N = 500$ and $b = 1000m$ while in the bottom plot $N = 1000$ and $b = 1500m$. The transmission radius of mobile nodes is assumed to be, $r_m = 75m$. From the simulation results, although it is seen that the approximations are quite far from simulation results, both give similar characteristics when λ_m is increasing, for both network sizes. In particular, it can be seen from Fig. 6.12 that, when the fraction of mobile nodes is increasing, the probability $P_c^{m,s}$ is getting smaller. That is because, when λ_m is increasing, the fraction of static nodes is decreasing (since we keep the total number of sensors constant) resulting a lower probability that a static node lies in the communication range of a mobile node. On the other hand, when mobile nodes are allowed to communicate with any other node

(mobile or static, $P_c^{m,any}$), a higher connectivity probability for all mobile nodes can be obtained compared to allowing to communicate with only static nodes. Note that from the results in subsections 9.6.1 and 6.6.2, it was observed that when the fraction of mobile nodes is getting larger, an improved performance gains are obtained in either perspective. But, in terms of the connectivity, increasing λ_m will reduce the probability of node connectivity in the hybrid network. Although figures are not included, it can be observed that by increasing the communication range of mobile nodes, r_m , a significant improvement in connectivity probabilities are obtained even at relatively large λ_m . However, allowing larger communication ranges for mobile nodes is not desirable. Therefore, it is required to properly design hybrid sensor networks taking these trade-offs between the probabilities of node connectivity and the performance gains into account.

6.7 Conclusions

In this Chapter, the impact of mobile node density on the detection performance in different perspectives and on the node connectivity of a hybrid sensor network consisting of both static and mobile nodes is addressed. We consider two random mobility models for mobile nodes where in the first one, mobile nodes move on a straight line after selecting a random direction initially and in the second one, mobile node follow a 2-D random walk. With the mobility model 1, we derived the detection performance, in terms of detection probability, detection latency and mean first contact distance for single-sensing and k -sensing detection models of the hybrid sensor network. With mobility model 2, we derived reasonable approximations for the average coverage area and the detection probability for single and k sensing detection models. We investigated the trade-off between the mobile node density and the desired (exact or approximated) performance gain with given constraints.

Moreover, the trade-off between mobile node density and the probability of node connectivity at a given time is analyzed based on the derived approximations for probability of node connectivity. The analytical results derived in this Chapter help to select design parameters in hybrid sensor networks for on-demand application requirements.

Chapter 7

Interactive Distributive Mobility Protocol for Mobile Node Navigation

7.1 Introduction

In Chapter 6, we considered a mobility assisted sensor network in which the node mobility is assumed to be random. As mentioned in Chapter 6, independent random mobility models are widely used in performance analysis in sensor networks especially when a priori information regarding the sensing field is unknown. Although random mobility models are desirable in many applications, and they need minimum coordinations among nodes, they may not always be ideal for hybrid networks consisting of both static and mobile nodes due to following reasons:

- In a hybrid sensor network, as mentioned earlier, a certain portion of the field is covered always (as shown by the union of checked circles in Fig. 7.1). Thus

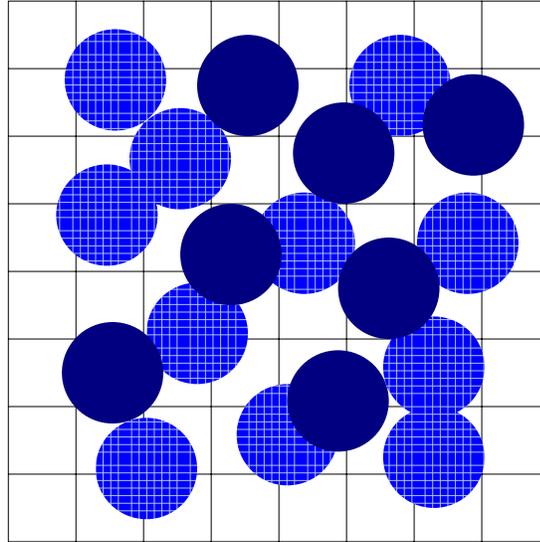


Figure 7.1: Hybrid sensor network consisting of both static and mobile nodes: solid circles-mobile nodes, checked circles-static nodes

it is required to use mobile nodes to cover only the areas uncovered by static nodes minimizing the overlapping between the mobile and static nodes' sensing ranges.

- When nodes are mobile, previously covered areas by mobile nodes become uncovered and uncovered areas become covered. Thus it is desirable to manage the mobility of the mobile nodes such that to minimize the duration that a particular location is uncovered. Random mobility schemes do not address these problems.
- If the network does not have any prior knowledge about the sensing field, it is desired that any point not covered by the static nodes is covered almost equally to maintain an approximately uniform coverage over time.

In this Chapter, we propose a new distributed mobility protocol for mobile node navigation in a hybrid sensor network by collaborating with static nodes to provide an efficient dynamic coverage for the area not covered by the static nodes. We assume that the sensor network is partitioned into square cells such that a node can cover a cell completely when it is located at the cell center. We divide these cells into two categories: *static* and *void* cells. We define a *static* cell as a cell in which there is at least one static node. A *void* cell is defined as a cell in which there is no any static node. Mobile nodes are directed move among these *void* cells based on a certain criteria. Each of these *void* cells is given a certain *base price* (which is similar to that in [134], however, the criteria for assigning *base price* is different from that in [134]). This *base price* is updated by static nodes based on the time that the *void* cell remains not-covered by at least one mobile node. At each movement step, mobile nodes communicate with their closest static nodes locally to search for *void* cells which are not covered for a long time. Static nodes provide necessary information for mobile nodes in their neighborhoods. At a given time, we assume that a mobile node can visit a certain number of candidate *void* cells from its current position. These candidate *void* cells are determined by the mobile node's maximum speed. Taking base prices (collected from neighboring static nodes) of the candidate *void* cells into account, each mobile node selects the best *void* cell to move by the next time step, as the one that is not covered for a long time. We show, from simulation results, the effectiveness of the proposed scheme in terms of the presence probability matrix and the average time that an arbitrary point in the network is not covered. The presence probability matrix contains the probabilities of the presence of at least one node at each cell at any given time instant.

We further analyze the effectiveness of the proposed mobility scheme in terms of the worst-case detection performance when the network is deployed for target detection applications, which will be addressed in Chapter 8.

The Chapter is organized as follows: Section 7.2 presents the motivation and the network model. In Section 7.3, the proposed mobility protocol is described. Performance results are shown in Section 7.4 and the concluding remarks are given in Section 7.5.

7.2 Motivation and System Model

7.2.1 Motivation

Consider a hybrid sensor network deployed in a square region as shown in Fig. 7.1, where the union of checked circles represents the area covered by static nodes while the union of solid circles represents the area covered by mobile nodes, respectively. When the nodes are first deployed in a region, a random placement is often desirable especially when a priori knowledge of the terrain is unavailable. However, such random deployment strategies may not result effective coverage always, since some nodes might be overly clustered while some of them might be sparsely located [158]. Use of node mobility to reconfigure the node locations to improve the coverage of such networks was addressed by some authors, for example in [134, 135, 158]. In these approaches, nodes move only during the deployment stage and the maximum coverage area achieved by the network after reconfiguration is limited by the number of total nodes and nodes' sensing ranges. For example, if the total number of nodes is relatively small, even by reconfiguration of mobile nodes to provide a uniform coverage, a large portion of the network may be remained not-covered. On the other hand, node failures after the initial reconfiguration might cause coverage holes in the network.

Thus in this Chapter, our focus is on the dynamic coverage provided by mobile nodes after the initial deployment stage. When mobile nodes are used for continuous

movements, locations in the sensing region flip between covered and uncovered as the time goes. Thus, an arbitrary location of the sensor field will have a time varying coverage. However, in a hybrid network a certain portion is covered always due to static nodes. Thus it is required to manage the node mobility such that the dynamic coverage at an arbitrary point, located outside the area covered by static nodes, is maximized. Motivating by these, in this Chapter we propose a distributed protocol for mobile nodes in a hybrid sensor network to provide an efficient dynamic coverage such that the average time that an arbitrary location remains uncovered is minimized.

7.2.2 Network model

We consider a hybrid sensor network made of N total sensor nodes deployed in a region \mathcal{R} with network dimension of $b \times b$. We assume that there are N_s number of static nodes and N_m number of mobile nodes. Denote $\lambda = \frac{N}{b^2}$ to be the spatial density of the nodes and $\lambda_m = \frac{N_m}{N}$ and $\lambda_s = \frac{N_s}{N}$ to be the fractions of mobile and static nodes respectively. Let \mathcal{V} be the set containing all node indices in the network and let \mathcal{V}_m and \mathcal{V}_s be the sets containing mobile and static node indices, respectively.

7.3 Interactive, Distributed Mobility Protocol

Suppose that the sensing region is divided into a square grid with a grid length of $l = \sqrt{2}r$ where r is the effective sensing radius of a sensor. We assume both static and mobile nodes have the same sensing radii and the analysis can be slightly modified to deal with different sensing radii for mobile and static nodes. When a sensor node is located at the center of a cell in the grid the corresponding cell is completely covered by the sensor node. Consider the hybrid network with only static nodes as shown in Fig. 7.2 (dropping the mobile nodes in Fig. 7.1). We denote the set of cells that is

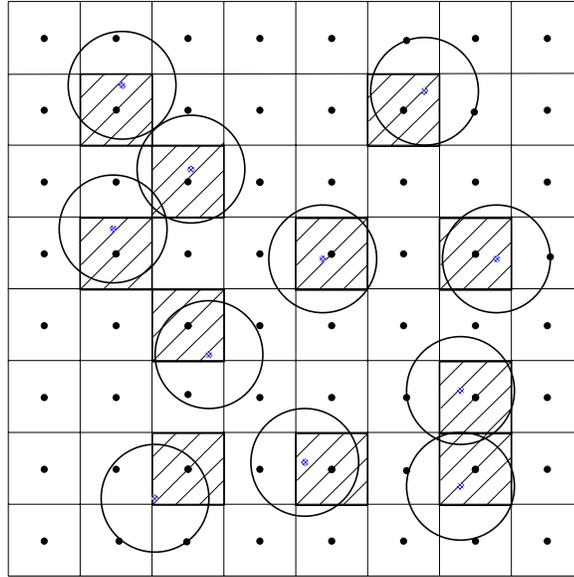


Figure 7.2: Sensor network with only static nodes

not covered by the static nodes as the set of *void* cells as shown in Fig. 7.2 with void squares. When a static node is located in a particular cell (crossed cells in Fig. 7.2) we consider that the corresponding cell is covered by the relevant static node and call that cell a *static* cell. However, note that since a static node does not necessarily locate at the middle of a cell, corresponding cell may not be completely covered by the static node. We address this problem later and for the moment assume that the cell is covered by the corresponding static node. Now the problem is how to use the mobile nodes efficiently to cover the *void* cells as shown in Fig. 7.2 over time such that revisiting time of any cell by at least one mobile node is maximized. In the following, we propose a new distributed interactive protocol to achieve the required task by collaborating among mobile and static nodes.

7.3.1 Distributed protocol for node mobility management

We assign a base price for each *void* cell according to the following rule. Initially, at time $t = 0$, we assign a base price $\mathcal{P} = 0$ for each *void* cell in which there is at least one mobile node. For all the other *void* cells we assign $\mathcal{P} = K$ where K is a large value. Let T_m be the time interval in which the mobility management is performed, which can be determined by the mobile node's maximum speed and the length of a grid. At each time step T_m , the base price of each *void* cell is updated considering the time it remains uncovered (or unvisited by a mobile node). More specifically, at each step T_m , if a particular cell is visited by a mobile node, its base price \mathcal{P} is set to zero and the base prices of all other *void* cells are increased by 1 unit.

Without loss of generality we assume that at time $t = 0$ each mobile node has moved to the cell center which it belongs to, and at each step T_m , mobile nodes move among cell centers. In the following we explain how a mobile node selects the best cell to be visited at each time step distributively by collaborating with static nodes.

Determining T_m

We assume that any mobile node can reach $L_c = 8$ number of closest distinct cell centers (and itself) as shown in Fig. 7.3 at any given time step. Then the maximum distant that a node has to move during time T_m is $2r$. Thus it is desirable to choose the time step T_m as $T_m = \lceil \frac{2r}{u_{max}} + \epsilon \rceil s$ where u_{max} is the maximum speed of a mobile sensor node and ϵ is a bias factor which accounts for the scenarios when it is needed to heal the lack of coverage at *static* cells which will be explained in subsection 7.3.4 in detail.

Let each cell (cell center) in the square grid is given an ID labeled by indices $1, 2, \dots, L_T$ where $L_T \approx \frac{b^2}{l^2}$ is the total number of cells. Let there is L_s number of cells covered by static nodes (*static* cells) and $L_v = L_T - L_s$ number of cells that

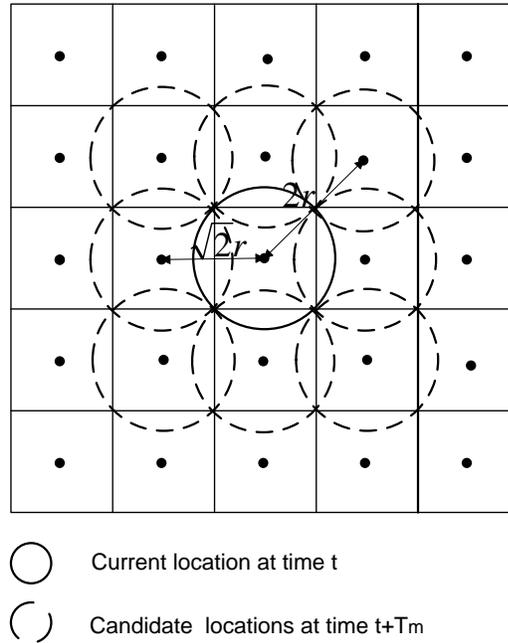


Figure 7.3: A mobile node's candidate locations at a given time

are not covered by static nodes (*void* cells). Also denote \mathcal{U} , \mathcal{U}_s and \mathcal{U}_v to be the sets containing all cell indices of the network, *static* cell indices and *void* cell indices, respectively.

We assign a certain number of cells to each static node in the network. Each static node in the network is responsible for updating the base price for each cell that belongs to it. Corresponding cells for each static node are assigned based on Voronoi partitions (as shown in Fig. 7.4). According to Voronoi partitions, any point inside a Voronoi polygon of a static node is closer to that node rather than to any other static node in the network. Thus for a given static node s_k , the cell centers belonging to its Voronoi polygon are closer to the static node s_k than any other static node in the network. We assume that each static node has the knowledge of the positions of the *void* cell centers belonging to itself. Note that at the initial

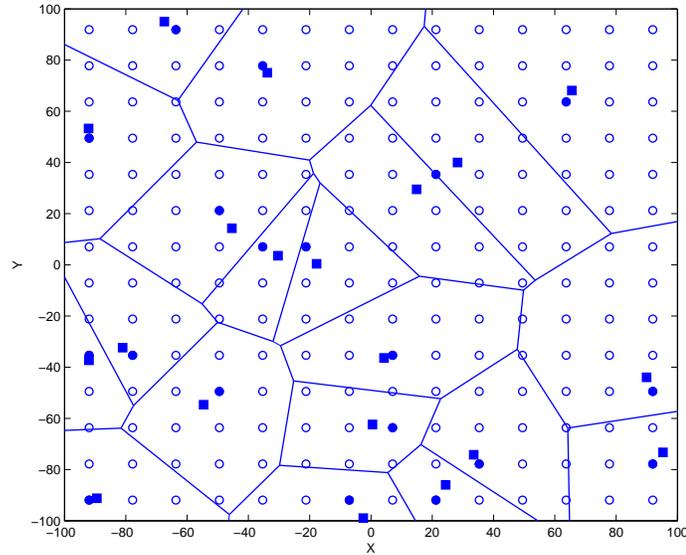


Figure 7.4: Voronoi polygons for each static node: Solid square-static node locations, solid circles-grid points (centers) corresponding to static nodes, void circles-grid points (centers) corresponding to grids not covered by static nodes

stage, static nodes can communicate with their Voronoi neighbors locally to construct Voronoi polygons. By knowing its own location, based on the grid length (in terms of the sensing range) each static node can determine the *void* cells in its Voronoi polygon. Denote \mathcal{U}_{s_k} to be the set of *void* cell indices belongs to the Voronoi polygon of the static node s_k for $s_k \in \mathcal{V}_s$ and $L_{s_k} = |\mathcal{U}_{s_k}|$ be the number of *void* cells (cell centers) belongs to static node s_k . Note that we have then $\mathcal{U}_v = \bigcup_{k \in \mathcal{V}_s} \mathcal{U}_{s_k}$. Further denote $\mathbf{g}_{s_k}(nT_m)$ to be the L_{s_k} -length vector containing the base prices for all *void* cells attached to the static node s_k at time nT_m for $k \in \mathcal{V}_s$. Each static node s_k is responsible for updating $\mathbf{g}_{s_k}(nT_m)$ at each time step $t = nT_m$ for $n = 1, 2, \dots$.

7.3.2 Updating $\mathbf{g}_{s_k}(nT_m)$

At time $t = 0$

At time $t = 0$, each mobile node broadcasts its current location (or equivalently current cell ID) to its neighborhood. The static nodes located close to the mobile node receive this information and if the corresponding mobile node's cell ID belongs to \mathcal{U}_{s_k} then the static node s_k sets the base price for the corresponding cell to zero. Base prices for all the other cells in \mathcal{U}_{s_k} are set to a large integer number K . Note that at time $t = 0$, all *void* cells which have no mobile node at time $t = 0$, have the same base price K .

At time $t = nT_m$, $n \geq 1$

At time $t = nT_m$, each mobile node broadcasts its location information (current cell ID) to its nearest static nodes. Based on this information, each static node updates base price vector $\mathbf{g}_{s_k}(nT_m)$ as follows: Let $N_{m,k}(nT_m)$ be the number of mobile nodes that the static node s_k receives location information at time nT_m and $\mathcal{U}_{m,k}(nT_m)$ be the set corresponding to those locations (cell indices). Then for a given static node s_k for all cell indices $c_j \in \mathcal{U}_{s_k}$, it checks whether c_j also belongs to $\mathcal{U}_{m,k}(nT_m)$. If yes static node s_k sets the base price of the cell c_j to be zero otherwise static node increases the c_j -th cell's base price by 1 unit.

After updating base price vector $\mathbf{g}_{s_k}(nT_m)$ at time nT_m at each static node s_k , the problem is to determine the next cell ID to be visited by each mobile node by time $t = (n + 1)T_m$ such that the cell-revisiting time is maximized. Denote $\mathcal{C}_{m,j}(nT_m)$ to be the set of candidate locations (cells) of the j -th mobile node at time nT_m . Also let $\mathcal{U}_{s_k}^{m_j}(nT_m)$ be the set of cell indices belongs to both $\mathcal{C}_{m,j}(nT_m)$ and \mathcal{U}_{s_k} . Note that the maximum size of the set $\mathcal{U}_{s_k}^{m_j}(nT_m)$ is $|\mathcal{U}_{s_k}^{m_j}(nT_m)|_{max} = L_c + 1 = 9$, since we assume

that each mobile node can move to one of the 8 distinct candidate locations and itself during a given movement step. For a given mobile node m_j from which the static node s_k receives the location information, the static node s_k checks whether any cell in m_j -th candidate set $\mathcal{C}_{m,j}(nT_m)$ belongs to \mathcal{U}_{s_k} at time $t = nT_m$. If not, static node s_k does not need to communicate with mobile node m_j at time nT_m . If yes, or in other words, if the set $\mathcal{U}_{s_k}^{m_j}(nT_m)$ is not empty, the static node s_k queries the mobile node m_j to check whether m_j is isolated with respect another mobile node. We call the mobile node m_j is isolated with respect to another mobile node, if there is no at least one mobile node within a distance d_t from its current location where d_t (equals to $4r$) is a threshold distance determined such that no duplicate covering occurs as discussed in subsection 7.3.3. We assume that the mobile node m_j can communicate locally with other mobile nodes within a distance of d_t to check whether it is isolated. Note that in the rest of the Chapter a mobile node is isolated means that the mobile node is isolated with respect to another mobile node. If m_j is isolated, static node s_k finds the cell from the set $\mathcal{U}_{s_k}^{m_j}(nT_m)$ which has the maximum base price and sends a message corresponding to the cell ID and the maximum corresponding base price. Note that at a given time, all the candidate cells for mobile node m_j may not belong to a one static node. In particular, they may belong to multiple near-by static nodes. Once the mobile node m_j gets maximum base prices from multiple static nodes in which its candidate cells belong to, it selects the best location for time $(n + 1)T_m$ by comparing the base prices it gets from different static nodes and selects the one with maximum base price.

If the mobile node m_j is not isolated (that is there is at least another mobile node very close to it) there might be situations which lead to duplicate covering; that is two or more mobile nodes may try to go to the same cell at time $(n + 1)T_m$. To combat this problem (as discussed in subsection 7.3.3), when a mobile node is not isolated, each static node s_k sends all the candidate cell IDs in the set $\mathcal{U}_{s_k}^{m_j}(nT_m)$ and their base prices to the mobile node to assist in resolving the duplicate covering

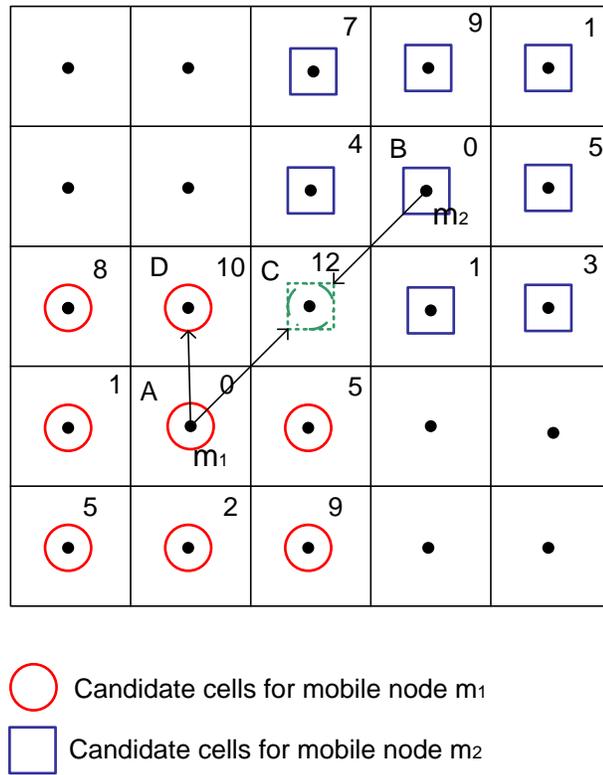


Figure 7.5: Duplicate covering at a given time

problem.

7.3.3 Duplicate covering at a given time

When two mobile nodes are close to each other there might be situations where both select the same *void* cell as the candidate location. For example, consider the scenario as depicted in Fig. 7.5 where two mobile nodes try to heal the same cell. It can be shown that this might happen when two mobile nodes are located within a maximum distance of $d_t = 2\sqrt{2}l = 4r$. Assume that two mobile nodes m_1 and m_2 are located

in cells represented by A and B at time $t = nT_m$ as shown in Fig. 7.5. According to the information received from closest static nodes, both mobile nodes can access to the base prices of all of their candidate cells, for example marked at the north-east corner of each candidate cell for both mobile nodes in Fig. 7.5. According to the base prices, both will try to select the cell C as the next location for time $(n + 1)T_m$ which has the highest base price from each mobile nodes' candidate sets. Since this will lead to inefficient coverage, we propose for two mobile nodes to exchange their local information to avoid duplicate covering. Since this phenomenon occurs when two mobile nodes are located close to each other, we assume that these two mobile nodes can exchange their information to check whether a duplicate covering is going to happen. If so, they exchange the next maximum base prices from their candidate sets, and check which mobile node has the second maximum base price. Accordingly, the node with the highest maximum second base price selects the corresponding cell as the candidate cell. According to Fig. 7.5, since the mobile node m_1 has the second maximum base price (compared to mobile node m_2), it moves to the corresponding cell (denoted by cell D) while the mobile node m_2 moves to the cell C . If the second maximum base price is same for both nodes, they can select either one of the nodes to move to the cell with the second maximum base price arbitrarily. When there are more than 1 mobile sensors within the distance d_t from node m_j , the same procedure can be extended by exchanging the relevant information among those nodes. In such cases it might be necessary to exchange, *2nd*, *3rd*,... highest base prices among neighboring mobile nodes.

7.3.4 Compensating for the lack of coverage in a static cell

As mentioned earlier in this section, since a static node might not be located at the center of a *static* cell in the grid, there might be certain uncovered portions of the corresponding cell. Note that this uncovered portion is maximum when a static node

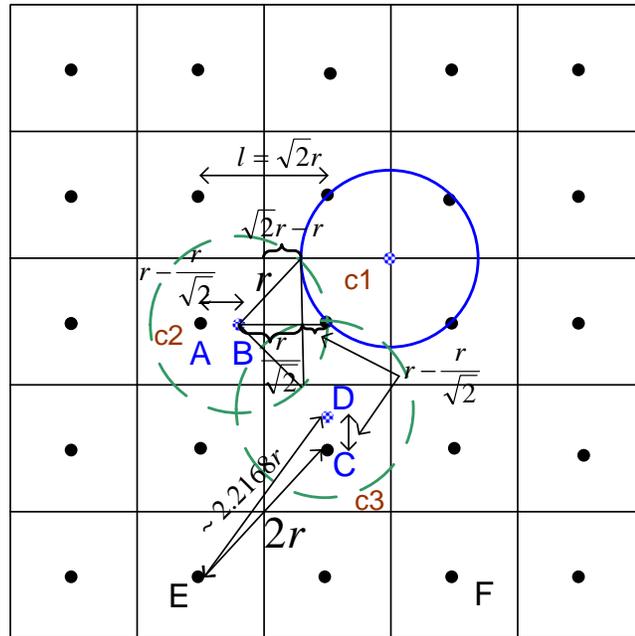


Figure 7.6: Compensating for the lack of coverage in *static* cells

is located very close to one of the cell corners in which it belongs to. Consider the scenario that the static node is located very close to the north-east corner of the cell it belongs to (denoted by c_1), as shown in Fig. 7.6 with a circle with solid line. To compensate for the lack of coverage in the corresponding cell, we propose the following procedure. It can be shown that with the relationship between the side length of a cell in the grid and the sensing range, when a mobile node comes to a cell located either to the left or to the bottom of the *static* cell, and if they are moved a distance of $r - \frac{r}{\sqrt{2}}$ (at the worst case) beyond the cell center towards the *static* cell, the corresponding *static* cell can be completely covered. This is illustrated in Fig. 7.6 where when a mobile node comes to either cell centers A or C, and if it is allowed to move a distance of $r - \frac{r}{\sqrt{2}}$ (i.e. to either B or D, respectively), the uncovered portion of the *static* cell can be totally covered. To address this problem,

at time nT_m , when a mobile node selects its candidate cell for time $(n+1)T_m$, it also checks whether there is a static node to the right, left, up or down to the selected cell. Then based on the static node location, it approximates the required distance it should move (maximum of $r - \frac{r}{\sqrt{2}}$) beyond the selected cell center to compensate for the lack of coverage of the *static* cell.

Note that according to the proposed mobility algorithm we allow mobile nodes to move between cell centers at consecutive time steps T_m . However, when we need to address this *static* cell compensating problem, mobile nodes have to move little far away from a cell center. When this happens (i.e. a mobile node may move to location B (or D) instead of A (or C) in Fig. 7.6), the mobile node may need to move a maximum distance of $\approx 2.2168r$ to reach its next candidate cell at next time step. As shown in Fig. 7.6, when the mobile node is at the point D in the cell c_3 , it can reach all its candidate cells by next time step, except E and F by moving a maximum distance of $2r$. To reach the candidate cells E and F it has to move a maximum distance of $\approx 2.2168r$. Thus when determining the time step T_m as pointed out in subsection 7.3.1, we need to take this scenario into account. Thus T_m is selected as, $T_m = \lceil \frac{2r}{u_{max}} + \epsilon \rceil s$ where $\epsilon = \frac{0.2168r}{u_{max}}$.

The proposed protocol for node mobility management of hybrid sensor network is summarized in Algorithm 3.

7.4 Performance Evaluation

To evaluate the effectiveness and efficiency of the proposed mobility protocol, we perform experiments to investigate how well the desired area is covered over time to minimize the time that a *void* cell is unvisited by a mobile node. We depict the results in different perspectives taking the factors, the probability that at least one

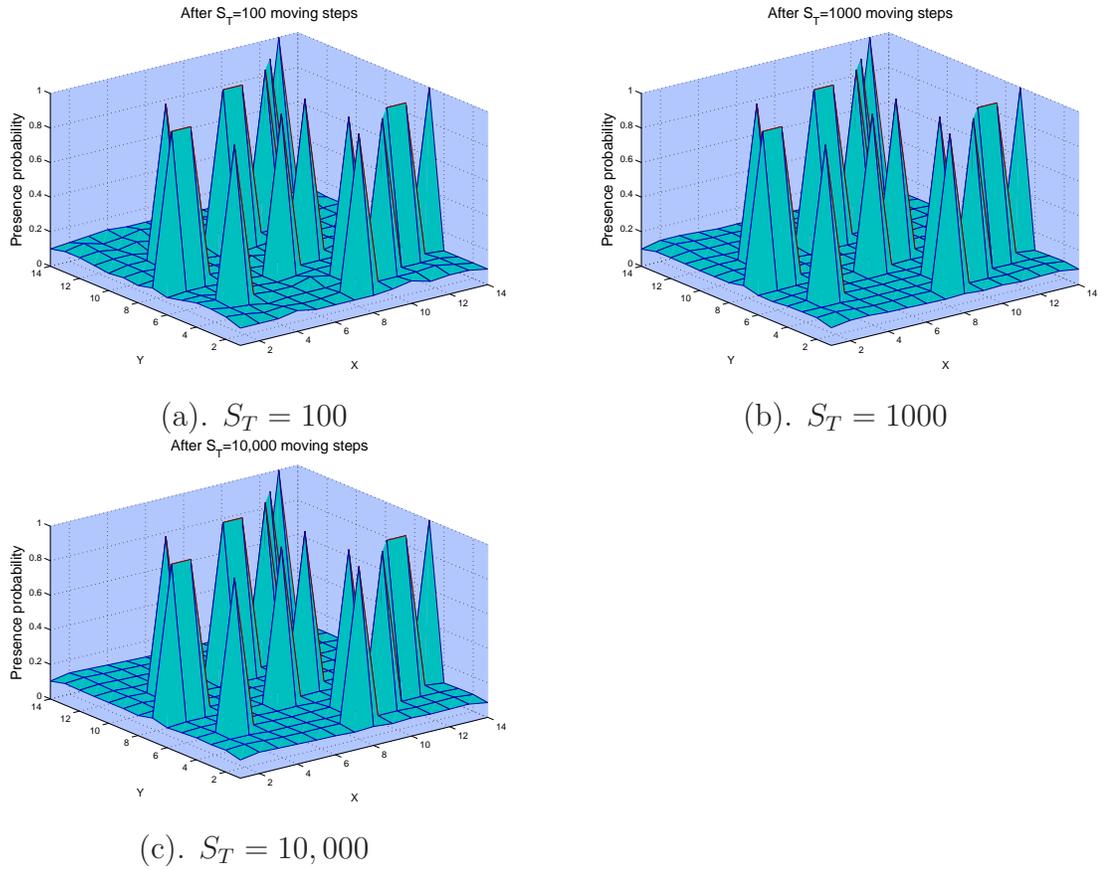


Figure 7.7: Presence probability matrix with proposed mobility protocol, $N = 40$, $\lambda_m = 0.5$, $u_{max} = 10m/s$ (a). after moving steps $S_T = 100$ (b). after moving steps $S_T = 1000$ (c). after moving steps $S_T = 10,000$

mobile node visits a particular cell at any given time instant, the average time that any arbitrary point in the network is unvisited, effect of the node speed and the fraction of mobile nodes, into account.

7.4.1 Presence probability at each cell

Denote p_{c_k} to be the probability that at least one node is present at the cell c_k at any given time. Let $\mathbf{\Lambda}$ be the presence probability matrix containing the probabilities of

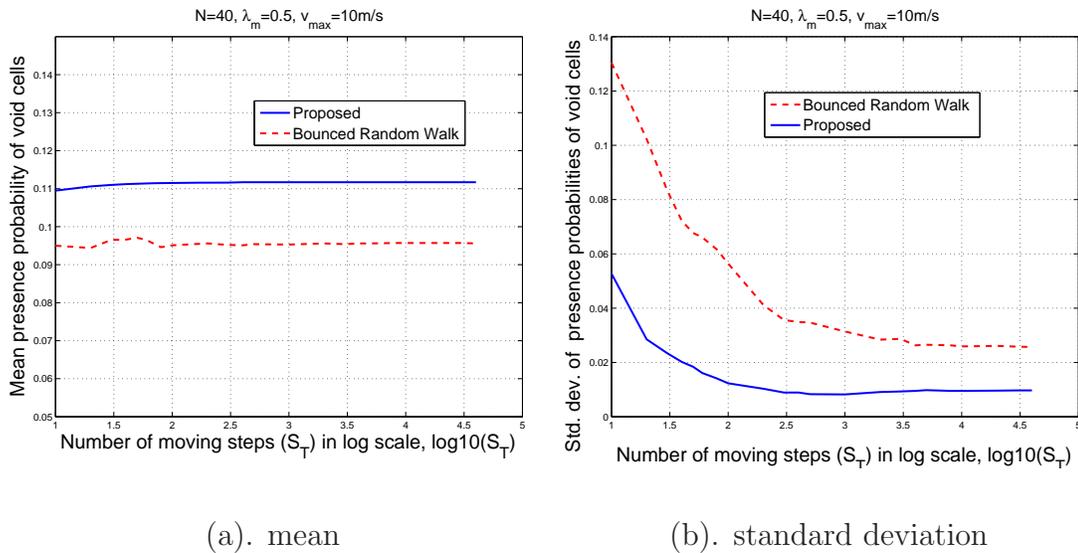


Figure 7.9: Mean and the standard deviation of presence probabilities at *void* cells vs. the number of movement steps S_T (in log scale) for proposed protocol and the bounced random walk mobility model, $N = 40$, $\lambda_m = 0.5$, $u_{max} = 10m/s$ (a). Mean (b). Standard deviation

random walk scheme, respectively. The presence probability matrices are shown after completing $S_T = 100$, $S_T = 1000$ and $S_T = 10,000$ moving steps, respectively, for $N = 40$ and $\lambda_m = 0.5$. Note that in Figs 7.7 and 7.8, the high peaks with presence probability 1 reflect the presence probability of *static* cells. Looking at the presence probabilities of *void* cells under two mobility schemes, from Fig. 7.7 it can be seen that the presence probabilities of *void* cells are becoming uniform after completing relatively a small number of steps compared to that with random walk model (Fig. 7.8). When the number of movements steps is large, it can be seen from Fig.7.8 that the presence probabilities of *void* cells under random walk mobility models are also becoming uniform, as expected. However, as can be seen from Figs. 7.7 and 7.8, in terms of the number of movement steps needed to achieve this uniformity the proposed protocol for hybrid sensor network outperforms the random mobility schemes.

