Energy efficient sensor scheduling and filtering algorithms for target tracking in wireless sensor networks

Lin, Jian Yong

2011

Lin, J. Y. (2011). Energy efficient sensor scheduling and filtering algorithms for target tracking in wireless sensor networks. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/44655>

<https://doi.org/10.32657/10356/44655>

**ENERGY EFFICIENT SENSOR SCHEDULING
AND FILTERING ALGORITHMS FOR
TARGET TRACKING IN WIRELESS SENSOR
NETWORKS**

LIN JIAN YONG

School of Electrical & Electronic Engineering

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

2011

Acknowledgements

Let me begin by first expressing my debt for my supervisor, Prof. Xie Lihua, who has given me the opportunity to embark on this project. I am also grateful to his direction, encouragement and patient guidance during my course of studies. It was a precious and rewarding experience working for and with Professor Xie.

I am also in the deepest gratitude to my co-supervisor, Dr. Xiao Wendong, who helps me carry out the project. His guidance and encouragement through years is very much appreciated.

Special thanks go to the team members in the project for their help, discussions and friendship. I also would like to thank my friends for their love, understanding, and support. It was happy and grateful time to be with them.

I thank the School of Electrical and Electronic Engineering of Nanyang Technological University for awarding me the postgraduate scholarship as my financial support in my PhD candidature period.

Finally, I own much to my parents who are always supporting me.

Table of Contents

Acknowledgements	i
Table of Contents	ii
Abstract	vii
List of Figures	ix
Symbols and Acronyms	xiii
1 Introduction	1
1.1 Motivations	1
1.2 Scope and Contributions of the Thesis	3
1.3 Organization of the Thesis	4
2 Literature Review	7
2.1 Overview of Wireless Sensor Networks	7
2.1.1 Hardware: Sensor Nodes and Fusion Center	7

2.1.2	Fundamental Issues in WSNs	10
2.2	Sensor Management in WSNs	13
2.3	Quantization in WSNs	15
2.4	KF and EKF Algorithms	19
2.4.1	Target Motion Model	19
2.4.2	Measurement Model	20
2.4.3	Kalman Filtering Algorithm	21
2.4.4	Extended Kalman Filtering Algorithm	23
2.5	Energy Model	23
3	Adaptive Single Sensor Scheduling	25
3.1	Introduction	25
3.2	Tracking Accuracy	26
3.3	Problem Formulation	27
3.4	Simulation Results	30
3.5	Conclusion	35
4	Multiple Sensor Scheduling Schemes	36
4.1	Introduction	36
4.2	Multi-Sensor with Fixed Sampling interval	37
4.2.1	Basic Idea	38

4.2.2	Detection Probability of a Single Sensor	40
4.2.3	Joint Detection Probability of Multiple Sensors	41
4.2.4	Monte Carlo Method for Approximation of Detection Probability	42
4.2.5	Leader and Tasking Sensors Selection	44
4.2.6	Experimental Results	44
4.3	Multi-sensor with Multi-hop configuration	49
4.3.1	Basic Idea	51
4.3.2	Energy Model and Minimum Energy Path	52
4.3.3	Selection of Next Tasking Sensors and Leader Sensor	53
4.3.4	Simulation Results	54
4.4	Multi-sensor with adaptive sampling interval	57
4.4.1	Problem Description	57
4.4.2	Sensor Detection Model	58
4.4.3	Mathematical Formulation	59
4.4.4	Deployment Density	62
4.4.5	Selection of Sampling Interval	63
4.4.6	Selection of Tasking Sensors	65
4.4.7	Selection of the Next Leader	67
4.4.8	Simulation Setup	67

4.4.9	Simulation Results	68
4.5	Conclusion	74
5	Quantized Kalman Filter	79
5.1	Introduction	79
5.2	Problem Formulation	80
5.3	Quantized Observations Kalman Filter	82
5.3.1	Quantization Strategy	82
5.3.2	Optimal Bandwidth Scheduling	84
5.4	Quantized Innovations Kalman Filter	89
5.5	Simulation	92
5.6	Conclusion	94
6	Compressed Kalman Filter	99
6.1	Introduction	99
6.2	Compressed Kalman Filter	101
6.2.1	Problem Formulation	101
6.2.2	General Suboptimal Solution	102
6.2.3	Boundedness of the Filtering Error Covariance of the CKF . .	105
6.3	Target Tracking Applications	108
6.3.1	Case 1: Target Moving in 1D Space	109

6.3.2	Case 2: Target Moving in 2D Space	111
6.4	Simulation Results	113
6.4.1	Energy Model	113
6.4.2	Case 1: Target Moving in 1D Space	114
6.4.3	Case 2: Target Moving in 2D Space	120
6.5	Conclusion	125
7	L-Banded Approximation and Drop Rate	127
7.1	Introduction	127
7.2	Problem Formulation	128
7.3	Centralized filter	130
7.4	Decentralized filter	132
7.5	L-Banded Approximation	137
7.6	A Numerical Example	138
7.7	Conclusion	140
8	Conclusion and Future Work	143
8.1	Conclusion	143
8.2	Future Work	145
	Author's Publications	146
	Bibliography	147

Abstract

In this thesis, the problems of sensor scheduling and state estimation over wireless sensor networks with target tracking application are investigated. Several energy efficient sensor scheduling schemes and filtering algorithms are presented with the consideration of limited communication bandwidth. The stability and performance robustness of the developed filters are addressed. The major outcomes of the work described in this thesis are summarized as follows:

First, a single sensor scheduling scheme is proposed to study the problem of sampling interval selection to achieve better energy efficiency while guarantee required tracking performance. Two sensor operation modes, namely fast tracking mode and tracking maintenance mode, are introduced to satisfy the tracking accuracy requirement and achieve better energy efficiency.

The single sensor scheduling scheme is then extended to multi-sensor scheduling. A uniform sampling interval is considered at first to illustrate the basic concepts introduced in multi-sensor scheduling schemes, such as the detection probabilities of a single sensor and multiple sensors. It is followed by a multi-sensor scheduling scheme with adaptive sampling interval. Three stages are designed for sampling interval determination, tasking sensors selection and the leader sensor selection. A scheme adopting a multi-hop configuration is also introduced to deal with the case when a target is moving at a relatively high speed such that any one-hop neighbor

sensor is not able to detect the target. The concept of minimum energy path is given to determine the relay sensors when exchanging the information between current tasking sensors and the selected tasking sensors at the next time step.

Third, the concept of quantized Kalman filter is developed to estimate a dynamic stochastic variable of interest in a wireless sensor network. Firstly, the quantized Kalman filter based on sensors' observations is presented. Approximate solutions for two optimal bandwidth allocation problems are given, where a tradeoff between the quantization level and the total bandwidth constraint or the total energy consumption is considered. To reduce the information loss introduced by quantization, a quantized Kalman filter is then developed, which is based on the sensors' innovations. The drawback of this filter is that it requires a feedback of one-step prediction of target state and the associated innovation variances from the fusion center. The effectiveness of both the quantized Kalman filters are illustrated by simulations.

Forth, a compressed Kalman filter is proposed. Instead of using dimension reduction, the proposed compressed Kalman filter reduces the totality of transmitted data by approximating the filtering error covariance matrix by a diagonal matrix. A suboptimal approximation method is derived to obtain the desired diagonal matrix. This method does not require much more computation than a standard Kalman filter and can be easily implemented in low cost simple sensor nodes. The stability of the proposed filter is guaranteed as well.

Finally, the relationship between the drop rate of transmitted data and the compression of data is investigated. Two types of transmission are discussed, transmitting raw measurement data and transmitting local processed estimate and its associated covariance. A banded matrix [10] is introduced to approximate the filter error covariance matrix $P(k|k)$. A numerical example is used to illustrate the effectiveness of the two types of data transmission.

List of Figures

2.1	A wireless sensor node	8
2.2	A close look of MicaZ node	9
2.3	WSN with a fusion center	9
2.4	Sensor network architecture	11
2.5	A quantized estimation system	16
2.6	A test bed of wireless sensor network	17
3.1	Adaptive sensor scheduling algorithm using two tracking modes . . .	29
3.2	Estimated target trajectory and the associated tasking sensor for each estimation using non-adaptive sensor scheduling scheme	32
3.3	Estimated target trajectory and the associated tasking sensor for each estimation using adaptive sensor scheduling scheme	33
3.4	Tracking accuracy achieved by different sensor scheduling schemes . .	33
3.5	Accumulated energy consumptions of different sensor scheduling scheme	34
3.6	Accumulated energy consumptions of different sensor scheduling scheme: with different parameters setting.	34

4.1	Target tracking by multiple sensors	38
4.2	Predicted detection probability of a single sensor	41
4.3	Detection probability of multiple sensors	42
4.4	Approximate the Gaussian distribution by 1000 random samples . . .	43
4.5	Tracking trajectory by single sensor scheduling scheme	47
4.6	Tracking trajectory by multi-sensor scheduling scheme	48
4.7	Evolution of the traces of the covariance matrix of the estimation . .	49
4.8	Number of sensors used in the multi-sensor scheduling scheme	50
4.9	Target loss time of the two sensor scheduling schemes	50
4.10	Tracking accuracy achieved in both schemes	56
4.11	Accumulated energy consumptions in both schemes	56
4.12	Tracking sustaining distance under different target speeds	57
4.13	Estimated and real target trajectory in a random distributed WSN. .	70
4.14	Evolution of the traces of the position uncertainty	71
4.15	Sampling interval varies at each time step	72
4.16	Number of sensors used at each time step	73
4.17	Sampling interval comparison for accurate and inaccurate q value . .	74
4.18	Tracking accuracies performance with respect to the given threshold, for accurate and inaccurate q value	75
4.19	Estimated and real target trajectory in a random distributed WSN using VigilNet method.	76

4.20	Evolution of the traces of the position uncertainty using VigilNet method.	77
4.21	Number of sensors used at each time step using VigilNet method. . .	77
4.22	Energy consumptions in single sensor scheduling schemes: $\Delta t_s = 0.1s$ means an uniform sampling interval of 0.1s scheme is adopted; $\Delta t_s = \text{Adaptive}$ means the adaptive sampling interval scheme is adopted.	78
4.23	Energy consumption in the multi-sensor scheduling schemes with uniform sampling intervals of 0.1s, 0.2s, 0.3s, 0.4s, and 0.5s cases and the proposed adaptive multi-sensor scheduling scheme	78
5.1	Decentralized estimation scheme based on quantized observations . .	81
5.2	Decentralized estimation scheme based on quantized innovations . . .	90
5.3	QKFQO: Tracking trajectory under $\sigma_{v_j}^2 = 0.1$	95
5.4	QKFQI: Tracking trajectory under $\sigma_{v_j}^2 = 0.1$	96
5.5	Tracking error comparison under $\sigma_{v_j}^2 = 0.1$	96
5.6	Tracking error comparison under $\sigma_{v_j}^2 = 0.01$	97
5.7	SOI: Tracking trajectory under $\sigma_{v_j}^2 = 0.01$	97
5.8	SOI: Tracking error comparison under $\sigma_{v_j}^2 = 0.01$	98
6.1	Target moving in 1D space	114
6.2	Position tracking in one dimension space	116
6.3	Position estimation error in one dimension space	117
6.4	Velocity tracking in one dimension space	117

6.5	Velocity estimation error in one dimension space	118
6.6	Computed filtering error variances for position of target moving in one dimension space case	118
6.7	Computed filtering error variances for velocity of target moving in one dimension space case	119
6.8	Energy consumption evaluation in one dimension space case	119
6.9	Tracking trajectories in two dimension space case	121
6.10	Position estimation error in two dimension space case	122
6.11	Velocity estimation in two dimension space case: x -direction	122
6.12	Velocity estimation error in two dimension space case: x -direction . .	123
6.13	Velocity estimation in two dimension space case: y -direction	123
6.14	Velocity estimation error in two dimension space case: y -direction . .	124
6.15	Energy consumption evaluation in two dimensional space case	124
7.1	A wireless sensor network	128
7.2	Tracking accuracy comparison	139
7.3	Communication channel of sensor 1	140
7.4	Communication channel of sensor 2	141
7.5	Tracking accuracy comparison, $\lambda = 0.8$	142
7.6	Tracking accuracy comparison, $\lambda = 0.2$	142

Symbols and Acronyms

CKF	Compressed Kalman Filtering
EKF	Extended Kalman Filtering
FC	Fusion Center
IDSQ	Information-Driven Sensor Querying
LEM	Leader Energy Measure function
LMI	Linear Matrix Inequality
KF	Kalman Filtering
MEMS	Micro-Electro-Mechanical Systems
MSE	Mean Square Error
PDA	Personal Digital Assistant
PDF	Probability Density Function
PIR	Passive Infrared Sensor
PMF	Probability Mass Function
QKFQI	Quantized Kalman Filter based on Quantized Innovations
QKFQO	Quantized Kalman Filter based on Quantized Observations
$s_L(k)$	Leader sensor node at k^{th} time step
STEM	Sparse Topology and Energy Management
$tr(A)$	Trace of matrix A
WSN(s)	Wireless Sensor Network(s)

Chapter 1

Introduction

1.1 Motivations

In recent years there has been considerable interest in wireless sensor networks (WSNs). WSN is generally a distributed system integrating sensors, microprocessors and communication and network devices and is used to gather information from the environment and provide data transmission between sensor nodes in order that users of different sites can realize resource sharing and coordinating manipulation.