To further investigate the relationship between the number of movement steps and the uniformity of presence probabilities of *void* cells, in Fig. 7.9 we plot the mean and the standard deviation of presence probabilities of *void* cells as the number of movement steps (S_T) is increasing for proposed and random walk mobility schemes. In Fig. 7.9 we use S_T in \log_{10} scale. From Fig. 7.9(a), it can be seen that the mean of the presence probabilities of *void* cells converges to a constant with a relatively small number of movement steps for both schemes and the corresponding mean value is relatively large with proposed scheme compared to that with the random mobility scheme. This essentially implies, with the proposed protocol, *void* cells are covered much efficiently over time compared to that with random mobility scheme. In Fig. 7.9(b), we plot the standard deviation of presence probabilities of *void* cells with $\log_{10} S_T$. Note that the standard deviation of presence probabilities of *void* cells acts as a measure of the quality of uniformness of the presence probabilities. From Fig. 7.9(b), it can be seen that the standard deviation of presence probabilities of *void* cells converges to a constant value for both mobility schemes and the threshold number of movement step that this happens is much more less with the proposed mobility protocol compared to that with the random mobility scheme. Moreover, the constant value of this convergence is less for proposed protocol compared to that with the bounced random walk scheme implying the effectiveness of the proposed scheme.

Although figures are not included, it can be seen that when the fraction of mobile nodes is increasing the presence probability of *void* cells is also increasing since then the frequency that any mobile node can visit a cell is also increasing.

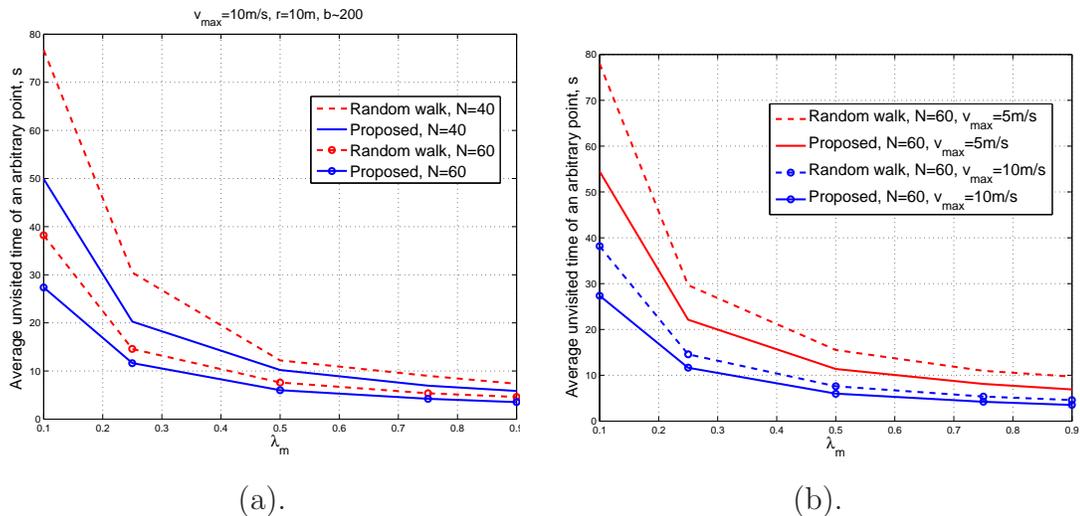


Figure 7.10: (a). Average time taken for an arbitrary point to be revisited for different network sizes N : $u_{max} = 10m/s$, $r = 10m$, $b \approx 200m$ (b). Average time taken for an arbitrary point to be revisited for different node speeds: $N = 60$, $r = 10m$, $b \approx 200m$

7.4.2 Average unvisited time of an arbitrary point

In the next experiment, we evaluate the performance of the proposed mobility scheme in terms of the average time that any arbitrary point is uncovered by the hybrid sensor network. We compare the results of the proposed scheme with a random mobility model. Figure 7.10(a) shows the average unvisited time of an arbitrary point in the network with the proposed mobility protocol and bounced random walk mobility model (with step size of l) for $N = 40$ and $N = 60$. In Fig. 7.10(a), we let $u_{max} = 10m/s$, $r = 10m$. It can be seen that when the fraction of mobile nodes is low, by the proposed mobility protocol for hybrid sensor network, a significant performance improvement can be obtain over bounced random walk mobility model. Note that due to extra cost needed for deploying mobile nodes compared to static nodes, this is the most interesting scenario. As mentioned earlier in the Chapter, random mobility models are not well suited for hybrid sensor networks specially for

lower λ_m 's since they may provide duplicate coverage, which results in an inefficient usage of mobile nodes. Since deploying mobile nodes is not as cost effective as deploying static nodes, it is more desirable to efficiently use the node mobility in order to improve the network coverage. However, from Fig. 7.10(a), it can be seen that when λ_m is increasing, the unvisited time with the proposed scheme is not much different from the random walk scheme since then there is a large number of mobile nodes compared to static nodes and thus the duplicate coverage caused by random walk mobility model is less. Also when the total number of nodes is increasing, it can be seen that even with a lower fraction of mobile nodes, relatively lower unvisited time can be obtained by the proposed scheme. The performance gain of the proposed scheme over the random walk mobility model is more significant when N is smaller, that is when the network is to be covered by a small number of total nodes.

Figure 7.10(b) shows the average unvisited time of an arbitrary point when the node speed is changing. In Fig. 7.10(b), we let $N = 60$, $r = 10m$ and the plots correspond to $u_{max} = 5m/s$ and $10m/s$. It can be seen that especially with a lower fraction of mobile nodes, the speed of mobile nodes affects the system performance significantly compared to that with a large fraction of mobile nodes. However, irrespective of the node speed, it can be seen that with relatively small fraction of mobile nodes, the proposed mobility scheme outperforms the random mobility schemes. For results in Fig. 7.10 we ran simulations for 10000s and averaged over 50,000 arbitrary points.

7.5 Conclusions

In this Chapter we proposed an interactive, distributed protocol for mobile node navigation in a hybrid sensor network to efficiently cover the area not-covered by static

nodes by maximizing the re-visiting time of an arbitrary point in the network. The proposed scheme can be implemented distributively by collaborating among static and mobile nodes locally, having only communicating in the local neighborhood. It was shown that the proposed scheme provides an approximate uniform coverage after completing relatively small number of movement steps compared to that with random walk model, which is desirable when the network is designed for detecting targets in which the existence is unknown. The proposed scheme also outperforms the random mobility schemes in terms of the average revisiting time of an arbitrary point in the network especially when the fraction of mobile nodes is small.

Algorithm 3 Mobility protocol

NOTATIONS:

- $\mathbf{g}_{s_k}(nT_m)$: base price vector at static node s_k at time $t = nT_m$
 \mathcal{U}_{s_k} : set of all *void* cell indices belongs to static node s_k
 $N_{m,k}(nT_m)$: number of mobile nodes from which the static node s_k receives locations information at time nT_m
 $\mathcal{C}_{m,j}(nT_m)$: set of cell indices corresponding to candidate cells of mobile node m_j at time nT_m
 $\mathcal{U}_{s_k}^{m_j}(nT_m)$: set of cell indices belongs to both $\mathcal{C}_{m,j}(nT_m)$ and \mathcal{U}_{s_k}
 $\mathbf{g}_{s_k}^{m_j}(nT_m)$: base price vector corresponding to cell indices in $\mathcal{U}_{s_k}^{m_j}$
 $P_{j,k}^*$: element with maximum value (maximum base price) in $\mathbf{g}_{s_k}^{m_j}(nT_m)$
 $c_{j,k}^*$: cell index corresponding to $P_{j,k}^*$

INITIALIZATION AT TIME $t = 0$:

- 1: Determine \mathcal{U}_{s_k} for all $k \in \mathcal{V}_s$ based on Voronoi partitions
- 2: Initialize $\mathbf{g}_{s_k}(0)$ as in subsection 7.3.2

AT STATIC NODE s_k AT TIME $t = nT_m$:

After receiving location (cell) information from neighboring mobile nodes:

- 1: Update the base price vector $\mathbf{g}_{s_k}(nT_m)$ as in subsection 7.3.2
- 2: **for** $j = 1 : N_{m,k}(nT_m)$ **do**
- 3: Check $\rightarrow \mathcal{U}_{s_k}^{m_j}(nT_m)$ is non-empty
- 4: **if** yes **then**
- 5: check $\rightarrow m_j$ is isolated
- 6: **if** yes **then**
- 7: Find $P_{j,k}^*$ and $c_{j,k}^*$ and transmit to mobile node m_j
- 8: **else** $\{m_j$ is not isolated $\}$
- 9: Send cell IDs and their base prices in the set $\mathcal{U}_{s_k}^{m_j}(nT_m)$ to mobile node m_j
- 10: **end if**
- 11: **else** $\{\text{no}\}$
- 12: Send nothing to mobile node m_j
- 13: **end if**
- 14: **end for**

AT MOBILE NODE m_j AT TIME $t = nT_m$:

- 1: Broadcast location information to neighboring static nodes

After receiving base prices for relevant candidate locations from neighboring static nodes:

- 1: check $\rightarrow m_j$ is isolated
- 2: **if** yes **then**
- 3: select candidate cell with maximum base price
- 4: **else** $\{\text{no}\}$
- 5: call *duplicate_covering*(m_j)
- 6: **end if**

After selecting candidate cell corresponding to time $(n + 1)T_m$:

- 1: Check \rightarrow need for *static* cell compensation
- 2: **if** yes **then**
- 3: Adjust the location to be moved in the selected candidate cell according to subsection 7.3.4
- 4: **else** $\{\text{no}\}$
- 5: Move to the center of the selected candidate cell by time $(n + 1)T_m$
- 6: **end if**

duplicate_covering(m_j)

- 1: Exchange local information with neighboring mobile nodes to check for duplicate covering
- 2: **if** yes:(duplicate covering) **then**
- 3: Exchange next highest base prices to determine the best candidate cell as in subsection 7.3.3
- 4: **else** $\{\text{no}:(\text{no duplicate covering})\}$
- 5: select candidate cell with maximum base price
- 6: **end if**

Chapter 8

Worst-Case Detection

Performance

8.1 Introduction

In this Chapter, we analyze the worst-case detection performance of a hybrid sensor network consisting of both static and mobile nodes when the target to be detected is trying to evade the sensing region with the minimum probability of being detected. We evaluate the worst-case detection performance in terms of the *exposure* [32,91,92,104], which reflects the quality of the sensor network when the target tries to evade the network with minimum detection probability. *Exposure* is defined in different contexts in the literature, and the general idea behind that is how can a target traverse the desired field with the minimum probability of being detected (or minimum detection time). To find the exposure path, different algorithms were

proposed in [91, 92, 104] considering different performance measures. For example, in [92], the exposure path was formulated in terms of the sensor field intensity. In [91], algorithms are presented to find exposure in terms of the worst case coverage. In the worst case coverage, the exposure path is found by maximizing the closest distance to any sensor node in the target traversal, based on Voronoi partitions and the graph theoretic techniques. In [104], a different definition is given for the exposure. The exposure path is defined as the one with the least probability of being detected and they have taken the measurement uncertainties at sensor nodes into account in finding the *exposure* path. The *exposure* in a mobile sensor network is addressed in [32] to minimize the probability of being detected, based on a given sensing architecture in which mobile nodes make noisy measurements on the emitted signals by the target at a given set of locations of the route of the mobile nodes. However, they did not consider specific mobility models for the mobile nodes.

To find the *exposure* in a hybrid sensor network, we develop an efficient sequential methodology based on the presence probability matrix. The proposed methodology to find *exposure* is valid for hybrid sensor networks with arbitrary mobility models as far as the knowledge of the presence probability matrix is available. We show that with the mobility management protocol presented in Chapter 7, a significant performance at the worst-case target exposure is achieved compared to random mobility schemes especially when the number of mobile nodes in the hybrid network is small.

The Chapter is organized as follows. Section 8.2 presents the sensor network and the target model. The worst case performance on target detection by the hybrid sensor network is addressed in Section 8.3. Section 8.4 shows performance results and concluding remarks are given in Section 8.5.

8.2 Network Model

We consider a hybrid sensor network made of N total sensor nodes deployed in a region \mathcal{R} with network dimension of $b \times b$. We assume that there are N_s number of static nodes and N_m number of mobile nodes. Denote $\lambda = \frac{N}{b^2}$ to be the spatial density of the nodes and $\lambda_m = \frac{N_m}{N}$ and $\lambda_s = \frac{N_s}{N}$ to be the fractions of mobile and static nodes respectively. Let \mathcal{V} be the set containing all node indices in the network and let \mathcal{V}_m and \mathcal{V}_s be the sets containing mobile and static node indices, respectively. We assume the same network model as in Chapter 7 in which the sensing region is divided into a square grid with a grid length of $l = \sqrt{2}r$ where r is the effective sensing radius of a sensor. We assume that any mobile node can reach $L_c = 8$ number of closest distinct cell centers (and itself) as discusses in Chapter 7 at any given time step. The time at which each movement step is taken by a mobile node is denoted by T_m and the maximum speed of a mobile node is u_{max} .

8.2.1 Target model

Without loss of generality we assume that the target traversal also is a sequence of cells in the grid formed in Chapter 7. We denote by \mathcal{S} , a set of cell sequences which forms a path for the target. We assume that a target can enter and leave the desired region from any boundary (boundary cell). Further we assume that the target should spend at least T_1 time after it enters the region to accomplish the required task, and has to leave the region before maximum of $T_2 \geq T_1$ time. The goal is to find the best path for the target to minimize the probability of being detected by the sensor network.

8.3 Worst Case Detection Performance

8.3.1 Detection probability

Let us assume that a target can visit 8 number of distinct candidate cells at a given time from its current cell as that with the mobile nodes. Let T_r be the time that the target needs to visit its candidate cells from its current position and $u_{r,max}$ be the maximum speed of the target. Note that if the target has the same speed as with mobile nodes, then we have $T_r \approx T_m$. When the target visits the cell c_k at time $t = nT_r$, the probability that the target is detected at time $t = nT_r$, $P(c_k, nT_r) = p_{c_k}$ where p_{c_k} is the presence probability of cell c_k , which is the probability that at least one node is present at the cell c_k at any given time instant. Note that $p_{c_k} = 1$ if c_k is a *static* cell. When a target traverses along the path S for n_0 time steps, where $T_1 \leq n_0T_r \leq T_2$ the probability that the target is detected by the sensor network is given by,

$$P(S, n_0) = 1 - \prod_{j=0}^{n_0} (1 - P(c_j, jT_r)), \quad (8.1)$$

where c_j is the cell index where the target is located at time jT_r . Let \mathcal{S} be the set of all cell sequences that the target can traverse by time $T_1 \leq n_0T_r \leq T_2$ starting at any boundary cell. Then the exposure is defined as [32],

$$\epsilon = \min_{S \in \mathcal{S}} P(S, n_0). \quad (8.2)$$

8.3.2 Evaluating worst-case target exposure

Note that minimizing $P(S, n_0)$ is equivalent to maximizing $\prod_{j=0}^{n_0} (1 - P(c_j, jT_r))$ and thus maximizing $\sum_{j=0}^{n_0} \log(1 - P(c_j, jT_r))$. Since $\log(1 - P(c_j, jT_r)) \leq 0$, we take

maximizing $P(S, n_0)$ as equivalent to minimizing $-\sum_{j=0}^{n_0} \log(1 - P(c_j, jT_r))$. As given in [32], to find the path with minimum exposure, we may convert the problem into a shortest path problem in a time expansion directed graph by assigning vertices and weights as follows.

For a given time $t = nT_r$, the vertices of the graph represent all the cell indices. We consider the same grid structure as given in section 7.3 which has a total of L_T number of cells. We represent vertices at time $t = nT_r$ as (c_k, nT_r) consisting of all cells where $c_k \in \mathcal{U}$. The weight assignment of the graph from time $t = nT_r$ to time $(n+1)T_r$ is performed as follows. If the cell c_k at time $t = nT_r$ (i.e. vertex (c_k, nT_r) in the expansion graph) is a non-boundary cell, it has 9 (including itself) outgoing edges to the corresponding neighboring cells. In particular, let $(c_{k1}, (n+1)T_r)$, $(c_{k2}, (n+1)T_r)$, $(c_{k3}, (n+1)T_r)$, $(c_{k4}, (n+1)T_r)$, $(c_{k5}, (n+1)T_r)$, $(c_{k6}, (n+1)T_r)$, $(c_{k7}, (n+1)T_r)$, $(c_{k8}, (n+1)T_r)$ and $(c_k, (n+1)T_r)$ be the vertices at time $(n+1)T_r$ corresponding to neighboring (candidate) cells of the cell c_k when the current time is $t = nT_r$. Then the vertex (c_k, nT_r) has outgoing edges to all vertices listed above at time $(n+1)T_r$ and the corresponding edge weights are given by $-\log(1 - P(c_{n+1}, (n+1)T_r))$, where c_{n+1} is the corresponding cell index at time $(n+1)T_r$. For a boundary cell, the number of candidate cells is less than that with non-boundary cells, and the vertices are connected only with the valid candidate cells. An illustration of vertex and edge assignments for a 3×3 grid is shown in Fig. 8.1 where edge weights are not marked. Since the target needs to exit the region by maximum of time T_2 , the graph is expanded at most T_2/T_r steps. Now the problem is to find the target traversal which will result the minimum weight $w = -\sum_{j=1}^{n_0} \log(1 - P(c_j, jT_r))$ for any $T_1 \leq n_0 \leq T_2$.

Note that in [32], an upper bound and a lower bound for the exposure were given instead of the exact exposure. In contrast, with the constraints that the target may have to exit the region within $[T_1, T_2]$, we present a sequential procedure to find the

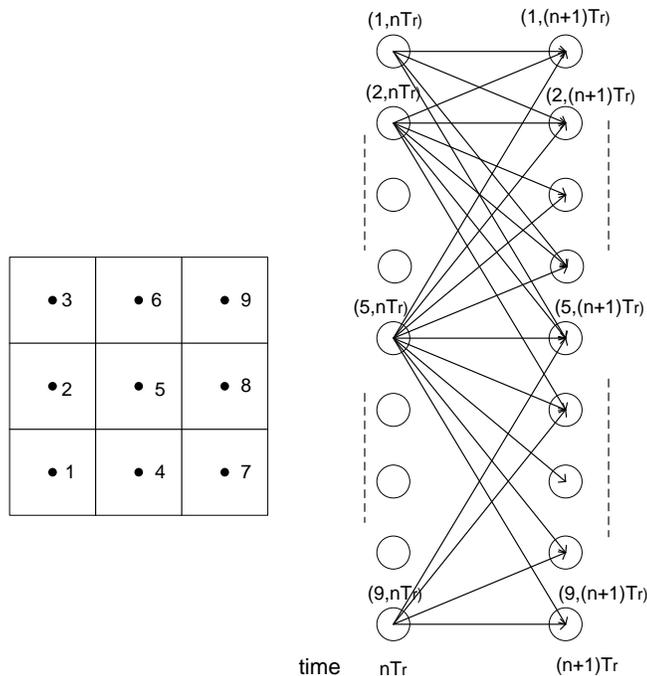


Figure 8.1: Vertex and edge assignment of the expansion graph from time nT_r to time $(n + 1)T_r$ for 3×3 square grid; edge weights are not marked. Note that the vertex $(5, nT_r)$ at time nT_r corresponds to a non-boundary cell of the considered grid and it has 9 outgoing edges from time nT_r to $(n + 1)T_r$. All the other vertices at time nT_r correspond to boundary-cells. For vertices $(1, nT_r)$, $(3, nT_r)$, $(7, nT_r)$, $(9, nT_r)$ at time nT_r , they have 4 outgoing edges while for vertices $(2, nT_r)$, $(4, nT_r)$, $(6, nT_r)$, $(8, nT_r)$, they have 6 outgoing edges from time nT_r to $(n + 1)T_r$.

exact exposure with reduced complexity using graph theoretic techniques.

Denote \mathcal{U}_b and \mathcal{U}_{nb} be the sets containing indices of boundary cells and non-boundary cells, respectively. Recall that we assume that the target may enter and exit from any boundary cell after spending T_1 time. Now the problem is to find the best path for the target such that it will give the minimum detection probability to exit the region after T_1 time, if it enters the region from a boundary cell. Note that based on the above graph theoretic view, the shortest path (cell sequence) that

any cell can be reached (from starting cell) by time $t = T_1$ can be found based on a single-source shortest path algorithm. For simplicity we assume that $T_1/T_r = q$ is an integer. Denote $s_k(qT_r)$ to be the shortest path (or cell sequence) for the target traversal with the destination being the cell c_k at time qT_r , and $w_k(qT_r)$ be the corresponding weight where $w_k(qT_r) = -\sum_{j=1}^q \log(1 - P(c_j^*, jT_r))$ where c_j^* 's are in the cell sequence of the corresponding path. Now we propose the following procedure to find the best traversal for the target.

Let $w_{k,b}^{min}(qT_r) = \min_{k \in \mathcal{U}_b} w_k(qT_r)$ be the minimum weight of all the shortest paths with a boundary cell being the destination cell at time $t = qT_r = T_1$ and $w_{k,nb}^{min}(qT_r) = \min_{k \in \mathcal{U}_{nb}} w_k(qT_r)$ be the minimum weight of all the shortest paths with a non-boundary cell being the destination cell at time $t = qT_r = T_1$. It can be shown that if $w_{k,nb}^{min}(qT_r) \geq w_{k,b}^{min}(qT_r)$, by expanding the graph beyond the time $t = qT_r = T_1$ will not result any shorter path with corresponding weight less than $w_{k,b}^{min}(qT_r)$. Thus if this condition stratifies at time qT_r (or (T_1)), the path with minimum weight, for the target enters at a particular boundary cell, is the path corresponding to $w_{k,b}^{min}(qT_r)$. If the condition is not satisfied, that will mean that there is a possibility to have a shorter path for the target to exit the region with a less weight (or less probability of detection) than the path corresponding to the weight $w_{k,b}^{min}(qT_r)$ which terminated by time $t = qT_r$. Then, if the above condition is not satisfied the graph is expanded to time $t = (q+1)T_r$ while keeping $w_{k,b}^{min}(qT_r)$ in the memory. Now the weight assignments for edges connecting vertices from time $t = qT_r$ to $t = (q+1)T_r$ are performed as follows.

From all the shortest paths with the destination cell as a non-boundary cell at time qT_r , we find the set of non-boundary cells which have the corresponding weights at time qT_r less than $w_{k,b}^{min}(qT_r)$. Then we connect only these non-boundary cells to their candidate cells at time $(q+1)T_r$. The point here is that, for other non-boundary cells at time qT_r where the corresponding weights of their shortest paths are greater

than $w_{k,b}^{min}(qT_r)$ by expanding the vertices corresponding to them beyond qT_r , we will not get any shorter path which will result in a less value than $w_{k,b}^{min}(qT_r)$ since any path with those cells being the cell at time qT_r , will always result the weight greater than $w_{k,b}^{min}(qT_r)$.

At time $(q + 1)T_r$, we follow two steps: (i). As in time qT_r , $w_{k,b}^{min}((q + 1)T_r)$ and $w_{k,nb}^{min}((q + 1)T_r)$ are computed. If $w_{k,b}^{min}((q + 1)T_r) \leq w_{k,b}^{min}(qT_r)$, $w_{k,b}^{min}(T_r)$ is deleted from the memory, since then it makes sure that there is a shorter path on or beyond time $(q + 1)T_r$ having a smaller weight than $w_{k,b}^{min}(qT_r)$. Then again as in time qT_r , the condition $w_{k,nb}^{min}((q + 1)T_r) \geq w_{k,b}^{min}((q + 1)T_r)$ is checked, and if true expansion is stopped by time $(q + 1)T_r$. If not, the same procedure follows as in time qT_r , to find the required set of non-boundary cells from which the edges are connected to time $(q + 2)T_r$ while keeping $w_{k,b}^{min}((q + 1)T_r)$ in the memory. (ii). If $w_{k,b}^{min}((q + 1)T_r) > w_{k,b}^{min}(qT_r)$, it checks whether the condition $w_{k,nb}^{min}((q + 1)T_r) \geq w_{k,b}^{min}(qT_r)$ is satisfied. If yes, the expansion is stopped by time $(q + 1)T_r$ resulting $w_{k,b}^{min}(qT_r)$ the minimum weight corresponding to shortest path for the target. If not, the graph is expanded to time $(q + 1)T_r$ after finding the required set of non-boundary cells from which the edges are connected to time $(q + 2)T_r$ (as in time qT_r) while keeping $w_{k,b}^{min}(qT_r)$ in the memory. The expansion is stopped at time q_0T_r if either one of the following criteria is met. (i). if $w_{k,nb}^{min}(q_0T_r) \geq \min\{w_{k,b}^{min}(qT_r), w_{k,b}^{min}((q + 1)T_r), \dots, w_{k,b}^{min}(q_0T_r)\}$ for $q \geq q_0 < T_2/T_r$ (ii). if $q_0 = T_2/T_r$, where the maximum time for expansion is reached.

Note that, with the proposed scheme, the complexity is greatly reduced since after time T_1 , at each time step a certain number of vertices corresponding to non-boundary cells does not need to be expanded. On the other hand, with the proposed mobility protocol, as can be observed from the simulation results, the graph does not need to expand a large number of time steps after time T_1 due to the approximately uniform nature of the presence probability matrix (for the *void* cells). This essentially

implies that after the required time is spent in the region (i.e. time T_1), by circulating inside the region to minimize the detection probability is not desirable for the target since, due to nearly uniform presence probability matrix, target will not find a safer area to avoid detection inside the region as time goes. Note that the above procedure is for the target traversal starting at a given boundary cell. Thus to find the worst case scenario over all starting boundary cells, the procedure can be repeated. The proposed procedure is summarized in Algorithm 4.

8.4 Performance Results

In this Section, we evaluate the worst-case detection probability based on the algorithm presented in Section 8.3. We compare the worst-case detection performance of the proposed mobility model with the bounced random walk model. To find the worst-case detection performance as given by Section 8.3, we find the presence probability matrix with random walk with a step size of l .

Figure 8.2 shows the worst-case detection performance with the proposed mobility scheme and with the random walk mobility model. In Fig. 8.2, we assume that the maximum speed of mobile nodes and the target is the same, where $u_{max} = u_{r,max} = 5m/s$. Note that higher the worst-case detection probability, less safe for the target to enter the sensing region. It can be seen from Fig. 8.2 that, with the proposed node mobility scheme, it is more dangerous for the target to enter the sensing region and is very less likely that it can find a safe path to exit. Also, it can be seen that more time the target has to be in the desired region (i.e. T_1 is increasing) to perform the required task, more vulnerable for the target, and the rate of vulnerability is higher as T_1 increases with the proposed mobility model when compared to that with random walk model.

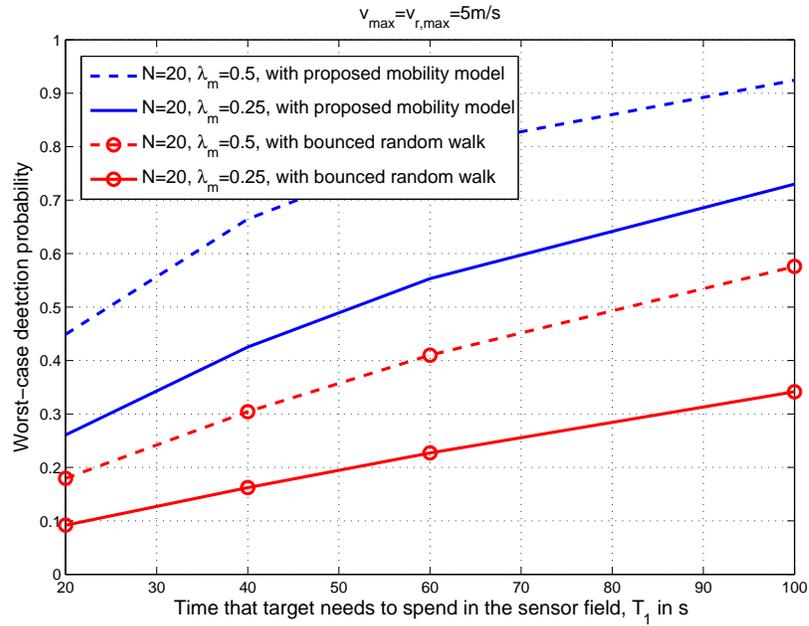


Figure 8.2: Worst-case detection probability

8.5 Conclusions

In this Chapter, the worst-case detection performance of a hybrid sensor network is evaluated when the target is trying to evade the sensing region by minimizing the probability of being detected, in terms of *exposure*. We developed an efficient sequential algorithm to find the worst-case target *exposure* based on graph-theoretic techniques which can be used to find exposure in mobile/hybrid sensor networks with arbitrary mobility models as far as the presence probability matrix is available. It was shown that with the proposed mobility protocol in Chapter 7, it is very less likely that a target would find a safe path to traverse through the sensing field without being detected, compared to that with random mobility models.

Algorithm 4 Procedure to find best target traversal

NOTATIONS: $q = T_1/T_r$: minimum number of time steps a target needs to spend in the region $s_k^*(qT_r)$: the shortest path (cell sequence) for the target traversal with the destination cell being the cell c_k at time qT_r $w_{k,b}^{min}(qT_r)$: (minimum) weight of the shortest path with a boundary cell being the destination cell at time qT_r $s_{k,b}^*(qT_r)$: corresponding shortest path (cell sequence) which results the weight $w_{k,b}^{min}(qT_r)$ $w_{k,nb}^{min}(qT_r)$: (minimum) weight of the shortest path with a non-boundary cell being the destination cell at time qT_r $s_{k,nb}^*(qT_r)$: corresponding shortest path (cell sequence) which results the weight $w_{k,nb}^{min}(qT_r)$ $\mathcal{U}'_{nb}(qT_r)$: set of non-boundary cells with the corresponding weights at time qT_r are less than $w_{k,b}^{min}(qT_r)$ $\bar{w}_{k,b}^{min}(nT_r)$: $\min\{w_{k,b}^{min}(qT_r), w_{k,b}^{min}((q+1)T_r), w_{k,b}^{min}(nT_r)\}$ is the minimum weight of a boundary cell over time qT_r to nT_r with $n \geq q$ AT TIME $t = qT_r$:

- 1: Construct the expansion graph over q time steps
- 2: Find $w_{k,b}^{min}(qT_r)$ and $w_{k,nb}^{min}(qT_r)$
- 3: **if** $w_{k,nb}^{min}(qT_r) \geq w_{k,b}^{min}(qT_r)$ **then**
- 4: end procedure: result \rightarrow shortest path $s_{k,b}^*(qT_r)$
- 5: **else** $\{w_{k,nb}^{min}(qT_r) < w_{k,b}^{min}(qT_r)\}$
- 6: Find $\mathcal{U}'_{nb}(qT_r)$
- 7: Expand the graph to time $(q+1)T_r$ by connecting edges from vertices corresponding to the cells in $\mathcal{U}'_{nb}(qT_r)$
- 8: Keep $\bar{w}_{k,b}^{min}(qT_r) = w_{k,b}^{min}(qT_r)$ in memory
- 9: **end if**

AT TIME $t = nT_r$ WITH $q < n < q_0$

- 1: compute $w_{k,b}^{min}(nT_r)$ and $w_{k,nb}^{min}(nT_r)$
 - 2: check $\rightarrow w_{k,b}^{min}(nT_r) \leq \bar{w}_{k,b}^{min}((n-1)T_r)$
 - 3: **if** yes **then**
 - 4: $\bar{w}_{k,b}^{min}(nT_r) = w_{k,b}^{min}(nT_r)$
 - 5: **else** {no}
 - 6: $\bar{w}_{k,b}^{min}(nT_r) = \bar{w}_{k,b}^{min}((n-1)T_r)$
 - 7: **end if**
 - 8: check $\rightarrow w_{k,nb}^{min}(nT_r) \geq \bar{w}_{k,b}^{min}(nT_r)$
 - 9: **if** yes **then**
 - 10: end procedure: result \rightarrow the shortest path corresponding to $\bar{w}_{k,b}^{min}(nT_r)$
 - 11: **else** {no}
 - 12: Find $\mathcal{U}'_{nb}(nT_r)$
 - 13: Expand the graph to time $(n+1)T_r$ by connecting edges from vertices corresponding to the cells in $\mathcal{U}'_{nb}(nT_r)$
 - 14: Keep $\bar{w}_{k,b}^{min}(nT_r)$ in memory
 - 15: **end if**
-

Chapter 9

Decision Fusion with Measurement

Uncertainty under Reactive

Mobility

9.1 Introduction

In Chapters 6 and 7, it was assumed that nodes perform continuous movements searching for targets when the target existence is unknown or uniformly located (if stationary as in Chapter 6) in the desired region. Moreover, in Chapters 6 and 7, the measurement uncertainty on the detection performance was not taken into account. In situations where achieving a continuous coverage is not affordable due to energy constraints, it might be desirable to keep mobile nodes in a stationary

configuration until a possible target is detected by the stationary configuration with certain confidence level, and allow nodes to be mobile only if necessary to detect the target with higher confidence level.

In this Chapter, we develop decision fusion models to detect randomly located stationary targets by the hybrid sensor network after detecting at a certain confidence level. Due to energy constraints, we assume that the mobile nodes are kept stationary until a target is detected with certain confidence level. Note that since mobile nodes are required to perform on-demand for different functionalities, it is not possible to locate them in a certain area for a specific task. We assume that, at each time step, a mobile node can move to a limited number of locations from its current position, where these candidate locations are determined by physical factors related to mobile sensors and the environment. After detecting at a lower confidence level, at each time step, mobile nodes move in a direction chosen based on the proposed mobility management schedule to maximize the detection probability during a desired delay constraint. At each time step, each node makes a local binary decision based on its observations and transmits it to a separate fusion center. The fusion center combines local decisions from all static and mobile nodes to reach at a final decision at the corresponding time instance. Specifically we develop two decision fusion models to make the final decision where in the first model, the impact of the node mobility is taken into account to update the decision at the fusion center, while in the second model, the impact of node mobility is taken at the node-level decisions. Since allowing a large number of nodes to be mobile increases the cost, we characterize analytically, the required minimum fraction of mobile nodes to be directed to move in order to achieve a desired performance level within a desired delay constraint. We investigate the performance gain achieved by the hybrid sensor network when the network parameters are changing and discuss the scenarios where the node mobility is essentially improves the network performance.

The organization of the Chapter is as follows: Section 9.2 explains the sensor network and the observation models, and presents the problem formulation. In Section 9.3, we develop a decision fusion model in which the fusion center updates the decisions over time while nodes make binary decisions based on the observations collected during one time step when the target location is random. Also, mobility management schedule is proposed to maximize the detection probability at the fusion center within a desired delay constraint. In this discussion, the effect of the node mobility is taken into account at the fusion center decision updating. In Section 9.4 a decision fusion model is developed in which the effect of the node mobility is taken into account at the node-level decisions. In Section 9.5, we develop an analytical procedure to find the minimum number of mobile nodes that should be incorporated with static nodes to achieve a desired performance level within a desired delay constraint. Performance results are given in Section 9.6 and the concluding remarks are given in Section 9.7.

9.2 Problem Formulation and System Model

We consider a hybrid sensor network made of N number of total sensors. We assume that there is N_s number of static nodes and a maximum of N_m number of mobile nodes initially deployed in a square region with dimensions $b \times b$. Note that when mobile nodes are not in the mobile configuration, they make measurements at their stationary configuration. Denote $\lambda_m = \frac{N_m}{N}$ and $\lambda_s = \frac{N_s}{N}$ to be the fractions of mobile and static nodes, respectively. Denote (x_{sk}, y_{sk}) to be the location of the k -th static node which is assumed to be fixed after initial deployment. Let \mathcal{V} be the set of all node indices in the network and let \mathcal{V}_m and \mathcal{V}_s to be the sets containing mobile and static node indices, respectively.

9.2.1 Problem formulation

In this Chapter, we assume that the network is kept stationary until a target is detected at a certain confidence level. We also assume that the network does not have any information regarding sensing field at the time of deployment. Information regarding possible target locations may be available to the network after initial deployment and, the target can be appeared in a particular target location during a certain period of time. Because of these factors, it is not possible to deploy mobile sensors to cover possible target locations at the time of deployment, and on the other hand, mobile nodes may be required to perform on-demand for different purposes. The key contributions in this Chapter are three-fold.

1. Develop decision fusion architectures for the target detection by hybrid sensor network when the target location is random. Specifically, we propose two decision fusion architectures where in the first one, the effect of the node mobility is taken into account for the decision updating at the fusion center and nodes make binary decisions based on the observations during one step movement. In the second model, nodes take the effect of the node mobility into account for node-level decision updating.
2. Manage node mobility to improve (maximize) the system performance within a desired delay constraint after a target is initially detected by the stationary configuration at certain confidence level.
3. The cost of mobile nodes is evaluated in terms of the minimum number of mobile nodes required to achieve a desired performance level within a desired delay constraint.

9.2.2 Node mobility model

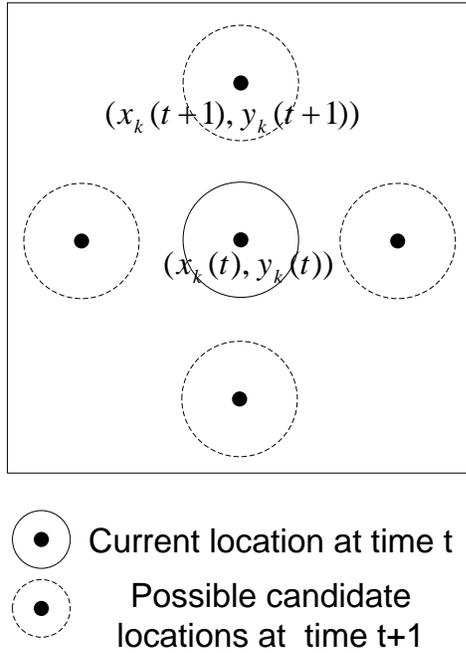
We assume limited mobility of mobile nodes where at each time step each mobile node can only move in one of the pre-determined set of locations (or directions) as shown in Fig. 9.1 (for example), and the maximum total distance it can move in any direction is bounded. This mobility model is justifiable in cases where a node can move to a limited number of locations from its current position due to terrain constraints. Let the velocity of mobile node k at time t be $\mathbf{u}_k(t) = (u_k(t), \theta_k(t)) = (u_k, \theta_k(t))$ where $u_k(t) = u_k$ is the speed of the node k that is assumed to be constant and $\theta_k(t)$ is the direction of node k at time t . Denote l_{max}^k is the maximum distance that the k -th mobile node can move with the available resources. At each time step T_s , mobile node k moves with an average speed of u_k in a direction θ_k selected from a set $\Theta = \{\theta^1, \theta^2, \dots, \theta^K\}$. Selection of θ_k at each time step T_s is considered in later sections. Let $(x_k(t), y_k(t))$ denote the location of the k -th mobile node at time t . Under this mobility model, the location $(x_k(t), y_k(t))$ of the k -th mobile node at time $jT_s \leq t \leq (j+1)T_s$ is given by

$$\begin{aligned} x_k(t) &= x_k(jT_s) + (t - jT_s)u_k \cos \theta_k(jT_s) \\ y_k(t) &= y_k(jT_s) + (t - jT_s)u_k \sin \theta_k(jT_s), \end{aligned}$$

for $k \in \mathcal{V}_m$, and $j = 0, 1, 2, \dots$, where $\theta_k(jT_s) \in \Theta$ is the selected direction at time jT_s and $(x_k(0), y_k(0))$ are X and Y coordinates of the initial location of the k -th mobile node.

9.2.3 Observation model

At each time step both mobile and static nodes make observations on the presence/absent of the target and make a binary decision on whether the target is present


 Figure 9.1: Candidate locations for a mobile node at time t

or absent. We consider the observation models for mobile and static nodes as given below at time $0 < t \leq nT_s$ under hypotheses H_1 (target present) and H_0 (target absent):

$$\begin{aligned}
 H_1 & : z_k(t) = m_k(t) + v_k(t); \text{ for } 0 < t \leq nT_s, \\
 H_0 & : z_k(t) = v_k(t); \text{ for } 0 < t \leq nT_s,
 \end{aligned} \tag{9.1}$$

for $k \in \mathcal{V}$, where $\{m_k(t), t \in (0, nT_s]\}$ is the signal strength received from the target at time t , $\{v_k(t), t \in [0, nT_s]\}$ is the measurement noise process at the k -th node which is assumed to be white Gaussian with mean zero and the auto-covariance function $C_n(t_1, t_2) = \sigma_v^2 \delta(t_1 - t_2)$, $t_1, t_2 \in [0, nT_s]$ where $\delta(\cdot)$ denotes the Dirac delta

function.

The received sensing signal $m_k(t)$ represents the attenuated (over distance) signal emitted by the target. Depending on the sensing modalities, (such as acoustic, seismic, IR, etc.), different models for received signal strength can be used. For this discussion, we assume the following model for the signal $m_k(t)$, which assumes that the signal emitted by the target decays as the distance from the target to the sensing node increases [71, 97]:

$$m_k(t) = \frac{A_0}{r_k^{\alpha'/2}(t)}, \text{ for } 0 \leq t \leq nT_s, \quad k \in \mathcal{V}_m \quad (9.2)$$

where A_0 is the signal strength emitted by the target,

$r_k(t) = \sqrt{(x_k(t) - x_0)^2 + (y_k(t) - y_0)^2}$ is the distance between the k -th mobile node and the target at time t for $0 \leq t \leq nT_s$, (x_0, y_0) is the location of the stationary target and α' is the path loss index that is assumed to be 2 throughout. Note that, for static nodes (9.2) reduces to

$$m_k(t) = \frac{A_0}{r_k^{\alpha'/2}}, \quad k \in \mathcal{V}_s, \quad (9.3)$$

where $r_k = \sqrt{(x_{sk} - x_0)^2 + (y_{sk} - y_0)^2}$. However, the results presented in this Chapter can be generalized to other sensing modalities as well.

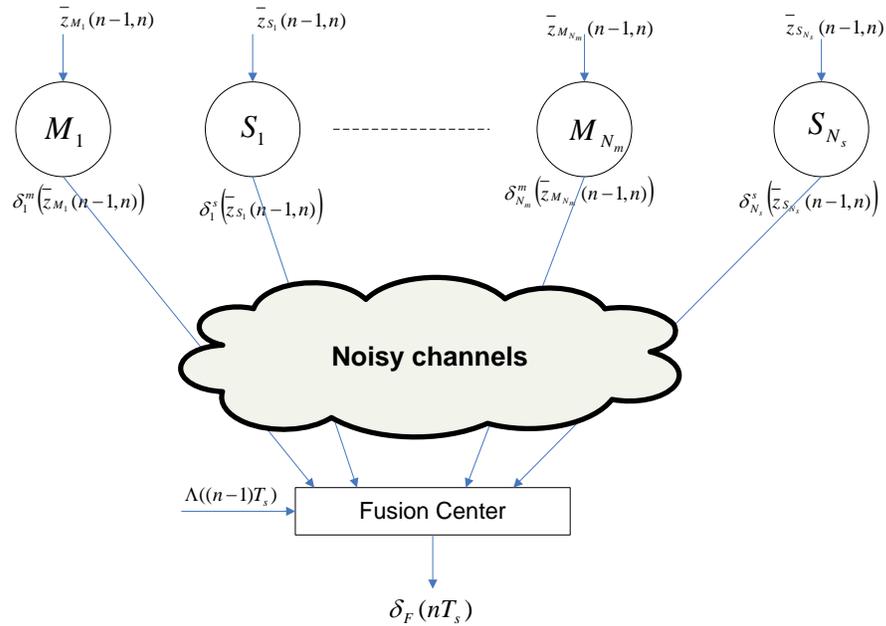


Figure 9.2: Decision fusion architecture for the hybrid sensor network with fusion center updating decisions over time

9.3 Detection Performance with Decision Fusion

Architecture 1: Fusion Center Updating Decisions Over Time

In this section we consider the performance dynamics of hybrid sensor network when the exact target location is unknown. At the stationary configuration, we assume that the network monitors the Field of Interest (FoI) continuously, and mobile nodes are directed to move when a possible target is detected with relatively lower confidence level by the stationary configuration. The target location coordinates, x_0 and y_0 are assumed to be random variables with known statistics.

After initially detected by a lower confidence level, we assume that mobile and static nodes make binary decisions at each time instant nT_s for $n = 1, 2, \dots$ based on the observations collected during the current time interval. Formally, the k -th node performs the following hypothesis testing problem at time $t = nT_s$:

$$\begin{aligned} H_1 & : z_k(t) = m_k(t) + v_k(t); \text{ for } (n-1)T_s < t \leq nT_s, \\ H_0 & : z_k(t) = v_k(t); \text{ for } (n-1)T_s < t \leq nT_s, \end{aligned} \quad (9.4)$$

Each node transmits its local decision to the fusion center over a noisy communication channel. The fusion center combines these local decisions from mobile and static nodes and the previous information at the fusion center to make a final decision. The corresponding decision fusion architecture is shown in Fig. 9.2 where the symbols used in Fig. 9.2 are defined in subsections 9.3.1 and 9.3.3. As shown in Fig. 9.2, at each time instant nT_s , each node performs a local detection based on the observations collected at the current time interval $(n-1)T_s \leq t < nT_s$. The corresponding local decisions are transmitted to the fusion center over a noisy communication channel at each time step $t = nT_s$. The fusion center updates the final decision based on the noisy corrupted decisions received from both static and mobile nodes at time nT_s , and the previous information at the fusion center at time $(n-1)T_s$ to reach a final decision. In this architecture, the impact of the mobility of mobile nodes is taken into account at the fusion center while the mobile nodes make a local decision based on observations collected at one-step movement.

9.3.1 Detection performance at k -th mobile node

Denote $z_k(t; n-1, n) = (z_k(t))_{t=(n-1)T_s}^{nT_s}$. Note that according to the signal model (9.2) assumed in the Chapter, the signal strength received by a sensor node is decreasing as the distance between the node location and the target location is increasing. If

a simple constant threshold testing is performed on the received signal strength [97] (or on energy [122]) at a sensor node to determine the target is present/absent, it can be seen that more false alarms will occur at the nodes located relatively far away from the target location if the threshold is chosen too small, or miss probability will be higher at sensors located closer to the target location, if the threshold is too large. [97] has provided an approach of selecting an optimal threshold such that the performance at the fusion center is maximized for a static sensor network. However, in this Chapter since mobile nodes are directed to move when required, maintaining a constant threshold test on signal strength (or energy) to determine the presence/absence of a target would not essentially reflect the performance gain achieved by node mobility. Thus it is required to have a dynamically varying threshold at sensor nodes to exploit the impact of node mobility in an effective way. Thus, in this Chapter we consider that k -th mobile node to perform likelihood ratio testing on its observations. Explicitly we assume that each node performs α -level Neyman-Pearson (N-P) optimum test to detect the presence/absence of the target at each time nT_s , since the ultimate goal in this Chapter is to manage node mobility to result optimal detection probability.

According to the detection problem at the k -th mobile node as given by (9.4), the log likelihood ratio based on the observations collected during time interval $((n-1)T_s, nT_s]$, $L_k(z_k(t; n-1, n))$, conditioned on the target location (x_0, y_0) , at the k -th mobile node can be shown to be [105],

$$\begin{aligned}
 & L_k(z_k(t; n-1, n)|(x_0, y_0)) \\
 = & \log \frac{dP_1}{dP_0}(z_k(t; n-1, n)) \\
 = & \frac{1}{\sigma_v^2} \int_{(n-1)T_s}^{nT_s} m_k(t; x_0, y_0) z_k(t) dt - \frac{1}{2\sigma_v^2} \int_{(n-1)T_s}^{nT_s} m_k^2(t; x_0, y_0) dt, \\
 = & \frac{\tilde{z}_k(n-1, n)}{\sigma_v^2} - \frac{E_k^m(n-1, n)}{2\sigma_v^2}, \tag{9.5}
 \end{aligned}$$

for $k \in \mathcal{V}_m$ where $\tilde{z}_k(l, n) = \int_{lT_s}^{nT_s} m_k(t; x_0, y_0) z_k(t) dt$ and

$E_k^m(l, n) = \int_{lT_s}^{nT_s} m_k^2(t; x_0, y_0) dt$ for $n = 1, 2, \dots$ and

$m_k(t; x_0, y_0) = \frac{A_0}{\sqrt{(x_k(t)-x_0)^2+(y_k(t)-y_0)^2}}$ as defined in (9.2). Computation of $E_k^m(l, n)$

for a given target location is given in subsection 9.4.1. Then the log likelihood ratio

$L_k(z_k(t; n-1, n))$ is given by,

$$\begin{aligned} L_k(z_k(t; n-1, n)) &= \mathbb{E}_{x_0, y_0} \left\{ \frac{\tilde{z}_k(n-1, n)}{\sigma_v^2} - \frac{E_k^m(n-1, n)}{2\sigma_v^2} \right\} \\ &= \frac{1}{\sigma_v^2} \int_{(n-1)T_s}^{nT_s} z_k(t) \overline{m_k(t)} dt - \frac{1}{2\sigma_v^2} \bar{E}_k^m(n-1, n), \end{aligned} \quad (9.6)$$

where $\overline{m_k(t)} = \mathbb{E}_{x_0, y_0} \{m_k(t; x_0, y_0)\}$ and $\bar{E}_k^m(n-1, n) = \mathbb{E}_{x_0, y_0} \{E_k^m(n-1, n)\}$. Com-

putation of $\overline{m_k(t)}$ is associated with the specific probabilistic model for the target

location distribution. For the evaluation used in this Chapter, the closed-form ex-

pression for $\overline{m_k(t)}$ with assumed target location distribution model is given in Section

9.6. Assuming no point masses in the pdf of $L_k(z_k(t; n-1, n))$, the optimal decision

rule at the k -th mobile node at time $t = nT_s$ for the hypothesis problem (9.4) is

given by (according to the N-P-criteria [105])

$$\delta_k^m(nT_s) = \begin{cases} 1 & \geq \\ \text{if } L_k(z_k(t; n-1, n)) & \eta_k^m(n) \\ 0 & < \end{cases} \quad (9.7)$$

where $\eta_k^m(n) \geq 0$ is uniquely determined such that, the false alarm probability at the

k -th node at time nT_s , $P_{f_k}^m(nT_s) = \alpha$ for $k \in \mathcal{V}_m$. Note that we assume that each

node performs same α -level N-P test at each time nT_s . The decision rule (9.7) can

be further simplified to,

$$\delta_k^m(nT_s) = \begin{cases} 1 & \geq \\ \text{if } \bar{z}_k(n-1, n) & \tau_k^m(n) \\ 0 & < \end{cases} \quad (9.8)$$

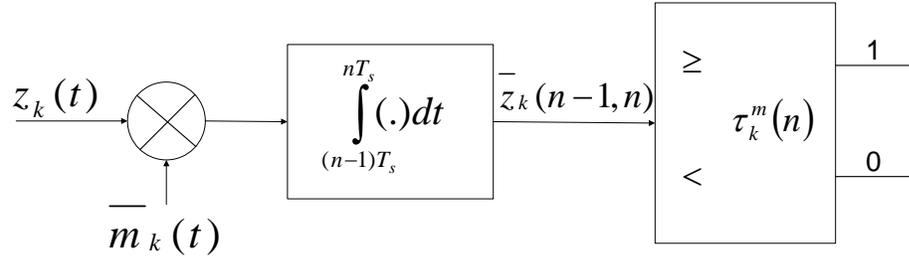


Figure 9.3: Detector structure at the k -th mobile node for the decision making based on the observations during time interval $((n-1)T_s, nT_s]$

where $\bar{z}_k(n-1, n) = \int_{(n-1)T_s}^{nT_s} z_k(t) \overline{m_k(t)} dt$ is the new decision statistic and $\tau_k^m(n) = \sigma_v^2 \eta_k^m(n) + \frac{1}{2} \bar{E}_k^m(n-1, n)$ is the new threshold, at the k -th mobile node for $k \in \mathcal{V}_m$.

Result 1. For α -level N - P test, the threshold $\tau_k^m(n)$ and the detection probability at the k -th mobile node at time nT_s are given by,

$$\tau_k^m(n) = \sigma_v Q^{-1}(\alpha) \sqrt{E_k^{\bar{m}}(n-1, n)} \quad (9.9)$$

and

$$\bar{P}_{d_k}^m(nT_s) = \mathbb{E}_{x_0, y_0} \left\{ Q \left(Q^{-1}(\alpha) - \frac{\tilde{E}_k(x_0, y_0; n-1, n)}{\sigma_v \sqrt{E_k^{\bar{m}}(n-1, n)}} \right) \right\} \quad (9.10)$$

respectively, where $\tilde{E}_k(x_0, y_0; n-1, n) = \int_{(n-1)T_s}^{nT_s} m_k(t; x_0, y_0) \overline{m_k(t)} dt$ and $E_k^{\bar{m}}(n-1, n) = \int_{(n-1)T_s}^{nT_s} \overline{m_k^2(t)} dt$.

Note that to evaluate $\bar{P}_{d_k}^m(nT_s)$ as in (9.10) at k -th mobile node, in general a 2-fold integration is required. The block diagram of the detector at the k -th mobile node is shown in Fig. 9.3.

9.3.2 Detection performance at k -th static node

Result 2. For static nodes, the optimal threshold and the detection probability for the α -level N - P test are given by,

$$\tau_k^s(n) = \tau_k^s = \sigma_v Q^{-1}(\alpha) \bar{m}_k \sqrt{T_s} \quad (9.11)$$

and

$$\bar{P}_{d_k}^s(nT_s) = \mathbb{E}_{x_0, y_0} \left\{ Q \left(Q^{-1}(\alpha) - \frac{m_k(x_0, y_0) \sqrt{T_s}}{\sigma_v} \right) \right\}. \quad (9.12)$$

Note that the detection threshold τ_k^s for a static node is a constant over the time.

9.3.3 Performance evaluation at fusion center with noisy communication

To evaluate the performance of the hybrid sensor network, let us assume that the nodes send their local decisions to the fusion center over binary symmetric channels (BSC) which can be used to model noisy channels [48, 97]. Denote $w_k(nT_s)$ to be the received signal at the fusion center from the k -th node at time $t = nT_s$. We assume that the k -th node transmits its local decision over a BSC with a cross-over probability p_k , and that the channels of N nodes are independent of each other. The received signals at the fusion center under the two hypotheses at time nT_s for

$n = 1, 2, \dots$ are given by

$$\begin{aligned}
 w_k(nT_s) &= \begin{cases} 1 & \text{with prob } \mu_{d_k}^m(nT_s) \\ 0 & \text{with prob } 1 - \mu_{d_k}^m(nT_s) \end{cases} \quad \text{for } k \in \mathcal{V}_m \\
 &= \begin{cases} 1 & \text{with prob } \mu_{d_k}^s(nT_s) \\ 0 & \text{with prob } 1 - \mu_{d_k}^s(nT_s) \end{cases} \quad \text{for } k \in \mathcal{V}_s
 \end{aligned} \tag{9.13}$$

under H_1 and

$$\begin{aligned}
 w_k(nT_s) &= \begin{cases} 1 & \text{with prob } \mu_{f_k}^m(nT_s) \\ 0 & \text{with prob } 1 - \mu_{f_k}^m(nT_s) \end{cases} \quad \text{for } k \in \mathcal{V}_m \\
 &= \begin{cases} 1 & \text{with prob } \mu_{f_k}^s(nT_s) \\ 0 & \text{with prob } 1 - \mu_{f_k}^s(nT_s) \end{cases} \quad \text{for } k \in \mathcal{V}_s
 \end{aligned} \tag{9.14}$$

under H_0 where $\mu_{d_k}^m(nT_s) = \bar{P}_{d_k}^m(nT_s)(1-p_k) + (1-\bar{P}_{d_k}^m(nT_s))p_k$, $\mu_{d_k}^s(nT_s) = \bar{P}_{d_k}^s(nT_s)(1-p_k) + (1-\bar{P}_{d_k}^s(nT_s))p_k$, $\mu_{f_k}^m(nT_s) = \alpha(1-p_k) + (1-\alpha)p_k$ and $\mu_{f_k}^s(nT_s) = \alpha(1-p_k) + (1-\alpha)p_k$.

The fusion center makes a final decision at time $t = nT_s$ using the majority rule based on the received signals from all nodes at time nT_s and the previous available at the fusion center at time $(n-1)T_s$. Denote the decision statistic at the fusion center to be $\Lambda(nT_s)$, where

$$\Lambda(nT_s) = \frac{1}{n} \left(\tilde{\Lambda}((n-1)T_s) + \sum_{k \in \mathcal{V}_m} w_k(nT_s) + \sum_{k \in \mathcal{V}_s} w_k(nT_s) \right),$$

where $\tilde{\Lambda}((n-1)T_s) = \sum_{j=1}^{n-1} \sum_{k \in \mathcal{V}_m} w_k(jT_s) + \sum_{j=1}^{n-1} \sum_{k \in \mathcal{V}_s} w_k(jT_s)$ which can be updated recursively over time. The final decision at the fusion center is then given by,

$$\delta_F(nT_s) = \begin{cases} 1 & \geq \\ \text{if } \Lambda(nT_s) & \rho, \\ 0 & < \end{cases} \tag{9.15}$$

where ρ is the threshold of the majority decision rule at the fusion. Note that, for general non-identical BSCs, $\Lambda(nT_s)$ is a sum of independent but non-identical binary random variables. Using the Lindberg-Feller Central Limit Theorem (LF-CLT) for independent random variables (under certain regularity conditions as shown in Appendix 9A) [38], it can be shown that $\Lambda(nT_s)$ is distributed under two hypotheses as,

$$\text{Under } H_1 : \Lambda(nT_s) \sim \mathcal{N}(\mu_{\Lambda_1}, \sigma_{\Lambda_1}^2)$$

and

$$\text{Under } H_0 : \Lambda(nT_s) \sim \mathcal{N}(\mu_{\Lambda_0}, \sigma_{\Lambda_0}^2).$$

where $\mu_{\Lambda_1}(nT_s) = \frac{1}{n} \sum_{j=1}^n \sum_{k \in \mathcal{V}_m} \mu_{d_k}^m(jT_s) + \frac{1}{n} \sum_{j=1}^n \sum_{k \in \mathcal{V}_s} \mu_{d_k}^s(jT_s)$,
 $\sigma_{\Lambda_1}^2(nT_s) = \frac{1}{n^2} \sum_{j=1}^n \sum_{k \in \mathcal{V}_m} \mu_{d_k}^m(jT_s)(1 - \mu_{d_k}^m(jT_s)) + \frac{1}{n^2} \sum_{j=1}^n \sum_{k \in \mathcal{V}_s} \mu_{d_k}^s(jT_s)(1 - \mu_{d_k}^s(jT_s))$,
 $\mu_{\Lambda_0}(nT_s) = \frac{1}{n} \sum_{j=1}^n \sum_{k \in \mathcal{V}_m} \mu_{f_k}^m(jT_s) + \frac{1}{n} \sum_{j=1}^n \sum_{k \in \mathcal{V}_s} \mu_{f_k}^s(jT_s)$ and
 $\sigma_{\Lambda_0}^2(nT_s) = \frac{1}{n^2} \sum_{j=1}^n \sum_{k \in \mathcal{V}_m} \mu_{f_k}^m(jT_s)(1 - \mu_{f_k}^m(jT_s)) + \frac{1}{n^2} \sum_{j=1}^n \sum_{k \in \mathcal{V}_s} \mu_{f_k}^s(jT_s)(1 - \mu_{f_k}^s(jT_s))$.
 Then the detection probability at the fusion center can be shown to be

$$P_D(nT_s) = Pr(\Lambda(nT_s) \geq \rho | H_1) = Q\left(\frac{\rho - \mu_{\Lambda_1}(nT_s)}{\sigma_{\Lambda_1}(nT_s)}\right). \quad (9.16)$$

9.3.4 Mobility management for mobile nodes

In this section, we find the best movement schedule for each mobile node in order to maximize the detection probability at the fusion center within a desired delay constraint. We assume that each mobile node moves with the same speed such that $u_k = \bar{u}$ for $k \in \mathcal{V}_m$. Note that, each mobile node can move a distance of $\bar{v}T_s$ during each time period of T_s in a direction selected from the set $\Theta = \{\theta^1, \theta^2, \dots, \theta^K\}$.

Let $\mathcal{C}_k(nT_s)$ be the candidate locations of mobile node k at time nT_s . Note that, if there are no terrain constraints such that nodes can move heading to the possible target locations on straight line, a certain number of steps can be made along a straight line as time goes, and there is only one direction. The following discussion is applicable, if mobile nodes are not in a position to direct towards the possible target locations on a straight line from their original locations due environmental and terrain constraints. Denote T_D to be the desired delay constraint. The requirement is to maximize the detection probability at the fusion center, $P_D(n_D T_s)$, where $n_D = \lfloor \frac{T_D}{T_s} \rfloor$, with optimal movement plan for each mobile node. Equivalently, we need to find the optimal direction that k -th mobile node should move at time jT_s , for $k \in \mathcal{V}_m$ and $j = 0, 1, 2 \dots, n_D$, to obtain the maximum (over all possible movements) detection probability at time $n_D T_s$. Denote $\hat{\Theta}_k = \{\hat{\theta}_k(T_s), \hat{\theta}_k(2T_s), \dots, \hat{\theta}_k(n_D T_s)\}$ be the optimal set of movement directions at each time step for node k . Now the problem can be formulated as,

$$\begin{aligned} & \text{Find the set } \{\hat{\Theta}_1, \hat{\Theta}_2, \dots, \hat{\Theta}_{N_m}\} \\ & \text{such that} \\ & P_D(n_D T_s) \text{ is maximized} \end{aligned} \tag{9.17}$$

If the fusion center were to compute the movement plan beforehand for each mobile node, in general, the optimization has to search over as many as $N_m \times |\Theta| \times n_D$ variables leading to a search space of size $2^{N_m |\Theta| n_D}$ where $|\Theta|$ is the cardinality of the set Θ . Although this brute-force approach will result in the optimal solution, it is computationally expensive. Thus, in the following we propose a near-optimal approach for each mobile node to select its best movement direction at each time step based on its own performance measure: i.e. each node moves in a direction at each time step which would lead to maximum individual performance at time $n_D T_s$.

Note that the detection probability at the fusion center at time nT_s is given by

(9.16). The required optimization problem is to find optimal movement plan for each mobile node k for each $j = 0, 1, 2, \dots, n_D$ such that $P_D(n_D T_s)$ is maximized. Maximizing $P_D(n T_s)$ in (9.16) is equivalent to minimizing the argument in the Q -function. Denote $f(\boldsymbol{\mu}_d)$ be the argument of the Q -function in (9.16) where,

$$f(\boldsymbol{\mu}_d(n T_s)) = \frac{\rho - \frac{1}{n} \boldsymbol{\mu}_d^T(n T_s) \mathbf{e}}{\sqrt{\frac{1}{n^2} \boldsymbol{\mu}_d^T(n T_s) \mathbf{e} - \frac{1}{n^2} \boldsymbol{\mu}_d^T(n T_s) \boldsymbol{\mu}_d(n T_s)}} \quad (9.18)$$

where \mathbf{e} is the Nn -length vector containing all ones,

$\boldsymbol{\mu}_d(n T_s) = [\mu_{d_1}^m(j T_s), \dots, \mu_{d_{N_m}}^m(j T_s), \mu_{d_1}^s(j T_s), \dots, \mu_{d_{N_s}}^s(j T_s), j = 1 \dots, n]^T$ is the Nn -length vector containing all the elements in the sum $\mu_{\Lambda_1}(n T_s)$ so that $\mu_{\Lambda_1}(n T_s) = \frac{1}{n} \boldsymbol{\mu}_d^T(n T_s) \mathbf{e}$. Then as given by [122], since $\mu_{d_k}^m(j T_s), \mu_{d_l}^s(j T_s) \in (0, 1)$ for $k \in \mathcal{V}_m, l \in \mathcal{V}_s$ and $j = 1 \dots, n$, using the first order Taylor series expansion around central point, $f(\boldsymbol{\mu}_d(n T_s))$ can be approximated as,

$$f(\boldsymbol{\mu}_d(n T_s)) \approx -\frac{2}{\sqrt{Nn}} \mu_{\Lambda_1}(n T_s) + \frac{2\rho\sqrt{n}}{\sqrt{N}} + H \quad (9.19)$$

where H denotes the second and higher order terms in the Taylor series expansion.

It is seen from (9.19) that if H and the sum

$\mu_{\Lambda_1}(n_D T_s) = \left[\frac{1}{n} \sum_{j=1}^{n_D} \left(\sum_{k \in \mathcal{V}_m} \mu_{d_k}^m(j T_s) + \sum_{k \in \mathcal{V}_s} \mu_{d_k}^s(j T_s) \right) \right]$ were to be independent of each other, then $f(\boldsymbol{\mu}_d(n T_s))$ will be monotonically decreasing with increasing $\mu_{\Lambda_1}(n T_s)$. ■

It was shown in [122] that, with high probability, $f(\boldsymbol{\mu}_d(n T_s))$ is indeed decreasing when the sum $\mu_{\Lambda_1}(n T_s)$ is increasing. Thus, with high probability, maximizing the detection probability at the fusion center at time $n T_s$ is equivalent to maximizing the sum $\mu_{\Lambda_1}(n T_s)$. Since each mobile and static node perform their detection problems independent of each other, maximizing $\sum_{j=1}^{n_D} \mu_{d_k}^m(j T_s)$ over all possible movement plans for $k \in \mathcal{V}_m$ will maximize the sum $\mu_{\Lambda_1}(n T_s)$ at time $n T_s$.

Note that $\sum_{j=1}^{n_D} \mu_{d_k}^m(jT_s)$ for the k -th mobile node is given by

$$\begin{aligned} \sum_{j=1}^{n_D} \mu_{d_k}^m(jT_s) &= \sum_{j=1}^{n_D} (\bar{P}_{d_k}^m(jT_s)(1 - p_k) + (1 - \bar{P}_{d_k}^m(jT_s))p_k) \\ &= p_k + (1 - 2p_k) \sum_{j=1}^{n_D} \bar{P}_{d_k}^m(jT_s). \end{aligned} \quad (9.20)$$

Hence maximizing $\sum_{j=1}^{n_D} \mu_{d_k}^m(jT_s)$ at k -th mobile node is equivalent to maximizing $\sum_{j=1}^{n_D} \bar{P}_{d_k}^m(jT_s)$ where $\bar{P}_{d_k}^m(jT_s)$ is given by (9.10).

Now the optimization problem is equivalent to finding the optimal set $\hat{\Theta}_k \equiv \{\hat{\theta}_k(T_s), \hat{\theta}_k(2T_s), \dots, \hat{\theta}_k(n_D T_s)\}$ which maximizes the sum of detection probabilities up to time $n_D T_s$ at the k -th mobile node as given in (9.10). Denote $\bar{P}_{d_k}^m(0, n_D) = \sum_{j=1}^{n_D} \bar{P}_{d_k}^m(jT_s)$ to be the sum of detection probabilities at k -th mobile node up to time $n_D T_s$ where $\bar{P}_{d_k}^m(jT_s)$ as given by (9.10) is the detection probability related to the decision made by k -th mobile node based on observations during time interval $((j-1)T_s, jT_s)$. In the following, we convert the required problem into an time expansion graph, such that the required problem becomes a shortest path problem and the optimal solution for the optimization problem can be obtained, for example via Dijkstra shortest path algorithm.

Let $\Theta(jT_s) = \{\theta^1(jT_s), \theta^2(jT_s), \dots, \theta^K(jT_s)\}$ be the state space at time (stage) jT_s for the k -th mobile node which represents the set of directions that the k -th mobile node can move at time jT_s . We assume that each mobile node has the same candidate set of directions that it can move at a given time step (However this assumption can be generalized to have different candidate sets for different mobile nodes).