WSN is an emerging technology for monitoring a physical world with a densely or sparsely deployed network of sensor nodes. The main advantages of WSN include its low cost, rapid deployment, self-organization, and fault tolerance. The advances in micro-electro-mechanical systems (MEMS), embedded microprocessors, wireless communications and networking technologies have made WSNs of large scale possible for a wide range of applications, such as navigation and controlling of moving vehicles, intelligent transportation and space exploration, environmental monitoring and diagnostics, industrial sensing, rescuing work on disaster, and body sensor network for medical/health care [1] [2] [3] [4] [5] [6] [7] [8] [9]. A critical scenario

is the distributed estimation over wireless sensor networks. This is useful in navigation and controlling of moving robot, as well as the prediction of disaster and environmental monitoring. In intelligent transportation, smart nodes can be tagged to mobile targets like containers or goods in a factory, so that these targets can be always traced and easily moved to desired locations. For environmental monitoring and diagnostics, ad hoc deployment of sensor nodes along the forest area can detect the the origin of a forest fire and relay the information to the end users before the fire is spread to a large area. The sensor nodes can also be deployed in the volcano area so that the development of eruption may be monitored and predicted. In a disaster area, huge number of sensor nodes can be deployed and the network can continuously monitor the area of large scale to allocate and save any survivors. For body sensor networks, it ranges from hospital care to personal and proactive care. The physiological data collected can be stored for a very long period of time and doctors and nurses can monitor the real time status of patients.

However, such a useful tool is constrained by many factors in the real world, such as the limited power, limited bandwidth and limited processing capability of sensor node. The design of a WSN need to overcome challenges in energy efficiency, bandwidth allocation, application specified requirements, and cost of sensor nodes which form the whole network. The hardware and software must be optimized to increase the efficiency of a WSN. Hardware includes the processing capability and limited storage space of low cost sensor nodes, while software addresses many other issues such as synchronization problem, communication protocol, network lifetime, fault tolerance, security, and self-organization.

1.2 Scope and Contributions of the Thesis

The thesis focuses on developing energy efficient sensor scheduling schemes and filtering algorithms for target tracking in WSNs, which are aimed to prolong the network lifetime, increase the robustness of WSNs, and strengthen the ability of self-organization. Major contributions of the thesis are summarized as follows:

- Several sensor scheduling schemes for target tracking in wireless sensor networks are proposed to improve the tracking accuracy, tracking reliability, and energy efficiency.
- A multi-hop configuration of WSN is studied for the case when the target is moving at a relatively high speed. An optimal tasking sensor selection method is proposed.
- Quantization problem in WSN is investigated. Two quantized Kalman filters are proposed: one is based on quantized observations, and the other is based on quantized innovations. The optimal bandwidth allocation is obtained and the information losses of these two methods are studied.
- A compressed Kalman filter for target tracking in WSN is developed. Our approach is based on reduction of state estimation error covariance. An approximated diagonal matrix that upper bounds the state estimation error covariance matrix is obtained. It is also proved that the filtering error covariance of the compressed Kalman filter is bounded. The uncertainty of the wireless communication channel is also taken into consideration.
- The relationship between the drop rate of transmitted data and the compression of data is investigated. Two types of transmission are discussed, transmitting raw measurements and transmitting local processed estimate and its

associated covariance. A banded matrix is introduced to approximate the filtering error covariance matrix.

1.3 Organization of the Thesis

The main contents of this thesis are organized as follows:

Chapter 2 gives an overview on wireless sensor networks. Some fundamental concerns of WSNs, such as synchronization, communication, localization, cooperation, power, and security problems, are reviewed. Existing research in estimation and control over networks are discussed as well. The Kalman filtering/Extended Kalman filtering algorithms (KF/EKF) are also briefly described in this chapter.

Chapter 3 discusses the effectiveness of using adaptive sampling intervals where a single sensor is scheduled at each time step. The sampling interval and the tasking sensor for the next time step are jointly selected to minimize the energy consumption while maintaining the required tracking accuracy. Two sensor operation modes are proposed in this part, the fast tracking mode and the tracking maintenance mode.

Chapter 4 introduces sensor scheduling schemes for target tracking application using multiple tasking sensors at each time step. The scheme using fixed sampling interval is studied firstly. It is then extended to adaptive sampling interval case. This scheme consists of three stages, the determination of the sampling interval, the selection of tasking sensors, and the selection of cluster head among the tasking sensors. It is shown that more tasking sensors can give much better performance, in the sense of improving the tracking accuracy and tracking reliability. This chapter also studies a multi-hop configuration for WSNs. This scheme deals with such a case that the communication range of the sensor nodes is relatively small while the target moves at a high speed. Tasking sensors should be selected from a wider range, that

is, not only select those sensors which are one-hop away from the current tasking sensors, but also select those sensors which are several hops away. A minimum energy transmission path is described in this chapter to forward the estimates from sensor to sensor.

Chapter 5 considers a quantized Kalman filter in a sensor network. Two types of quantized Kalman filters are investigated, namely, quantizing the sensors' observations (QKFQO) and quantizing the sensors' innovations (QKFQI). Approximate solutions for bandwidth scheduling are given, taking into consideration the constraints on energy consumption and bandwidth usage. It concludes that compared with QKFQO, QKFQI is better in saving power and reducing information loss since the values of the innovations are usually smaller than those of the observations.

Chapter 6 presents a compressed Kalman filter for target tracking in WSNs, which is concerned with reducing the totality of transmission data, instead of quantizing the sensors' observations or innovations, or reducing the system dimension by introducing an extra dimension reduction matrix. A diagonal matrix is used to approximate the filtering error covariance matrix $P(k|k)$, which reduces the totality of transmitted data from $n(n+1)/2 + n$ to $2n$, where the first term $n(n+1)/2$ is the number of transmitted elements of the filtering error covariance matrix, due to its symmetry property, and the second term n is the number of transmitted elements of the estimated state. A suboptimal solution for approximating $P(k|k)$ by a diagonal matrix is proposed. Most importantly, the boundedness of the filtering error covariance of the proposed filter is proved. It is also shown that the proposed CKF can achieve better energy efficiency.

Chapter 7 investigates the relationship between the drop rate of transmitted data and the compression of data. Two types of transmission are discussed, transmitting raw measurements and transmitting local processed estimate and its associated

covariance. A banded matrix [10] is introduced to approximate the filtering error covariance matrix $P(k|k)$. A numerical example is used to illustrate the effectiveness of the two types of data transmission.

Chapter 8 summarizes the results and gives conclusions as well as some topics for further investigation.

Chapter 2

Literature Review

2.1 Overview of Wireless Sensor Networks

A WSN typically consists of a large number of sensor nodes, which are working together to monitor a field of interest. The sensor nodes are usually densely deployed within the monitored region to obtain information about the environment. WSNs have a great potential for many applications and it is believed that, in future, WSNs will be an integral part of our daily lives. In the following, we will give an introduction to WSN and highlight some key research issues in a WSN.

2.1.1 Hardware: Sensor Nodes and Fusion Center

During the past few decades, advancements in electronic devices and wireless communication technologies have enabled the development of low cost and multi-functional sensor nodes. There are many kinds of sensor nodes available in the market, such as thermal, visual, infrared, acoustic and magnetic sensors, which can monitor various environmental conditions, including temperature, sound, vibration, pressure, or in-

tensity of light. Fig. 2.1 shows a wireless sensor node as an example, which combines one MicaZ and sensor board on the top for communication with other sensor node and/or fusion center, one 360 degree sonar array at the middle portion for actively gathering data about the environment, a PIR sensor array at the bottom for passive sensing, as well as a battery pack providing energy to the whole node. Fig.2.2 shows a close look of a MicaZ node [11].

Figure 2.1: A wireless sensor node

Depending on its configuration, a WSN may have one or several fusion centers (FC), or no FC at all. For a centralized WSN, there is at least one FC, which can be a PDA, a laptop, or even a powerful supercomputer. This center collects all the information sensed by those sensor nodes in the WSN, and then performs data fusion. The controlling signal delivered to each sensor node is also generated by this center (see Fig. 2.3) [12]. The advantage of this kind of WSN is that there is no computing requirement on each sensor node, i.e. sensor nodes only perform the basic measurement tasks, and then transmit the raw measurements to the fusion

Figure 2.2: A close look of MicaZ node

center, while complicated and energy consuming computation can be carried out at the FC site.

Figure 2.3: WSN with a fusion center

This kind of sensor nodes is cheaper in price, smaller in volume, and less in power consumption. However, due to the limited bandwidth available, it may result in information loss in quantizing the raw measurements and limited data rate assigned

to each sensor node. The data loss due to unreliable communication channel can be another big problem for the centralized system, as the lost data will never be recovered. What is more, this kind of WSN exhibits poor scalability. The complexity of computation and communication grows rapidly with the total number of sensors, making these schemes impractical for sensor-rich systems.

While for a decentralized WSN, there is no FC. The sensor nodes in the system have to perform computations locally, and the controlling signal is generated locally as well. A key issue in distributed sensor networks is scalability. Desirably, sensor networks must simultaneously track multiple phenomena, working within tight communication bandwidth, energy, and processing speed limits. Thus, it is critical to distribute the workload in an equitable way across only the “relevant” sensors, and leave other sensors available for other tasks [12]. This may be the only way to achieve the large scales needed for some applications. What is more, a decentralized WSN is inherently more robust against individual sensor node or link failures, because of redundancy in the network.

One disadvantage of decentralized WSN is that it requires higher quality sensor nodes, which are complex in design and more expensive. Besides the basic sensing function, the sensor nodes also need to store the data either got from parent sensor nodes or obtained locally, and perform some computation and pass the result on to the next node.

2.1.2 Fundamental Issues in WSNs

WSN is emerging as an exciting interdisciplinary research area, including aspects of signal processing, networking, distributed algorithms, and MEMS sensor technology. One big advantage of WSNs is that they allow a large scale deployment, which

is impractical for wired networks, considering the thousands of miles of wire that would be needed for the connections. A WSN can be easily deployed in place where there is no a priori sensing infrastructure. This flexibility has led to an increasing interest in using these networks for large-scale applications such as habitat monitoring [13], environmental monitoring, security surveillance, and battlefield awareness, extended from the original applications in target tracking and navigation and controlling moving vehicles. Other important fields of applications include home and building automation and military applications. Self-configurable, ubiquitous, easy to deploy, secure, undetectable WSNs are an ideal technology to employ in intelligence operations and war scenarios for detecting movements of enemy troops and artillery, and for monitoring and managing friendly resources. Fig. 2.4 shows the architecture of a typical WSN.

Figure 2.4: Sensor network architecture

The research community has quickly acknowledged the importance of large-scale ad hoc networks and has developed several services to support applications. Time services [14] provide the network with a globally consistent notion of time; localization services [15] allow computing nodes to acquire their coordinates relative to each

other; routing services [16] [17] reliably deliver packets while dynamically adapting to the ever-changing network topology; and tracking services [18] [19] follow objects moving through the network.

There are several fundamental issues that need to be considered.

- **Time:** The notion of time presents two distinct problems. First, coordinating sensing and actuating in a physical world requires either a sense of global time or the ability to resolve different time measurements to a meaningful representation. This can be seen as a synchronization problem of a WSN. Second, many existing design techniques assume that the computation of control signal and the processing of sensing and actuation occur within a negligible amount of time, thus new design and analysis techniques for WSNs are required, in which the time consumption for computation and communication is taken into consideration.
- **Communication:** It is expected that a network of sensor nodes will span a spatial area significantly greater than a single sensor node's maximum communication area. For a sensor node to send a message to another distant node, intermediate sensor nodes must be able to relay the message. Additionally, because sensor nodes can go offline without warning, the underlying communication protocol must be robust to network changes.
- **Localization:** Sensing and actuating events in a physical world must be paired with the relative or absolute location of the sensor node to be useful to control algorithms. That location must be assumed, provided, or deduced.
- **Cooperation:** Tasks that require the combined effort of two or more sensor nodes, such as any form of distributed sensing or distributed computing, require protocols and structures that provide handshaking, coordination, and

possibly hierarchy.