For clarity, let us write the sum of detection probabilities $\bar{P}_{d_k}^m(0, n_D)$ as,

$$\bar{P}_{d_k}^m(0, n_D) = \sum_{j=1}^{n_D} \bar{P}_{d_k}^m((j-1), j), \quad (9.21)$$

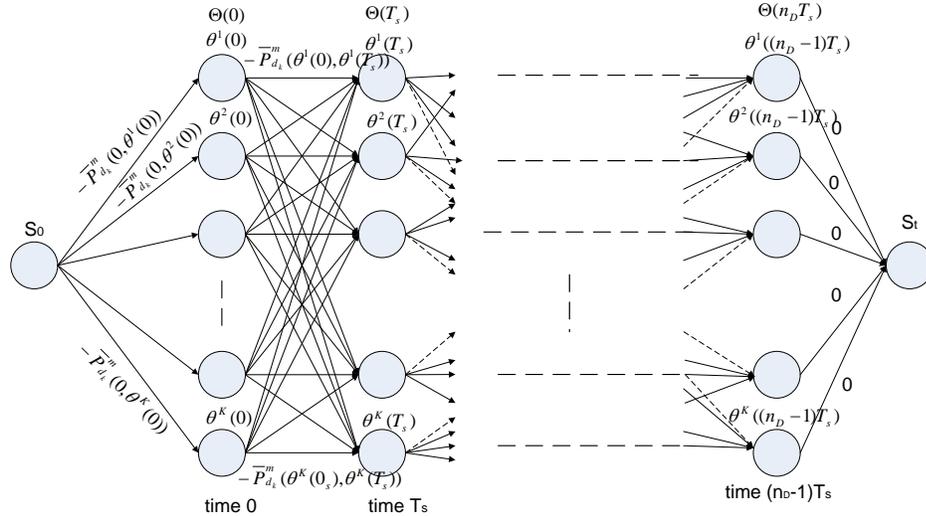


Figure 9.4: Shortest path representation for finding maximum detection probability at time $n_D T_s$ at k -th mobile node

where $\bar{P}_{d_k}^m((j-1), j)$ is the average detection probability corresponding to the decision made based on the observations during the interval $((j-1)T_s, jT_s)$ which is given by (9.10). Now we construct a trellis as shown in Fig. 9.4 where the states of the trellis at time (stage) jT_s represents the directions (states) from the finite set $\Theta(jT_s)$. In Fig. 9.4, the trellis diagram is preceded by s_0 and followed by s_t which are two dummy nodes. Denote $\bar{P}_{d_k}^m(\theta^p(j-1), \theta^q(j))$ represents the detection probability for the decision based on observations collected during transition from state $\theta^p(j-1) \in \Theta((j-1)T_s)$ to $\theta^q(j) \in \Theta(jT_s)$. This represents the detection probability for the decision based on the observations collected during the time interval $(jT_s, (j+1)T_s)$ when the k -th mobile node selects the direction $\theta^q(j)$ at time instant jT_s given that the direction selected at time $(j-1)T_s$ is $\theta^p(j-1)$. Now, branch from s_0 to $\theta^p(0) \in \Theta(0)$ is assigned the metric $-\bar{P}_{d_k}^m(0, \theta^p(0))$ where $\bar{P}_{d_k}^m(0, \theta^p(0))$ represents the average detection probability for the decision based on observations collected by the k -th mobile node if it selects the direction $\theta^p(0)$ from its original location. Branch from state $\theta^p(j-1) \in \Theta((j-1)T_s)$ to $\theta^q(j) \in \Theta(jT_s)$ is assigned

the metric $-\bar{P}_{d_k}^m(\theta^p(j-1), \theta^q j)$ for $j = 1, 2, \dots, n_D - 1$. Then finding maximum value of the sum $\bar{P}_{d_k}^m(0, n_D) = \sum_{j=1}^{n_D} \bar{P}_{d_k}^m((j-1), j)$ from time 0 to time $n_D T_s$ over all possible directions is equivalent to finding the shortest path between the node s_0 to s_t as in the graph shown in Fig. 9.4, and the optimal $\hat{\Theta}_k$ is the set of states in the shortest path, which can be computed, for example, using Dijkstra shortest path algorithm.

Note that in this algorithm, which yields a near optimal solution for the original optimization problem in (9.17), the movement plan for each mobile node needs to be computed beforehand at time $t = 0$ which also requires the knowledge of the candidate set of locations at each time. In the following we propose a sequential approach where the k -th mobile node determines its movement direction at time jT_s based on only its current information and expected information at time $(j+1)T_s$.

We consider the following approach where mobile nodes select best direction to move at time jT_s sequentially. The idea is to select the best location for the k -th mobile node at time step jT_s such that the observations collected during time interval $[jT_s, (j+1)T_s]$ would lead to best detection performance over all possible directions. According to the signal model (9.2), when a mobile node is getting closer to the target, the SNR at the node is increased, subsequently increasing the detection probability at the k -th mobile node. Hence, the direction at time $t = jT_s$ is chosen as in the following:

$$\begin{aligned} \theta_k(jT_s) &= \operatorname{argmax}_{\theta_i \in \Theta} \{ \bar{P}_{d_k}^m((j+1)T_s, \theta_i) \}, \\ &\text{for } j = 0, 1, 2, \dots, n_D, \end{aligned} \tag{9.22}$$

where $\bar{P}_{d_k}^m((j+1)T_s, \theta_i)$ is the average detection probability at the k -th mobile node at time step $(j+1)T_s$ if the direction $\theta_i \in \Theta$ is selected at time jT_s , n_D is the step index at which $P_D(n_D T_s) \geq \xi_D$ for the first time. The average detection probability

at k -th mobile node at time jT_s is as given by (9.10). From the simulation results, we see that, when the candidate set of directions that any mobile node can move at a given time is the same, and a node moves at the same speed in all directions, the performance of this scheme coincides with the near-optimal scheme which is computed based on shortest path algorithm.

9.4 Detection Performance with Decision Fusion

Architecture 2: Nodes Updating Decisions

Over Time

In this section, develop an alternate formulation for decision fusion in the hybrid sensor network when the nodes are updating decisions over time, where the impact of the node mobility is taken into account at the node level decisions. After a target is detected with a lower confidence level, mobile and static nodes perform the following hypothesis testing problem, at time $t = nT_s$, based on the observations collected until time nT_s :

$$\begin{aligned} H_1 & : z_k(t) = m_k(t) + v_k(t); 0 \leq t \leq nT_s, \\ H_0 & : z_k(t) = v_k(t); 0 \leq t \leq nT_s. \end{aligned} \tag{9.23}$$

Note that in this section we consider that each node performs hypothesis testing (9.23) based on the observations collected during the interval $[0, nT_s]$, contrast to Section 9.3. The decision fusion architecture in this case is shown in Fig. 9.5. As shown in Fig. 9.5, at each time instant nT_s , each distributed node performs a local detection based on the observations collected at the current time interval

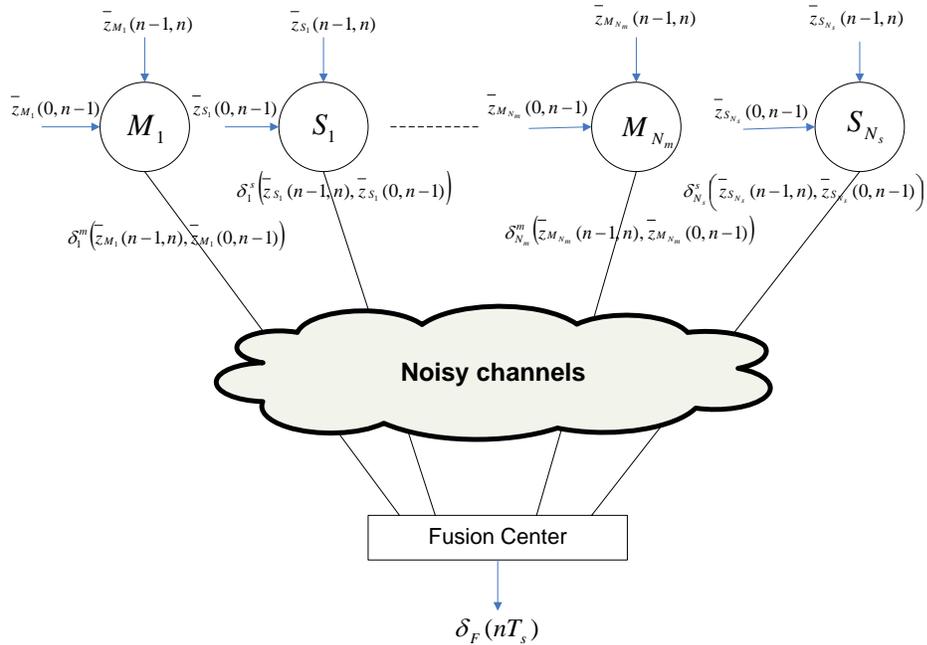


Figure 9.5: Decision fusion architecture for the hybrid sensor network with nodes updating decisions over time

$(n-1)T_s \leq t < nT_s$ and previous observations up to time $(n-1)T_s$ which can be computed recursively for $n = 1, 2, \dots$. These local decisions are transmitted to the fusion center over a noisy communication channel at each time step $t = nT_s$. The fusion center combines these noise corrupted decisions to reach at a final decision on whether the target is present or absent.

9.4.1 Detection performance at k -th mobile node

Similar to Section 9.3, we assume that each node performs α -level N-P detector to decide whether the target is present or absent based on the observations collected during time interval $(0, nT_s]$. The decision statistic and the threshold for the N-P

detector corresponding to (9.8) now can be shown as,

$$\delta_k^m(nT_s) = \begin{cases} 1 & \geq \\ \text{if } \bar{z}_k(0, n) & \tau_k^m(n) \\ 0 & < \end{cases}, \quad (9.24)$$

where $\bar{z}_k(0, n) = \int_0^{nT_s} \overline{z_k m_k(t)} dt = \bar{z}_k(0, n-1) + \int_{(n-1)T_s}^{nT_s} \overline{z_k m_k(t)} dt$ is the decision statistic which can be computed recursively at each time step and $\tau_k^m(n) = \sigma_v^2 \eta_k^m(n) + \frac{1}{2} \bar{E}_k^m(0, n)$ is the corresponding threshold, for $k \in \mathcal{V}_m$ where $\bar{E}_k^m(0, n) = \mathbb{E}_{x_0, y_0} \{E_k^m(0, n)\}$ as defined in subsection 9.3.1.

Result 3. *With decision fusion model 2, for α -level N - P test, the threshold and the detection probability at the k -th mobile node, at time nT_s are given by ,*

$$\tau_k^m(n) = \sigma_v Q^{-1}(\alpha) \sqrt{E_k^m(0, n)} \quad (9.25)$$

and

$$P_{d_k}^m(nT_s) = \mathbb{E}_{x_0, y_0} \left\{ Q \left(Q^{-1}(\alpha) - \frac{\tilde{E}_k(x_0, y_0; 0, n)}{\sigma_v \sqrt{E_k^m(0, n)}} \right) \right\}. \quad (9.26)$$

Note that $E_k^m(0, n)$ at the k -th mobile node is $E_k^m(0, n) = \sum_{j=1}^n \int_{(j-1)T_s}^{jT_s} m_k^2(t) dt$ which is essentially the instant total signal energy received during the period $0 \leq t \leq nT_s$, for a given target location. Then we have,

$$\begin{aligned} E_k^m(j-1, j) &= \int_{(j-1)T_s}^{jT_s} m_k^2(t) dt = \int_{(j-1)T_s}^{jT_s} \frac{A_0^2}{r_k^2(t)} dt \\ &= \int_{(j-1)T_s}^{jT_s} A_0^2 \left((x'_k(j-1) + u_k(t - (j-1)T_s) \cos \theta_k(jT_s))^2 \right. \\ &\quad \left. + (y'_k(j-1) + u_k(t - (j-1)T_s) \sin \theta_k(jT_s))^2 \right)^{-1} dt \end{aligned}$$

$$= \frac{2A_0^2}{\sqrt{\Delta_k(j-1)}} \arctan \left(\frac{2a_k T_s \sqrt{\Delta_k(j-1)}}{\Delta_k(j-1) + b_k(j-1)(b_k(j-1) + 2a_k T_s)} \right) \quad (9.27)$$

where $x'_k(j-1) = x_k((j-1)T_s) - x_0$, $y'_k(j-1) = y_k((j-1)T_s) - y_0$, $a_k = u_k^2$, $b_k(j-1) = 2u_k(x'_k(j-1) \cos \theta_k(jT_s) + y'_k(j-1) \sin \theta_k(jT_s))$, $c_k(j-1) = x_k'^2(j-1) + y_k'^2(j-1)$ and $\Delta_k(j-1) = 4a_k c_k(j-1) - b_k^2(j-1)$. Note that (9.27) holds only if $\Delta_k(j-1) > 0$ which is shown to be true in the following.

$$\begin{aligned} \Delta_k(j-1) &= 4u_k^2(x_k'^2(j-1) + y_k'^2(j-1)) \\ &\quad - 4u_k^2(x_k'(j-1) \cos \theta_k(jT_s) + y_k'(j-1) \sin \theta_k(jT_s))^2 \\ &= 4u_k^2(x_k'(j-1) \sin \theta_k(jT_s) - y_k'(j-1) \cos \theta_k(jT_s))^2 \\ &> 0. \end{aligned}$$

Then $E_k^m(0, n)$ is given by (9.28).

$$E_k^m(0, n) = \sum_{j=1}^n \frac{2A_0^2}{\sqrt{\Delta_k(j-1)}} \arctan \left(\frac{2a_k T_s \sqrt{\Delta_k(j-1)}}{\Delta_k(j-1) + b_k(j-1)(b_k(j-1) + 2a_k T_s)} \right). \quad (9.28)$$

9.4.2 Detection performance at the k -th static node

Similarly, for the k -th static node, the α -level N-P threshold and the detection probability at time nT_s are given by the following result.

Result 4. *For static nodes, the optimal threshold and the detection probability for the α -level NP test are given by,*

$$\tau_k^s(n) = \sigma_v Q^{-1}(\alpha) \bar{m}_k \sqrt{nT_s} \quad (9.29)$$

and

$$P_{d_k}^s(nT_s) = \mathbb{E}_{x_0, y_0} \left\{ Q \left(Q^{-1}(\alpha) - \frac{m_k(x_0, y_0) \sqrt{nT_s}}{\sigma_v} \right) \right\}. \quad (9.30)$$

9.4.3 Decision fusion performance with noisy communication

Similar to subsection 9.3.3, we evaluate the decision fusion performance at the fusion center at time nT_s with BSC channels. Now, since the effect of the mobility is taken at the node level, the decision statistic at the fusion center is taken as, $\Lambda(nT_s)$, where

$$\Lambda(nT_s) = \sum_{k \in \mathcal{V}_m} w_k(nT_s) + \sum_{k \in \mathcal{V}_s} w_k(nT_s),$$

where $w_k(nT_s)$ is same as given by (9.13) and (9.14) under two hypotheses where now $\mu_{d_k}^m(nT_s) = P_{d_k}^m(nT_s)(1 - p_k) + (1 - P_{d_k}^m(nT_s))p_k$, $\mu_{d_k}^s(nT_s) = P_{d_k}^s(nT_s)(1 - p_k) + (1 - P_{d_k}^s(nT_s))p_k$, $\mu_{f_k}^m(nT_s) = \alpha(1 - p_k) + (1 - \alpha)p_k$ and $\mu_{f_k}^s(nT_s) = \alpha(1 - p_k) + (1 - \alpha)p_k$ with $P_{d_k}^m(nT_s)$ and $P_{d_k}^s(nT_s)$ are given by (9.26) and (9.30), respectively. The detection probability corresponding to the decision rule based on majority rule is given by (following a similar approach as in subsection 9.3.3),

$$P_D(nT_s) = Pr(\Lambda(nT_s) \geq \rho' | H_1) = Q \left(\frac{\rho' - \mu_\Lambda(nT_s)}{\sigma_\Lambda(nT_s)} \right), \quad (9.31)$$

where $\mu_\Lambda(nT_s) = \sum_{k \in \mathcal{V}_m} \mu_{d_k}^m(nT_s) + \sum_{k \in \mathcal{V}_s} \mu_{d_k}^s(nT_s)$ and $\sigma_\Lambda^2(nT_s) = \sum_{k \in \mathcal{V}_m} \mu_{d_k}^m(nT_s)(1 - \mu_{d_k}^m(nT_s)) + \sum_{k \in \mathcal{V}_s} \mu_{d_k}^s(nT_s)(1 - \mu_{d_k}^s(nT_s))$ and ρ' is the threshold of the majority rule. In obtaining (9.31), it is required to show that conditions for LF-CLT are held. Following a similar approach as in subsection 9.3.3, it can be shown that the sufficient conditions for LF-CLT are held and details are omitted for brevity.

9.4.4 Mobility management for mobile nodes

Similar to the scenario in Section 9.3, we need to find the best movement schedule for each mobile node in order to maximize the detection probability at the fusion center within a desired delay constraint, when the target location is exactly known. The idea is to find the optimal movement schedule for each mobile node k such that the detection probability at the fusion center within a desired delay constraint, is maximized. As in subsection 9.3.4, denote T_D to be the desired delay constraint and $\hat{\Theta}_k = \{\hat{\theta}_k(T_s), \hat{\theta}_k(2T_s), \dots, \hat{\theta}_k(n_D T_s)\}$ be the optimal set of movement directions at each time step for node k . following a similar approach as in subsection 9.3.4, it can be shown that with high probability, maximizing the detection probability at the fusion center at time nT_s (9.31) is equivalent to maximizing the sum $\mu_\Lambda(nT_s) = \sum_{k \in \mathcal{V}_m} \mu_{d_k}^m(nT_s) + \sum_{k \in \mathcal{V}_s} \mu_{d_k}^s(nT_s)$. Since each mobile and static node performs their detection problems independent of each other, maximizing each $\mu_{d_k}^m(nT_s)$ for $k \in \mathcal{V}_m$ over all possible movement plans will maximize the sum $\mu_\Lambda(nT_s)$ at time nT_s . Similar to subsection 9.3.4, it can be shown that maximizing $\mu_{d_k}^m(nT_s)$ at k -th mobile node is equivalent to maximizing $P_{d_k}^m(nT_s)$ at the k -th mobile node, given by (9.26).

Note that if the exact target location is known, then maximizing (9.26) at the k -th mobile node is equivalent to maximizing the total energy collected during the interval $(0, n]$, $E_k^m(0, n)$ as given in (9.28). Then the approach given in subsection 9.3.4, can be directly used to find the optimal movement directions at each time step, where now the metrics of branches of the trellis in Fig. 9.4 are replaced by $-E_k^m(\theta^p(j-1), \theta^q j)$ which represents the energy collected during transition from state $\theta^p(j-1) \in \Theta((j-1)T_s)$ to $\theta^q(j) \in \Theta(jT_s)$.

9.5 Minimum Set of Mobile Nodes

Since allowing nodes to be mobile is expensive in terms of energy, it is important to determine the minimum number of mobile nodes (from the set \mathcal{V}_m) that should be directed to move to achieve a certain detection probability during a given delay constraint. In the following, we consider the problem of finding the smallest set of mobile nodes in order to maintain the maximum detection probability achieved by time T_D is greater than some threshold value. For the discussion given below, we assume the case where exact target location is known with the decision fusion model as given by Section 9.4 where nodes are updating decisions over time.

Let the required detection probability threshold at time T_D be ξ_D . The problem is to find the minimum set of mobile nodes, that should be used in the network to reach the desired performance level by the desired delay constraint. Formally, we can write the optimization problem as,

$$\begin{aligned}
 & \min |\mathcal{S}_m| \\
 & \text{such that } \mathcal{S}_m \subset \mathcal{V}_m \\
 & \text{and } P_D(n_D T_s) \geq \xi_D
 \end{aligned} \tag{9.32}$$

where, as before, $n_D = \lfloor \frac{T_D}{T_s} \rfloor$. Assuming that $\xi_D \geq \frac{1}{2}$, the inequality (9.32) can be further simplified as given below (For simplicity, we assume perfect communication channels such that $p_k = 0$ for all $k \in \mathcal{V}$):

$$\begin{aligned}
 & P_D(n_D T_s) \geq \xi_D \\
 \equiv & Q\left(\frac{\rho' - \mu_\Lambda(n_D T_s)}{\sigma_\Lambda(n_D T_s)}\right) \geq \xi_D \\
 \equiv & \frac{\rho' - \mu_\Lambda(n_D T_s)}{\sigma_\Lambda(n_D T_s)} \leq \beta
 \end{aligned} \tag{9.33}$$

$$\Rightarrow \rho' - \mu_\Lambda(n_D T_s) \leq \beta \sigma_\Lambda(n_D T_s) \tag{9.34}$$

$$\Rightarrow (\rho' - \mu_\Lambda(n_D T_s))^2 \geq \beta^2 \sigma_\Lambda^2(n_D T_s) \tag{9.35}$$

where $\mu_\Lambda(n_D T_s) = \sum_{k \in \mathcal{S}_m} P_{d_k}^m(n_D T_s) + \sum_{k \in \mathcal{V} \setminus \mathcal{S}_m} P_{d_k}^s(n_D T_s)$ and $\sigma_\Lambda^2(n_D T_s) = \sum_{k \in \mathcal{S}_m} P_{d_k}^m(n_D T_s)(1 - P_{d_k}^m(n_D T_s)) + \sum_{k \in \mathcal{V} \setminus \mathcal{S}_m} P_{d_k}^s(n_D T_s)(1 - P_{d_k}^s(n_D T_s))$ and $\beta = Q^{-1}(\xi_D)$. Note that here $P_{d_k}^m(n_D T_s)$ and $P_{d_k}^s(n_D T_s)$ are given by (9.26) and (9.30) without the outer expectation with respect to target locations (since we assume exact target locations for this analysis). Since mobile nodes which are not moving also make observations at their stationary configuration, we will have the set of static nodes as $\mathcal{V} \setminus \mathcal{S}_m$ when the set of mobile nodes is \mathcal{S}_m . Note that (9.34) is obtained from (9.33) since $\sigma_\Lambda(n_D T_s) > 0$ and (9.35) is obtained from (9.34) since, $\beta < 0$ for $\xi_D \geq \frac{1}{2}$ and $\rho' - \mu_\Lambda(n_D T_s) < 0$ under certain conditions. We can show that when ρ' is chosen as the threshold for majority rule ($\rho' = N/2$) and under the assumption of $P_{d_k}^s(n_D T_s), P_{d_k}^m(n_D T_s) \geq \frac{1}{2}$, at $n_D T_s$ for $k \in \mathcal{V}$ such that $\mu_\Lambda(n_D T_s) \geq \frac{N}{2}$, then $\rho' - \mu_\Lambda(n_D T_s) < 0$. The inequality (9.35) can be further simplified as in (9.36).

$$\begin{aligned}
 & - (2\rho' + \beta^2) \sum_{k \in \mathcal{S}_m} P_{d_k}^m(n_D T_s) + \left(\sum_{k \in \mathcal{S}_m} P_{d_k}^m(n_D T_s) \right)^2 + \beta^2 \sum_{k \in \mathcal{S}_m} P_{d_k}^{m2}(n_D T_s) \\
 & + 2 \sum_{k \in \mathcal{S}_m} P_{d_k}^m(n_D T_s) \sum_{k \in \mathcal{V} \setminus \mathcal{S}_m} P_{d_k}^s(n_D T_s) \\
 & \geq (2\rho' + \beta^2) \sum_{k \in \mathcal{V} \setminus \mathcal{S}_m} P_{d_k}^s(n_D T_s) - \beta^2 \sum_{k \in \mathcal{V} \setminus \mathcal{S}_m} P_{d_k}^{s2}(n_D T_s) \\
 & - \left(\sum_{k \in \mathcal{V} \setminus \mathcal{S}_m} P_{d_k}^s(n_D T_s) \right)^2 - \rho'^2. \tag{9.36}
 \end{aligned}$$

The problem is to find the minimum size set \mathcal{S}_m such that, inequality (9.36) is satisfied. To find this, in general we need to search over a maximum of 2^{N_m} possibilities.

In the following we will show how to obtain the solution with reduced complexity under certain conditions. Note that as discussed in subsection 9.4.4, the maximum

$P_{d_k}^m(n_D T_s)$ for each $k \in \mathcal{V}_m$ at time $n_D T_s$ can be computed. Without loss of generality, let us arrange $P_{d_k}^m(n_D T_s)$'s in descending order for $k \in \mathcal{V}_m$ such that $P_{d_1}^m(n_D T_s) \geq P_{d_2}^m(n_D T_s) \geq \dots \geq P_{d_{N_m}}^m(n_D T_s)$. Then the set denoted by \mathcal{S}_m^k consists of the indices of first k mobile nodes. Now define two functions $f_1(k)$ and $f_2(k)$ such that,

$$f_1(k) = (2\rho' + \beta^2) \sum_{j=1}^k P_{d_j}^m(n_D T_s) \quad (9.37)$$

and

$$\begin{aligned} f_2(k) &= \left(\sum_{j=1}^k P_{d_j}^m(n_D T_s) \right)^2 + \beta^2 \sum_{j=1}^k P_{d_j}^{m2}(n_D T_s) \\ &+ 2 \sum_{j=1}^k P_{d_j}^m(n_D T_s) \sum_{j \in \mathcal{V} \setminus \mathcal{S}_m^k} P_{d_j}^s(n_D T_s) \end{aligned} \quad (9.38)$$

for $k = 0, 1, 2, \dots, N_m$ with $f_1(0) = f_2(0) = 0$.

Denote $\mathcal{V}_s^k = \mathcal{V} \setminus \mathcal{S}_m^k$ to be the set containing all static node indices and the mobile node indices from $k + 1$ to N_m , for $k = 1, \dots, N_m$. Clearly, $\mathcal{V}_s^0 = \mathcal{V}$ and $\mathcal{V}_s^{N_m} = \mathcal{V}_s$. Define $K_s(n_D, k)$ to be,

$$\begin{aligned} K_s(n_D, k) &= (2\rho' + \beta^2) \sum_{j \in \mathcal{V}_s^k} P_{d_j}^s(n_D T_s) - \beta^2 \sum_{j \in \mathcal{V}_s^k} P_{d_j}^{s2}(n_D T_s) \\ &- \left(\sum_{j \in \mathcal{V}_s^k} P_{d_j}^s(n_D T_s) \right)^2 - \rho'^2. \end{aligned} \quad (9.39)$$

Theorem 9. *If $P_{d_k}^m(n_D T_s)$'s are arranged in descending order and $f_1(k)$, $f_2(k)$ and $K_s(n_D, k)$ are defined as in (9.37), (9.38) and (9.39), respectively, then we can find*

a unique K_0 such that,

$$f_2(k) - f_1(k) \geq K_s(n_D, k), \text{ for } k \geq K_0 \quad (9.40)$$

and

$$f_2(k) - f_1(k) < K_s(n_D, k), \text{ for } k < K_0. \quad (9.41)$$

Then K_0 is the minimum number of mobile nodes that should be used to meet the desired criteria where the minimum set $\mathcal{S}_m^{K_0} = \{1, 2, \dots, K_0\}$.

Proof. See Appendix 9B. □

9.6 Performance Results

In this section, we evaluate the performance of the proposed target detection schemes using a hybrid sensor network. We assume there is a total of 30 sensors deployed in a square region of area $100 \times 100m^2$ where the center is at $(0, 0)$. We assume that mobile node speed is a constant for all directions and the same for all nodes. The time step $T_s = 1s$ and each mobile node's speed is $\bar{u} = 1m/s$. We define the nominal SNR at each node to be $\gamma_0 = A_0^2/\sigma_v^2$. We also assume that the communication between nodes and the fusion center is over i.i.d. BSC's such that $p_k = p$ for all $k \in \mathcal{V}$. At each time step, we assume that a mobile node can move a distance of $\bar{u}T_s$ in directions corresponding to due-east, north-east, due-north, north-west, due-west, south-west, due-south and south-east or remain at the current location. To

illustrate the detection performance of proposed schemes, we assume that X and Y coordinates of the target location are with the following distribution:

$$x_0 = \begin{cases} x_a & \text{with } q_x \\ x_b & \text{with } 1 - q_x \end{cases} \quad (9.42)$$

and

$$y_0 = \begin{cases} y_a & \text{with } q_y \\ y_b & \text{with } 1 - q_y \end{cases} \quad (9.43)$$

This type of target location model is justifiable in situations when there is a finite number of possible surveillance locations that the target can appear probabilistically, based on the prior knowledge or new information received after initial deployment.

Note that with this target location model, $\overline{m_k(t)}$ is given by,

$$\begin{aligned} \overline{m_k(t)} &= \frac{A_0}{\sqrt{(x_k(t) - x_a)^2 + (y_k(t) - y_a)^2}} q_x q_y + \frac{A_0}{\sqrt{(x_k(t) - x_a)^2 + (y_k(t) - y_b)^2}} q_x (1 - q_y) \\ &+ \frac{A_0}{\sqrt{(x_k(t) - x_b)^2 + (y_k(t) - y_a)^2}} (1 - q_x) q_y \\ &+ \frac{A_0}{\sqrt{(x_k(t) - x_b)^2 + (y_k(t) - y_b)^2}} (1 - q_x) (1 - q_y) \end{aligned}$$

However, the proposed scheme is valid for any target distribution and the complexity of the local detectors is determined by the complexity of evaluating $\overline{m_k(t)}$. For a general target location distribution model with $x_0 \sim f_a(x_0)$ and $y_0 \sim f_b(y_0)$, we can find $\overline{m_k(t)}$ as,

$$\overline{m_k(t)} = \int \int_{x_0 y_0} \frac{A_0}{\sqrt{(x_k(t) - x_0)^2 + (y_k(t) - y_0)^2}} f_a(x_0) f_b(y_0) dx_0 dy_0. \quad (9.44)$$

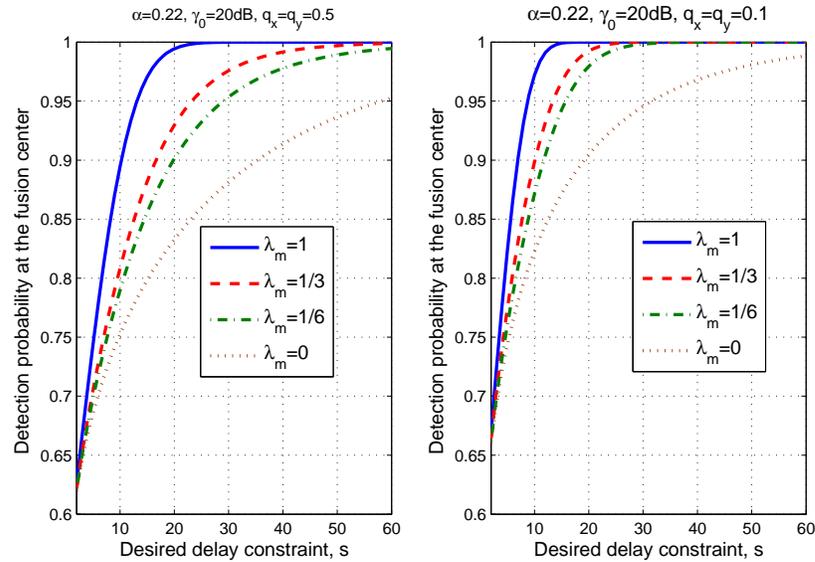


Figure 9.6: Detection probability at the fusion center with desired detection delay with perfect communication, with decision fusion model 1: $T_s = 1s$, $\bar{u} = 1m/s$, $\gamma_0 = 20dB$, $p = 0$.

9.6.1 Performance evaluation with decision fusion architecture 1

In Fig.9.6, the time varying detection performance is shown when the target location is random based on the decision fusion scheme discussed in Section 9.3 where the nodes make binary decision based on the observations collected during current time interval and the fusion center updates the decisions incorporating the current local decisions and previous information at the fusion center. In Fig. 9.6, we let $\gamma_0 = 20dB$, $\alpha = 0.22$, and $x_a = -25, x_b = 25, y_a = -25, y_b = 25$. The detection performance is shown with varying the fraction of mobile nodes and for $p = 0$. Subplot in the left hand side shows the performance for $q_x = q_y = 0.5$ while the one in right hand side is corresponding to $q_x = q_y = 0.1$. It can be seen that with the decision fusion model as described in Section 9.3, by allowing even a small fraction of mobile

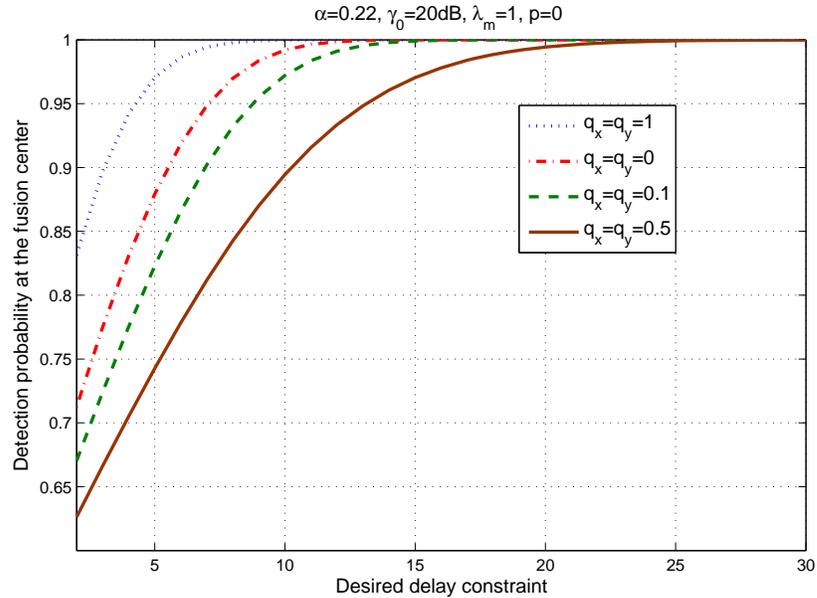


Figure 9.7: Comparing the detection performance with exact and random target locations, with decision fusion model 1 : $T_s = 1s$, $\bar{u} = 1m/s$, $\gamma_0 = 20dB$, $\alpha = 0.22$, $p = 0$.

nodes will improve the detection performance significantly when compared to a all-static network. As time goes, since mobile nodes getting closer to possible target locations, according to the given probability distribution for target locations, a mobile node can make a binary decision based on the information collected at current time interval with a higher confidence level, when compared to a stationary sensor. On the other hand, stationary sensors make binary decisions based on its observations collected during current time interval, and the quality of these decisions remains the same over time since nodes are stationary. Then, according to the decision fusion model described in Section 9.3, fusion center receives binary decisions over the time with higher confidence level when there are mobile sensors, resulting an improved performance compared to a all-stationary network.

Figure 9.7 shows the performance of decision fusion scheme presented in Section

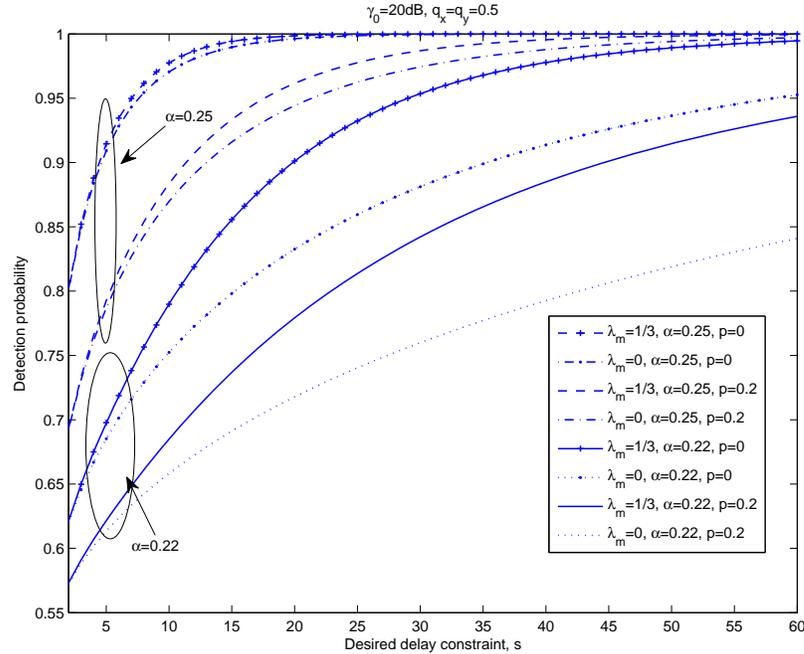


Figure 9.8: Detection performance with varying α and p , with decision fusion model 1: $T_s = 1s$, $\bar{u} = 1m/s$, $\gamma_0 = 20dB$, $q_x = q_y = 0.5$.

9.3 when q_x and q_y are varying. In Fig. 9.7, we have let $\alpha = 0.22$, $\gamma_0 = 20dB$ and $\lambda_m = 1$. Note that when $q_x = q_y = 1$ and $q_x = q_y = 0$, the target location is exactly known. From Fig. 9.7, it can be seen that, when the target location is exactly known, the detection performance for given parameters such as number of mobile nodes λ_m , γ_0 and α , is increased compared to that with random target location. This due to the fact that, when the local nodes perform non-coherent detectors (unknown target location), a higher SNR is required to achieve the same performance level compared to that with a coherent detector (known target location) [105].

Note that the effect of node mobility for the system performance is also dependent on the parameters α , γ_0 and the static node and initial mobile node locations as well. Figure 9.8 shows the detection performance when α is varying while keeping

γ_0 constant. In Fig. 9.8, we consider imperfect communication as well such that the cross over probability of BSC channels is $p = 0.2$. For simplicity, to explore the effect of mobility when α is varying we consider only $\lambda_m = 0$ and $\lambda_m = 1/3$. It seen that when α is increasing, the performance gain achieved by allowing nodes to be mobile compared to stationary network is decreasing. Note that, according to (9.9), the threshold of the N-P detector depends on α and the average energy collected during the corresponding time interval. Specifically, for a constant α , the threshold is increasing if the energy collected during the current time is increasing while for a constant energy, threshold is decreasing if α is increasing. Thus, when mobile nodes are getting closer to the target locations, even if more energy is collected for a higher α value, the detector threshold might be lower. Thus in such situations, the decision quality of mobile nodes will not improve significantly even though they move towards target locations. Therefore, when α is set to a higher value, the node mobility does not much affect the performance improvement of hybrid network. Although figures are not included, it also can be seen that when γ_0 is getting larger, the effect of the node mobility for the system performance is getting less significant compared to a all-static network. This phenomenon is intuitively makes sense, since when γ_0 is increasing, the local decisions at static nodes are made with a higher confidence level. This is further discussed in the next subsection.

9.6.2 Performance evaluation with decision fusion architecture 2

The performance evaluations with the decision fusion architecture 2 is shown in this subsection. We assume same network parameters as in subsection 9.6.1. Figure 9.9 shows the detection probability at the fusion center with the desired delay constraint, when the fraction of mobile nodes is varied for two different γ_0 values. In Fig. 9.9,

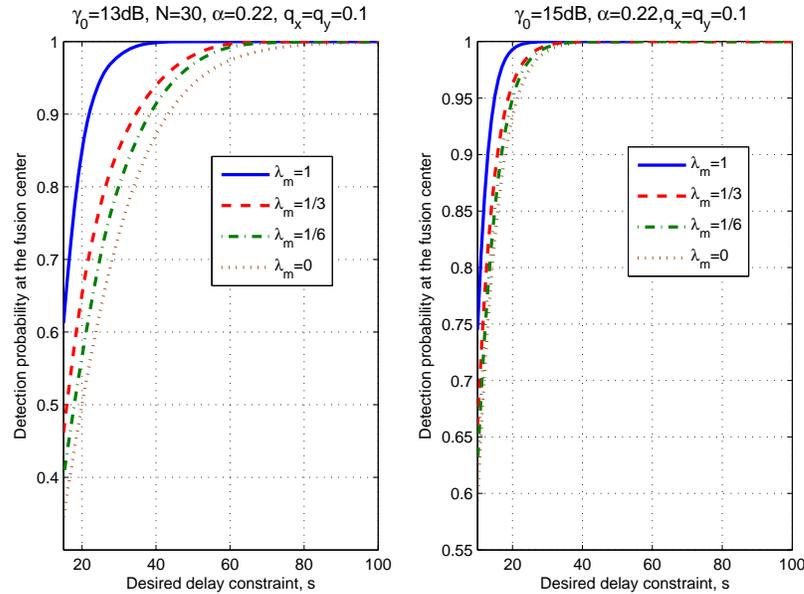


Figure 9.9: Detection performance at the fusion center with decision fusion model 2: $\alpha = 0.22$, $N = 30$, $\bar{u} = 1\text{m/s}$, $p_k = p = 0$

we let $\alpha = 0.22$ and two subplots are corresponding to nominal SNR $\gamma_0 = 13\text{dB}$ and for $\gamma_0 = 15\text{dB}$. Note that $\lambda_m = 0$ represents the network when all nodes are static. It can be seen that, with the decision fusion model 2, where the nodes update decisions over time based on all observations collected up to current time, the effect of node mobility becomes insignificant when the nominal SNR γ_0 is getting larger. This is because, according to the decision fusion model 2, static nodes also collect energy over the time and decisions are getting more accurate as the time goes. When nominal SNR, γ_0 , is larger, static nodes may collect sufficient energy at their stationary locations compared to that is collected by mobile nodes while moving towards possible target locations, since for larger γ_0 , even sensors located far away from the target location will receive signals with considerable strength. However, for lower γ_0 values, (which is more important scenario), it can be seen that with the decision fusion model 2, by allowing a small fraction of nodes to be mobile, relatively

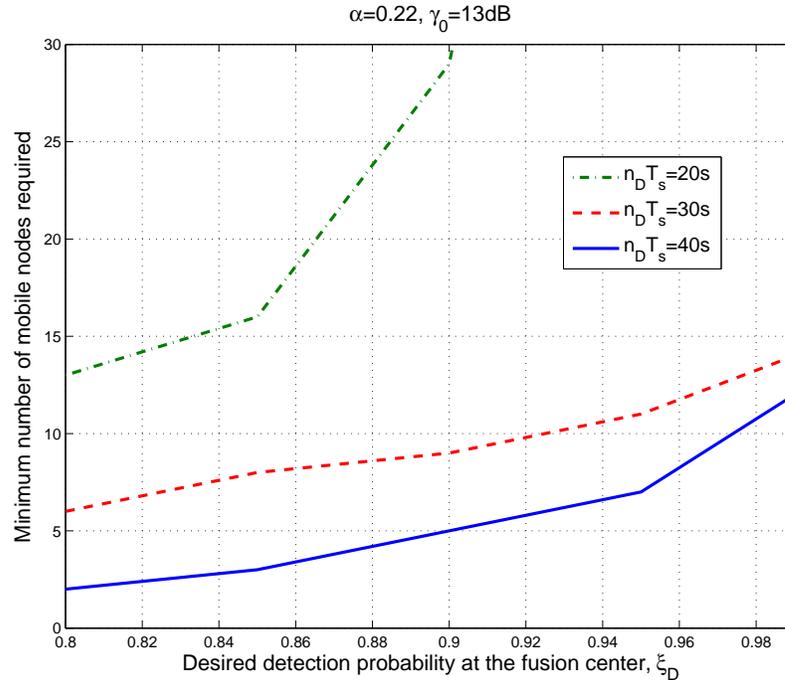


Figure 9.10: Minimum number of mobile nodes required to achieve a desired performance level within a desired delay constraint, for $p = 0$, with decides fusion model 2

a significant performance gain can be achieved compared to that with a all-static network. Also it can be observed that the when α is getting large, the performance gain achieved by allowing nodes to be mobile, is getting smaller.

Although, the network has a maximum number of possible mobile nodes, it might not be required to operate them all depending on the requirement. Figure 9.10 shows the minimum number of mobile nodes (from maximum of 30) that should be directed to move to achieve a desired detection probability at the fusion center during a desired delay constraint, for $\alpha = 0.22$ and $\gamma_0 = 13dB$. Figure 9.10 is corresponding to decision fusion model 2, when the exact target location is known. Without loss of generality, we assume that the target is located at the center point $(0, 0)$. If the desired delay constraint is T_D , we define $n_D = \lfloor \frac{T_D}{T_s} \rfloor$ as before. In Fig. 9.10, the

minimum number of required mobile nodes is shown for $n_D = 20$ and $n_D = 30$ and $n_D = 40$. Figure 9.10 essentially depicts the proper trade-off between the desired detection delay and the minimum number of nodes to be mobile to achieve a desired performance level. In particular, for the target to be detected in a shorter time with higher confidence level, the effect of node mobility is more significant. However, as larger the detection delay required, less significance of adding mobile nodes for the detection.

When comparing the two decision fusion architectures developed in this Chapter, it can be seen that the effect of node mobility has a significant performance improvement over all-stationary network even with a relatively small fraction of mobile nodes with the architecture 1 compared to that with architecture 2. With architecture 2, where nodes update decisions over time, adding mobility improves the system performance significantly only at the lower local nominal SNR range and the performance improvement is also not as significant as with decision fusion architecture 1. The reason is that, with architecture 1, since nodes make local decisions F only with observations collected during the current time interval, the quality of the decisions remains same for static nodes over time while the quality of decisions of mobile nodes is getting much better as time goes, compared to that with static nodes. Thus, adding mobile nodes (even a small number) will enhance the overall performance significantly. This architecture can be regarded as a memoryless architecture at node level, since nodes do not have to use the previous information for their current decision. On the other hand, with decision fusion architecture 2, both nodes make decisions from the observations collected until the current time interval. In this case, eventhough mobile nodes may collect more energy by moving compared to static nodes, static nodes also will be able to make decisions with a higher quality as time goes since their decisions are based on the observations collected during whole time period. Thus for a relatively higher local nominal SNR region, it can be seen that allowing nodes be mobile is fairly useless with the architecture 2. However,

due to energy and computational limitations in sensor nodes, performing decision fusion architecture 2 may have practical limitations since nodes have to use the previous information to update the current decisions. Thus the choice of two decision fusion models may depend on the factors such as node energy and computational constraints, affordable number of mobile nodes and the performance metrics.

9.7 Conclusions

In this Chapter, we proposed two decision fusion models for target detection using a hybrid sensor network in which the node mobility is taken into account at node-level and at the fusion center and analyzed the impact of node mobility to the overall performance under both schemes. The mobile nodes in the network are kept stationary until a target is detected with a low confidence level or statistical information on target locations are available and are directed to move to maximize the detection probability during a desired delay constraint once a target is detected within a certain confidence level. We proposed a node mobility management scheme in order to maximize the detection probability within a desired delay constraint. Since deploying mobile nodes in a sensor network is not as cost effective as deploying static nodes, we evaluate the cost of allowing nodes to be mobile in terms of the minimum number of mobile nodes required to achieve a desired performance level within desired delay constraint.

9.8 Appendix 9A

Regularity conditions to apply L-F central limit theorem in subsection 9.3.3

L-F Central Limit Theorem: Suppose $S_m = X_1 + \dots + X_m$ is a sum of m independent random variables with $\mathbb{E}\{X_k\} = \eta_k$ and $Var\{X_k\} = \nu_k^2$. The L-F CLT states that under certain regularity conditions the sum S_m converges in distribution to a Gaussian random variable with mean $\sum_{j=1}^m \eta_k$ and the variance $\sum_{j=1}^m \nu_k^2$ as $m \rightarrow \infty$ [38]. For the applicability of LF-CLT, it was shown in [103] two sufficient conditions should be satisfied:

- $\nu_k^2 > B_1$
- $\mathbb{E}\{|X_k - \mathbb{E}\{X_k\}|^3\} < B_2$

for $k = 1, \dots, m$ where B_1 and B_2 are positive values.

To apply the LF-CLT in subsection 9.3.3, first we prove that the sufficient conditions are satisfied under H_1 . Note that we can write $\Lambda(nT_s)$ as,

$$\begin{aligned} \Lambda(nT_s) &= \frac{1}{n} \left(\sum_{j=1}^n \sum_{k \in \mathcal{V}_m} w_k(jT_s) + \sum_{j=1}^n \sum_{k \in \mathcal{V}_s} w_k(jT_s) \right) \\ &= \frac{1}{n} \sum_{j,k} X_{j,k}^m + X_{j,k}^s \end{aligned} \quad (9.45)$$

where $X_{j,k}^m = w_k(jT_s)$ for $k \in \mathcal{V}_m$ and $X_{j,k}^s = w_k(jT_s)$ for $k \in \mathcal{V}_s$. Under H_1 , it can be seen from (9.13) that $X_{j,k}^m$ is a Binary random variable with mean $\mu_{d_k}^m(jT_s)$ and variance $\mu_{d_k}^m(jT_s)(1 - \mu_{d_k}^m(jT_s))$. Similarly, $X_{j,k}^s$ is a Binary random variable

with mean $\mu_{d_k}^s(jT_s)$ and the variance $\mu_{d_k}^s(jT_s)(1 - \mu_{d_k}^s(jT_s))$. Then assuming perfect communication channels (such that $p_k = 0$) we have,

$$\begin{aligned} \text{Var}(X_{j,k}^m) &= \mu_{d_k}^m(jT_s)(1 - \mu_{d_k}^m(jT_s)) \\ &= P_{d_k}^m(jT_s)(1 - P_{d_k}^m(jT_s)) \end{aligned} \quad (9.46)$$

and

$$\begin{aligned} \mathbb{E}\{|X_{j,k}^m - \mathbb{E}\{X_{j,k}^m\}|^3\} &= P_{d_k}^m(jT_s)(1 - P_{d_k}^m(jT_s)) \left(P_{d_k}^{m^2}(jT_s) + (1 - P_{d_k}^m(jT_s))^2 \right) \\ &= P_{d_k}^m(jT_s)(1 - P_{d_k}^m(jT_s)) (1 - 2P_{d_k}^m(jT_s)(1 - P_{d_k}^m(jT_s))) \\ &> P_{d_k}^m(jT_s)(1 - P_{d_k}^m(jT_s)) \end{aligned} \quad (9.47)$$

where last inequality results because $1 - 2P_{d_k}^m(jT_s)(1 - P_{d_k}^m(jT_s)) < 1$. Note that from (9.10), if the local false alarm probability α is set such that $0 < \alpha < 1$, $P_{d_k}^m(jT_s)$ is positive and finite for any j, k . Let $B_1^m = \min_{j,k} \{P_{d_k}^m(jT_s)(1 - P_{d_k}^m(jT_s))\}$ and $B_2^m = \max_{j,k} \{P_{d_k}^m(jT_s)(1 - P_{d_k}^m(jT_s))\}$. Then we have $\text{Var}(X_{j,k}^m) > B_1^m$ and $\mathbb{E}\{|X_{j,k}^m - \mathbb{E}\{X_{j,k}^m\}|^3\} < B_2^m$ for $j = 1, \dots, n, k \in \mathcal{V}_m$. Similarly we can show that we can find two positive values B_1^s and B_2^s such that $\text{Var}(X_{j,k}^s) > B_1^s$ and $\mathbb{E}\{|X_{j,k}^s - \mathbb{E}\{X_{j,k}^s\}|^3\} < B_2^s$ for $j = 1, \dots, n, k \in \mathcal{V}_s$. Following a similar procedure, it can be shown that the two regularity conditions are satisfied under H_0 as well.

9.9 Appendix 9B

Proof of theorem 9

When $k = 0$, all mobile nodes are in stationary mode. Then we have,

$$\begin{aligned}
 K_s(n_D, 0) &= (2\rho' + \beta^2) \sum_{j \in \mathcal{V}} P_{d_j}^s(n_D T_s) - \beta^2 \sum_{j \in \mathcal{V}} P_{d_j}^{s,2}(n_D T_s) \\
 &\quad - \left(\sum_{j \in \mathcal{V}} P_{d_j}^s(n_D T_s) \right)^2 - \rho'^2
 \end{aligned} \tag{9.48}$$

If $K_s(n_D, 0) \leq 0$, we have $f_2(k) - f_1(k) \geq K_s(n_D, k)$ for $k = 0$ since $f_2(0) - f_1(0) = 0$ from the definition of $f_1(\cdot)$ and $f_2(\cdot)$. Also from the claim 1 (given below), we can see that then $f_2(k) - f_1(k) \geq K_s(n_D, k)$ for all $k \geq 0$. Then we have $K_0 = 0$, where no need for any node to be mobile to achieve the desired performance level within a desired delay constraint. Now, if $K_s(n_D, 0) > 0$, in the following we prove that, we can find a unique $1 \geq K_0 \leq N_m$ such that $f_2(k) - f_1(k) \geq K_s(n_D, k)$ for $k \geq K_0$ and $f_2(k) - f_1(k) < K_s(n_D, k)$ for $k < K_0$. The uniqueness of such K_0 is followed from claim 1. If $f_2(k) - f_1(k) < K_s(n_D, k)$ for all $k = 1, 2, \dots, N_m$, it implies that the required performance level can not be achieved within the desired delay constraint even if all mobile nodes are directed to move.

To prove the uniqueness of K_0 , we prove the following.

Claim 1. *If $f_2(k) - f_1(k) \geq K_s(n_D, k)$ then $f_2(k+1) - f_1(k+1) \geq K_s(n_D, k+1)$ for $k = 1, \dots, N_m$. Also, if $f_2(k) - f_1(k) \leq K_s(n_D, k)$ we have $f_2(k-1) - f_1(k-1) \leq K_s(n_D, k-1)$.*

Proof. First we prove that if $f_2(k) - f_1(k) \geq K_s(n_D, k)$ then $f_2(k+1) - f_1(k+1) \geq K_s(n_D, k+1)$. Note that, when necessary, we use $P_{d_k}^s(n_D T_s)$ and $P_{d_k}^m(n_D T_s)$ to denote the detection probability at the k -th mobile node at time $n_D T_s$ at its stationary configuration and mobile configuration, respectively. We have,

$$\begin{aligned}
 f_2(k+1) - f_1(k+1) &= \left(\sum_{j=1}^{k+1} P_{d_j}^m(n_D T_s) \right)^2 \\
 &+ \beta^2 \sum_{j=1}^{k+1} P_{d_j}^{m2}(n_D T_s) + 2 \sum_{j=1}^{k+1} P_{d_j}^m(n_D T_s) \sum_{j \in \mathcal{V}_s^k} P_{d_j}^s(n_D T_s) \\
 &- (2\rho' + \beta^2) \sum_{j=1}^{k+1} P_{d_j}^m(n_D T_s) \\
 &= f_2(k) - f_1(k) + (1 + \beta^2) P_{d_{k+1}}^{m2}(n_D T_s) \\
 &+ 2P_{d_{k+1}}^m(n_D T_s) \left(\sum_{j=1}^k P_{d_j}^m(n_D T_s) + \sum_{j \in \mathcal{V}_s^{k+1}} P_{d_j}^s(n_D T_s) \right) \\
 &- 2P_{d_{k+1}}^s(n_D T_s) \sum_{j=1}^k P_{d_j}^m(n_D T_s) - (2\rho' + \beta^2) P_{d_{k+1}}^m(n_D T_s) \tag{9.49}
 \end{aligned}$$

Now adding and subtracting $K_s(n_D, k)$ to the right hand side of (9.49), we will get,

$$\begin{aligned}
 f_2(k+1) - f_1(k+1) &= [f_2(k) - f_1(k) - K_s(n_D, k)] + K_s(n_D, k) + (1 + \beta^2) P_{d_{k+1}}^{m2} \\
 &+ 2P_{d_{k+1}}^m \left(\sum_{j=1}^k P_{d_j}^m + \sum_{j \in \mathcal{V}_s^{k+1}} P_{d_j}^s \right) - 2P_{d_{k+1}}^s \sum_{j=1}^k P_{d_j}^m - (2\rho' + \beta^2) P_{d_{k+1}}^m \tag{9.50}
 \end{aligned}$$

where we dropped argument $n_D T_s$ such that $P_{d_k}^s(n_D T_s) = P_{d_k}^s$ and $P_{d_k}^m(n_D T_s) = P_{d_k}^m$ for simplicity. Substituting for $K_s(n_D, k)$ from (9.39) and using the fact that

$f_2(k) - f_1(k) \geq K_s(n_D, k)$, after simplification (9.50) reduces to (9.51).

$$\begin{aligned} & f_2(k+1) - f_1(k+1) \\ \geq & K_s(n_D, k+1) + (P_{d_{k+1}}^m - P_{d_{k+1}}^s) \left[\beta^2(P_{d_{k+1}}^m + P_{d_{k+1}}^s - 1) + 2 \left(\sum_{j=1}^{k+1} P_{d_j}^m + \sum_{j \in \mathcal{V}_s^k} P_{d_j}^s - \rho' \right) \right] \end{aligned} \quad (9.51)$$

Note that we use $P_{d_k}^s(n_D T_s)$ to denote the detection probability at the k -th mobile node at time $n_D T_s$ at its stationary configuration, as mentioned before. In (9.51) Since mobility towards the target improves the detection probability at the k -th mobile node, we have $P_{d_{k+1}}^m(n_D T_s) - P_{d_{k+1}}^s(n_D T_s) \geq 0$. Using this fact, and with the assumption that $P_{d_k}^m(n_D T_s), P_{d_k}^s(n_D T_s) \geq \frac{1}{2}$ for $k \in \mathcal{V}$ (which holds true in practice for sufficient $n_D T_s$) the second term of the right hand side of the inequality (9.51) is positive. Then we have,

$$f_2(k+1) - f_1(k+1) \geq K_s(n_D, k+1) \quad (9.52)$$

as required. Following a similar approach, we can prove that $f_2(k-1) - f_1(k-1) \leq K_s(n_D, k-1)$ if $f_2(k) - f_1(k) \leq K_s(n_D, k)$. \square

Chapter 10

Particle Filter Based Target

Tracking with Reactive Mobility

10.1 Introduction

The Bayesian approach provides the general framework to solve dynamic state estimation problems, in which the key is to construct the probability density function (pdf) of the underlying state vector based on available observations. When the state dynamics and observation models are linear and Gaussian, the optimal minimum mean squared error (MMSE) estimator is given by the well known Kalman filter [105]. However, in most real world applications, dynamic state estimation problems are non-linear and non-Gaussian. Under the Bayesian approach, for non-linear and non-Gaussian problems, obtaining the optimal solution in closed-form is not tractable. In such cases, several sub-optimal approaches such as extended Kalman

filter [5] and Gaussian sum filter [121] are used to solve the problem with certain approximations. However, these sub-optimal approaches become inefficient when the system is highly non-linear and non-Gaussian. Numerical techniques based on sequential Monte-Carlo methods are proposed as an alternative to solve this problem where the idea is to approximate the required posterior pdfs by discrete random measures. [49] proposed the idea of particle filtering where the required pdf is represented as a set of random sampling. A detailed description of particle filtering can be found in [8, 40].

Use of particle filtering for target tracking was addressed by many authors in the literature with static sensor networks, [21, 35, 41, 54, 100, 101]. For example, in [41] target tracking based on binary observations in a static sensor network was considered, where the tracking is performed at a central fusion center. In [100], tracking algorithm based on particle filters incorporating imperfect communication between sensor nodes and the fusion center is proposed. When the target tracking is performed at a centralized fusion center, each node in the network has to forward its raw or locally processed observations to the central unit, perhaps via long-range communication, which indeed consumes a large transmit power at nodes. Since many practical sensor networks are operated with sensor nodes which have limited battery power, it is desirable that the tracking is performed distributively utilizing the limited resources efficiently. Distributed implementation of particle filters is proposed in some recent work [12, 36, 52, 53, 84, 115, 160]. In [12], the computational burden of a centralized particle filter is divided into distributed stations. In [36], a distributed particle filter (DPF) for a sensor network is proposed when the observation likelihood can be factorized, where each node maintains a separate particle filter. In [115], another approach for distributed particle filter was proposed for multiple target tracking where the DPFs are performed at different uncorrelated sensor cliques which are formed based on different target trajectories. The main feature of this scheme is that the central posterior distribution can be represented as a product of posterior pdfs at

different uncorrelated sensor cliques. The locally estimated posterior distributions (beliefs) based on particle filters are transmitted between different sensor cliques via low dimensional Gaussian mixture model. In [52, 53], a distributed consensus based particle filters are proposed where each distributed node performs PF in parallel at each node. In all above distributed particle filter approaches, all nodes in the network are active at every time and participate in the tracking task. However, all the nodes in the network might not have rich information regarding the target state as the target moves. Thus it is of interest to obtain observations from sensors which have useful information and allow the rest of the nodes to be idle saving energy.

On the other hand, most of the above work on target tracking considered static sensor networks. Target tracking in mobile sensor networks is addressed recently in, [75, 76, 99, 159]. In [99, 159], the tracking task was performed based on Kalman filters assuming linear dynamic models and information driven approaches for mobility management for the mobile nodes are presented. Target tracking with particle filters in a mobile sensor network based on a centralized approach was considered in [75, 76]. Use of hybrid sensor networks consisting of both static and mobile nodes for target tracking is addressed in [68] where the mobile nodes are used to aid the data propagation when the communication ranges of static nodes are limited.

In this Chapter, our focus is on developing a target tracking algorithm based on distributed particle filtering in a hybrid sensor network consisting of both static and mobile nodes. We exploit the node mobility to compensate for the lack of coverage resulted by static nodes dynamically. To the best of our knowledge, a tracking algorithm in a hybrid sensor network exploiting the node mobility dynamically to compensate for the lack of coverage by static nodes, is not addressed in the literature. In the proposed scheme, the network is partitioned into clusters and each time only one cluster is active. The corresponding cluster head performs the tracking task at the given time. Since nodes have to communicate with only their cluster heads,

they do not necessarily need to have long communication ranges. The active cluster head is selected based on the predicted target locations. When the active cluster head is selected, the associate sensor nodes are activated and asked to send their local measurements to the corresponding cluster head. An example illustration of cluster based network for target tracking consisting of both static and mobile nodes is shown in Fig. 10.1 in which cluster heads are equi-spaced. In the proposed tracking algorithm, the main idea is to maintain a certain *coverage level* for the predicted target location at each time step. By *coverage level*, we mean that each predicted target location at time k , is covered by exactly or approximately by a certain number (say β) of sensor nodes. The predicted target location is β covered essentially means that there is at least β number of nodes located within a certain distance (which is a design parameter and is discussed in Section 10.4 in detail) from the predicted target location. The terms *exact* and *approximate* coverages are explained in a later section of the Chapter. If the predicted position at time k is already covered by β -number of static nodes, then mobile nodes are not needed to move during time k to $k + 1$. Otherwise, to maintain the β -coverage (exact or approximate) mobile nodes are moved taking the energy and speed constraints into account.

We consider cases where nodes transmit their raw observations as well as binary decisions to the cluster head to perform the tracking task. In the binary sensor observation model, only 1 bit of information is transmitted to the cluster head by each node representing the present/absent of the target. In the proposed scheme, since nodes communicate only with their cluster heads, the node transmit power is drastically reduced compared to that with centralized approach. On the other hand, only a certain portion of nodes are making measurements at a given time while the rest of the nodes are inactive saving node energy. Thus the proposed tracking algorithm is robust against node as well as cluster head failures. We also derive the Posterior Cramér-Rao lower bounds (PCRLB) for the states and compare the performance of the proposed tracking algorithm with the derived lower bound.

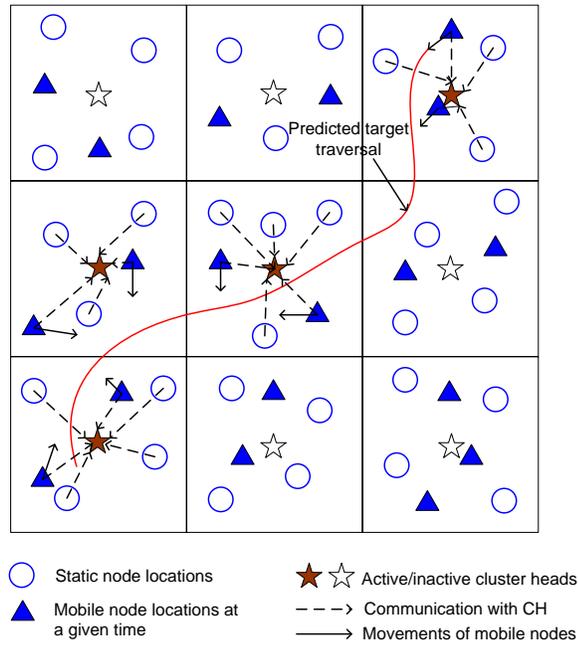


Figure 10.1: Cluster based target tracking with reactive mobility

The Chapter is organized as follows. Section 10.2 presents the system model and problem formulation. In Section 10.3, the cluster based distributed target tracking by particle filtering for raw as well as binary observations, is explained. Proposed node mobility management scheme is discussed in Section 10.4. PCRLB analysis for state estimates for both raw and binary observations is given in Section 10.5. Performance results are shown in 10.6 and concluding remarks are given in 10.7

10.2 Problem Formulation and System Model

10.2.1 Sensor network model

We consider a hybrid sensor network consisting of n_s number of static nodes and n_m number of mobile nodes. Denote $n = n_s + n_m$ to be the total number of nodes. In general we assume that $n_m \ll n_s$, since deploying a large number of mobile nodes is not as cost effective as deploying static nodes. Let \mathcal{V} be the set containing all node indices in the network and \mathcal{V}_m and \mathcal{V}_s to be the sets containing mobile and static node indices, respectively.

10.2.2 State dynamics model

We consider the problem of tracking a single target which is moving in 2-dimensional $X \times Y$ plane. Denote $\mathbf{x}_k = [x_{1k} \ x_{2k} \ \dot{x}_{1k} \ \dot{x}_{2k}]^T$ to be the target state vector at time k where first two elements represent the target position and the latter two elements of \mathbf{x}_k represent the speed of the target in X and Y directions, respectively. We assume following linear dynamic model for the target state:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \boldsymbol{\nu}_k, \quad (10.1)$$

for $k = 1, 2, \dots$, with the initial known distribution, $p(\mathbf{x}_0)$ for \mathbf{x}_0 where \mathbf{F}_k is a 4×4 matrix that models the state kinematics and is defined as, [29]

$$\mathbf{F}_k = \begin{pmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where T_s the time difference between two consecutive measurements (or sampling period). The noise vector $\boldsymbol{\nu}_k$ is assumed to be zero mean Gaussian with covariance matrix Q where Q is given by [29],

$$Q = \sigma_\nu^2 \begin{pmatrix} \frac{T_s^3}{3} & 0 & \frac{T_s^2}{2} & 0 \\ 0 & \frac{T_s^3}{3} & 0 & \frac{T_s^2}{2} \\ \frac{T_s^2}{2} & 0 & T_s & 0 \\ 0 & \frac{T_s^2}{2} & 0 & T_s \end{pmatrix},$$

which models the acceleration terms in X and Y directions. σ_ν^2 is a scalar which controls the intensity of the process noise. We assume that the dynamic model (10.1) performs a Markov transition and is represented by the conditional transition probability density, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. The above state dynamical model is justifiable in modeling targets' acceleration components using random noise and targets move with a constant velocity [29, 101].

10.2.3 Observation model

We assume that the signal emitted by the target is attenuated with the distance from the target according to the following model:

$$\begin{aligned} z_{j,k} &= h_{j,k}(\mathbf{x}_k) + v_{j,k}, \\ &= \frac{A_0}{|\mathbf{r}_{j,k} - \tilde{\mathbf{x}}_k|^{\alpha/2}} + v_{j,k}, \text{ for } j \in \mathcal{V} \end{aligned} \quad (10.2)$$

where $z_{j,k}$ is the j -th node's observation at time k , $\tilde{\mathbf{x}}_k = [\mathbf{x}_k(1) \ \mathbf{x}_k(2)]^T = [x_{1k} \ x_{2k}]^T$, $\mathbf{r}_{j,k} = (r_{j,k}, s_{j,k})$ is the position of the j -th node at time k , A_0 is the amplitude of the signal emitted by the target and $v_{j,k}$ is the observation noise which is assumed to be zero mean Gaussian with variance σ_v^2 and independent across sensor nodes, and

α is the path-loss attenuation index, which is assumed to be 2 throughout. Note that for static node locations, we use $\mathbf{r}_{j,k} = \mathbf{r}_j = (r_j, s_j)$ by dropping the time index since static node locations do not change over time. The observation model 10.2 is justifiable for targets in which the emitted signals follow the power attenuation model that inversely proportional to the distance (e.g. acoustic sources) [41, 100, 114].

10.2.4 Node mobility model

Initially, mobile nodes are assumed to be stationary and are directed to move to improve the tracking performance when necessary. When directed at time k , mobile node j moves with a speed of $u_{j,k} \in [u_{min}, u_{max}]$ in a direction $\theta_{j,k} \in [0, 2\pi)$. Let $(r_{j,k}, s_{j,k})$ denotes the location of the j -th mobile node at time k , as defined before. Then the location $(r_{j,k+1}, s_{j,k+1})$ of the j -th mobile node at time $k + 1$ is given by

$$\begin{aligned} r_{j,k+1} &= r_{j,k} + T_s u_{j,k} \cos \theta_{j,k} \\ s_{j,k+1} &= s_{j,k} + T_s u_{j,k} \sin \theta_{j,k} \end{aligned}$$

for $j \in \mathcal{V}_m$.

10.3 Cluster Based Target Tracking by Particle Filters

10.3.1 Centralized particle filter

With raw observations

In the centralized approach, we assume that each node j sends its raw observations given by (10.2) to a central fusion center where the fusion center performs the tracking algorithm based on the observations obtained from sensor nodes. We assume that the observations are sent to the fusion center over AWGN channel. The received signal at the fusion center from the j -th node at time k is,

$$y_{j,k} = z_{j,k} + \epsilon_{j,k}, \text{ for } j \in \mathcal{V}, \quad (10.3)$$

where $\epsilon_{j,k}$ is the received noise which is assumed to be Gaussian with mean zero and the variance σ_ϵ^2 .

With binary observations

Since sensor nodes are equipped with limited battery power and limited computation and communication capabilities, it is desired that raw observations are processed locally to produce a summary of the observations and transmit. In the following we consider the case where each local node makes a binary decision on target present/absent at each time k , as given below and transmits only 1 bit information to the fusion center if the target is present.

We assume that each node j makes a binary decision on the target present/absent at each time k based on its own observation $z_{j,k}$ according to the following rule:

$$\delta_{j,k} = \begin{cases} 1 & \text{if } z_{j,k} \geq \tau_{j,k} \\ 0 & \text{if } z_{j,k} < \tau_{j,k} \end{cases}, \text{ for } j \in \mathcal{V}. \quad (10.4)$$

where $\tau_{j,k}$ is the detection threshold at the j -th node at time k . If a constant threshold test is performed at each node at each time, we set $\tau_{j,k} = \tau$ for all $j \in \mathcal{V}$ and for all k . The received signal at the fusion center from the j -th node at time k is,

$$y_{j,k} = \delta_{j,k} + \epsilon_{j,k}, \text{ for } j \in \mathcal{V}. \quad (10.5)$$

where $\epsilon_{j,k}$ is the received noise which is assumed to be Gaussian with mean zero and the variance σ_ϵ^2 , as before.