- **Power:** Energy is a valued resource in a WSN. Service and performance guarantees provided by a WSN must be balanced against overall power consumption.
- **Security:** To prevent numerous potential abuses of a WSN, a communication security layer must provide known guarantees for access control, message integrity, and confidentiality.

Although these issues are all related to WSN, researchers working in WSN, usually only consider those issues related to time, communication, localization, cooperation, and power, at the most of time, without the consideration of issues about security. However, all the parts described above are not isolated to each other.

2.2 Sensor Management in WSNs

Nowadays, a typical WSN consists of a large number of geographically distributed sensor nodes with limited energy resources and communication capability. Since sensor nodes have only small size batteries, minimization of sensor power consumption is important to ensure the lifespan of a WSN. Due to the limited resources for sensing, communication and computation, the network must rely on sensor management to balance the tracking accuracy and energy consumption [18].

Authors in [20] firstly applied the information theory in sensor management. It uses the change in entropy as a measure of information, and the sign of the information becomes the indicator of whether the information is gained or lost, which can be expressed as

$$\text{Information} = H_{\text{Source}} - H_{\text{Source after observation}}$$

where H is the entropy. A greedy heuristic approach, which is also based on entropy, is proposed in [21]. This approach sequentially selects tasking sensor to reduce target estimation uncertainty. In [22], some information utility measures are used to optimize sensor selection and incremental belief update in a centralized system. Authors in [23] developed several tree structures for in-network target tracking. A central unit is required to store the network topology. An adaptive threshold control scheme is developed in [24] in which the evolutionary programming technique is used to develop a control law, such that the energy consumption is small while the field-level probability of detection is maintained at a desired level. The detection thresholds are adjusted dynamically, which is computationally heavy and time consuming for simple low cost sensor nodes.

Sensor selection for a decentralized sensing system is introduced in [25], which is based on expected information gain. The information-driven sensor querying (IDSQ) technique is then introduced [26] [27] [28], which selects the next tasking sensor based on maximizing information utility and minimizing resource cost. This technique uses a general form of information utility, modeling both the information contents and the configuration of the network. What is more important, it is a distributed algorithm, which is applicable to WSNs of large scale. Note that the sampling interval in [29], defined as the time difference between any two successive time steps, is of same length at each time step, an adaptive sensor scheduling scheme for target tracking in WSNs is proposed in [29], in which a varying sampling interval concept is introduced. The sampling interval is adaptively selected within a given range to satisfy the tracking accuracy with less energy consumption. These existing approaches only select one tasking sensor node at each time step.

Generally, more sensors can achieve better tracking accuracy, especially when there exist some uncertainties in sensor measurements. It is believed that the quality

of target estimate improves with increasing number of measurements from different sensor nodes. It is, therefore, desirable to have many sensor nodes involved in localizing a target. However, more sensor nodes involved in measurement usually lead to more energy consumption. Hence, it is important to consider the tradeoff between target estimation accuracy and overall energy consumption of sensor nodes. A WSN should be self-organized into a topology that involves only the necessary subset of sensor nodes, in response to a query. It is observed that a small number of sensor nodes are sufficient to achieve this goal [30]. The work of [30] assumes a bounded uncertainty sensing model, where the exact probability distribution $p(z|x)$ is unknown but for a given x , the set of possible values for the measurement z is bounded. A hierarchical sensor network is introduced in [31]. It has a static backbone of sparsely placed powerful sensor nodes, which act as cluster heads to collect data from neighboring low-end sensors. In the VigilNet project [32], multiple sensors are activated at each time step to get the measurements and report their locations to a leader sensor, and the leader sensor estimates the target location by taking average of those reporting sensors' locations. However, this simple estimation method leads to a large estimation error due to the ignorance of motion information of the target. Another multi-sensor scheduling scheme is proposed in [33], which selects tasking sensors at each time step based on calculated target detection probabilities. The sampling interval, however, is pre-fixed. Quite often such uniform sampling interval based sensor scheduling schemes are difficult to satisfy a specific tracking accuracy [34], and their energy efficiency is usually low.

2.3 Quantization in WSNs

Research on quantization is not a new topic in control and signal processing fields. A typical quantized estimation system is shown in Fig. 2.5. The system consists of

N wireless sensor nodes and a FC. Each sensor node processes its measurement and communicates with the FC over a digital communication channel.

Figure 2.5: A quantized estimation system

A standard assumption in the classical control theory is that data transmission required by a control or state estimation algorithm can be performed with infinite precision. However, due to the growth in communication technology, it is becoming more common to employ limited capacity digital communication networks for exchange of information between system components. The resources available in such systems for communication between sensors, controllers and actuators can be severely limited due to size or cost. Since the radio spectrum is limited, communication constraint is a real concern [35]. This is also proved in Stilwell and Bishop's work, which showed that the major difficulty in controlling platoon of autonomous underwater vehicles is the bandwidth limitation on communication between them [36].

Fig. 2.6 shows a test bed of general sensor network configuration implemented in our lab. The low cost sensor nodes have only very limited energy, hence hash quantization of measured data may be helpful to prolong the lifespan of network, but the estimation may suffer heavy losses in precision. Some other scenarios in complex networked sensor systems which contain a very large number of low power

sensor nodes and MEMS also showed that classical optimal control and estimation theory cannot be applied since the control signals and the state information are sent via a limited capacity digital communication channel. It is thus very important to balance estimation accuracy and energy usage, which is mainly consumed by communication among sensor nodes and/or FC.

Figure 2.6: A test bed of wireless sensor network

Some significant results have been achieved during the past few decades. An important fundamental result about the minimum data rate is obtained by Nair and Evans [37] [38] [39]. They investigated a stochastic setting for the problem of stabilization for any linear time-invariant system via a perfect noiseless digital channel. They proved that in order to produce a bounded response, the data rate R must satisfy the following inequality,

$$R > \sum_{\lambda^u(A)} \max\{0, \log_2 |\lambda^u(A)|\},$$

where $\lambda^u(A)$ means the unstable eigenvalue of A . This inequality shows that the required data rate should be larger for a system which has a larger magnitude of unstable poles. These results were further extended to lossy communication channels case [40] [41].

Kim and Ortega [42] [43] proposed a quantizer design algorithm which is optimized for source localization in sensor network. It uses a cost function that takes into account the distance between the actual source position and the position estimated based on quantized data. Binary quantization has also been studied in the literature. An optimal one-bit quantizer has been proposed in [44], which is based on the maximum likelihood estimate. The innovation of sensor measurement is severely quantized and represented using only one bit. Their result shows that the variances of the estimator using the one-bit quantizer are close to that of estimate with un-quantized measurements. A simple probabilistic local quantization rule was proposed by Luo in a decentralized sensor network [6]. It is showed in [6] that the local quantization strategy together with a fusion rule can guarantee a bounded Mean Square Error (MSE), and that the average length of local messages is bounded as well. Moreover, the proposed scheme is isotropic and universal in the sense that the local quantization rules and the final fusion rules are independent of sensor ID, noise distribution, network size, or topology. It allows sensor nodes to operate identically and autonomously even when the network undergoes changes in size or topology.

Cui et al. proposed the optimal coded and un-coded transmission strategies for sensor networks in order to minimize the required energy per bit [45] [46]. Xiao et al. considered the problem of optimal power allocation for decentralized estimation [47] where sensor measurements are corrupted by additive spatially uncorrelated noises, while communication links from sensors to the fusion center differ in quality. An adaptive modulation scheme is considered for message transmission which leads

to the exponential dependence of energy on the message size. The optimal power and quantization levels for sensor nodes are then decided jointly at the fusion center. Sensor observations corrupted by additive noise [48] which can have arbitrary spatial correlation are considered by Krasnopeev et al.. They showed that the total sensor transmitted power can be minimized while meeting target Mean Square Error requirement if quantization levels are determined jointly by the fusion center using the knowledge of noise covariance matrix.

2.4 Kalman Filtering and Extended Kalman Filtering Algorithms

In this thesis, a single target tracking problem is investigated, and at each time instant single/multiple sensor(s) for detection and generating measurements will be selected. For linear systems, Kalman filter (KF) is used as the estimation method, while Extended Kalman filter (EKF) is used for nonlinear systems.

2.4.1 Target Motion Model

The target motion is modeled as the following linear state equation

$$x(k+1) = F(\Delta t_k)x(k) + w(\Delta t_k), \quad (2.1)$$

where $x(k) \in \mathbb{R}^n$ is the state of the target at the k^{th} time instant, $F(\Delta t_k)$ is the transition matrix, Δt_k is the sampling interval defined as the difference between the $(k+1)^{th}$ and k^{th} time instants, i.e. $\Delta t_k = t_{k+1} - t_k$, and $w(\Delta t_k)$ is the process noise.

For a 2-dimensional constant velocity model with $x(k) = [x_p(k), x_v(k), y_p(k), y_v(k)]'$

where $x_p(k)$ and $y_p(k)$ are the x - and y -coordinates of the target at time instant t_k , $x_v(k)$ and $y_v(k)$ are the velocities of the target along x - and y -directions at k^{th} time instant, the state transit matrix $F(\Delta t_k)$ is given by

$$F(\Delta t_k) = \begin{bmatrix} 1 & \Delta t_k & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t_k \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.2)$$

The matrix $Q(\Delta t_k)$ is assumed to be

$$Q(\Delta t_k) = q \begin{bmatrix} \frac{1}{3}\Delta t_k^3 & \frac{1}{2}\Delta t_k^2 & 0 & 0 \\ \frac{1}{2}\Delta t_k^2 & \Delta t_k & 0 & 0 \\ 0 & 0 & \frac{1}{3}\Delta t_k^3 & \frac{1}{2}\Delta t_k^2 \\ 0 & 0 & \frac{1}{2}\Delta t_k^2 & \Delta t_k \end{bmatrix}, \quad (2.3)$$

where q is a known scalar that determines the intensity of the process noise [49].

2.4.2 Measurement Model

The measurement model for a single sensor s_j , which takes the measurement $z_j(k)$ at the k^{th} time instant, is given by

$$z_j(k) = h_j(x(k)) + v_j(k), \quad (2.4)$$

where $h_j(\cdot)$ is a (generally non-linear) measurement function of sensor s_j and $v_j(k)$ is its measurement noise. $w(\Delta t_k)$ and $v_j(k)$ are assumed to be independent with zero-mean, white, Gaussian probability distributions. The covariance matrices of $w(\Delta t_k)$ and $v_j(k)$ are $Q(\Delta t_k)$ and $R_j(k)$ respectively.

While if multiple sensors $s_1, s_2, \dots$, and s_N provide the measurements $z_1(k), z_2(k), \dots$, and $z_N(k)$ at the k^{th} time instant, the measurement model is then given by

$$Z(k) = \begin{bmatrix} z_1(k) \\ z_2(k) \\ \vdots \\ z_N(k) \end{bmatrix} = \begin{bmatrix} h_1(x(k)) \\ h_2(x(k)) \\ \vdots \\ h_N(x(k)) \end{bmatrix} + \begin{bmatrix} v_1(k) \\ v_2(k) \\ \vdots \\ v_N(k) \end{bmatrix} = H(x(k)) + V(k). \quad (2.5)$$

$w(\Delta t_k)$ and $V(k)$ are assumed to be independent with zero-mean, white, Gaussian probability distributions. The covariance matrices of $w(\Delta t_k)$ and $V(k)$ are $Q(\Delta t_k)$ and $R(k)$, respectively.

2.4.3 Kalman Filtering Algorithm

KF is used for linear measurement model and operates in the following way: Given the estimate $\hat{x}(k|k)$ of $x(k)$ at the k^{th} time instant with covariance $P(k|k)$ and assuming sensor s_j is used for taking measurement at the $(k+1)^{th}$ time instant, the predicted state $\hat{x}(k+1|k)$ of the target can be calculated as

$$\hat{x}(k+1|k) = F(\Delta t_k)\hat{x}(k|k), \quad (2.6)$$

with the predicted error covariance

$$P(k+1|k) = F(\Delta t_k)P(k|k)F'(\Delta t_k) + Q(\Delta t_k), \quad (2.7)$$

and the predicted measurement of sensor s_j is

$$\hat{z}_j(k+1|k) = h_j(\hat{x}(k+1|k)). \quad (2.8)$$

For a multiple number of sensors which are taking the measurements at $(k + 1)^{th}$ time instant, the predicted measurement is similar to (2.5), i.e.,

$$\hat{Z}(k + 1|k) = H(\hat{x}(k + 1|k)). \quad (2.9)$$

In the rest of this thesis, $H(\cdot)$ is used to represent the measurement function $h_j(\cdot)$ or $H(\cdot)$, according to the number of sensors used. Then the innovation is given by

$$\gamma(k + 1) = Z(k + 1) - \hat{Z}(k + 1|k), \quad (2.10)$$

with the covariance

$$S(k + 1) = H(k + 1)P(k + 1|k)H'(k + 1) + R(k + 1). \quad (2.11)$$

The KF gain is given by

$$K(k + 1) = P(k + 1|k)H'(k + 1)S^{-1}(k + 1), \quad (2.12)$$

and the state estimation will be updated as

$$\hat{x}(k + 1|k + 1) = \hat{x}(k + 1|k) + K(k + 1)\gamma(k + 1), \quad (2.13)$$

with the covariance matrix

$$P(k + 1|k + 1) = P(k + 1|k) - K(k + 1)S(k + 1)K'(k + 1). \quad (2.14)$$

2.4.4 Extended Kalman Filtering Algorithm

EKF is used for nonlinear measurement model and operates in a similar way as KF, except the $H(\cdot)$ defined in (2.9). In EKF, the $H(\cdot)$ is defined as the Jacobian matrix of the measurement function at $(k+1)^{th}$ time instant with respect to the predicted state $\hat{x}(k+1|k)$. Similarly, the EKF gain is given then by

$$K(k+1) = P(k+1|k)H'(k+1)S^{-1}(k+1), \quad (2.15)$$

and the state estimation will be updated as

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + K(k+1)\gamma(k+1), \quad (2.16)$$

with the covariance matrix

$$P(k+1|k+1) = P(k+1|k) - K(k+1)S(k+1)K'(k+1). \quad (2.17)$$

2.5 Energy Model

Energy is consumed mainly during data communication, sensing, and data processing. The energy model used here is extended from [50].

Suppose sensor node s_j is currently performing a sensing task. We assume that the energy cost of s_j for sensing at each time step is E_s which is considered as a constant. The energy for transmission of b bits from sensor node s_j to s_k is

$$E_t(s_j, s_k) = (e_t + e_d r_{jk}^{\alpha_c})b, \quad (2.18)$$

where e_t and e_d are decided by the specifications of the transmitter of s_j , r_{jk} is the

distance between sensor s_j and sensor s_k , and α_c depends on the channel characteristics and is assumed to be known. The energy consumption for receiving data of b bits is

$$E_r(s_k) = e_r b, \quad (2.19)$$

where e_r is decided by the specifications of the receiver of sensor node s_k .

By using lower-power detection sensors such as PIR sensors, the energy consumption for the initial detection of target presence is negligible. As reported, a PIR sensor consumes much less (about one-fortieth) energy than an active sensor such as an ultrasonic sensor [51]. The energy consumption of the ultra-low power channel [52] to activate the tasking sensors is also ignored.

Chapter 3

Adaptive Single Sensor Scheduling

3.1 Introduction

Target tracking is one of the most important applications of WSNs in which, due to the limited resources for sensing, communication and computation, the network must rely on sensor management to balance the tracking accuracy and energy consumption [18].

In most of the existing sensor scheduling approaches, usually the tasking sensors are scheduled based on the uniform sampling interval, ignoring the changing of the target dynamics and obtained estimation accuracy. Therefore the sensor scheduling problem is degraded to a sensor selection problem. For example, in the information-driven sensor querying (IDSQ) [26] [28] approach, at each time step, one tasking sensor is selected for the next time step according to criteria of information gain and communication cost. In the entropy based sensor selection approach [21], an additional sensor is selected such that fusion of its measurements with existing information would yield the greatest entropy reduction of the target location distribution. Quite often such uniform sampling interval based sensor scheduling schemes are dif-

difficult to satisfy a specific tracking accuracy [34], and their energy efficiency is usually low.

This chapter addresses the more general sensor scheduling problem which answers the questions about which sensor should be selected as the next tasking sensor and when it should undertake the sensing task. Based on the Extended Kalman Filter (EKF) estimation technique [49] and a linear energy consumption model, an adaptive sensor scheduling strategy is introduced by jointly selecting the next tasking sensor and determining the sampling interval according to the predicted tracking accuracy and the tracking cost. In addition, the sensors are scheduled in two tracking modes using different objective functions, i.e., the fast tracking mode when the tracking accuracy is not satisfactory, and the tracking maintenance mode when the tracking accuracy is satisfactory.

In this chapter, the single target tracking problem is considered, and only one single sensor is selected for detection and generating measurement at each time instant. The target motion and the measurement model are given in Section 2.4.1 and Section 2.4.2. The sampling interval $\Delta t_k = t_{k+1} - t_k$ is adaptively chosen at each time step. EKF, as described in Section 2.4.4, is used as the estimation algorithm for this section as the measurement model is a nonlinear one.

3.2 Tracking Accuracy

Various measures can be defined based on the state estimation to represent the tracking accuracy, such as the trace, the determinant of the covariance matrix [28], Fischer information defined on the Fischer information matrix which is the inverse of the state estimation covariance [28], eigenvalues of the difference between the desired and the predicted covariance matrices [49] and entropy of the state estimation

distribution [21]. Based on the constant velocity model, we define the tracking accuracy $\Phi(k)$ at time step t_k as the position estimation error variance which is based on the covariance matrix $P(k|k)$ and can be expressed as

$$\Phi(k) = \sigma_x^2 + \sigma_y^2, \quad (3.1)$$

where σ_x^2 and σ_y^2 are the $(1, 1)$ and $(3, 3)$ elements of $P(k|k)$.

Given a predefined threshold Φ_0 , the tracking accuracy at time step t_k is considered as satisfactory if $\Phi(k) \leq \Phi_0$, and unsatisfactory otherwise.

3.3 Problem Formulation

Suppose the current time step is t_k and the current tasking sensor is s_i which receives the state estimate $\hat{x}(k-1|k-1)$ and its associated covariance matrix $P(k-1|k-1)$ from its parent tasking sensor. It first updates the state estimation by incorporating its new measurement $z_i(k)$ using the EKF algorithm. Then it uses the sensor scheduling algorithm to select the next tasking sensor s_j and the next sampling interval Δt_k such that the sensor s_j can undertake the sensing task at the time $t_{k+1} = t_k + \Delta t_k$. We suppose Δt_k should be in the range $[T_{min}, T_{max}]$. In practice, T_{min} must be larger than the total time for all operations, including sensing time, transmission delay, channel access time, as well as the data processing time. On the other hand, T_{max} can not be too large in order to deal with possible target maneuvering. The objective function is thus defined as:

$$J(s_j, \Delta t_k) = \omega \Phi_j(k) + (1 - \omega) \frac{E_t(s_i, s_j) + E_r(s_j) + E_s}{\Delta t_k}, \quad (3.2)$$

where $\Phi_j(k)$ is the predicted tracking accuracy, $E_t(s_i, s_j) + E_r(s_j) + E_s$ is the corresponding predicted energy consumption as described in (2.18) and (2.19) in Section 2.5, $\frac{E_t(s_i, s_j) + E_r(s_j) + E_s}{\Delta t_k}$ is considered as the averaged energy consumption over the period Δt_k , and $\omega \in [0, 1]$ is the weighting parameter used to balance the tracking accuracy and the energy consumption.

The sensors are scheduled in the following two tracking modes:

- Fast Tracking Mode: Based on the prediction, if none of the sensors can achieve the required tracking accuracy using any sampling interval within the range $[T_{min}, T_{max}]$, then Δt_k is set to the minimal sampling interval T_{min} and the sensor is selected according to

$$s_j^* = \arg_{s_j \in A} \min \{J(s_j, T_{min})\}, \quad (3.3)$$

where A is the candidate sensors that can be selected by sensor s_i . The purpose of this mode is to drive the tracking accuracy to be satisfied as soon as possible with consideration of the energy consumption.

- Tracking Maintenance Mode: Based on the prediction, if at least one sensor can achieve the desired tracking accuracy, the optimal sensor and sampling interval $(s_j^*, \Delta t_k^*)$ is determined by

$$(s_j^*, \Delta t_k^*) = \arg_{s_j \in A^*, \Phi_j(k) \leq \Phi_0} \min \left\{ \frac{E_t(s_i, s_j) + E_r(s_j) + E_s}{\Delta t_k} \right\}, \quad (3.4)$$

where A^* is the set of sensors that can achieve the required tracking accuracy. (3.4) utilizes the objective function (3.2) with $\omega = 0$. The basic idea of this mode is that when the predicted tracking accuracy can be achieved, the sensor and the sampling interval are selected according to the energy efficiency.

The sampling interval is selected from a predefined N values $\{T_t\}_1^N$ which is obtained by evenly dividing the interval $[T_{min}, T_{max}]$ into $N - 1$ sub-intervals such that $T_1 = T_{min}$, $T_N = T_{max}$, and $T_{t_1} < T_{t_2}$ if $t_1 < t_2$.

The proposed sensor scheduling algorithm is described in Fig. 3.1, which is inspired by the adaptive Radar revisiting interval design scheme in [50]. In the algorithm, $Flag_j$ is defined for sensor s_j to indicate, after prediction, whether it can achieve the required tracking accuracy. Step 2-2 is used to get the maximal sampling interval Δt_k^j for sensor s_j with the required predicted tracking accuracy.

Figure 3.1: Adaptive sensor scheduling algorithm using two tracking modes

3.4 Simulation Results

The proposed adaptive sensor scheduling scheme is used in a target tracking scenario. The network is formed using randomly distributed ranging sensors as shown in Fig. 3.2 as an example, where the black dot stands for the location of a sensor node. The detection region of a sensor is assumed to be the circular area centered at the node and with radius of 25 meters, which is not shown in the figure. The monitoring area U is $100m \times 100m$ with coordinates from $(0,0)$ to $(100,100)$ and covered by 20 randomly placed sensors (i.e., $N_s = 20$). The measurement model for sensor s_j is given by

$$z_j(k) = \frac{a}{\|(x(k), y(k)) - (x_s(j), y_s(j))\|} + v_j(k), \quad (3.5)$$

where $a \in \mathfrak{R}$ is the assumed known amplitude of the sound source, $(x(k), y(k))$ is the location of the target to be estimated at the k^{th} time step, $(x_s(j), y_s(j))$ is the known position of sensor s_j , $\|(x(k), y(k)) - (x_s(j), y_s(j))\|$ is the Euclidean distance between the target and sensor s_j , and $v_j(k)$ is the zero-mean Gaussian measurement noise of sensor s_j with variance σ_j^2 .

Simulations are done to verify the performance of the proposed adaptive sensor scheduling algorithm. The sound source produces sound with a constant amplitude $a = 40$ and is traveling at a constant speed $v = 10m/s$ along the straight diagonal line starting from $(20,20)$ and ending at $(80,80)$. We assume $\sigma_j^2 = 0.001$ for any sensor s_j and $q = 50$ in the covariance matrix of the process noise. For initialization, we assume the nearest sensor detects the sound source and initiates the tracking procedure. The initial estimation of the sound source is set $(10, 0, 25, 0)'$ with the covariance $25I$ where I is the identity matrix. For the energy model described in Section 2.5, the following parameters are used: $\alpha_c = 2$, $e_t = 45 \times 10^{-9}$ J/bit, $e_r = 135 \times 10^{-9}$ J/bit, and $e_d = 10 \times 10^{-9}$ J/bit-m². The sensing energy cost is

assumed to be $E_s = 8 \times 10^{-7} \text{J}$. The number of bits for transmission is assumed to be $b = 1024$. The sampling interval is bounded within $[T_{min}, T_{max}] = [0.1s, 0.5s]$. The weighting factor ω is set to be 0.16 for the objective function (3.2) and the threshold of the tracking accuracy is set as $\Phi_0 = 5$.