Now the problem is to estimate the state vector \mathbf{x}_k based on the observation vector $\mathbf{y}_{1:k}$ up to time k , where $\mathbf{y}_{1:k} = [y_{1,1:k}, y_{2,1:k}, \dots, y_{n,1:k}]$.

According to Bayes formulation, to estimate the target state at time k , it is required to construct the posterior distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ with the initial pdf $p(\mathbf{x}_0|\mathbf{y}_0) = p(\mathbf{x}_0)$. The pdf $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ is constructed in two stages: prediction and update.

prediction

If the pdf $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ is available at time $k-1$, the predicted pdf $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ is obtained as [8]

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1}. \quad (10.6)$$

update

When the observations \mathbf{y}_k are available at time k , the required posterior distribution to estimate \mathbf{x}_k is updated as [8],

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})}. \quad (10.7)$$

If the state dynamic model and the observation model are linear and Gaussian, an analytical solution for the optimal MMSE estimate $\hat{\mathbf{x}}_{k|k}$ for the state vector can be obtained based on well known Kalman filter. However, although the state dynamic model (10.1) is linear and Gaussian, the observation model used in this Chapter is non-linear. In such situations, there is no analytically tractable solution for optimal estimator based on (10.6) and (10.7). Thus we use a sequential Monte Carlo approach to approximate the posterior pdf (10.7) with particle filters [8].

Particle filters for centralized target tracking

Denote $\mathcal{X}_k = \{\mathbf{x}_k^i, w_k^i\}_{i=1}^S$ to be the random measure that characterizes the posterior pdf $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ where S is the number of samples. Then $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ can be approximated as [8],

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \sum_{i=1}^S w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i),$$

where $\delta(\cdot)$ here is the Dirac delta function. Then the target estimate at time k , (which is the mean of the posterior pdf $p(\mathbf{x}_k|\mathbf{y}_{1:k})$) can be obtained by the sum,

$$\hat{\mathbf{x}}_{k|k} \approx \sum_{i=1}^S w_k^i \mathbf{x}_k^i$$

and the covariance matrix $U_{k|k}$ of the estimate is,

$$U_{k|k} = \sum_{i=1}^S w_k^i (\mathbf{x}_k^i - \hat{\mathbf{x}}_{k|k}) (\mathbf{x}_k^i - \hat{\mathbf{x}}_{k|k})^T.$$

The predicted state $\hat{\mathbf{x}}_{k+1|k}$ and the covariance matrix $U_{k+1|k}$ can be obtained by the state dynamic model in (10.1):

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k} &= \mathbf{F} \hat{\mathbf{x}}_{k|k} \\ U_{k+1|k} &= \mathbf{F} U_{k|k} \mathbf{F}^T + Q. \end{aligned}$$

10.3.2 Distributed cluster based target tracking for hybrid sensor network

As well known, the problem of the centralized tracking algorithm is that the reliability of the network fusion depends on the reliability of the central unit. Also since all nodes in the network have to forward their local measurements to the central unit, a large communication power is needed for nodes.

In the following we propose a cluster-based tracking algorithm for mobile target tracking in the hybrid sensor network. In the proposed algorithm, we assume that there are few static nodes with high processing capabilities which act as cluster heads. The cluster head formation can be performed at the deployment stage, for example, based on Voronoi partitions. Note that there are both static and mobile nodes in the network. Thus, when a cluster is formed at the deployment stage, there is a fixed number of static nodes belong to each cluster. Depending on the mobile nodes' mobility, the number of mobile nodes belonging to a particular cluster may change over time. Thus, for a given time there is a variable number of total nodes for a

given cluster head. In the proposed tracking algorithm, we assume that each cluster head keeps track on the mobile nodes entering and leaving the corresponding cluster at each time.

Let C be the total number of clusters (or cluster heads) in the network and $n_{c,k}$ be the number of total nodes belongs to the cluster c at time k , for $c = 1, \dots, C$. In the proposed tracking algorithm, we assume that only one cluster head performs tracking on the moving object at a given time. The active cluster head is selected as the closest one to the predicted target position at each time. In the cluster based approach, each node belonging to the cluster c , sends its observation to the cluster head of the c -th cluster, CH_c for $c = 1, 2, \dots, C$. Since nodes have to communicate only with their cluster heads, a significant reduction of transmit power can be achieved compared to the centralized approach. On the other hand, the cluster head performing the tracking task is changing over time according to the predicted target locations. Thus one cluster head does not have to be active all the time. The proposed cluster based approach is summarized in the following with the assumptions:

- There is a total of C clusters (or cluster heads).
- Each cluster has a fixed number of static nodes and a varying number of mobile nodes at a given time.
- Each cluster head is assumed to have the knowledge of locations of static nodes and the mobile nodes belong to itself at a given time.
- Only the nodes in the active cluster makes observations on the target state, while other nodes in the network are in inactive mode.
- At each time k , the active cluster head performs particle filtering and makes an estimate of the underlying state.

- Once the target state is estimated at time k , the next active cluster head is selected based on the predicted target position. The current cluster head talks to its neighbors to determine the next cluster head. If the next cluster head is different from the current cluster head, current CH transmits the necessary information to the next CH, to perform the tracking task.
- Cluster heads can communicate with neighboring cluster heads, but slave nodes in a particular cluster communicate only with the cluster head.

Denote $\mathbf{y}_{c,1:k} = [y_{1,1:k}, y_{2,1:k}, \dots, y_{n_{c,k},1:k}]$ be the observation vector at CH_c up to time k . We propose to use *Sampling importance resampling* (SIR) particle filter [8] at the active cluster head, due to its implementation convenience. According to SIR filter, the particles are generated from the state transition probability:

$$\mathbf{x}_k^i \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}^i) \quad (10.8)$$

and the corresponding weights are updated according to

$$w_k^i \propto p(\mathbf{y}_{c,k} | \mathbf{x}_k^i) \quad (10.9)$$

where $\mathbf{y}_{c,k}$ is the observation vector at time k at CH_c .

With raw observations

With raw observations, from (10.2) and (10.3), the conditional pdf $p(\mathbf{y}_{c,k} | \mathbf{x}_k)$ for the c -th cluster head is given by,

$$p(\mathbf{y}_{c,k} | \mathbf{x}_k) = \mathcal{N}(\tilde{\mathbf{a}}_{c,k}, \tilde{R}_{c,k}) \quad (10.10)$$

where $\tilde{\mathbf{a}}_{c,k} = [a_{1,k}, a_{2,k}, \dots, a_{n_{c,k},k}]^T$ is a $n_{c,k}$ -length vector with,

$$a_{j,k} = \frac{A_0}{|\mathbf{r}_{j,k} - \tilde{\mathbf{x}}_k|^{\alpha/2}}, \text{ for } j = 1, 2, \dots, n_{c,k}, \quad (10.11)$$

and $\tilde{R}_{c,k} = (\sigma_v^2 + \sigma_\epsilon^2)\mathbf{I}_{c,k}$ where $\mathbf{I}_{c,k}$ is the $n_{c,k} \times n_{c,k}$ identity matrix.

Then the weight update of the SIR filter at CH_c at time k can be performed as

$$w_{c,k}^i \propto p(\mathbf{y}_{c,k} | \mathbf{x}_k^i) = \mathcal{N}(\tilde{\mathbf{a}}_{c,k}(\mathbf{x}_k^i), \tilde{R}_{c,k}).$$

With binary observations

With binary observations, the conditional observation likelihood function is given by,

$$p(\mathbf{y}_{c,k} | \mathbf{x}_k) = \prod_{j=1}^{n_{c,k}} \left(p_{j,k}^\delta \mathcal{N}(y_{j,k}; 1, \sigma_\epsilon^2) + (1 - p_{j,k}^\delta) \mathcal{N}(y_{j,k}; 0, \sigma_\epsilon^2) \right) \quad (10.12)$$

where $p_{j,k}^\delta(\mathbf{x}_k) = Q\left(\frac{\tau_{j,k} - a_{j,k}(\mathbf{x}_k)}{\sigma_v}\right)$ and we use $\mathcal{N}(x; \mu, \sigma^2)$ to denote that x is distributed as Gaussian with mean μ and variance σ^2 . Then the weight updating of the SIR particle filter with binary observations at cluster head CH_c at time k is given by,

$$w_{c,k}^i \propto \prod_{j=1}^n \left(p_{j,k}^\delta \mathcal{N}(y_{j,k}; 1, \sigma_\epsilon^2) + (1 - p_{j,k}^\delta) \mathcal{N}(y_{j,k}; 0, \sigma_\epsilon^2) \right)$$

Note that we drop x in the term $\mathcal{N}(x; \mu, \sigma^2)$ in the rest of the Chapter for clarity, when there is no ambiguity. Denote $\hat{\mathbf{x}}_{c,k|k}$ and $U_{c,k|k}$ be the state estimator and the error covariance matrix at the c -th cluster head at time k . Then the estimator and

the error covariance matrix at the cluster head CH_c are given by,

$$\hat{\mathbf{x}}_{c,k|k} \approx \sum_{i=1}^S w_{c,k}^i \mathbf{x}_k^i$$

and

$$U_{c,k|k} \approx \sum_{i=1}^S w_{c,k}^i (\mathbf{x}_k^i - \hat{\mathbf{x}}_{c,k|k})(\mathbf{x}_k^i - \hat{\mathbf{x}}_{c,k|k})^T,$$

respectively. The predicted estimator and the error covariance matrices are given by,

$$\begin{aligned} \hat{\mathbf{x}}_{c,k+1|k} &= \mathbf{F} \hat{\mathbf{x}}_{c,k|k} \\ U_{c,k+1|k} &= \mathbf{F} U_{c,k|k} \mathbf{F}^T + Q. \end{aligned}$$

Once the estimator is computed at time k at CH_c , we assume that the cluster head that should perform the tracking task at time $k+1$ is selected based on the predicted position estimate, $\hat{\mathbf{x}}_{c,k+1|k}$. Denote \mathbf{r}_c be the location of the c -th cluster head for $c = 1, \dots, C$. Then, the cluster head that should perform the tracking task at time $k+1$ is selected as,

$$\hat{CH} = \underset{CH_i, i \in \mathcal{N}_c}{\operatorname{argmin}} |\hat{\mathbf{x}}_{c,k+1|k} - \mathbf{r}_i|, \quad (10.13)$$

where \mathcal{N}_c is the indices of the neighboring cluster heads of the cluster head CH_c including itself. Note that, by selecting the next CH as in (10.13), that is the one which is closest the predicted target position at time k , the next cluster head is able to collect a set of rich observations on the target state. Let $\hat{CH} = CH_d$. If the selected CH , $CH_d \neq CH_c$, then, CH_c transmits its estimator to CH_d . Then for the particle filtering, CH_d samples particles from $p(\mathbf{x}_{k+1} | \mathbf{x}_k = \hat{\mathbf{x}}_{c,k|k})$ and weight updating is performed based on the observation likelihood at CH_d at time $k+1$. Further, for the

discussion in this dissertation, we assume that communication between cluster heads is via perfect channels. This assumption is justifiable, since we assume that there are few cluster heads in the networks and they have sufficient power and processing capabilities such that two neighboring cluster heads can communicate each other with a sufficient transmit power such that the messages transmitted by one cluster head can be reconstructed with a negligible error at the receiving cluster head.

10.4 Node mobility management

In this Section we discuss the proposed node mobility scheme for the target tracking in the hybrid sensor network for dynamic coverage improvement as required. The idea is to maintain a continuous β -coverage (exact or approximate) on the predicted trace of the moving target's trajectory at each time k . We call a point \mathbf{r}_0 is β -covered if there is at least β -number of sensors located within the disk denoted by $D(\mathbf{r}_0, r_D)$ centered at \mathbf{r}_0 with a radius of r_D where r_D is a design parameter.

When the target state is estimated at time k , the idea is to maintain a β -coverage for the next target location at time $k + 1$ such that it can be tracked with high accuracy. Note that with the assumed observation model (10.2), the signal strength emitted by the target decays as the distance from the target is increasing. Thus the sensor nodes located closer to the target position at a given time make rich observations while those located far away from the current target location make poor observations. To better track the target at every time step, it is necessary to maintain a certain number of nodes very close to the target location at each time such that they receive rich observations. Let r_D be the distance from the target where we call the nodes within this distance receive rich observations. r_D can be selected such that the signal strength received at any node located within the disk

$D(\mathbf{r}_0, r_D)$ from a target when the target is located at \mathbf{r}_0 , exceeds a certain threshold. One straight-forward way of maintaining a β -coverage with a static network is to deploy a large number of nodes in the desired region such that each point in the region is covered by β -number of nodes. However, when it is necessary to cover a large region, to achieve this a large number of static nodes may be required. On the other hand, after the initial deployment if the nodes become inactive (due to node power failure or node breakage, etc..) the required β -coverage at each point cannot be achieved. Thus it is of interest to allow a certain number of mobile nodes with static nodes such that the lack of coverage results in a static network is compensated by the node mobility when necessary, taking their speed and energy limitation into account.

When a predicted target location at a given time is covered by at least β -number of static nodes we say *exact* β -coverage is achieved at the predicted target location. If the required position is not covered by β -number of static nodes, the required number of mobile nodes are directed to move the minimum distance needed to provide a β -coverage. At time k , if the predicted target location is $\hat{\mathbf{x}}_{c,k+1|k}$, if all the required number of mobile nodes can reach the desired destination to provide β -coverage (i.e. they can move such that after T_s time they can be within $D(\hat{\mathbf{x}}_{c,k+1|k}, r_D)$), we call the position $\hat{\mathbf{x}}_{c,k+1|k}$ achieves *exact* β -coverage. At sometimes, due to speed limitations of mobile nodes, some mobile nodes may not be able to reach the disk $D(\hat{\mathbf{x}}_{c,k+1|k}, r_D)$ even if they move with their maximum speed to provide β -coverage. However, in this case, since they move the maximum distance they can move towards the predicted target location so that the signal strength received at their destination is higher than that of the original location. If this happens, we call an *approximate* β -coverage is achieved at the predicted target location.

The basic idea of the proposed mobility model is illustrated in Fig. 10.2. In Fig. 10.2, the predicted target location at time k is denoted by $\mathbf{P}_{k+1|k}$ for clarity.

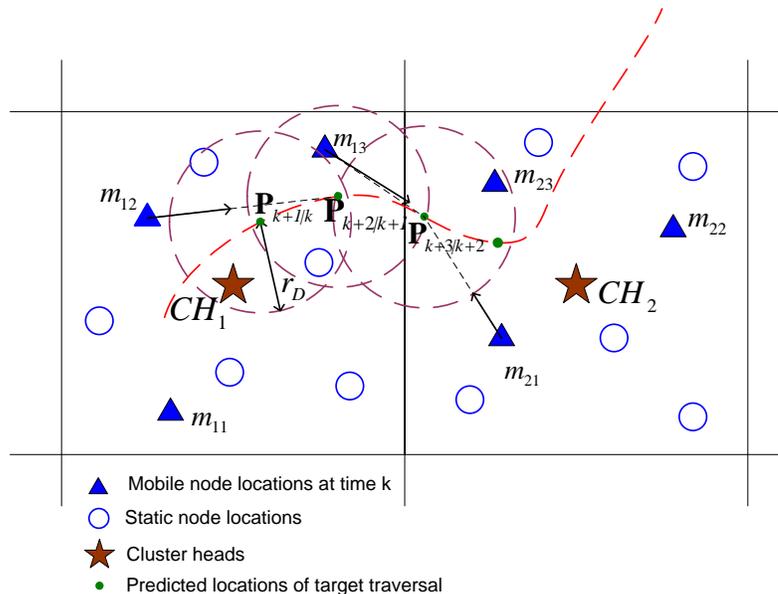


Figure 10.2: Illustration of providing $\beta = 3$ coverage at predicted target locations

Assume that the mobile node locations of two clusters CH_1 and CH_2 at time k are represented by solid triangles. Suppose we need to achieve $\beta = 3$ coverage (*exact* or *approximate*) at each predicted target location. According to static and mobile node locations, at time k , it can be seen from Fig. 10.2 that the disk $D(\mathbf{P}_{k+1|k}, r_D)$ has three nodes (including cluster heads). Thus no mobile node needs to be moved from time k to $k + 1$ to cover the predicted location $\mathbf{P}_{k+1|k}$ to achieve the desired coverage level. At time $k + 1$, it can be seen that the disk $D(\mathbf{P}_{k+2|k+1}, r_D)$ has only two nodes (one mobile and one static). Then to provide $\beta = 3$ coverage at $\mathbf{P}_{k+2|k+1}$, the cluster head CH_1 selects 1 more mobile node to move the desired distance towards $\mathbf{P}_{k+2|k+1}$. According to Fig. 10.2, mobile node m_{12} is more likely to be selected to move towards $\mathbf{P}_{k+2|k+1}$ during time $k + 1$ to $k + 2$. At time $k + 2$, it can be seen that the predicted target position $\mathbf{P}_{k+3|k+2}$ belongs to second cluster (with cluster head CH_2) and the disk $D(\mathbf{P}_{k+3|k+2}, r_D)$ has only one (mobile) node. Thus two more

mobile nodes are required to be moved towards $D(\mathbf{P}_{k+3|k+2}, r_D)$. Now, if the cluster head CH_2 has the knowledge of mobile node locations only inside its own cluster, it may direct mobile nodes m_{21} and m_{23} towards $\mathbf{P}_{k+3|k+2}$ in time duration $k + 2$ to $k + 3$. However, if CH_2 has the knowledge of mobile node locations of neighboring clusters it is more likely that CH_2 selects node m_{13} to move towards $\mathbf{P}_{k+3|k+2}$ instead of m_{22} since m_{13} is more likely to provide exact coverage by moving a less distance compared to m_{22} . Note that when the predicted target location is closer to the boundaries of the current cluster, there might be situations, as observed just now, that it might be more efficient to move mobile nodes located closer to predicted target location in neighboring clusters rather than moving mobile nodes located far away from the predicted target location in its own cluster. However, to get the information about mobile node locations in neighboring clusters, it might need to have communication between neighboring clusters, which will result an additional cost. Thus there is a trade-off between this additional cost and the performance gain achieved by using mobile nodes in neighboring clusters in such situations. In this Chapter we limit our discussion to the case where a cluster head has the knowledge of mobile nodes only in its own cluster. However, we assume that the current cluster head communicates with neighboring cluster heads to get location information of near-by of mobile nodes to the predicted target position (outside its own cluster) only if the current cluster head does not have sufficient number of mobile nodes to provide the required β -coverage.

The proposed node mobility management scheme has 4 basic steps:

- The active cluster head checks the number of static nodes $n_{s,D}$ within the disk $D(\hat{\mathbf{x}}_{c,k+1|k}, r_D)$. If $n_{s,D} \geq \beta$, mobile nodes in the corresponding cluster remain stationary.
- If $n_{s,D} < \beta$, the difference $(\beta - n_{s,D})$ is determined.

- Determine $(\beta - n_{s,D})$ number of mobile nodes which should be directed to move. If the current cluster does not have $(\beta - n_{s,D})$ number of mobile nodes, the cluster head communicates with neighboring clusters (which are located close to the predicted target location) to determine the required number of mobile nodes needed from neighboring clusters.
- Determine the speed and direction of these selected mobile nodes.

Let CH_d be the candidate cluster head for time $k + 1$ which is selected at time k and CH_c be the active cluster head which performs the tracking task at time k . Once CH_c determines the predicted location $\hat{\mathbf{x}}_{c,k+1|k}$, it is transmitted to CH_d (if it is different from CH_c). Note that if $\hat{\mathbf{x}}_{c,k+1|k} \in CH_c$, then $CH_d = CH_c$. The cluster head CH_d is responsible for managing mobile node mobility to maintain β -coverage for the predicted location, before start making measurements. Let $n_{m,d}^k$ be the total number of mobile nodes belonging to CH_d at time k and $\mathcal{V}_{m,d}^k$ be the set containing corresponding mobile node indices. If $n_{s,D} < \beta$, the cluster head CH_d selects $(\beta - n_{s,D})$ number of mobile nodes which are closest to $\hat{\mathbf{x}}_{c,k+1|k}$. If $(\beta - n_{s,D}) > n_{m,d}^k$, CH_d selects closest $n_{m,d}^k - (\beta - n_{s,D})$ number of mobile nodes from neighboring clusters (located closer to the predicted target location) by communicating locally with neighboring cluster heads. Denote $\tilde{\mathcal{V}}_{m,d}^k$ be the set containing indices of $(\beta - n_{s,D})$ number of mobile nodes which are closest to $\hat{\mathbf{x}}_{c,k+1|k}$. Note that $\tilde{\mathcal{V}}_{m,d}^k \subseteq \mathcal{V}_{m,d}^k$ only if $n_{m,d}^k \geq (\beta - n_{s,D})$.

According to the assumption, a mobile node j can move in a direction $\theta_{j,k} \in [0, 2\pi)$ with a speed of $u_{j,k} \in [u_{min}, u_{max}]$ from time k to $k + 1$. Now the objective is to determine the optimal $\theta_{j,k}$ and $u_{j,k}$ for $j \in \tilde{\mathcal{V}}_{m,d}^k$ such that corresponding mobile nodes move the minimum distance to provide a β -coverage for $\hat{\mathbf{x}}_{c,k+1|k}$ at time $k + 1$.

The optimal $\theta_{j,k}$ and $u_{j,k}$ for the j -th mobile node for $j \in \tilde{\mathcal{V}}_{m,d}^k$ are given by the following theorem:

Theorem 10. *The optimal speed and the direction for the j -th mobile node at time k is given as follows. If $(|\mathbf{r}_{j,k} - \hat{\mathbf{x}}_{c,k+1|k}| - r_D) > 0$ and $(|\mathbf{r}_{j,k} - \hat{\mathbf{x}}_{c,k+1|k}| - r_D) \leq u_{max}T_s$,*

$$\hat{u}_{j,k} = \max \left\{ \frac{1}{T_s} |\mathbf{r}_{j,k} - \hat{\mathbf{x}}_{c,k+1|k}|, u_{min} \right\} \quad (10.14)$$

and

$$\hat{\theta}_{j,k} = \text{atan2} \left(\frac{s_{j,k} - \hat{x}_{c,2(k+1|k)}}{r_{j,k} - \hat{x}_{c,1(k+1|k)}} \right) \quad (10.15)$$

If $(|\mathbf{r}_{j,k} - \hat{\mathbf{x}}_{c,k+1|k}| - r_D) > 0$ and $(|\mathbf{r}_{j,k} - \hat{\mathbf{x}}_{c,k+1|k}| - r_D) > u_{max}T_s$,

$$\hat{u}_{j,k} = u_{max} \quad (10.16)$$

and

$$\hat{\theta}_{j,k} = \text{atan2} \left(\frac{s_{j,k} - \hat{x}_{c,2(k+1|k)}}{r_{j,k} - \hat{x}_{c,1(k+1|k)}} \right) \quad (10.17)$$

If $(|\mathbf{r}_{j,k} - \hat{\mathbf{x}}_{c,k+1|k}| - r_D) < 0$,

$$\hat{u}_{j,k} = 0 \quad (10.18)$$

where $\text{atan2}(x)$ is the four quadrant inverse tangent of x , and $\hat{x}_{c,1(k+1|k)}$ and $\hat{x}_{c,2(k+1|k)}$ are the X and Y coordinates of the predicted target location $\hat{\mathbf{x}}_{c,k+1|k}$.

Proof. See Appendix 10A. □

The pseudocode for the proposed node mobility algorithm at time k is given in Algorithm 5.

Note that in the proposed tracking algorithm, the current cluster head decides how many nodes to move and where to move prior to making measurements, based on the node's location information and the predicted target position. Since the node mobility algorithm does not depend on the measurements (or statistics) residing at individual nodes, the algorithm can be implemented with minimal communication costs between mobile nodes and the cluster head. Also, it should be noted that the mobile node locations can be updated at cluster heads timely manner based on the mobility management algorithm, thus nodes do not necessarily communicate to cluster heads to inform their current locations.

10.5 PCRLB Analysis

The Cramér-Rao bound provides a lower limit on the mean squared estimation error for non-random parameter estimation [105]. Analogous to this CRB, posterior CRLB provides a lower bound for the mean squared error of random parameter estimation [126]. For non-linear systems, a recursive formulation for computing PCRLB is given in [126].

10.5.1 With raw observations

In this section we analyze the PCRLB for the derived estimator based on raw observations. The mean squared error matrix $U_{k|k}$ at time k is lower bounded by the

PCRLB, such that, [42]

$$U_{k|k} \geq J_k^{-1}$$

where J_k is the $d_x \times d_x$ Fisher-information matrix. According to [126], J_k can be computed recursively via,

$$J_{k+1} = D_k^{22} - D_k^{21} (J_k + D_k^{11})^{-1} D_k^{12}, \quad (10.19)$$

where

$$\begin{aligned} D_k^{11} &= \mathbb{E}\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\} \\ D_k^{12} &= \mathbb{E}\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_{k+1}} \log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\} \\ D_k^{21} &= [D_k^{12}]^T \\ D_k^{22} &= \mathbb{E}\{-\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} \log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\} \\ &\quad + \mathbb{E}\{-\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} \log p(\mathbf{y}_{k+1}|\mathbf{x}_k)\} \end{aligned}$$

where $\Delta_{\Phi}^{\Theta} = \nabla_{\Phi} \nabla_{\Theta}^T$ and ∇ is the Laplacian operator. The initial condition for the recursion (10.19) is,

$$J_0 = \mathbb{E}\{-\Delta_{\mathbf{x}_0}^{\mathbf{x}_0} \log p(\mathbf{x}_0)\}$$

If $p(\mathbf{x}_0) \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$, J_0 becomes,

$$J_0 = \boldsymbol{\Sigma}_0^{-1}.$$

Note that the terms D_k^{11} , D_k^{12} and D_k^{22} do not depend on the observation model, but only on the state dynamic model. According to the state dynamic model in

$$\begin{aligned}
 \left(\tilde{D}_k^{22}\right)_{11} &= \frac{1}{(\sigma_v^2 + \sigma_\epsilon^2)} \mathbb{E}_{x_{1k+1}, x_{2k+1}} \left\{ \sum_{j=1}^{n_{c,k}} \frac{A_0^2 (r_{j,k+1} - x_{1k+1})^2}{((r_{j,k+1} - x_{1k+1})^2 + (s_{j,k+1} - x_{2k+1})^2)^3} \right\} \\
 \left(\tilde{D}_k^{22}\right)_{12} &= \left(\tilde{D}_k^{22}\right)_{21} = \frac{1}{(\sigma_v^2 + \sigma_\epsilon^2)} \mathbb{E}_{x_{1k+1}, x_{2k+1}} \left\{ \sum_{j=1}^{n_{c,k}} \frac{A_0^2 (r_{j,k+1} - x_{1k+1})(s_{j,k+1} - x_{2k+1})}{((r_{j,k+1} - x_{1k+1})^2 + (s_{j,k+1} - x_{2k+1})^2)^3} \right\} \\
 \left(\tilde{D}_k^{22}\right)_{13} &= \left(\tilde{D}_k^{22}\right)_{14} = 0 \\
 \left(\tilde{D}_k^{22}\right)_{22} &= \frac{1}{(\sigma_v^2 + \sigma_\epsilon^2)} \mathbb{E}_{x_{1k+1}, x_{2k+1}} \left\{ \sum_{j=1}^{n_{c,k}} \frac{A_0^2 (s_{j,k+1} - x_{2k+1})^2}{((r_{j,k+1} - x_{1k+1})^2 + (s_{j,k+1} - x_{2k+1})^2)^3} \right\} \\
 \left(\tilde{D}_k^{22}\right)_{23} &= \left(\tilde{D}_k^{22}\right)_{24} = 0 \\
 \left(\tilde{D}_k^{22}\right)_{31} &= \left(\tilde{D}_k^{22}\right)_{32} = \left(\tilde{D}_k^{22}\right)_{33} = \left(\tilde{D}_k^{22}\right)_{34} = 0 \\
 \left(\tilde{D}_k^{22}\right)_{41} &= \left(\tilde{D}_k^{22}\right)_{42} = \left(\tilde{D}_k^{22}\right)_{43} = \left(\tilde{D}_k^{22}\right)_{44} = 0
 \end{aligned} \tag{10.20}$$

(10.1), we can determine them as,

$$\begin{aligned}
 D_k^{11} &= \mathbf{F}^T Q^{-1} \mathbf{F} \\
 D_k^{12} &= [D_k^{21}]^T = -\mathbf{F}^T Q^{-1}.
 \end{aligned}$$

With raw observations, the observation likelihood function at CH_c at time k is given by (10.10). Then D_k^{22} is given by [126],

$$D_k^{22} = Q^{-1} + \tilde{D}_k^{22}$$

where $\tilde{D}_k^{22} = \mathbb{E}\{[\nabla_{\mathbf{x}_{k+1}} \tilde{\mathbf{a}}_{c,k+1}^T] R_{c,k}^{-1} [\nabla_{\mathbf{x}_{k+1}} \tilde{\mathbf{a}}_{c,k+1}^T]^T\}$. Note that \tilde{D}_k^{22} is a 4×4 matrix and we can verify that the matrix elements are given by (10.20).

10.5.2 With binary observations

Note that in computing PCRLB for the state estimate with binary observations, the fisher information matrix updating is given by (10.19) where now only the term D_k^{22} differs from subsection 10.5.1. With binary observations, D_k^{22} is given by,

$$D_k^{22} = Q^{-1} + \tilde{D}_k^{22}$$

where $\tilde{D}_k^{22} = \mathbb{E}\{-\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} \log p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1})\}$. The corresponding matrix elements of \tilde{D}_k^{22} are given in Appendix 10B.

10.6 Performance Analysis

10.6.1 With raw observations

In this section we analyze the performance of the proposed tracking algorithm when the raw observations are transmitted by sensor nodes to the active cluster head. For the simulations we assume a sensor network deployed in a square region of area $200m \times 200m$ as shown in Fig. 10.3. The network is assumed to be consisting of 4 cluster heads. We assume that there are 36 number of static nodes (including cluster heads) are deployed in a grid.

The initial target state is assumed to be Gaussian with mean $\boldsymbol{\mu}_0$ and covariance matrix $\boldsymbol{\Sigma}_0$. We assume $\boldsymbol{\mu}_0 = [-80 \ -80 \ 1 \ 1]^T$ and $\boldsymbol{\Sigma}_0 = \text{diag}([10 \ 10 \ 0.1 \ 0.1]^T)$. Sampling time is assumed to be $T_s = 1s$. The intensity of the state process noise $\sigma_v^2 = 0.4$. Observation noise variance at individual nodes, and cluster heads, σ_v^2 and σ_c^2 are set to 0.1. The target amplitude $A_0 = 100$. The tracking is performed for 60s and the number of particles in the particle filter is set to $S = 1000$. Particle

generation is performed as follows: As before, let CH_c be the current cluster head at time k . At time k , if the selected next CH , $CH_d \neq CH_c$, then, CH_c transmits its estimator $\hat{\mathbf{x}}_k$ to CH_d . At time $k + 1$, CH_d samples particles from the distribution, $p(\mathbf{x}_{k+1} | \mathbf{x}_k = \hat{\mathbf{x}}_{c,k|k})$.

The performance measure is taken as the root mean square error (RMSE) of the target position estimate given by,

$$RMSE_k = \sqrt{((x_k - \hat{x}_k)^2 + (y_k - \hat{y}_k)^2)}.$$

The RMSE is compared with the square root of the PCRLB components of the position error:

$$PCRLB_k^p = \sqrt{([J_k^{-1}]_{11} + [J_k^{-1}]_{22})}.$$

Figures 10.3 and 10.4 show the performance of the proposed scheme when there are 12 mobile nodes in the network. Then the average number of mobile nodes per a cluster is 3. From the total nodes for a cluster, the fraction of mobile nodes is 1/4. In both figures we assume that $r_D = 5m$ and $u_{max} = 10ms^{-1}$. Mobile nodes are directed to move at each time k such that to provide a β -coverage (exactly or approximately) for the predicted target position. In Fig. 10.3 estimated and true trajectories are shown with the assumed parameters. It can be seen that by allowing 3 nodes per cluster to be mobile, the target trajectory can be tracked with high accuracy compared to a static network. The results in Figs 10.3 and 10.4 are averaged over 50 Monte Carlo trials. In Fig. 10.3, we also compare the performance of particle filter based tracking algorithm to that with Extended Kalman Filter (EKF) when all nodes are static. It can be seen that, with the non-linear observation model considered in this Chapter, the performance with particle filter outperforms the performance with EKF. Moreover, from 50 trials it was observed that there were

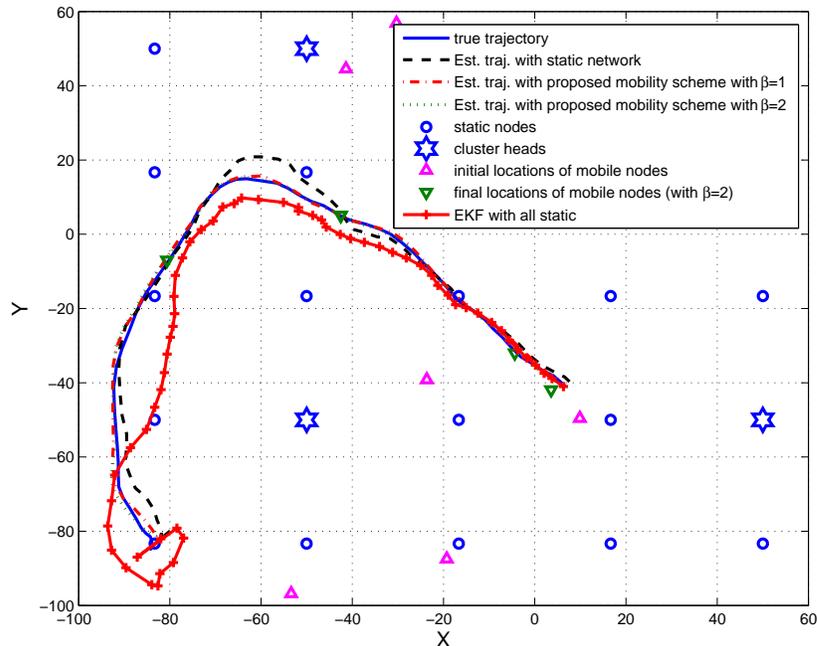


Figure 10.3: Estimated trace of the target trajectory with proposed mobility management scheme; $n_s = 36$, $n_m = 12$, $C = 4$, $u_{max} = 10ms^{-1}$, $r_D = 5m$

20 number of missing tracks with EKF and 0 missing tracks with particle filters. This essentially reflects the suitability of selecting PF as the tracking algorithm for the non-linear tracking problem considered in the Chapter. For the static network performance, we assume that all mobile nodes make measurements at their initial locations.