We compare the performance of the proposed adaptive sensor scheduling scheme with the IDSQ non-adaptive sensor scheduling scheme which is implemented by selecting the optimal next task sensor using the objective function (3.2). Fig. 3.2 shows the estimated target trajectory of the non-adaptive sensor scheduling with the sampling interval T_{min} . The associated tasking sensor of each estimated target position is indicated by a line between them. Similarly Fig. 3.3 shows the estimated target trajectory and the associated tasking sensor of each estimated target position using the adaptive sensor scheduling. Fig. 3.4 shows the tracking accuracy achieved by such two different sensor scheduling schemes, and Fig. 3.5 shows their accumulated energy consumptions. The non-adaptive sensor scheduling scheme with the sampling interval T_{min} schedules sensors with high frequency and can achieve better tracking accuracy (Fig. 3.4). The adaptive sensor scheduling has a similar performance at the initial stage due to the adaption of the fast tracking mode. Its first 2 sampling intervals are $0.1s$ and $0.3s$ respectively, then its tracking accuracy becomes satisfactory (Fig. 3.4) and the algorithm works in the tracking maintenance mode by using bigger and changing sampling intervals. The mean sampling interval of the adaptive sensor scheduling is $0.35s$. The main advantages of the adaptive sensor scheduling scheme is that it spends much less energy ($6.04mJ$ in this simulation) than the non-adaptive sensor scheduling scheme ($8.55mJ$ in this simulation) while meeting the required tracking accuracy, as illustrated in Fig. 3.5, which shows the energy saving is about 29.36%. The energy consumption for non-adaptive sensor scheduling scheme whose sampling interval is $0.2s$ is also shown in this figure. It shows that the adaptive scheme uses less energy as compared with the non-adaptive

Figure 3.2: Estimated target trajectory and the associated tasking sensor for each estimation using non-adaptive sensor scheduling scheme

scheme.

Regarding the sensitivity of the proposed adaptive scheme to the simulation parameters, some more simulations are carried out. Fig. 3.6 shows the energy consumption comparison with different parameter settings. When the threshold of the tracking accuracy is set as $\Phi_0 = 1$, it is shown in this figure that the energy consumption for the adaptive scheme is almost the same as the non-adaptive one with sampling interval $t = 0.2s$, and it consumed more energy as compared with the scheme having larger threshold of the tracking accuracy. It is desired since the requirement of tracking accuracy is higher, the sensor network needs to measure the target more frequently, hence more energy is consumed. When the minimal value of the sampling interval is changed to $T_{min} = 0.01s$, the energy consumption as shown in Fig. 3.6 is even less. Therefore, the proposed adaptive scheme can effectively save the energy and the performance is not sensitive to the parameters changing.

Figure 3.3: Estimated target trajectory and the associated tasking sensor for each estimation using adaptive sensor scheduling scheme

Figure 3.4: Tracking accuracy achieved by different sensor scheduling schemes

Figure 3.5: Accumulated energy consumptions of different sensor scheduling scheme

Figure 3.6: Accumulated energy consumptions of different sensor scheduling scheme: with different parameters setting.

3.5 Conclusion

In this chapter, an adaptive single sensor scheduling scheme for collaborative target tracking in WSNs was presented. This scheme jointly selects the next tasking sensor and determines the sampling interval for the next time step based on the predicted tracking accuracy and tracking cost. Two operation modes of tasking sensors, the fast tracking mode and the tracking maintenance mode, were introduced. Simulation results showed that this scheme can achieve significant energy efficiency while meeting the tracking accuracy requirement.

Chapter 4

Multiple Sensor Scheduling Schemes

4.1 Introduction

Due to the limited resources for sensing, communication and computation, WSNs must rely on sensor management to balance the tracking accuracy and energy consumption [18]. Generally, more sensors can achieve better tracking accuracy, especially when there is a possibility that sensor nodes may have false detection. It is observed that a small number of sensor nodes are sufficient to achieve this goal [30], in which a bounded uncertainty sensing model is assumed. The WSN should be self-organized into a topology that involves only the necessary subset of sensor nodes, in response to a query. A hierarchical sensor network is introduced in [31]. It has a static backbone of sparsely placed powerful sensor nodes, which act as leaders to collect data from neighboring low-end sensors. In the VigilNet project [32], the target location is obtained by taking average of those reporting sensors' own locations, which leads to a large estimation error due to the ignorance of motion

information of the target.

To overcome the limitations of the existing work, we will develop a novel distributed adaptive multi-sensor scheduling scheme for collaborative target tracking in WSNs such that the tracking accuracy satisfies a pre-specified requirement with low energy consumption. Prior to the adaptive scheme, a multi-sensor scheduling scheme with fixed sampling interval will be first presented in Section 4.2. In Section 4.3, we further explore this scheme in a WSN with multi-hop configuration, in which we will discuss the energy consumption and the tracking reliability of using multi-hop configuration. The adaptive scheduling scheme is described in Section 4.4.

4.2 Multi-Sensor Scheduling with Fixed Sampling Interval

In this section, we will consider the single target tracking problem, and at each time instant we will select multiple sensors for detection and generating measurements. The target motion and the measurement model are given in Section 2.4.1 and Section 2.4.2. Particularly, the sampling interval $\Delta t_k = t_{k+1} - t_k$ is the same for each time step, which can be denoted as $\Delta t_k = T$.

EKF, as described in Section 2.4.4, is used as the estimation algorithm, when measurements from multiple sensors are available, for prediction of the detection probability of a specific sensor, and for the prediction of the joint detection probability of the selected sensors.

4.2.1 Basic Idea

As illustrated in Fig. 4.1, the human being is acting as the target to be tracked. Those small circles represent wireless sensor nodes. In this chapter, we assume that each sensor node is with a single sensor and the target is initially detected by some low-power sensors, such as passive infrared (PIR) sensors [51], when it enters the monitoring area. The first sensor detects the target and initializes the tracking process. This sensor broadcasts its location information as the initial state of the target, with an initial state error covariance. At each time step, multiple selected sensors form a temporary tracking group, with one of them designated as the leader. All the selected sensors are assumed to take measurements simultaneously at the instant of the current time step. Those sensors that detect the targets send their measurements to the leader. The leader performs the state update for the target using these measurements, selects tasking sensors for the next time step, send the measurement command to the selected tasking sensors, and forward the current estimation (including the covariance matrix) to the next leader.

Figure 4.1: Target tracking by multiple sensors

The operations of a tasking sensor s_j at the k^{th} time step mainly include the following

three steps:

1. to get measurement $z_j(k)$ for the target,
2. to calculate the prediction of its measurement $\hat{z}_j(k)$ and innovation $\gamma_j(k) = z_j(k) - \hat{z}_j(k)$,
3. to transmit the calculated innovation $\gamma_j(k)$ to the leader sensor $s_L(k)$.

Besides performing the same sensing and calculation tasks as the other member sensors, the leader needs to perform the following additional tasks:

1. to combine the innovations from other tasking sensors and itself into a vector form for estimating the target state using the EKF algorithm;
2. to do a prediction on the target state $\hat{x}(k+1|k)$ and its associated error covariance $P(k+1|k)$;
3. to select a group of tasking sensors $G(k+1)$ for time step $k+1$;
4. to designate new leader $s_L(k+1)$ for time step $k+1$;
5. to pass the predicted target state $\hat{x}(k+1|k)$ and its error covariance $P(k+1|k)$ to the tasking group $G(k+1)$.

We assume that sensor nodes are normally in the sleeping mode and are triggered to perform sensing tasks by using an ultra-low power channel when they are scheduled to perform the sensing tasks. For each time step, energy consumption happens mainly in the following operations:

1. sensing by tasking sensors;
2. transmitting/receiving measurement data from tasking sensors to leader;

3. broadcasting/receiving by the current leader and tasking sensors in the next tasking group.

4.2.2 Detection Probability of a Single Sensor

Practically, there is no guarantee for a sensor to detect a target within its detection range. The following sensor detection model is assumed: when the distance between the sensor and the target is not larger than R_1 , the detection probability density function (PDF) is 1; when the distance increases from R_1 to R_2 , the detection PDF decreases from 1 to 0; when the distance is larger than R_2 , the detection PDF is 0. For example, in [53], an exponential decay model is adopted when the distance change from R_1 to R_2 . Here a simplified sensor detection model for a given sensor s_j is used, i.e., for a target location $u = (x, y)$,

$$p_{s_j}(u) = \begin{cases} \alpha, & \text{if } \mathbf{d}(u, (x_{s_j}, y_{s_j})) \leq r_j \\ 0, & \text{otherwise} \end{cases} \quad (4.1)$$

where (x_{s_j}, y_{s_j}) is the location of s_j , $\mathbf{d}(u, (x_{s_j}, y_{s_j}))$ is the Euclidean distance between the sensor s_j and the location point u , and r_j is the detection range of the sensor s_j . Without loss of generality, here we suppose the target location is 2-dimensional. When $\alpha = 1.0$, the target can always be detected if it is located inside this circle.

When a leader selects sensors, the prediction of the detection probability of a given sensor (say sensor s_j) is required. According to (2.6) and (2.7), the prediction of the location of the target is Gaussian distributed. We denote its expectation and the associated covariance matrix as μ and Σ , which are sub-vector and sub-matrix of $\hat{x}(k+1|k)$ and $P(k+1|k)$, respectively. Then the PDF function of the location

Figure 4.2: Predicted detection probability of a single sensor

$u = (x, y)$ will be

$$p(u) = \frac{1}{2\pi|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(u - \mu)' \Sigma^{-1}(u - \mu)\right). \quad (4.2)$$

As shown in Fig. 4.2, due to the limited sensing coverage of a sensor and the probabilistic nature of the target movement, the target may move to a location that can not be reached by a scheduled sensor, resulting in the loss of the target which is very dangerous for the tracking system. In addition, even when the target location u is within the detection coverage of the sensor, it is still possible that the sensor can not detect the target due to uncertain detection probability as given in (4.1).

The target detection probability of sensor s_j for the next time step can be given by

$$P_{s_j} = \iint p(u) p_{s_j}(u) dx dy. \quad (4.3)$$

4.2.3 Joint Detection Probability of Multiple Sensors

As shown in Fig. 4.3, by using multiple sensors, the target is more likely to be detected. Suppose $s_1, s_2, \dots$, and s_M are the sensors selected, then the prediction

Figure 4.3: Detection probability of multiple sensors

of the joint detection PDF for a target location $u = (x, y)$ will be

$$p_{s_1, s_2, \dots, s_m}(u) = 1 - \prod_{j=1}^m (1 - p_{s_j}(u)), \quad (4.4)$$

and the joint target detection probability is

$$P_{s_1, s_2, \dots, s_M} = \iint p(u) p_{s_1, s_2, \dots, s_M}(u) dx dy. \quad (4.5)$$

4.2.4 Monte Carlo Method for Approximation of Detection Probability

Analytically to calculate the detection probability for a single sensor or the joint detection probability for multiple sensors is difficult. Therefore we adopt the Monte Carlo simulation method to generate random samples to approximate the Gaussian distribution (4.2) of the target location prediction. Fig. 4.4 demonstrates an example for approximating a 2-dimensional location Gaussian distribution by 1000 random samples. Within a unit area, there will be more sample points if the dou-

ble integral of the probability density function over that area is larger. Hence, by counting the number of sampling within a certain area, it is possible to know an approximate value of the probability for that area without performing a more complicated double integral operation. Suppose the total number of samples is K , then we approximate the Gaussian distribution using the discrete probability mass function (PMF) $\bar{p}(x_k, y_k)$ with value $1/K$ for each sample location. Accordingly, the detection probability P_{s_j} of s_j is approximated by

$$\bar{P}_{s_j} = \sum_{k=1}^K \bar{p}(x_k, y_k) p_{s_j}(x_k, y_k) = \frac{1}{K} \sum_{k=1}^K p_{s_j}(x_k, y_k). \quad (4.6)$$

Similarly, the joint detection probability $P_{s_1, s_2, \dots, s_M}$ is approximated by

$$\bar{P}_{s_1, s_2, \dots, s_M} = \sum_{k=1}^K \bar{p}(x_k, y_k) p_{s_1, s_2, \dots, s_M}(x_k, y_k) = \frac{1}{K} \sum_{k=1}^K p_{s_1, s_2, \dots, s_M}(x_k, y_k). \quad (4.7)$$

Figure 4.4: Approximate the Gaussian distribution by 1000 random samples

Remark 4.1. *Most time consuming job in the Monte Carlo method is to calculate the eigenvalues and eigenvector for a given square matrix, which is commonly needed in many numerical computational approaches. A set of Gaussian distributed random numbers also needs to be generated, and the size of the set depends on the number of sampling points used for the approximation. This Monte Carlo method was not*

implemented in our WSN platform, due to the limited computation capability and memory size of our sensors.

4.2.5 Leader and Tasking Sensors Selection

The leader is selected as the sensor which is of the maximal detection probability. As for selection of other tasking sensors, we use a heuristic incremental sensor selection scheme where at each time the sensor with the maximal increase of joint detection probability (i.e., the sensor with the maximal joint detection probability together with the already selected sensors) is selected. It is implemented in the Monte Carlo method framework based on (4.6) and (4.7). The selection procedure terminates when the joint detection probability is larger than a given threshold (denoted as ϕ_D).

4.2.6 Experimental Results

We apply the proposed multi-sensor scheduling scheme to tracking of a moving sound source (the target) using a network of acoustic amplitude sensors. The measurement model for sensor s_j is

$$z_j(k) = \frac{a}{\|(x(k), y(k)) - (x_{s_j}, y_{s_j})\|} + v_j(k), \quad (4.8)$$

where $a \in \Re$ is the assumed known amplitude of the sound source, $(x(k), y(k))$ is the location of the sound source at time step t_k to be estimated, (x_{s_j}, y_{s_j}) is the known position of sensor s_j , and $v_j(k)$ is the zero-mean Gaussian measurement noise of sensor s_j with variance σ_j^2 . We use the constant velocity model as the target motion model.

The monitored field is $100m \times 100m$ with the coordinates from $(0, 0)$ to $(100, 100)$. Totally 25 sensors are distributed in this area uniformly, the distance between any two sensors is 20 and the detection range R is $10\sqrt{2}$. As a result, the whole area can just be covered by these sensors, as shown in Fig. 4.5, where the “*” stands for the location of sensor, while the circle stands for the detection range of sensor. We assume $\alpha = 1.0$, i.e., if the target is within the circle, the sensor can always detect it.

The sound source produces sound with a constant amplitude $a = 40$ and travels at a constant speed $v = 10m/s$ along the straight diagonal line starting from $(20, 20)$ and ending at $(80, 80)$. We assume $\sigma_j^2 = 0.001$ for any sensor s_j and $q = 50$ for the process noise. For initialization, we assume the nearest sensor detects the sound source and initiates the tracking process. The initial estimate of the sound source is assumed to be $(18, 5, 23, 5)'$ with covariance matrix $15I$ where I is the identity matrix.

The sampling interval is set to $T = 0.1s$. We also assume the joint detection probability $\phi_D = 0.95$, and the total number of samples K used in the Monte Carlo approximation is 1000.

We compare the performance of the proposed multi-sensor scheduling scheme with the single-sensor scheduling scheme which is implemented by selecting the sensor with the maximal detection probability at each step according to (4.6). Figures 4.5 - 4.7 are based on a single simulation run. Fig. 4.5 shows the estimated target trajectory of the single-sensor scheduling scheme. The associated tasking sensor of each estimated target position is indicated by a line between them. The duration before the target is lost by the tracking system is defined as sustaining tracking time. It is shown in Fig. 4.5 that the sustaining tracking time of this simulation by the single-sensor tracking scheme is $5.1s$, which is also shown in Fig. 4.7. Fig.

4.6 shows the estimated target trajectory and the associated tasking sensor of each estimated target position using the multi-sensor scheduling scheme. It is shown that the sustaining tracking time has been improved to the whole tracking course, and the estimated trajectory is closer to the true target trajectory. Fig. 4.7 shows the evolutions of the tracking accuracy achieved by such two different sensor scheduling schemes.

Fig. 4.8 shows the number of sensors used at each time step. At time step 0, the sensor network has little information on the target. In order to successfully detect the target (fulfill the detection probability threshold), more sensors are needed. Based on the calculation, two sensors are scheduled at time step 0 in the given scenario. At time step 49, the target is crossing the boundary of neighbor sensors. The uncertainty of the target location is high. It is insufficient to schedule only one sensor at this time step, which can be described by the detection probability, i.e. the detection probability threshold cannot be fulfilled no matter which sensor is to be scheduled. Based on the calculation in the proposed algorithm, there exist two sensors whose joint detection probability can satisfy the required threshold, therefore there is no need to select more than these two sensors for the sake of energy saving.

More simulations were carried out to compare the sustaining tracking time of both the tracking schemes and the statistical results for 50 simulation runs are given in Fig. 4.9, from which we can find that the multi-sensor scheduling scheme can track the target throughout the entire course of tracking for 39 times (i.e. 78%), and for the other runs the target is lost, the sustaining tracking time is mostly around 5.5s. However the single scheduling scheme can only track the target throughout the course of tracking for 18 times (i.e., 36%), and for the other runs the target is lost, most sustaining tracking times are around 2s - 3.6s, and 5s - 5.4s. It is clear that the multi-sensor scheduling scheme significantly improves the tracking reliability (Fig.

Figure 4.5: Tracking trajectory by single sensor scheduling scheme

Figure 4.6: Tracking trajectory by multi-sensor scheduling scheme

Figure 4.7: Evolution of the traces of the covariance matrix of the estimation

4.9) and the tracking accuracy (Fig. 4.7).

4.3 Multi-Sensor Scheduling Scheme with Multi-hop Configuration

In most existing work for sensor scheduling in WSNs, a single hop configuration is assumed where the next tasking sensors are selected from the single hop neighbor nodes of the current tracking leader. If the communication range is relatively small and the target moves with high speed, the sensor network using single hop configuration may fail to track a target when the target moves outside the detection range of any neighbor at the next time step.

Figure 4.8: Number of sensors used in the multi-sensor scheduling scheme

Figure 4.9: Target loss time of the two sensor scheduling schemes

On the other hand, for a WSN with sensor nodes densely deployed, neighbor nodes can be very close to each other. Hence, multi-hop communication in sensor networks can be expected which will consume less power than the traditional single hop communication. Multi-hop communication may also get around some of the signal propagation effects experienced in long-distance wireless communication.

In this section, a multi-sensor scheduling scheme with multi-hop configuration is presented based on the state prediction of the target movement where the next tasking sensors are not necessary to be 1-hop neighbors of the current tracking leader. An algorithm for minimal energy transmission path is described and the next tracking leader is selected based on energy consumption calculated from the minimal energy path.

4.3.1 Basic Idea

Due to the probabilistic characteristic of the target location, multiple sensors are needed at each time step to ensure successful tracking of the target. The distribution of the target location is assumed Gaussian. It is described by the predicted target state and its associated error covariance. A state-centric tasking sensor selection strategy is adopted where the next tasking sensors are selected based on the state prediction of the target, no matter how many communication hops a selected tasking sensor is away from the current tracking leader. One of these tasking sensors is selected as the next tracking leader, based on an energy cost measurement (as defined in Section 4.3.2).

All the sensors in this tasking group perform the sensing operation. The successful detection results from those member sensors are then transmitted to the tracking leader. The tracking leader combines the measurements and updates the target state

estimation, and then selects the next tasking sensors as well as the next tracking leader for the next time step based on its prediction. The predicted target state and its associated covariance are then forwarded to the tracking leader of the next tasking group. Transmission of message between any two sensor nodes is through the path with minimal energy consumption.

4.3.2 Energy Model and Minimum Energy Path

The basic energy model is described in Section 2.5. In this section, we further define the following parameters.

Definition 4.1. Let $l_e(s_{j_1}, s_{j_2})$ denote the energy cost per bit while transmitting data from sensor s_{j_1} to its 1-hop neighbor sensor s_{j_2} , then

$$l_e(s_{j_1}, s_{j_2}) = e_t + e_d r_{s_{j_1}, s_{j_2}}^\alpha + e_r, \quad (4.9)$$

where e_t , e_d , e_r , $r_{s_{j_1}, s_{j_2}}$ and α are defined in Section 2.5.

If sensor s_{j_n} is not within the communication range of sensor s_{j_1} , a multi-hop transmission is then necessary. The total communication energy associated with a path is equal to the sum of the communication energy of each link along this path. Usually there are multiple paths for two nodes in the network. Here we calculate such an end-to-end minimum energy path using the Dijkstra's shortest path algorithm.

Definition 4.2. Let $p_e(s_{j_1}, s_{j_2}, \dots, s_{j_n})$ denote the energy cost per bit while transmitting data from sensor s_{j_1} to sensor s_{j_n} via $(n-1)$ -hop path $s_{j_1}, s_{j_2}, \dots, s_{j_n}$, then

$$p_e(s_{j_1}, s_{j_2}, \dots, s_{j_n}) = \sum_{i=1}^{n-1} l_e(s_{j_i}, s_{j_{i+1}}), \quad (4.10)$$

where $l_e(s_{j_i}, s_{j_{i+1}})$ is defined above.

Hereinafter we let $E_{sp}(s_{j_1}, s_{j_n}, b_c)$ stand for the minimum energy consumption between sensors s_{j_1} and s_{j_n} for transmission of b_c bits, or

$$E_{sp}(s_{j_1}, s_{j_n}, b_c) = \min \{p_e(s_{j_1}, s_{j_2}, \dots, s_{j_n}) \cdot b_c\}. \quad (4.11)$$

4.3.3 Selection of Next Tasking Sensors and Leader Sensor

In the following, we denote $G(k)$ as the set of tasking sensors and $s_L(k)$ as the tracking leader at time step t_k . The tasking sensor selection procedure is based on the detection probability and the joint detection probability of sensors, as described in (4.1)-(4.4). A sensor with the highest detection probability is selected as the first member of $G(k+1)$. The rest members of the group are heuristically selected to achieve highest joint detection probability. The sensor selection procedure is terminated when either one of the following two conditions is satisfied:

1. $P_{s_{j_1}, s_{j_2}, \dots, s_{j_m}} \geq \theta_d$, i.e., the joint detection probability is larger than a given threshold θ_d .
2. $|P_{s_{j_1}, s_{j_2}, \dots, s_{j_{m+1}}} - P_{s_{j_1}, s_{j_2}, \dots, s_{j_m}}| \leq \theta_0$, i.e., the joint detection probability does not increase significantly while any new sensor is introduced into the tasking group, here θ_0 is a predefined value.

In order to save the energy consumption, an energy cost measurement function is used for selection of leader sensor in the tasking group $G(k+1)$. Given $s_{j_m} \in G(k+1)$, if s_{j_h} is the tracking leader, the energy cost measurement function associated with s_{j_h} is given by

$$f_d(s_{j_h}) = \sum_{s_{j_m} \in G(k+1) - \{s_{j_h}\}} E_{sp}(s_{j_m}, s_{j_h}, b_m), \quad (4.12)$$

where b_m is the number of bits of the measurement of a tasking sensor. The tracking leader at time step t_{k+1} is selected according to

$$s_L(k+1) = \arg_{s_h} \min(f_d(s_{j_h})), \quad (4.13)$$

with the fusion energy as $E_{fusion}(s_L(k+1)) = f_d(s_L(k+1))$.

It is assumed that the current tracking leader sends a unicast control message to each selected next tasking sensor for activation, the total energy consumption for this activation operation is

$$E_{act}(s_L(k), G(k+1)) = E_{sp}(s_L(k), s_L(k+1), b_{est}) + \sum_{s_j \in G(k+1)} E_{sp}(s_L(k), s_j, b_{ctrl}), \quad (4.14)$$

where b_{ctrl} is the number of bits in the control message which contains the sensor ID of the tasking sensor and its assigned sensing time, and b_{est} is the number of bits in the packet transmitted to the new tracking leader which contains the current estimation.

4.3.4 Simulation Results

We apply our proposed scheme to track a target using a network of ranging sensors. The measurement model for sensor s_j is

$$z_j(k) = \|(x(k), y(k)) - (x_{s_j}, y_{s_j})\| + v_j(k), \quad (4.15)$$

where $(x(k), y(k))$ is the location of the target at the time step t_k , (x_{s_j}, y_{s_j}) is the known position of sensor s_j , and $v_j(k)$ is the zero-mean Gaussian measurement noise of sensor s_j with variance σ_j^2 . We assume that all the sensors in this sensor network have the same characteristics, i.e., the same sensing radius r_s , the same

communication radius R_c and the same measurement noise covariance σ_j^2 . In the simulation, we assume $r_s = 5m$, $R_c = 2r_s$, and $\sigma_j^2 = 0.001$. The query area is $100m \times 100m$ with the coordinate from $(0, 0)$ at the bottom left to $(100, 100)$ at the upper right. This area is covered by 196 grid placed sensors. The desired sensor detection threshold is assumed to be $\theta_d = 0.95$. The sampling interval is $T = 0.2s$. For the energy model, $e_t = e_r = e_s = 5 \times 10^{-8}$ in J/bit, and $e_d = 1.0 \times 10^{-9}$ in J/bit-m² are used. In this simulation, we also assume $b_s = 16$ and $b_{ctrl} = b_{est} = b_m = 1024$.

We compare the performance of our proposed scheme with the single-hop multi-sensor scheduling scheme ($t = 0.2s$). The target is moving at a constant velocity of $v = 10$ m/s starting from the base point $(89.1, 76.2)$, to the destination point $(45.6, 1.9)$. Fig. 4.10 illustrates the tracking performance (defined to be the trace of the state error covariance) in both schemes, from which we can find that the proposed multi-hop scheme can achieve better tracking accuracy (smaller error) and longer tracking time. Fig. 4.11 demonstrates the accumulated energy consumption in both schemes. From this figure, it clearly shows that our proposed algorithm can significantly reduce the energy consumption in the target tracking scenario. Take the accumulated energy consumption at $t = 6.2$ s for comparison, the scheme with single-hop configuration used up to 2.7mJ energy, while our proposed scheme only used about 0.71mJ, which saves about 73.7% energy. We also compared the tracking sustaining time performance with respect to different target speeds for these two schemes. The result is shown in Fig. 4.12, which shows that our proposed scheme achieves much better tracking sustaining distance with respect to desired tracking distance (starting from the base to the destination).