The RMSE and PCRLB analysis for proposed scheme for $u_{max} = 10ms^{-1}$ and $r_D = 5m$ are shown in Fig. 10.4 with PF. The results are shown for $\beta = 1$ and $\beta = 2$. It can be seen that when the objective is to maintain at least $\beta = 2$ number of nodes within the disk $D(\hat{\mathbf{x}}_{c,k+1|k}, r_D = 5m)$, the target trajectory can be tracked with lower RMSE. Also in that case, it can be seen that the RMSE performance gets very closed to the derived Posterior Cramer-Rao lower bound. For $\beta = 1$,

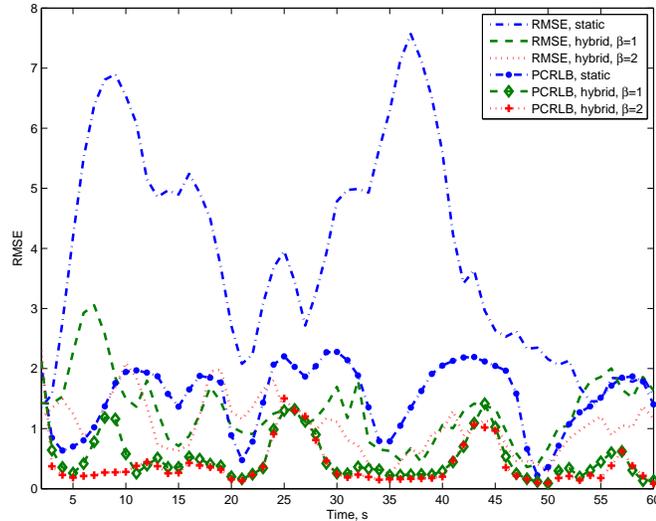


Figure 10.4: RMSE and PCRLB for the estimated target position when β is changing; $n_s = 36$, $n_m = 12$, $C = 4$, $u_{max} = 10m/s$, $u_{min} = 0$, $r_D = 5m$

that is to maintain at least 1 node within the disk $D(\hat{\mathbf{x}}_{c,k+1|k}, r_D = 5m)$ at each k , a considerable performance gain can be achieved compared to the static network. Note that there is always a trade-off between the value β and the energy consumption of mobile nodes, since when β is getting larger the number of mobile nodes to be moved is also increasing although it provides a high performance gain. On the other hand, it is of interest to investigate the performance metrics, when the maximum speed that a mobile node is varying.

In the next experiment, we investigate the effect of the maximum node speed for the proposed tracking algorithm when r_D is fixed. Figure 10.5 depicts the performance metrics when the maximum speed of a mobile node is varying. The results in Fig. 10.5 are corresponding to $u_{max} = 5m/s$ and $u_{max} = 10m/s$, and $\beta = 2$. It can be seen that for low values of u_{max} the performance gain is quite decreasing compared to higher values of u_{max} . This is due to the fact that for lower u_{max} values the maximum distance that a mobile node can move from one time period

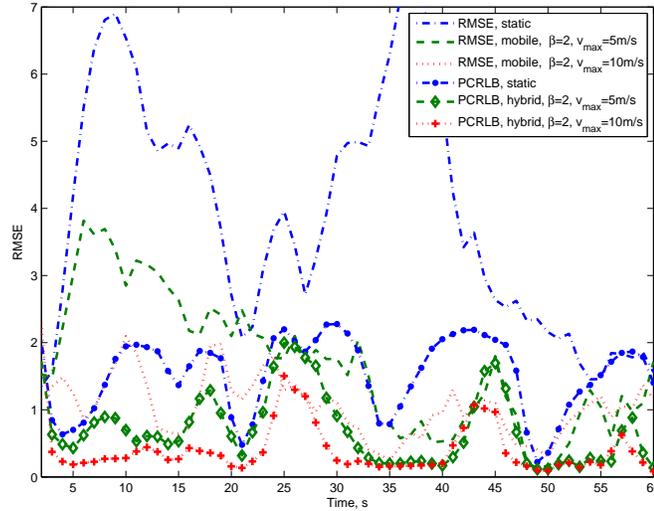


Figure 10.5: RMSE and PCRLB for the estimated target position when u_{max} is varying; $n_s = 36$, $n_m = 12$, $C = 4$, $u_{min} = 0$, $r_D = 5m$, $\beta = 2$

is lower, thus almost it might provide an approximate β -coverage rather than exact β -coverage. Also, with lower u_{max} , a considerable performance gain is achieved compared to all-static network, for a given r_D . However, as mentioned earlier, there is always a trade-off among the parameters u_{max} , β , r_D and the required performance gain.

Figure 10.6 shows the performance of the tracking algorithm when the number of mobile nodes is increasing. (However, note that we kept the total number of nodes constant). The fraction of mobile nodes is now allowed to be $5/12$. In Fig. 10.6, we compare the performance of the static network when β and u_{max} is varying. From Fig. 10.6, it can be seen that when the number of mobile nodes is large, the tracking performance with a given β ($\beta = 2$) does not have much effect on the maximum speed of mobile nodes. This is due to the fact that when there is relatively large number of nodes, it is more likely that there are sufficient number of nodes around the required position and they can reach the required location by moving small distances.

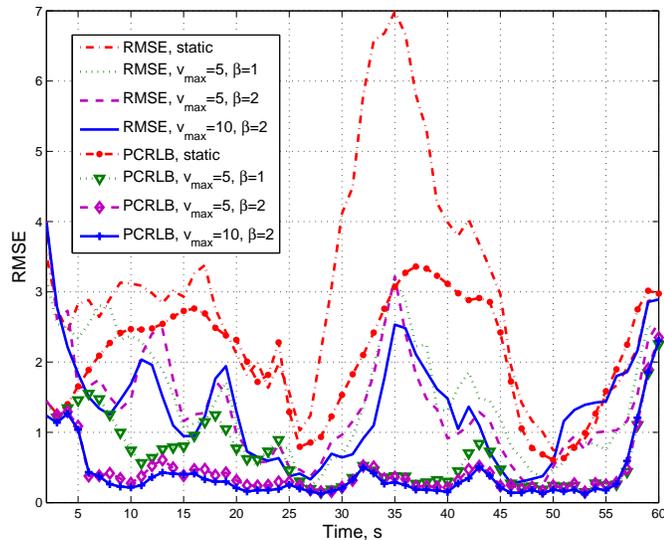


Figure 10.6: RMSE and PCRLB for the estimated target position when the number of mobile nodes is increasing; fraction of mobile nodes is $5/12$; $n_s = 28$, $n_m = 20$, $C = 4$, $u_{min} = 0$, $r_D = 5m$

Table 10.1 shows the average distance (during the same tracking period considered above) that a mobile node needs to move in order to provide β -coverage according to the proposed mobility algorithm. We set the number of static nodes to $n_s = 24$, the maximum speed of a node $u_{max} = 10m/s$ and the number of mobile nodes is varied. From Table 10.1, it can be seen that for a given number of static nodes, when the number of mobile nodes is increasing, the average total distance that a mobile nodes has to move to provide the required β -coverage dynamically, is significantly reduced. This is because, for large n_m , it has more flexibility to find mobile nodes in the close proximity of the predicted target position which would provide the desired β coverage by moving a small distance at any given time. As expected, it can be seen that when β is increasing, a mobile node has to move a larger total distance in average to provide the required coverage.

Table 10.1: Average distance (in m) a mobile node needs to move to provide β -coverage

	$\beta = 1$	$\beta = 2$
$n_m = 12$	71.7993	92.1394
$n_m = 16$	42.8655	66.8339

10.6.2 With binary observations

With constant threshold

According to (10.4), if the binary decisions at each node are made considering a constant threshold, i.e. $\tau_{j,k} = \tau$ for all $j \in \mathcal{V}$ and $k = 1, 2, \dots$, from simulation results we see that a relatively poor tracking performance is obtained with the SIR filter as seen in Fig. 10.7. In that case, even with the proposed node mobility model, a significant performance gain is not observed. With a constant threshold testing, the gain achieved by allowing nodes to be mobile towards the predicted target locations essentially is not reflected efficiently. Thus in the following we consider the tracking performance of the hybrid sensor network with dynamic threshold testing at sensor nodes.

With dynamic threshold

In [129], target tracking in a binary sensor network with adaptive threshold was considered for a static network, where they have proposed two schemes for adjusting threshold when the transmitted power level by the target is known and unknown. In this work, we consider different approach for maintaining dynamic threshold at each node at each time in the hybrid sensor network. The idea is that, at each time, each node makes the binary decision such that to minimize the average error in making the decision. At time k , if the threshold at node j is selected such that

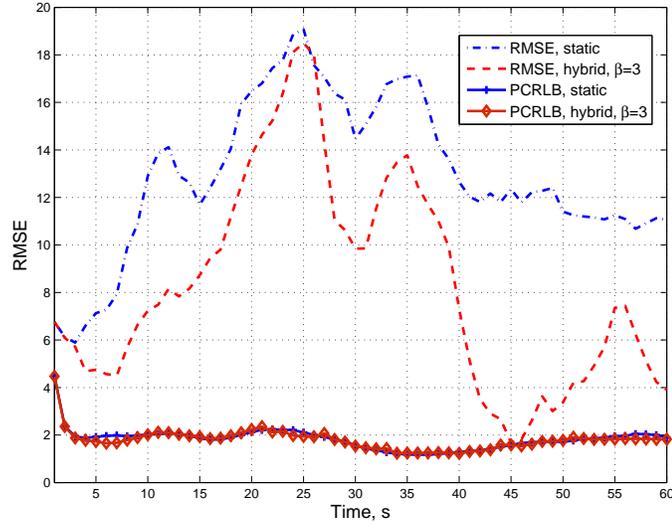


Figure 10.7: RMSE and PCRLB for the estimated target position with binary observations (constant threshold); fraction of mobile nodes is $5/12$; $n_s = 28$, $n_m = 20$, $C = 4$, $u_{min} = 0$, $r_D = 10m$, $\tau = 2$

$\tau_{j,k} = \frac{a_{j,k}}{2} = \frac{A_0}{2|\mathbf{r}_{j,k} - \bar{\mathbf{x}}_k|}$, from the Bayesian framework it can be seen that the average error associated with the decision (10.4) is minimized. However, it is to be noted that the threshold corresponding to time k at the j -th should be computed at time $k - 1$. If the cluster head CH_d is selected at time $k - 1$ as the lead cluster for time k , from the node mobility model discussed in Section 10.4, CH_d can compute the mobile node locations at time k . Since CH_d is assumed to know the locations of static nodes belonging to corresponding cluster, CH_d can compute the j -th node's threshold for time k as,

$$\hat{\tau}_{j,k} = \frac{A_0}{2|\mathbf{r}_{j,k} - \hat{\mathbf{x}}_{k|k-1}|}, \text{ for } j \in \mathcal{V}_d^k \quad (10.21)$$

where $\hat{\mathbf{x}}_{k|k-1}$ is the predicted target position at time $k - 1$ and \mathcal{V}_d^k is the set containing all node indices belonging to CH_d at time k .

The tracking performance of the hybrid sensor network with dynamic thresholds

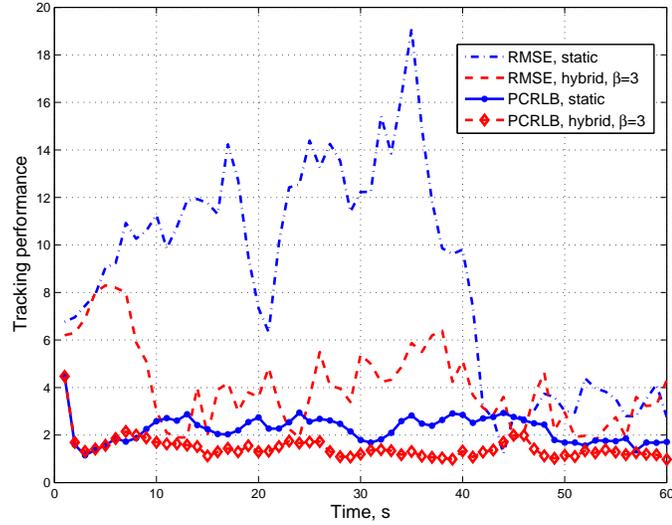


Figure 10.8: RMSE and PCRLB for the estimated target position with binary observations (dynamic threshold); fraction of mobile nodes is $5/12$; $n_s = 28$, $n_m = 20$, $C = 4$, $u_{min} = 0$, $r_D = 10m$

is shown in Fig. 10.8. In Fig. 10.8, we let $n_m = 20$, $n_s = 28$, $u_{max} = 10m/s$ and $\beta = 3$. It can be seen that despite of the inefficiency of the SIR filter with binary observations, an improved performance can be obtained with the proposed mobility management scheme, with the dynamic threshold.

10.7 Conclusions

In this Chapter we proposed a cluster based target tracking algorithm incorporating node mobility in a hybrid sensor network consisting of both static and mobile nodes, based on particle filtering. The tracking task is performed at a cluster head where the nodes need to only communicate with their cluster heads. We considered target tracking based on both raw as well as binary observations received at cluster heads

from individual nodes. In the proposed scheme, node mobility is exploited to maintain a required *coverage level* on the trace of the target at each time dynamically as the target moves. It can be seen that when the mobile nodes are directed to move to achieve relatively higher *coverage level*, RMSE performance of the target position estimate is getting much closer to PCRLB even with a relatively small number of mobile nodes. The trade-off between node velocity, required *coverage level* and the performance gain was shown. The proposed scheme is robust against cluster head or node failures in the network.

10.8 Appendix 10A

Proof of Theorem 10

Suppose that the predicted location of the target at time k is denoted by $\mathbf{P}_{k+1|k}$, as before, in Fig. 10.9. If any node is located inside the disk centered at $\mathbf{P}_{k+1|k}$ with a radius of r_D , it can cover the point $\mathbf{P}_{k+1|k}$. Thus the requirement is to move the required number of mobile nodes with the minimum distance in order to cover the point $\mathbf{P}_{k+1|k}$. Denote $d_j = |\mathbf{r}_{j,k} - \hat{\mathbf{x}}_{c,k+1|k}|$ to be the Euclidean distance between the predicted target position and the j -th mobile node. As seen in Fig. 10.9, if j -th mobile node locates inside the disk, it already covers the point $\mathbf{P}_{k+1|k}$, so it does not need to move. If the j -th node is analogous to the node m_2 , in the figure, to cover the point $\mathbf{P}_{k+1|k}$, it should move along the straight line towards $\mathbf{P}_{k+1|k}$ from its original location at least until it touches the perimeter of the disk. However, with the maximum speed that it can move, if it can not reach the perimeter by moving from time k to $k + 1$, it stops after moving $u_{max}T_s$ distance towards $\mathbf{P}_{k+1|k}$ from its original location. In that case, the optimal speed and direction (from geometry) of

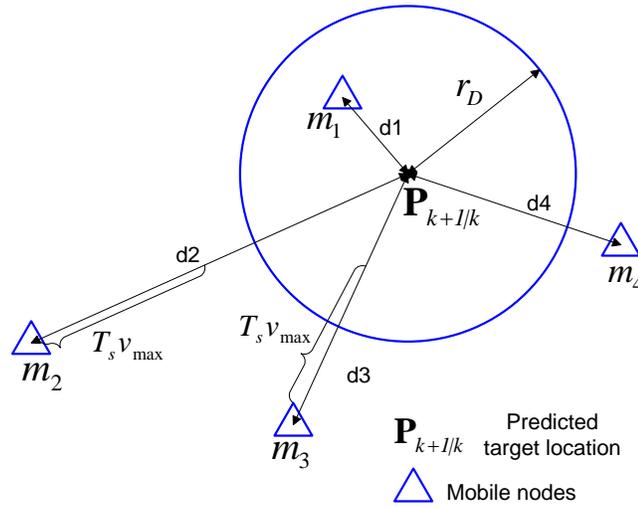


Figure 10.9: Illustration of mobile sensor locations and predicted target position in a particular cluster

the j -th mobile node at time k are given by

$$\hat{u}_{j,k} = u_{max}$$

and

$$\hat{\theta}_{j,k} = \text{atan2} \left(\frac{s_{j,k} - \hat{x}_{c,2(k+1|k)}}{r_{j,k} - \hat{x}_{c,1(k+1|k)}} \right), \quad (10.22)$$

where $\hat{x}_{c,1(k+1|k)}$ and $\hat{x}_{c,2(k+1|k)}$ are the X and Y coordinates of the predicted target location $\hat{\mathbf{x}}_{c,k+1|k}$. On the other hand, if the node j can reach the perimeter of the disk (as node m_3 in the figure) within the desired time, it moves the minimum distance such that it can cover the point $\mathbf{P}_{k+1|k}$. That is a distance of $|\mathbf{r}_{j,k} - \hat{\mathbf{x}}_{c,k+1|k}| - r_D$. In that case, the optimal speed is given by

$$\hat{u}_{j,k} = \max \left\{ \frac{1}{T_s} |\mathbf{r}_{j,k} - \hat{\mathbf{x}}_{c,k+1|k}|, u_{min} \right\}$$

and the optimal direction is as given by (10.22).

Note that if the condition $|\mathbf{r}_{j,k} - \hat{\mathbf{x}}_{c,k+1|k}| - r_D > u_{max} T_s$ is satisfied for all $j \in \tilde{\mathcal{V}}_{m,d}^k$, exact β -coverage cannot be achieved as desired. However, in this case, since the node moves a certain distance towards the predicted target position, it receives a higher signal strength compared to its original location. Thus in such cases, we call that the predicted target position is *approximately* β -covered.

10.9 Appendix 10B

With binary observations, the conditional likelihood function of observation at cluster head CH_c is given by (10.12). The logarithm of the likelihood function is given by (omitting the cluster head index for clarity)

$$\log p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}) = \sum_{j=1}^n \log p(y_{j,k+1}|\mathbf{x}_{k+1}),$$

where $p(y_{j,k+1}|\mathbf{x}_{k+1}) = p_{j,k+1}^\delta(\mathbf{x}_{k+1})\mathcal{N}(1, \sigma_\epsilon^2) + (1 - p_{j,k+1}^\delta(\mathbf{x}_{k+1}))\mathcal{N}(0, \sigma_\epsilon^2)$, where,

$$p_{j,k+1}^\delta(\mathbf{x}_{k+1}) = Q\left(\frac{\tau_{j,k+1} - a_{j,k+1}}{\sigma_v}\right).$$

Then the first partial derivative of $\log p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1})$ is given by,

$$\nabla_{\mathbf{x}_{k+1}} \log p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}) = \sum_{j=1}^n \frac{1}{p(y_{j,k+1}|\mathbf{x}_{k+1})} \frac{\partial p(y_{j,k+1}|\mathbf{x}_{k+1})}{\partial \mathbf{x}_{k+1}}. \quad (10.23)$$

Note that $\nabla_{\mathbf{x}_{k+1}}(\cdot)$ in (10.23) is a 4×1 vector, where the elements are given by,

$$\begin{aligned} (\nabla_{\mathbf{x}_{k+1}})_{11} &= \sum_{j=1}^n \frac{1}{p(y_{j,k+1}|\mathbf{x}_{k+1})} \frac{\partial p(y_{j,k+1}|\mathbf{x}_{k+1})}{\partial x_{1k+1}} \\ &= \sum_{j=1}^n \frac{(\mathcal{N}(1, \sigma_\epsilon^2) - \mathcal{N}(0, \sigma_\epsilon^2))}{p(y_{j,k+1}|\mathbf{x}_{k+1})} \frac{\partial p_{j,k+1}^\delta}{\partial x_{1k+1}} \end{aligned}$$

where $\frac{\partial p_{j,k+1}^\delta}{\partial x_{1k+1}}$ is given by,

$$\frac{\partial p_{j,k+1}^\delta}{\partial x_{1k+1}} = \frac{e^{-\frac{(\tau_{j,k+1} - a_{j,k+1}(\mathbf{x}_{k+1}))^2}{2\sigma_v^2}}}{\sigma_v^2} \frac{A_0(r_{j,k+1} - x_{1k+1})}{[(r_{j,k+1} - x_{1k+1})^2 + (s_{j,k+1} - x_{2k+1})]^{\frac{3}{2}}}$$

The second element $(\nabla_{\mathbf{x}_{k+1}})_{21}$ is given by,

$$(\nabla_{\mathbf{x}_{k+1}})_{21} = \sum_{j=1}^n \frac{(\mathcal{N}(1, \sigma_\epsilon^2) - \mathcal{N}(0, \sigma_\epsilon^2))}{p(y_{j,k+1}|\mathbf{x}_{k+1})} \frac{\partial p_{j,k+1}^\delta}{\partial x_{2k+1}}$$

where $\frac{\partial p_{j,k+1}^\delta}{\partial x_{2k+1}}$ is given by,

$$\frac{\partial p_{j,k+1}^\delta}{\partial x_{2k+1}} = \frac{e^{-\frac{(\tau_{j,k+1} - a_{j,k+1}(\mathbf{x}_{k+1}))^2}{2\sigma_v^2}}}{\sigma_v^2} \frac{A_0(s_{j,k+1} - x_{2k+1})}{[(r_{j,k+1} - x_{1k+1})^2 + (s_{j,k+1} - x_{2k+1})]^{\frac{3}{2}}}$$

and $(\nabla_{\mathbf{x}_{k+1}})_{31} = (\nabla_{\mathbf{x}_{k+1}})_{41} = 0$. Note that $\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} \log p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1})$ is a 4×4 matrix where only $(\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}})_{11}$, $(\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}})_{12}$, $(\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}})_{21}$, $(\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}})_{22}$ are non-zero. The (1,1)-th element is given by,

$$\begin{aligned} (\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}})_{11} &= \sum_{j=1}^n (\mathcal{N}(1, \sigma_\epsilon^2) - \mathcal{N}(0, \sigma_\epsilon^2)) \left(\frac{1}{p(y_{j,k+1}|\mathbf{x}_{k+1})} \frac{\partial^2 p_{j,k+1}^\delta}{\partial^2 x_{1k+1}} \right. \\ &\quad \left. - \frac{1}{p^2(y_{j,k+1}|\mathbf{x}_{k+1})} \frac{\partial p(y_{j,k+1}|\mathbf{x}_{k+1})}{\partial x_{1k+1}} \frac{\partial p_{j,k+1}^\delta}{\partial x_{1k+1}} \right) \end{aligned}$$

where $\frac{\partial^2 p_{j,k+1}^\delta}{\partial^2 x_{1k+1}}$ is given by (10.24).

$$\begin{aligned} \frac{\partial^2 p_{j,k+1}^\delta}{\partial^2 x_{1k+1}} &= \frac{e^{-\frac{(\tau_{j,k+1} - a_{j,k+1}(\mathbf{x}_{k+1}))^2}{2\sigma_v^2}}}{\sigma_v^2} \left(\frac{A_0(2(r_{j,k+1} - x_{1k+1})^2 - (s_{j,k+1} - x_{2k+1})^2)}{[(r_{j,k+1} - x_{1k+1})^2 + (s_{j,k+1} - x_{2k+1})]^{\frac{5}{2}}} + \right. \\ &\quad \left. \frac{(\tau_{j,k+1} - a_{j,k+1})}{\sigma_v^2} \frac{A_0^2(r_{j,k+1} - x_{1k+1})^2}{[(r_{j,k+1} - x_{1k+1})^2 + (s_{j,k+1} - x_{2k+1})]^3} \right) \end{aligned} \quad (10.24)$$

Similarly, The (2, 2)-th element is given by,

$$\begin{aligned} (\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}})_{22} &= \sum_{j=1}^n (\mathcal{N}(1, \sigma_\epsilon^2) - \mathcal{N}(0, \sigma_\epsilon^2)) \left(\frac{1}{p(y_{j,k+1} | \mathbf{x}_{k+1})} \frac{\partial^2 p_{j,k+1}^\delta}{\partial^2 x_{2k+1}} \right. \\ &\quad \left. - \frac{1}{p^2(y_{j,k+1} | \mathbf{x}_{k+1})} \frac{\partial p(y_{j,k+1} | \mathbf{x}_{k+1})}{\partial x_{2k+1}} \frac{\partial p_{j,k+1}^\delta}{\partial x_{2k+1}} \right) \end{aligned}$$

where $\frac{\partial^2 p_{j,k+1}^\delta}{\partial^2 x_{2k+1}}$ is given by (10.25).

$$\begin{aligned} \frac{\partial^2 p_{j,k+1}^\delta}{\partial^2 x_{2k+1}} &= \frac{e^{-\frac{(\tau_{j,k+1} - a_{j,k+1}(\mathbf{x}_{k+1}))^2}{2\sigma_v^2}}}{\sigma_v^2} \left(\frac{A_0(2(s_{j,k+1} - x_{2k+1})^2 - (r_{j,k+1} - x_{1k+1})^2)}{[(r_{j,k+1} - x_{1k+1})^2 + (s_{j,k+1} - x_{2k+1})]^{\frac{5}{2}}} + \right. \\ &\quad \left. \frac{(\tau_{j,k+1} - a_{j,k+1})}{\sigma_v^2} \frac{A_0^2(s_{j,k+1} - x_{2k+1})^2}{[(r_{j,k+1} - x_{1k+1})^2 + (s_{j,k+1} - x_{2k+1})]^3} \right). \end{aligned} \quad (10.25)$$

Following a similar approach, we have,

$$\begin{aligned} (\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}})_{12} &= (\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}})_{21} \\ &= \sum_{j=1}^n (\mathcal{N}(1, \sigma_\epsilon^2) - \mathcal{N}(0, \sigma_\epsilon^2)) \left(\frac{1}{p(y_{j,k+1} | \mathbf{x}_{k+1})} \frac{\partial^2 p_{j,k+1}^\delta}{\partial x_{1k+1} \partial x_{2k+1}} \right. \\ &\quad \left. - \frac{1}{p^2(y_{j,k+1} | \mathbf{x}_{k+1})} \frac{\partial p(y_{j,k+1} | \mathbf{x}_{k+1})}{\partial x_{2k+1}} \frac{\partial p_{j,k+1}^\delta}{\partial x_{1k+1}} \right) \end{aligned}$$

where, $\frac{\partial^2 p_{j,k+1}^\delta}{\partial x_{1k+1} \partial x_{2k+1}}$ is given by (10.26).

$$\frac{\partial^2 p_{j,k+1}^\delta}{\partial x_{1k+1} \partial x_{2k+1}} = \frac{2A_0}{\sigma_v^2} \frac{((r_{j,k+1} - x_{1k+1})(s_{j,k+1} - x_{2k+1}))}{[(r_{j,k+1} - x_{1k+1})^2 + (s_{j,k+1} - x_{2k+1})]^3} e^{-\frac{(\tau_{j,k+1} - a_{j,k+1}(\mathbf{x}_{k+1}))^2}{2\sigma_v^2}} \left[1 + \frac{A_0(\tau_{j,k+1} - a_{j,k+1})}{2\sigma_v^2} \right] \quad (10.26)$$

Now the elements of \tilde{D}_k^{22} are given by,

$$\begin{aligned} \left(\tilde{D}_k^{22} \right)_{11} &= \mathbb{E}\{-(\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}})_{11}\} \\ \left(\tilde{D}_k^{22} \right)_{12} &= \left(\tilde{D}_k^{22} \right)_{21} = \mathbb{E}\{-(\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}})_{12}\} \\ \left(\tilde{D}_k^{22} \right)_{22} &= \mathbb{E}\{-(\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}})_{22}\} \\ \left(\tilde{D}_k^{22} \right)_{23} &= \left(\tilde{D}_k^{22} \right)_{24} = 0 \\ \left(\tilde{D}_k^{22} \right)_{31} &= \left(\tilde{D}_k^{22} \right)_{32} = \left(\tilde{D}_k^{22} \right)_{33} = \left(\tilde{D}_k^{22} \right)_{34} = 0 \\ \left(\tilde{D}_k^{22} \right)_{41} &= \left(\tilde{D}_k^{22} \right)_{42} = \left(\tilde{D}_k^{22} \right)_{43} = \left(\tilde{D}_k^{22} \right)_{44} = 0 \end{aligned}$$

Algorithm 5 Node mobility management algorithm for tracking in the hybrid sensor network

INPUT: Predicted target location: $\hat{\mathbf{x}}_{c,k+1|k}$, number of static nodes for CH_d : $n_{s,d}^k$, number of mobile nodes for CH_d at time k : $n_{m,d}^k$

OUTPUT: Optimal speed $\hat{u}_{j,k}$ and the direction $\hat{\theta}_{j,k}$ of j -th mobile node for $j \in \tilde{\mathcal{V}}_{m,d}^k$

PROCEDURE:

- 1: Find number of static nodes inside the disk $(\hat{\mathbf{x}}_{c,k+1|k}, r_D)$, $n_{s,D}$,
 - 2: Check $\rightarrow n_{s,D} \geq \beta$ [i.e. $\hat{\mathbf{x}}_{c,k+1|k}$ is β -covered by static nodes]
 - 3: **if** yes (i.e. $n_{s,D} \geq \beta$) **then**
 - 4: $\mathbf{r}_{j,k+1} = \mathbf{r}_{j,k}$ for $j \in \mathcal{V}_{m,d}^k \Rightarrow$ no mobile node needs to move
 - 5: **else** {no (i.e. $n_{s,D} < \beta$)}
 - 6: Check $\rightarrow n_{m,d}^k \geq (\beta - n_{s,D})$ (i.e. to check whether CH_d has sufficient mobile nodes)
 - 7: **if** yes **then**
 - 8: Find $\tilde{\mathcal{V}}_{m,d}^k$ from $\mathcal{V}_{m,d}^k$
 - 9: **else** {no}
 - 10: Communicate with local CH s to find mobile node locations in neighboring clusters to form set $\tilde{\mathcal{V}}_{m,d}^k$ as described in Section 10.4.
 - 11: **end if**
 - 12: **end if**
 - 13: **for** $j = 1 : size(\tilde{\mathcal{V}}_{m,d}^k)$ **do**
 - 14: $f(j) = |\mathbf{r}_{j,k} - \hat{\mathbf{x}}_{c,k+1|k}|$
 - 15: **if** $f(j) - r_D \leq 0$ **then**
 - 16: $\mathbf{r}_{j,k+1} = \mathbf{r}_{j,k}$
 - 17: **else** { $f(j) - r_D > 0$ & $f(j) - r_D > T_s u_{max}$ }
 - 18: Compute $\hat{u}_{j,k}$ and $\hat{\theta}_{j,k}$ from (10.16) and (10.17)
 - 19: **else** { $f(j) - r_D > 0$ & $f(j) - r_D < T_s u_{max}$ }
 - 20: compute $\hat{u}_{j,k}$ and $\hat{\theta}_{j,k}$ from (10.14) and (10.15)
 - 21: **end if**
 - 22: **end for**
-

Chapter 11

Summary of Results and Future

Work

In this dissertation, we addressed research challenges in a resource constrained mobility assisted sensor network in signal processing and communication perspectives. In this Chapter we summarize our contribution in this dissertation highlighting the main results. We further discuss some of the possible directions that the work can be extended.

11.1 Summary of Results

In Chapter 3, we derived the optimal fusion performance for distributed detection at the fusion center when the local observations at individual nodes are correlated. We derived optimal power allocation schemes at sensor nodes to meet desired per-

formance level at the fusion center with independent as well as correlated observations [139, 140, 142, 143]. It was shown that the optimal scheme suggests to turn off the sensors with poor observation qualities and the bad communication channels, and only the sensor nodes with higher observation qualities and the good communication channels should transmit their information to the fusion center. The number of such active nodes is determined by certain factors such as the desired performance levels, total number of sensors in the network and the quality of received signals at the fusion center. Optimal power management schemes for distributed estimation with correlated observations is discussed in Chapter 4 [141].

In Chapter 5, a sequential methodology is developed for parameter estimation, incorporating the channel noise in inter-node communication links, when the final decision is made at an arbitrary sensor node without depending on a central fusion center [144]. In such a sequential process, since among all available sensor nodes in the network, not all nodes provide useful information which improves the accuracy of the estimator, we proposed distributed algorithms to select the best sequence of nodes to be participated in the sequential estimation process. It was shown that the proposed sequential estimation algorithm can be implemented distributively having information at local neighborhood.

In Chapters 6-10, adding mobility for performance improvement in hybrid sensor networks was considered for target detection and tracking, in different perspectives. In Chapter 6, we investigated the cost of deploying mobile nodes in a hybrid sensor network in terms of the minimum required fraction of mobile nodes to achieve different performance metrics, which are important in performance evaluations in hybrid sensor networks [145]. Derivation of different performance measures in Chapter 6 is based on random node mobility models.

In Chapter 7, we proposed a new interactive distributive mobility protocol for node mobility, collaborating between static and mobile nodes, in a hybrid sensor

network to provide efficient time varying coverage. The proposed mobility protocol is suitable for target detection when the target existence is unknown. From the performance analysis based on the proposed mobility protocol, we showed that the area uncovered by the static nodes, is approximately uniformly covered over time and the time that any arbitrary point is uncovered, is minimized. In Chapter 8, we developed an efficient sequential procedure to determine the worst case detection performance based on graph theoretic techniques, when smart targets try to evade the sensing region such that to minimize the probability of being detected by the sensor network.

Although, node mobility can be exploited for continuous coverage in hybrid sensor networks, it might cause significant cost since deploying mobile nodes is not as cost effective as deploying static nodes. Thus it might be desired to allow nodes to be mobile only if a detection is made with a certain confidence level at stationary configuration. In such situations, mobile nodes are directed to move, only if a decision on target present/absent can not be made by the stationary configurations with the desired quality. To that end, in Chapter 9 we developed two decision fusion models incorporating measurement uncertainties to enhance the detection performance by node movements, if the mobile nodes are directed to move from their stationary configurations. It was shown that, according to proposed decision fusion models, when the effect of node mobility is taken at node level decision updating, node mobility is only beneficial at lower nominal signal-to-noise ratio (SNR) region at local nodes. On the other hand, if the effect of node mobility is taken into account at the decision updating at the fusion center, it was shown that allowing even a small number of nodes to be mobile can boost the system performance significantly irrespective of the quality of nominal SNR at local nodes. The choice of two decision fusion models, however depends on certain factors such as computation capabilities at local nodes, number of affordable mobile nodes and the quality of desired performance metrics.

In Chapter 10, we developed a cluster-based distributed tracking algorithm based on particle filters in hybrid sensor networks with reactive mobility. The key feature in the proposed tracking algorithm is that the predicted target locations at each time step are maintained under a certain *coverage level* dynamically with the assistance of mobile nodes. The quality of the tracking performance with the proposed tracking algorithm was compared with the PCRLB, and it was seen that the corresponding tracking quality is getting much closer to PCRLB by adding a small number of mobile nodes compared to a all-static network.

11.2 Future Work

In this Section, we point out some future research directions paved by this dissertation.