Figure 4.10: Tracking accuracy achieved in both schemes

Figure 4.11: Accumulated energy consumptions in both schemes

Figure 4.12: Tracking sustaining distance under different target speeds

4.4 Multi-Sensor Scheduling with Adaptive Sampling Interval

The multi-sensor scheduling with adaptive sampling interval is formulated as an optimization problem and solved by a sequential three-step heuristic algorithm to determine the sampling interval, tasking sensors and tracking leader. Due to novel strategies for adaptive calculation of sampling intervals, compared with existing works such as VigilNet [32], our proposed scheme can achieve much better trade-off between tracking accuracy and energy consumption.

4.4.1 Problem Description

Similar to the multi-sensor scheduling scheme with fixed sampling interval described in Section 4.2, at the k^{th} time step, a group of sensor nodes $G(k)$ are activated to

perform sensing tasks, with one of them as the leader $s_L(k)$. Our goal is to jointly select, for each time step, sensors and the time instant for them to perform the sensing tasks, such that the tracking accuracy is satisfied and the network consumes less energy. The sampling interval significantly affects tracking accuracy and the energy efficiency of the network. For example, with a short sampling interval, the target can be measured more frequently but may lead to high energy consumption. While with a long sampling interval, more sensors may be required to reduce the uncertainty of the state estimation of the target, and the communication cost will also be high.

4.4.2 Sensor Detection Model

When the sampling interval is varying, the predicted target location is given as

$$\mu(\Delta t_k) = \begin{bmatrix} \hat{x}(k+1|k) \\ \hat{y}(k+1|k) \end{bmatrix} = \begin{bmatrix} \hat{x}(k|k) + \Delta t_k \cdot \hat{x}_v(k|k) \\ \hat{y}(k|k) + \Delta t_k \cdot \hat{y}_v(k|k) \end{bmatrix}, \quad (4.16)$$

with the associated predicted position error covariance

$$\Sigma(\Delta t_k) = \begin{bmatrix} \sigma_{11} + 2\Delta t_k \sigma_{12} + \Delta t_k^2 \sigma_{22} & \sigma_{13} + \Delta t_k (\sigma_{14} + \sigma_{23}) + \Delta t_k^2 \sigma_{24} \\ \sigma_{13} + \Delta t_k (\sigma_{14} + \sigma_{23}) + \Delta t_k^2 \sigma_{24} & \sigma_{33} + 2\Delta t_k \sigma_{34} + \Delta t_k^2 \sigma_{44} \end{bmatrix}, \quad (4.17)$$

where σ_{ij} is the element of the filtering error covariance matrix $P(k|k)$. Different from (4.2), the PDF of the target located at $u = (x, y)$ will be modified as

$$p(u, \Delta t_k) = \frac{1}{2\pi |\Sigma(\Delta t_k)|^{1/2}} \exp\left(-\frac{1}{2}(u - \mu(\Delta t_k))' \Sigma^{-1}(\Delta t_k)(u - \mu(\Delta t_k))\right). \quad (4.18)$$

Hence, the predicted detection probability of a sensor s_j is given as

$$P_{s_j} = \iint p(u, \Delta t_k) p_{s_j}(u) dx dy. \quad (4.19)$$

For a target located at $u = (x, y)$ covered by m sensors $s_{j_1}, s_{j_2}, \dots, s_{j_m}$, its detection probability is given by

$$p_{s_{j_1}, s_{j_2}, \dots, s_{j_m}}(u) = 1 - \prod_{j=j_1}^{j_m} (1 - p_{s_j}(u)), \quad (4.20)$$

and by considering the PDF of the target located at $u = (x, y)$, the joint detection probability for these m tasking sensors is given by

$$P_{s_{j_1}, s_{j_2}, \dots, s_{j_m}} = \iint p_{s_{j_1}, s_{j_2}, \dots, s_{j_m}}(u) \cdot p(u, \Delta t_k) dx dy. \quad (4.21)$$

4.4.3 Mathematical Formulation

Based on the sensor detection probability model and the energy model given above, the sensor scheduling problem can be formulated as follows: at the k^{th} time step, based on the updated target state $\hat{x}(k|k)$ and its error covariance $P(k|k)$ which represents the accuracy of the tracking, the sampling interval Δt_k , tasking sensors and leader sensor are determined such that

$$\begin{aligned} \min \quad & \left(\frac{E_{b,k+1} + E_{s,k+1} + E_{c,k+1}}{\Delta t_k} \right) \\ \text{subject to:} \quad & \text{tr}(\Sigma(\Delta t_k)) \leq \Phi_0, \\ & P_{s_{j_1}, s_{j_2}, \dots, s_{j_m}} \geq \theta_d \end{aligned}$$

where $E_{b,k+1}$ is the predicted broadcasting energy consumption of the leader $s_L(k)$ and the energy cost at the receiving tasking nodes, $E_{s,k+1}$ is the predicted energy

consumption for sensing at the $(k+1)^{th}$ time step, $E_{c,k+1}$ is the predicted communication energy consumption while members of the tasking group $G(k+1)$ transmit innovations to the new leader $s_L(k+1)$, $\Sigma(\Delta t_k)$ is the predicted position error covariance matrix as defined by (4.17), $\text{tr}(A)$ means the trace of the matrix A , Φ_0 is the user designed accuracy threshold, $P_{s_{j_1}, s_{j_2}, \dots, s_{j_m}}$ is the joint detection probability of the selected tasking sensors $s_{j_1}, s_{j_2}, \dots, s_{j_m}$, and θ_d is the threshold of target detection probability determined by users.

More precisely, the energy consumption for the above three different types of activities can be expressed as follows: Assuming the current leader at the k^{th} time step is sensor $s_L(k)$, from $s_L(k)$'s point of view, the furthest sensor in the tasking group $G(k+1)$ is $s_f = \arg_{s_j \in G(k+1)} \max(\mathbf{d}(s_j, s_L(k)))$, then

$$E_{b,k+1} = E_{c1}(s_L(k), s_f) + \sum_i m_i^{(k+1)} \cdot E_r(s_i), \quad (4.22)$$

$$E_{s,k+1} = \sum_i m_i^{(k+1)} \cdot E_s, \quad (4.23)$$

$$E_{c,k+1} = \sum_{i,j} m_i^{(k+1)} \cdot (1 - l_j^{(k+1)}) \cdot E_{c2}(s_i, s_j), \quad (4.24)$$

where $\mathbf{d}(\cdot)$ is the Euclidean distance, $G(k+1)$ is the set of tasking sensors at the $(k+1)^{th}$ time step, $E_{c1}(\cdot, \cdot)$ is the communication energy cost when the previous leader at the k^{th} time step broadcasts the estimates $\hat{x}(k+1|k)$ and $P(k+1|k)$, $E_{c2}(\cdot, \cdot)$ is the communication energy between two sensors within the tasking group $G(k+1)$ (for both transmission and receiving), $m_j^{(k+1)}$ is the membership indicator which takes value of 1 or 0 ($m_j^{(k+1)} = 1$ means that sensor s_j is selected to perform the measurement at the $(k+1)^{th}$ time step, while $m_j^{(k+1)} = 0$ means not), $l_j^{(k+1)}$ is the leadership indicator which takes value of 1 or 0 ($l_j^{(k+1)} = 1$ means that sensor s_j is selected to be the local fusion center, the leader, at the $(k+1)^{th}$ time step, while $l_j^{(k+1)} = 0$ means not), and E_s is the sensing energy cost for a single sensor.

Due to the complexity in computing the detection probability and the coupling effect of the sampling interval and the tasking sensors, a three-stage sub-optimal algorithm is designed to approximate this optimization problem.

At the first step, based on the predicted state error covariance $P(k+1|k)$, the maximum sampling interval Δt_k is obtained to satisfy the given prediction accuracy specification Φ_0 . The corresponding optimization problem is

$$\begin{aligned} \max \quad & \Delta t_k \\ \text{subject to:} \quad & \text{tr}(\Sigma(\Delta t_k)) \leq \Phi_0 \end{aligned}$$

where $\Sigma(\Delta t_k)$ is defined in (4.17).

At the second step, a number of sensors are selected based on the determined sampling interval to form a temporary tasking group to achieve the target detection probability threshold. This decomposed problem can be expressed in the following mathematical form:

$$\begin{aligned} \min \quad & \left(\sum_j m_j^{(k+1)} \cdot E_s \right) \\ \text{subject to:} \quad & \iint (1 - \prod_{m_j^{(k+1)}=1} (1 - p_{s_j}(u))) \cdot p(u, \Delta t_k) dx dy \geq \theta_d \end{aligned}$$

where E_s is the sensing energy cost, $m_j^{(k+1)}$ is the membership indicator which takes value of either 1 or 0, $p_{s_j}(u)$ is the sensor detection probability at point $u = (x, y)$ as defined in (4.1), and $p(u, \Delta t_k)$ is the PDF of the target location at point $u = (x, y)$ as defined in (4.18). As E_s is a constant, the above optimization problem can be

reformulated as

$$\begin{aligned} \min \quad & \sum_j m_j^{(k+1)} \\ \text{subject to:} \quad & \iint (1 - \prod_{m_j^{(k+1)}=1} (1 - p_{s_j}(u))) \cdot p(u, \Delta t_k) dx dy \geq \theta_d \end{aligned}$$

which means to select the minimal number of tasking sensors subject to the joint detection probability constraint.

At the final step, a sensor from the selected tasking group is designated as the leader sensor at the $(k+1)^{th}$ time step, based on a *Leader Energy Measure* (LEM) function $f_L(\cdot)$, where $f_L(s_j)$ is the predicted total energy consumption used for all the selected tasking sensors to transmit their innovations (each with b_c bits) to sensor s_j if s_j is designated as the leader. Mathematically,

$$\begin{aligned} f_L(s_j) &= \sum_{s_i \in G(k+1) - \{s_j\}} E_c(s_i, s_j) \\ &= \sum_{s_i \in G(k+1) - \{s_j\}} ((e_t + e_d r_{ij}^{\alpha_c} + e_r) \cdot b_c), \end{aligned} \quad (4.25)$$

where $G(k+1)$ is the group of tasking sensors at the $(k+1)^{th}$ time step. Hence, the mathematical formulation for the leader selection is

$$s_L(k+1) = \arg_{s_j} \min \{f_L(s_j) : s_j \in G(k+1)\}. \quad (4.26)$$

4.4.4 Deployment Density

Due to the uncertainty in sensor detections, sensors need to be densely deployed into a field to ensure a high probability of successful detection. It is important to determine the minimum deployment density of sensors in a certain area, which is

defined as the number of sensors in a unit area.

If a point $u = (x, y)$ is inside the sensing region of m sensors, then the probability that the target located at $u = (x, y)$ can be successfully detected by at least one of these sensors is

$$p_{s_{j_1}, s_{j_2}, \dots, s_{j_m}}(u) = 1 - (1 - \alpha)^m.$$

Hence if $p_{s_{j_1}, s_{j_2}, \dots, s_{j_m}}(u) \geq \theta_d$, then the target can be detected with the detection probability larger or equal to θ_d . Thus, by setting $p_{s_{j_1}, s_{j_2}, \dots, s_{j_m}}(u) \geq \theta_d$, we obtain the required number of sensors as

$$m \geq \frac{-\log(1 - \theta_d)}{-\log(1 - \alpha)}. \quad (4.27)$$

For example, if $\alpha = 0.9$, $\theta_d = 0.99$, then we have $m \geq 2$; if $\alpha = 0.78$, $\theta_d = 0.99$, then we have $m \geq 3$. Thus in order to ensure the successful detection of the target, we can let the sensor deployment density ρ satisfy

$$\rho \geq m/R_d, \quad (4.28)$$

where R_d is the area of the detection region. For example, the area of a circular detection region is $R_d = \pi r_s^2$, where r_s is the detection radius.

4.4.5 Selection of Sampling Interval

In many existing works, the sampling interval Δt_k is uniform for each time step [26], [28], [54], which is not power efficient and not adaptive to the changing motion of the target. In [29], the sampling interval is adaptively determined in the interval $[T_{min}, T_{max}]$. Based on the predicted covariance matrix of the measurement update and in view of the difficulty in calculating the optimal sampling interval, a dis-

crete search algorithm is adopted by discretizing the interval $[T_{min}, T_{max}]$ into finite sub-intervals in [29]. Different from the above work, the determination of optimal sampling interval in our proposed scheme does not require pre-defined sub-intervals.