Power management with joint source-channel coding with correlated observations: In Chapters 3 and 4, we considered the optimal power allocation for detection and estimation with correlated observations. In these studies we assumed that the local nodes perform amplify-and-forward local processing on their observations. It is interesting to address the problem of designing source-channel coding taking the correlations among node observations into account, to optimize the transmit power. Distributed implementations of the optimal power scheduling algorithms presented in Chapters 3 and 4 are based on the assumption that the channel state information (CSI) is available at sensor nodes. However, estimating CSI may require to consume additional amount of power. Thus it is interesting to investigate the performance degradation, when the exact CSI is not available at sensor nodes.

Detection performance with node mobility under time varying number of mobile nodes: In Chapter 6, the performance measures are derived assuming a fixed number

of total mobile nodes. However, in a mobility assisted sensor network, it is more likely to remove or add mobile nodes in timely manner due to energy depletion. In such scenarios, the total number of mobile nodes at a given time can be modeled as a random variable. It is interesting to extend the work to derive the time varying performance measures of a hybrid network under these conditions.

Analysis of steady state distribution of mobility protocol for hybrid sensor networks: We have developed a distributed mobility model in Chapter 7 for mobile node navigation in hybrid sensor networks in which a steady state uniform distribution is achieved with a relatively small number of moving steps compared to random walk mobility model. For analytical performance analysis in hybrid sensor networks, it is useful to obtain analytical formulas for the parameters which characterize the steady state probability distribution of the proposed model. In the proposed mobility model, it is noted that the node movements are not independent across mobile nodes. Thus deriving marginal probability distributions of individual node's mobility patterns is not satisfactory. It is an important problem to consider the joint steady state probability density function of the proposed mobility protocol.

Node scheduling for multiple target tracking in a hybrid sensor network: In Chapter 10, we proposed a cluster based target tracking algorithm with reactive mobility in which each predicted target location is covered to a desired *coverage level* exploiting node mobility. The open road ahead is to consider multiple target tracking. It is an interesting problem to optimally scheduling mobile nodes belonging to multiple clusters for multiple target tracking.

Impact of node mobility in MAC (Medium Access Control) and upper layers: In the work in this dissertation, sources of energy wasted in MAC layer such as collisions of data in the communication channels (between nodes and the fusion center or among nodes), idle listening and overhearing are not considered. Thus the results presented in this dissertation are desirable when node transmissions are scheduled

over collision free protocols. On the other hand, in traditional MAC protocols designed to maximize bandwidth utilization and fair usage of channels by all nodes in sensor networks, stationary nodes are considered. It is an important research direction to design MAC protocols for hybrid sensor networks taking the node mobility with different realistic mobility models into account.

References

- [1] <http://www.xbow.com>.
- [2] K. M. Abadir and J. R. Magnus. *Matrix Algebra*. Cambridge University Press, New York, USA, 2005.
- [3] K. Altarazi. Asymptotic fusion performance in a power constrained, distributed wireless sensor network. Master's thesis, Wichita State University, Wichita, KS, April 2006.
- [4] K. Altarazi, S. K. Jayaweera, and V. Aravinthan. Performance of decentralized detection in a resource-constrained sensor network. In *39th Annual Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2005.
- [5] B. D. Anderson and J. B. Moore. *Optimal Filtering*. Prentice Hall, Englewood Cliffs, NJ, 1979.
- [6] Swaroop Appadwedula, V. V. Veeravalli, and Douglas L. Jones. Energy efficient detection in sensor networks. *IEEE J. Select. Areas Commun.*, 23(4):693–702, Apr 2005.
- [7] V. Aravinthan, S. K. Jayaweera, and K. Al Tarazi. Distributed estimation in a power constrained sensor network. In *IEEE 63rd Vehicular Technology Conference (VTC) 2006 Spring*, Melbourne, Australia, May 2006.
- [8] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. on Signal Processing*, 50:174–188, Feb. 2002.
- [9] T. C. Aysal and K. E. Barner. Blind decentralized estimation for bandwidth constrained wireless sensor networks. *IEEE Trans. on Wireless commun.*, 7(5):1466–1471, May 2008.

- [10] I. Bahceci and A. K. Khandani. Energy-efficient estimation of correlated data in wireless sensor networks. In *40th Annual Conf. on Information Sciences and Systems (CISS)*, pages 973–978, Princeton, NJ, March 2006.
- [11] G. Barriac, R. Mudumbai, and U. Madhow. Distributed beamforming for information transfer in sensor networks. In *Third Int. Symp. on Information Processing in Sensor Networks (ISPN 2004)*, volume 26-27, pages 81–88, Berkeley, CA, April 2004.
- [12] A. S. Bashi, V. P. Jilkov, X. R. Li, and H. Chen. Distributed implementations of particle filters. In *Proc. of the Sixth Int. Conf. of Information Fusion*, pages 1164–1171, Cairns, Australia, July 2003.
- [13] C. Bettstetter. On the connectivity of wireless multihop networks with homogeneous and inhomogeneous range assignment. In *Proc. IEEE Vehicular Technology Conf. (VTC)*, volume 3, pages 1706–1710, Sept. 2002.
- [14] C. Bettstetter, G. Resta, and P. Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Trans. on Mobile Computing*, 2(3):257–269, Sep. 2003.
- [15] A. Bharathidasan and V. A. S. Ponduru. Sensor networks: An overview. In <http://www.csif.cs.ucdavis.edu/bharathi/sensor/survey.pdf>.
- [16] N. Bisnik, A. Abouzeid, and V. Isler. Stochastic event capture using mobile sensors subject to a quality metric. *IEEE Trans. on Robotics*, 23(4):676–692, Aug. 2007.
- [17] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. In *Proc. Wireless Communications and Mobile Computing (WCMC)*, volume 2, Nov. 2002.
- [18] Q. Cao, T. Yan, J. Stankovic, and T. Abdelzaher. Analysis of target detection performance for wireless sensor networks. In *Int. Conf. on Distributed Computing in Sensor Systems (DCOSS)*, pages 276–292, June-July 2005.
- [19] A. Carlisle and G. Doizier. An off-the-shelf PSO. In *Proc. 2001 Workshop on Particle Swarm Optimization*, Indianapolis, IN, 2001.
- [20] S. Caser and H. J. Hilhorst. Topology of the support of the two-dimensional lattice random walk. *Phys. Rev. Lett.*, 77(6):992–995, Aug. 1996.
- [21] V. Cevher, R. Velmurugan, and J. H. McClellan. Acoustic multitarget tracking using direction-of arrival batches. *IEEE Trans. on Signal Processing*, 55(6):2810–2825, June 2007.

- [22] J. F. Chamberland and V. V. Veeravalli. Asymptotic results for decentralized detection in power constrained wireless sensor networks. *IEEE J. Select. Areas Commun.*, 22(6):1007–1015, Aug. 2004.
- [23] J. F. Chamberland and V. V. Veeravalli. Decentralized detection in wireless sensor systems with dependent observations. In *Proc. Int. Conf. Comput., Commun. and Contr. Technol. (CCCT)*, Austin, TX, Aug. 2004.
- [24] J. F. Chamberland and V. V. Veeravalli. The impact of fading on decentralized detection in power constrained wireless sensor networks. In *Proc. Acoust., Speech, Signal Processing 2004. (ICASSP '04)*, volume 3, pages 837–840, 2004.
- [25] J. F. Chamberland and V. V. Veeravalli. How dense should a sensor network be for detection with correlated observations? *IEEE Trans. Inform. Theory*, 52(11):5099–5106, Nov. 2006.
- [26] J. F. Chamberland and V. V. Veeravalli. Wireless sensors in distributed detection applications. *IEEE Signal Processing Magazine*, pages 16–25, May 2007.
- [27] S. Chellappan, X. Bai, B. Ma, and D. Xuan. Sensor network deployment using flip-based sensors. In *Proc. of IEEE Mobile Sensor and Ad-hoc and Sensor Systems (MASS)*, pages 291–298, Nov. 2005.
- [28] B. Chen, W. B. Heinzelman, M. Liu, and A. T. Campbell. Editorial of special issue on wireless sensor networks. *EURASIP Journal on Wireless Commun. and Networking*, (4):459–461, 2005.
- [29] A. S. Chhetri, D. Morrell, and A. Papandreou-Suppappola. Scheduling multiple sensors using particle filters in target tracking. In *IEEE Workshop on Statistical Signal Processing*, pages 549–552, Sept. 2003.
- [30] J.-C. Chin, Y. Dong, W.-K. Hon, and D. K. Y. Yau. On intelligent mobile target detection in a mobile sensor network. In *Proc. of Int. Conf. on Mobile Ad-hoc and Sensor Systems (MASS)*, Pisa, Italy, Oct. 2007.
- [31] T.-L. Chin, P. Ramanathan, and K. K. Saluja. Analytical modeling of detection latency in mobile sensor networks. In *Conference on Information Processing in Wireless Sensor Networks (IPSN)*, April 2006.
- [32] T.-L. Chin, P. Ramanathan, K. K. Saluja, and K.-C. Wang. Exposure for collaborative detection using mobile sensor networks. In *IEEE Int. Conf. on Mobile Adhoc and Sensor Systems*, Washington, DC, Nov. 2005.
- [33] C.-Y. Chong and S. P. Kumar. Sensor networks: Evolution, opportunities, and challenges. *Proc. of IEEE*, 91(8):1247–1256, Aug. 2003.

- [34] M. Chu, H. Haussecker, and F. Zhao. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *Int. J. High-Perform. Comput. Applicat.*, 16(3):90–110, 2002.
- [35] P. Closas, C. Ferndndez-Prades, and J. A. Ferndndez-Rubio. A particle filtering tracking algorithm for GNSS synchronization using laplaces method. In *IEEE Int. Conf. on Acoust., Speech, Signal Processing (ICASSP)*, pages 3409 – 3412, Las Vegas, Nevada, U.S.A., Apr. 2008.
- [36] M. Coates. Distributed particle filters for sensor networks. In *Third Int. Symp. on Information Processing in Sensor Networks, (IPSN)*, pages 99–107, Berkeley, CA, Apr. 2004.
- [37] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Willey and Sons, Inc., NY, 2006.
- [38] H. Cramer. *Mathematical Methods of Statistics*. Princeton University Press, Princeton, New Jersey, 1946.
- [39] S. Cui, J.-J. Xiao, A. J. Goldsmith, Z.-Q. Luo, and H. V. Poor. Estimation diversity and energy efficiency in distributed sensing. *IEEE Trans. on Signal Processing*, 55(9):4683 – 4695, Sept. 2007.
- [40] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez. Particle filtering. *IEEE Signal Processing Magazine*, pages 19–38, Sept. 2003.
- [41] P. M. Djurić, M. Vemula, and M. F. Bugallo. Target tracking by particle filtering in binary sensor networks. *IEEE. Trans. on Signal Processing*, 56(6):2229–2238, June 2008.
- [42] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.
- [43] O. Dousse and C. Tavoularis nad P. Thiran. Delay of intrusion detection in wireless sensor networks. In *Proc. of the 7th ACM int. symp. on Mobile ad hoc networking and computing*, pages 155 – 165, May 2006.
- [44] E. Drakopoulos and C. C. Lee. Optimum multisensor fusion of correlated local decisions. *IEEE Trans. Aerospace and Electronic Systems*, 27(4):593–606, July 1991.
- [45] T. M. Duman and M. Salehi. Optimal quantization for distributed estimation via a multipleaccess channel. In *IEEE Int. Symp. on Inform. Theory (ISIT)*, Whistler, Canada, Sept. 1995.

- [46] R. C. Eberhart and Y. Shi. Evolving artificial neural networks. In *Proc. 1998 Int. Conf. Neural Networks and Brain*, Beijing, P.R.C, 1998.
- [47] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the world with wireless sensor networks. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, (ICASSP)*, Salt Lake City, Utah, May 2001.
- [48] N. Farvardin and V. Vaishampayan. Optimal quantizer design for noisy channels: An approach to combined source - channel coding. *IEEE Trans. on Info. Theory*, 33(6):827–838, Nov. 1987.
- [49] N. J. Gordon, D. J. Salmond, and A.F. M. Smith. Novel approach to non-linear/non-gaussian bayesian state estimation. *IEE Proc-F*, 140(2):107–113, 1993.
- [50] R. M. Gray. Toeplitz and circulant matrices: A review. <http://ee.stanford.edu/gray/toeplitz.pdf>, Aug. 2002.
- [51] M Grossglauser and D. Tse. Mobility increases the capacity of ad-hoc wireless networks. In *Proc. IEEE INFOCOM*, 2001.
- [52] D. Gu. Distributed particle filter for target tracking. In *IEEE Int. Conf. on Robotics and Automation*, pages 3856–3861, Roma, Italy, Apr. 2007.
- [53] D. Gu, J. Sun, and H. Li. Consensus based distributed particle filter in sensor networks. In *Proc. of IEEE Int. Conf. on Information and Automation*, pages 302–307, Zhangjiajie, China, June 2008.
- [54] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Trans. on Signal Processing*, 50(2):425 – 437, Feb. 2002.
- [55] P. Hall. *Introduction to the Theory of Coverage Processes*. John Wiley and Sons, 1988.
- [56] H. R. Hashemi and I. B. Rhodes. Decentralized sequential detection. *IEEE Trans. Inform. Theory*, 35(3):509–520, May 1989.
- [57] A. Howard, M. J. Mataric, and G. S. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *6th Int. Symp. on Distributed Autonomous Robotics Systems (DARS02)*, Fukouka, Japan, June 2002.

- [58] P. Ishwar, R. Puri, K. Ramchandran, and S. S. Pradhan. On rate-constrained distributed estimation in unreliable sensor networks. *IEEE Jour. on Selected Areas of Communications*, 23(4):765 – 775, April 2005.
- [59] S. K. Jayaweera. Large sensor system performance of decentralized detection in noisy, bandlimited channels. In *IEEE 61st Vehicular Technology Conference (VTC) 2005 Spring*, Stockholm, Sweden, May 2005.
- [60] S. K. Jayaweera. Large system decentralized detection performance under communication constraints. *IEEE Commun. Lett.*, 9:769–771, Sep. 2005.
- [61] S. K. Jayaweera. Bayesian fusion performance and system optimization in distributed stochastic gaussian signal detection under communication constraints. *IEEE Trans. Signal Processing.*, 55(4):1238–1250, April 2007.
- [62] S. K. Jayaweera. Optimal decision fusion wireless sensor networks for distributed detection of a randomly-located target. In *Military Communications Conference MILCOM '07*, Orlando, FL, Oct. 2007.
- [63] S. K. Jayaweera and C. Mosquera. Distributed sequential estimation with noisy, correlated observations. *IEEE Signal Processing Letters*, 15:741–744, 2008.
- [64] D. Jensen. SIVAM: Communication, navigation and surveillance for the amazon. *Avionics Magazine (Online)*, June 2002. Available online: <http://www.aviationtoday.com>.
- [65] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proc. IEEE Conf. Neural Networks IV*, Piscataway, NJ, 1995.
- [66] J. Kennedy and W. M. Spears. Matching algorithms to problems: An experimental test of the particle swarm and some genetic algorithms on multi modal problem generator. In *Proc. IEEE Int. Conf. Evolutionary Computation*, 1998.
- [67] G. Kesidis, T. Konstantopoulos, and S. Phooha. Surveillance coverage of sensor networks under a random mobility strategy. In *IEEE Proc. Sensors*, volume 2, pages 961– 965, Oct. 2003.
- [68] O. Kosut, A. Turovsky, J. Sun, M. Ezovski, L. Tong, and G. Whipps. Integrated mobile and static sensing for target tracking. In *Military Communication Conference, MILCOM*, pages 1–7, Oct. 2007.
- [69] A. Krasnopeev, J.-J. Xiao, and Z.-Q. Luo. Minimum energy decentralized estimation in a wireless sensor network with correlated sensor noises. *EURASIP Journal on Wireless Commun. and Networking*, 2005(4):473–482, 2005.

- [70] L. Lazos, R. Poovendran, and J. A. Ritcey. Probabilistic detection of mobile targets in heterogeneous sensor networks. In *6th Int. Symp. on Information Processing in Sensor Networks, (IPSN)*, pages 519–528, 2007.
- [71] D. Li and Y. H. Hu. Energy-based collaborative source localization using acoustic microsensor array. *EURASIP Journal on Applied Signal Processing*, (4):321–337, 2003.
- [72] J. Li and G. AlRegib. Energy-constrained distributed estimation in wireless sensor networks. In *Military Communication Conference, MILCOM*, Oct. 2007.
- [73] J. Li and G. AlRegib. Rate-constrained distributed estimation in wireless sensor networks. *IEEE Trans. on Signal Processing*, 55(5):1634–1643, May 2007.
- [74] W. Li and H. Dai. Distributed detection in wireless sensor networks using a multiple access channel. *IEEE Trans. Signal Processing.*, 55(3):822–833, March 2007.
- [75] Y. Li and P. M. Djurić. Particle filtering for target tracking with mobile sensors. In *Proc. Acoust., Speech, Signal Processing (ICASSP)*, pages II–1101–II–1104, Honolulu, Hawai'i, U.S.A., Apr. 2007.
- [76] Y. Li and P. M. Djurić. Target tracking with mobile sensors using cost-reference particle filtering. In *Proc. Acoust., Speech, Signal Processing (ICASSP)*, pages 2549–2552, Las Vegas, Nevada, U.S.A., Apr. 2008.
- [77] L. Lima and J. Barros. Random walks on sensor networks. In *5th Int. Symp. Modeling and Optimization in Mobile, Ad hoc and wireless networks*, Apr. 2007.
- [78] Y. Lin, B. Chen, and L. Tong. Distributed detection over multiple access channels. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, (ICASSP)*, pages III–541–III–544, 2007.
- [79] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley. Mobility improves coverage of sensor networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 300–308, 2005.
- [80] Y. Liu, H. Ngan, and L. M. Ni. Power-aware node deployment in wireless sensor networks. In *IEEE Int. Conf. on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06) - Vol 1*, pages 128–135, 2006.
- [81] Z.-Q. Luo. An isotropic universal decentralized estimation scheme for a bandwidth constrained ad hoc sensor network. *IEEE Journal on Selected areas in Commun.*, 23(4):735–744, April 2005.

- [82] Z.-Q. Luo and J.-J. Xiao. Universal decentralized estimation in a bandwidth constrained sensor network. *IEEE Trans. Inf. Theory*, 51(6):2210–2219, June 2005.
- [83] D. Lymberopoulos and A. Savvides. XYZ: a motion-enabled, power aware sensor node platform for distributed sensor network applications. In *IPSN*, 2005.
- [84] G. Gordon M. Rosencrantz and S. Thrun. Decentralized sensor fusion with distributed particle filters. In *Proceedings of the Conf. on Uncertainty in AI (UAI)*, Z Acapulco, Mexico, 2003.
- [85] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *ACM Int. Workshop on Wireless Sensor Networks and Applications (WSNA)*, Atlanta, GA, Sept. 2002.
- [86] J. Mammen and D. Shah. Throughput and delay in random wireless networks with restricted mobility. *IEEE Trans. on Information Theory*, 53(3):1108–1116, Mar. 2007.
- [87] S. Marano, V. Matta, L. Tong, and P. Willett. A likelihood-based multiple access for estimation in sensor networks. *IEEE Trans. on Signal Processing*, 55(11):5155–5166, Nov. 2007.
- [88] S. Marano, V. Matta, and P. Willett. Distributed estimation in large wireless sensor networks via a locally optimum approach. *IEEE Trans. on Signal Processing*, 56(2):748–756, Feb. 2008.
- [89] M. McGuire. Stationary distributions of random walk mobility models for wireless ad hoc networks. In *Proc. 6th ACM Int. Symp. on Mobile Ad Hoc Networking and Computing*, pages 90–98, 2005.
- [90] V. Megalooikonomou and Y. Yesha. Quantizer design for distributed estimation with communication constraints and unknown observation statistics. *IEEE Trans. Communications*, 48(2):181–184, Feb. 2000.
- [91] S. Megerian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. Worst and best-case coverage in sensor networks. *IEEE Trans. on Mobile Computing*, 4(1):84–92, Jan./Feb. 2005.
- [92] S. Megerian, F. Koushanfar, G. Qu, and M. Potkonjak. Exposure in wireless ad-hoc sensor networks. In *Proc. of MOBICOM*, pages 139–150, July 2001.

- [93] G. Mergen, V. Naware, and L. Tong. Asymptotic detection performance of type-based multiple access over multiaccess fading channels. *IEEE Trans. Signal Processing.*, 55(3):1081–1092, March 2007.
- [94] G. Mergen and L. Tong. Type based estimation over multiaccess channels. *IEEE Trans. Signal Processing.*, 54(2):613–626, Feb. 2006.
- [95] R. Negi and A. Rajeswaran. Capacity of power constrained ad-hoc networks. In *Proc. IEEE Infocom*, pages 443–453, May 2004.
- [96] R. Niu and P. K. Varshney. Performance analysis of distributed detection in a random sensor field. *IEEE Trans. on Signal Processing*, 56(1):339 – 349, Jan. 2008.
- [97] R. Niu and P. K. Varshney. Performance analysis of distributed detection in a random sensor field. *IEEE Trans. on Signal Processing*, 56(1):339 – 349, Jan. 2008.
- [98] R. Niu, P. K. Varshney, M. Moore, and D. Klammer. Decision fusion in a wireless sensor network with large number of sensors. In *The 7th International Conference on Information Fusion*, Stockholm, Sweden, July 2004.
- [99] R. Olfati-Saber. Distributed tracking for mobile sensor networks with information-driven mobility. In *Proc. of American Control Conf.*, pages 4606–4612, New York City, NY, July 2007.
- [100] O. Ozdemir, R. Niu, and P. K. Varshney. Tracking in wireless sensor networks using particle filtering: Physical layer considerations. *IEEE. Trans. on Signal Processing*, 57(5):1987–1999, May 2009.
- [101] O. Ozfidan, U. Bayazit, and H. A. Curpan. Cluster based sensor scheduling in a target tracking application with particle filtering. In *IEEE Int. Symp. on Industrial Electronics (ISIE)*, pages 1741 – 1746, June 2007.
- [102] K. E. Parsopoulos and M. N. Vrahatis. *Intelligent Technologies: New Trends in Intelligent Technologies, Book Chapter: Particle Swarm Optimization Method for Constrained Optimization Problems: pages 214-220*. IOS press, 2002.
- [103] P. Z. Peebles. *Probability, Random Variables, and Random Signal Principles*. McGraw-Hill, 3rd Edition, NJ, USA, 1993.
- [104] V. Phipatanasuphorn and P. Ramanathan. Vulnerability of sensor networks to unauthorized traversal and monitoring. *IEEE. Trans. on Computers*, 53(4):364–369, Mar. 2004.

- [105] H. V. Poor. *An Introduction to Signal Detection and Estimation*. Springer-Verlag, New York, 1994.
- [106] S. S. Rao. *Optimization: Theory and Applications*. Wiley Eastern Limited, New Delhi, India, 1995.
- [107] A. Ribeiro and G. B. Giannakis. Bandwidth-constrained distributed estimation for wireless sensor networks- part II: Unknown probability density function. *IEEE Trans. on Signal Processing*, 54(7):2784–2796, July 2006.
- [108] A. Ribeiro and G. B. Giannakis. Bandwidth-constrained distributed estimation for wireless sensor networks-part I: Gaussian case. *IEEE Trans. on Signal Processing*, 54(3):1131–1143, Mar. 2006.
- [109] J. Robinson and Y. R. Samii. Particle swarm optimization in electromagnetism. *IEEE Trans. Antennas Propagat.*, 52(2):397–407, Feb. 2004.
- [110] L. Ronnie, M. Johansson, N. Xiong, and H. I. Christensen. A game theoretic model for management of mobile sensors. In *Proceedings of the Sixth International Conference of Information Fusion, 2003*, 2003.
- [111] I. D. Schizas, G. B. Giannakis, and Z-Q. Luo. Distributed estimation using reduced-dimensionality sensor observations. *IEEE Trans. on Signal Processing*, 55(55):4284 – 4299, Aug. 2007.
- [112] R. Serfozo. *Introduction to Stochastic Networks*. Springer, 1999.
- [113] S. D. Servetto and G. Barrenechea. Constrained random walks on random graphs: routing algorithms for large scale wireless sensor networks. In *Proc. First ACM Inter. Workshop on Wireless Sensor Networks and Applications (WSNA-02)*, New York, Sept. 2002.
- [114] X. Sheng and Y.-H. Hu. Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks. *IEEE. Trans. on Signal Processing*, 53(1):44–53, Jan. 2005.
- [115] X. Sheng, Y-H. Hu, and P. Ramanathan. Distributed particle filter with GMM approximation for multiple targets and localization and tracking in wireless sensor network. In *4th Int. Symp. on Information Processing in Sensor Networks, (IPSN)*, Apr. 2005.
- [116] G. T. Sibley, M. H. Rahimi, and G. S. Sukhatme. Robomote: A tiny mobile robot platform for large-scale ad-hoc sensor networks. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1143–1148, Washington, DC, May 2002.

References

- [117] L. Snidaro, R. Niu, P. Varshney, and G. L. Foresti. Sensor fusion for video surveillance. In *Proc. 7th Int. Conf. Information Fusion*, Stockholm, June 2004.
- [118] S.-H. Son, S. R. Kulkarni, S. C. Schwartz, and M. Roan. Communication-estimation tradeoffs in wireless sensor networks. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, (ICASSP)*, volume 5, pages v/1065–v/1068, Mar. 2005.
- [119] A. Speranzon, C. Fischione, and K. H. Johansson. Distributed and collaborative estimation over wireless sensor networks. In *Proc. 45th IEEE Int. Conf. on Decision and Control (CDC)*, pages 1025–1030, San Diego, CA, Dec. 2006.
- [120] Y. Sung, L. Tong, and H. V. Poor. Neyman-pearson detection of gauss-markov signals in noise: Closed-form error exponent and properties. *IEEE Trans. Info. Theory*, 52(4):1354–1365, Apr. 2006.
- [121] W. I. Tam and D. Hatzinakos. An adaptive gaussian sum algorithm for radar tracking. In *IEEE Int. Conf. on Communications 'Towards the Knowledge Millennium' (ICC)*, volume 3, pages 1351–1355, June 1997.
- [122] R. Tan, G. Xing, J. Wang, and H. C. So. Collaborative target detection in wireless sensor networks with reactive mobility. In *16th Int. Workshop on Quality of Service (IWQoS)*, University of Twente, Enschede, Netherlands, June 2008.
- [123] D. Teneketzis and Y. C. Ho. The decentralized Wald problem. *Inform. Computation*, 73(1):23–44, 1987.
- [124] C. Thejaswi and R. K. Patro. Distributed detection in bandwidth constrained wireless sensor networks. In *IEEE Radio and Wireless Symposium (RWS)*, pages 95–98, San Diego, CA, Jan. 2006.
- [125] Q. Tian and E. J. Coyle. Optimal distributed estimation in clustered sensor networks. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, Toulouse, France, May 2006.
- [126] P. Tichavský, C. H. Muravchik, and A. Nehorai. Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. *IEEE Trans. on Signal Processing*, 46(5):1386–1396, May 1998.
- [127] A. Tiwari, F. L. Lewis, and S. S. Ge. Wireless sensor network for machine condition based maintenance. In *Proc. 8th Control, Automation, Robotics and Vision Conference*, volume 1, pages 461–467, Dec. 2004.

References

- [128] John N. Tsitsiklis. Decentralized detection. *Advances in Statistical Signal Processing*, 2:297–344, 1993.
- [129] M. Vemula, M. F. Bugallo, and P. M. Djurić. Particle filtering-based target tracking in binary sensor networks using adaptive thresholds. In *2nd IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing, (CAMPASAP)*, pages 17–20, Dec. 2007.
- [130] S. Verdu. *Multiuser Detection*. Cambridge University Press, Cambridge, United Kingdom, 1998.
- [131] A. Verma, H. Sawant, and J. Tan. Selection and navigation of mobile sensor nodes using a sensor network. In *Third IEEE International Conference on Pervasive Computing and Communications (PerCom'05)*, pages 41–50, Pisa, Italy, 2005.
- [132] R. Vishwanathan and P. K. Varshney. Distributed detection with multiple sensors 1. fundamental. *Proc. IEEE*, 85(1):54–63, Jan. 1997.
- [133] R. Viswanathan and V. Aalo. On counting rules in distributed detection. *IEEE Trans. Acoustics, Speech and Signal Processing*, 37(5):772–775, May. 1999.
- [134] G. Wang, G. Cao, and T. LaPorta. A bidding protocol for deploying mobile sensors. In *Proc. 11th IEEE Int. Conf. on Network Protocols*, pages 315–324, Nov. 2003.
- [135] G. Wang, G. Cao, and T. La Porta. Movement assisted sensor deployment. *IEEE Trans. Mobile Computing*, 5:640–652, 2006.
- [136] W. Wang, V. Sirinivasan, and K-C. Chua. Trade-offs between mobility and density for coverage in wireless sensor networks. In *Proc. 13th annual ACM int. conf. on Mobile computing and networking*, pages 39–50, Montreal, Quebec, Canada, 2007.
- [137] Y. Wang, X. Wang, B. Xie, D. Wang, and D. P. Agrawal. Intrusion detection in homogeneous and heterogeneous wireless sensor networks. *IEEE. Trans. on Mobile Computing*, 7(6):698–711, June 2008.
- [138] P. Willet, P. F. Swaszek, and R. S. Blum. The good, bad, and ugly: Distributed detection of a known signal in dependent Gaussian noise. *IEEE Trans. Sig. Proc.*, 48(12):3266 – 3279, December 2000.
- [139] T. Wimalajeewa and S. K. Jayaweera. Optimal power scheduling for data fusion in inhomogeneous wireless sensor networks. In *IEEE International Conference on Video and Signal Based Surveillance, (AVSS)*, pages 73–78, Nov. 2006.

References

- [140] T. Wimalajeewa and S. K. Jayaweera. Power efficient analog forwarding for correlated data fusion in wireless sensor networks. In *IEEE 66th Vehicular Technology Conference, (VTC)*, pages 367–371, Sept.-Oct. 2007.
- [141] T. Wimalajeewa and S. K. Jayaweera. Power efficient distributed estimation in a bandlimited wireless sensor network. In *41st Asilomar Conference on Signals, Systems and Computers (ACSSC)*, pages 2156–2160, Pacific Grove, CA, Nov. 2007.
- [142] T. Wimalajeewa and S. K. Jayaweera. PSO for constrained optimization: Optimal power scheduling for correlated data fusion in wireless sensor networks. In *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, (PIMRC)*, pages 1–5, Sept. 2007.
- [143] T. Wimalajeewa and S. K. Jayaweera. Optimal power scheduling for correlated data fusion in wireless sensor networks via constrained PSO. *IEEE Trans. on Wireless commun.*, 7(9):3608–3618, Sept. 2008.
- [144] T. Wimalajeewa and S. K. Jayaweera. Sequential estimation over noisy channels with distributed node selection. In *Military Communication Conference, MILCOM*, Boston, MA, Oct. 2009.
- [145] T. Wimalajeewa and S. K. Jayaweera. Trade-off between mobile node density and the detection performance in mobility-assisted sensor networks. In *Fifth Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Melbourne, Australia, Dec. 2009, to appear.
- [146] J. Wu and S. Wang. Smart: A scan based movement-assisted deployment method in wireless sensor networks. In *IEEE Conf. on Computer Communications (INFOCOM)*, Miami, FL, March 2005.
- [147] J.-J. Xiao, S. Cui, and A. J. Goldsmith. Power-efficient analog forwarding transmission in an inhomogeneous gaussian sensor network. In *6th Workshop on IEEE Signal Processing Advances in Wireless Commun.*, pages 121–125, June 2005.
- [148] J.-J. Xiao, S. Cui, Z.-Q. Luo, and A. J. Goldsmith. Joint estimation in sensor networks under energy constraints. In *Proc. IEEE first conf. Sensor and Ad Hoc commun. and Networks*, Santa Clara, CA, Oct. 2004.
- [149] G. Xing, J. Wang, K. Shen, Q. Huang, X. Jia, and H. C. So. Mobility-assisted spatiotemporal detection in wireless sensor networks. In *28th Int. Conference on Distributed Computing Systems (ICDCS)*, Beijing, China, June 2008.

- [150] K. Xu, H. Hassanein, G. Takahara, and Q. Wang. Relay node deployment strategies in heterogeneous wireless sensor networks: single-hop communication case. In *IEEE Global Telecommunications Conference (GLOBECOM)*, pages 21–25, Nov.-Dec. 2005.
- [151] J. M. Yang, Y. P. Chen, J. T. Horng, and C. Y. Kao. *Applying Family Competition to Evolution Strategies for Constrained Optimization. Lecture Notes in Computer Science*. Springer-Verlag, Berlin Heidelberg, New York, 1997.
- [152] K. Zhang and X. R. Li. Optimal sensor data quantization for best linear unbiased estimation fusion. In *Proc. 43rd IEEE Int. Conf. on Decision and Control (CDC)*, pages 2656–2661, Atlantis, Paradise Island, Bahamas, Dec. 2004.
- [153] X. Zhang, H. V. Poor, and M. Chiang. Optimal power allocation for distributed detection in wireless sensor networks. *IEEE Trans. Signal Processing.*, 56(9):4124–4140, Sept. 2008.
- [154] F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration. *IEEE Signal Processing Magazine*, 19(2):61–72, Mar. 2002.
- [155] L. Zhao, L. Peng, and J. Tan. A distributed dynamic model of mobile sensor network. In *Proc. 7th Int. Conf. on Machine Learning and Cybernetics*, Kunming, China, July 2008.
- [156] T. Zhao and A. Nehorai. Distributed sequential bayesian estimation of a diffusive source in wireless sensor networks. *IEEE Trans. Signal Processing.*, 55(4):1511–1524, Apr. 2007.
- [157] T. Zhao and A. Nehorai. Information-driven distributed maximum likelihood estimation based on Gauss-Newton method in wireless sensor networks. *IEEE Trans. on Signal Processing*, 55(9):4669–4682, Sept. 2007.
- [158] Y. Zou and K. Chakrabarty. Sensor deployment and target localization based on virtual forces. *Proc. INFOCOM*, pages 1293–1303, 2003.
- [159] Y. Zou and K. Chakrabarty. Distributed mobility management for target tracking in mobile sensor networks. *IEEE. Trans. on Mobile Computing*, 6(8):872–887, Aug. 2007.
- [160] L. Zuo, K. Mehrotra, P. K. Varshney, and C. K. Mohan. Bandwidth-efficient target tracking in distributed sensor networks using particle filters. In *9th Int. Conf. on Information Fusion*, pages 1–4, July 2006.

References

- [161] L. Zuo, R. Niu, and P. K. Varshney. Posterior CRLB based sensor selection for target tracking in sensor networks. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages II-1041-II-1044, Honolulu, Hawaii, USA, April 2007.