Following the representations in (4.16) and (4.17), the predicted position uncertainty is given by

$$\begin{aligned}\Phi(k+1|k) &= \text{tr}(\Sigma(\Delta t_k)) \\ &= (\sigma_{11} + \sigma_{33}) + (2\sigma_{12} + 2\sigma_{34})\Delta t_k + (\sigma_{22} + \sigma_{44})\Delta t_k^2 + \frac{2}{3}q\Delta t_k^3 \\ &= \Phi(k|k) + (2\sigma_{12} + 2\sigma_{34})\Delta t_k + (\sigma_{22} + \sigma_{44})\Delta t_k^2 + \frac{2}{3}q\Delta t_k^3, \quad (4.29)\end{aligned}$$

where $\Phi(k|k) = \sigma_{11} + \sigma_{33}$. Thus, the maximum sampling interval can be solved from the equation $\Phi(k+1|k) = \Phi_0$.

The next result shows that the existence of a positive real root is guaranteed if the filtered state error covariance matrix $P(k|k) > 0$, the covariance matrix of the process noise $Q(k) > 0$, and $\Phi_0 > \Phi(k|k) > 0$.

Theorem 4.1. *There is one and only one positive real root for $\Phi(k+1|k) = \Phi_0$, if the filtered state error covariance matrix $P(k|k) > 0$, the covariance matrix of the process noise $Q(k) > 0$, and $\Phi_0 > \Phi(k|k) > 0$.*

Proof. Let $f(\Delta t_k) = \Phi(k+1|k)$, i.e.,

$$f(T) = \Phi(k|k) + (2\sigma_{12} + 2\sigma_{34})T + (\sigma_{22} + \sigma_{44})T^2 + \frac{2}{3}qT^3.$$

Clearly, $f(0) = \Phi(k|k)$, thus $0 < f(0) < \Phi_0$. Further,

$$f'(T) = (2\sigma_{12} + 2\sigma_{34}) + 2(\sigma_{22} + \sigma_{44})T + 2qT^2.$$

Since $\sigma_{22} + \sigma_{44} \geq 0$, $f'(T)$ is a monotonic function of T . Thus there exists a T_0 such that $f'(T_0) > 0$, $\forall T \geq T_0$. Hence there exists $T_1 > T_0$ to satisfy the equation $f(T_1) = \Phi_0$. For any other T_2 , if $f(T_2) = \Phi_0$, due to $f(0) < \Phi_0$ and the monotonic property of $f'(T)$, we have $T_2 = T_1$. $\square$

Based on the solution to $\Phi(k+1|k) = \Phi_0$, we use two operation modes similar to the chapter of adaptive single sensor scheduling:

1. Fast tracking mode which is operated when the position uncertainty $\Phi(k|k) > \Phi_0$, i.e., the position uncertainty threshold cannot be satisfied. This often happens at the beginning of the tracking or when there is a large target maneuvering. In this case, the sampling interval at the next time step is chosen to be T_{min} which acts as the fastest possible sampling rate.
2. Tracking maintenance mode which is operated when $\Phi(k+1|k) < \Phi_0$. Denote the solution to $\Phi(k+1|k) = \Phi_0$ by T_1 , then the sampling interval is selected as follows:

$$T_{opt} = \begin{cases} T_{min}, & T_1 \leq T_{min} \\ \min\{T_1, T_{max}\}, & T_1 > T_{min} \end{cases}$$

4.4.6 Selection of Tasking Sensors

After the sampling interval for the next time step is determined, tasking sensors will be selected. We assume that a sensor node keeps necessary information about sensing characteristics of its neighboring nodes, such as their locations and their detection regions.

According to (4.19), the current leader $s_L(k)$ calculates the predicted detection probability of each neighbor sensor, based on the predicted target location and

associated covariance matrix. Denoting the set of neighbor sensors as S_C , then the sensor with the maximum predicted detection probability at the next time step is selected as the first member of the tasking group at time step $k + 1$, i.e.,

$$g_1^{k+1} = \arg_{s_j} \max \{P_{s_j} : s_j \in S_C\}. \quad (4.30)$$

The subscript 1 of g_1^{k+1} means the first member of $G(k + 1)$, and its superscript $(k + 1)$ means the $(k + 1)^{th}$ time step. Based on the joint detection probability as defined in (4.20) and (4.21), the second member sensor of this tasking group is selected according to

$$g_2^{k+1} = \arg_{s_j} \max \left\{ P_{g_1^{k+1}, s_j} : s_j \in S_C \setminus \{g_1^{k+1}\} \right\}, \quad (4.31)$$

where $P_{g_1^{k+1}, s_j}$ is the joint detection probability of sensors g_1^{k+1} and s_j , and $S_C \setminus \{g_1^{k+1}\}$ is the set of all members of S_C excluding g_1^{k+1} . The rest member sensors of this tasking group are selected in a similar way, for example, the i^{th} tasking sensor is selected as

$$g_i^{k+1} = \arg_{s_j} \max \left\{ P_{g_1^{k+1}, g_2^{k+1}, \dots, g_{i-1}^{k+1}, s_j} : s_j \in S_C \setminus \bigcup_{y=1}^{i-1} \{g_y^{k+1}\} \right\}. \quad (4.32)$$

Based on (4.21), the selection is terminated if the joint detection probability is larger than or equal to the given threshold θ_d , i.e., if $P_{g_1^{k+1}, g_2^{k+1}, \dots, g_m^{k+1}} \geq \theta_d$, then the tasking sensor selection is terminated and the tasking group for the next time step $k + 1$ is formed as

$$G(k + 1) = \{g_i^{k+1} : i = 1, 2, \dots, m\}. \quad (4.33)$$

4.4.7 Selection of the Next Leader

The selection of the next leader is based on the Leader Energy Measure function, as shown in (4.26). The current leader $s_L(k)$ calculates the LEM for each selected tasking sensor and selects the one with the smallest LEM as the next leader $s_L(k+1)$.

4.4.8 Simulation Setup

The proposed adaptive multi-sensor scheduling scheme is used in a target tracking scenario. The network is formed using randomly distributed ranging sensors as shown in Fig. 4.13, where “*” stands for the location of a sensor node. The detection region of a sensor is assumed to be the circular area surrounding it, which is not shown in the figure. The required sensor deployment density is determined according to (4.28). The monitoring area U is $100m \times 100m$ with coordinates from $(0, 0)$ to $(100, 100)$. The measurement model for sensor s_j is given by

$$z_j(k) = \|(x(k), y(k)) - (x_s(j), y_s(j))\| + v_j(k), \quad (4.34)$$

where $(x(k), y(k))$ is the location of the target to be estimated at the k^{th} time step, $(x_s(j), y_s(j))$ is the known position of sensor s_j , $\|(x(k), y(k)) - (x_s(j), y_s(j))\|$ is the Euclidean distance between the target and sensor s_j , and $v_j(k)$ is the zero-mean Gaussian measurement noise of sensor s_j with variance σ_j^2 .

We assume that all the sensors in the network have the same sensing parameters, i.e., the same detection probability α , the same sensing radius r_s and the same measurement noise covariance σ_j^2 with $\alpha = 0.8$, $r_s = 15m$, and $\sigma_j^2 = 0.001$. The desired sensor detection threshold is assumed to be $\theta_d = 0.9$. We use $N = 80$ sensors for simulation so that the required sensor deployment density is satisfied.

The target travels with a constant speed $v = 12m/s$ along an ‘S’ shape trajectory as shown in Fig. 4.13. The intensity of the process noise is $q = 50$. When the target enters the sensor network, a sensor node next to this point detects the target by its PIR sensor, and then initializes the target tracking process with its position as the target initial estimation of the position, the initial covariance matrix is assumed to be $15I$, where I is the identity matrix. The number of samples used in Monte Carlo method is 100. The tracking accuracy threshold is $\Phi_0 = 5$. For the energy model, the following parameters are used: $\alpha_c = 2$, $e_t = 45 \times 10^{-9}$ J/bit, $e_r = 135 \times 10^{-9}$ J/bit, and $e_d = 10 \times 10^{-9}$ J/bit-m². The sensing energy cost is $E_s = 8 \times 10^{-7}$ J. For simplicity, the number of bits for inter-sensor communication is assumed to be $b_c = 264$, and the number of bits used in local processing is assumed to be the same as $b_p = 16$. The sampling interval is bounded within $[T_{min}, T_{max}] = [0.1s, 0.5s]$.

4.4.9 Simulation Results

In this simulation, the target travels in the WSN with a curved trajectory for 20 seconds. It is considered as a successful tracking if the sensor network is able to track the target throughout the 20 seconds period. If the target can not be detected by any scheduled sensor nodes for 3 successive time steps, then the tracking algorithm will be terminated and it is considered as a failed tracking.

Fig. 4.13 shows a successful tracking process of the proposed algorithm. The estimated trajectory lies very closely to the true trajectory of the target. The tracking error, which is defined as the distance between the estimated target position and the true position of the target, with respect to the given tracking error threshold, is shown in Fig. 4.14. The tracking error is large at the beginning of tracking; hence the sampling intervals should be small at these time steps. The sampling interval and number of sensors used at each time step are shown in Fig. 4.15 and

Fig. 4.16 respectively. As shown in Fig. 4.15, the sampling interval is equal to the minimum value ($0.1s$) at the beginning of tracking. This means that the network operates in the fast tracking mode. While operating in the tracking maintenance mode, the sampling interval can be as high as $0.45s$, and hence the power efficiency is improved. In this particular run of simulation, the average sampling interval is $0.324s$. At the first time step, 3 sensor nodes are activated to initialize the tracking. The average number of sensor used is 2.355 per time step.

As described in (2.3), the intensity of the process noise q can be adjusted to represent the uncertainty in the target motion model. The accuracy of the estimation of the intensity of the process noise q affects the tracking performance of the proposed algorithm. To illustrate the robustness of the proposed algorithm, the performance of the algorithm for inaccurate estimation of the intensity of the process noise q is shown in Fig. 4.17 and Fig. 4.18.

In Fig. 4.17, the dot line shows the sampling intervals at each time step when the estimation of the intensity of the process noise q is accurate ($q = 50$, denoted as ‘Accurate q ’), which is actually the curve shown in Fig. 4.15. The solid line shows the case when the estimation of the intensity of the process noise q is inaccurate (denoted as ‘Inaccurate q ’), i.e., the real value is 75, i.e. there exists a 50% deviation from the estimated value $q = 50$. The longest sampling interval for the inaccurate q case is only $0.41s$ and the average sampling interval is about $0.306s$, which are both less than their corresponding ones for the accurate q case, $0.45s$ and $0.333s$, respectively. Although the performance for the inaccurate q is not as good as the one for the accurate q , the proposed algorithm is still able to track the target for $20s$, i.e., complete the tracking task. The corresponding tracking errors with respect to the threshold are shown in Fig. 4.18 which clearly shows that the proposed algorithm performs well for the inaccurate estimation of the intensity of the process noise q .

Figure 4.13: Estimated and real target trajectory in a random distributed WSN.

The performance of the proposed adaptive multi-sensor scheduling scheme is compared with the multi/single-sensor scheduling scheme with uniform sampling interval, the VigilNet method [32] with uniform sampling interval, and the adaptive single-sensor scheduling scheme. The uniform sampling interval is set to be $0.1s$, $0.2s$, $0.3s$, $0.4s$, and $0.5s$ for the multi-sensor uniform sampling interval scheme. For the single sensor case and VigilNet project, the uniform sampling interval is set to be $0.1s$ which is the fastest sampling rate.

Fig. 4.19 - Fig. 4.21 show the tracking performance when the VigilNet project method is adopted, which estimates the target position by averaging the sensors' own locations. The sampling interval is set at $0.1s$ and passive detection mode is used for the sensors, which means a sensor is triggered if the target is within the sensor's detection area. Although the sampling interval is the fastest, the tracking

Figure 4.14: Evolution of the traces of the position uncertainty

error still exceeds the required threshold, as shown in Fig. 4.20. What is more, the tracking error is fluctuating and does not converge with time. The number of sensors used at each time step is highly determined by the density of sensors, shown in Fig. 4.21. The average number of sensor used for this particular simulation is 5.307 per time step.

Fig. 4.22 and Fig. 4.23 show the comparison results in terms of their accumulated

	[0s, 5s)	[5s, 10s)	[10s, 15s)	[15s, 20s)	Success (20s)
$\Delta t = 0.1s$	0%	0%	1%	0%	99%
$\Delta t = 0.2s$	1%	1%	2%	1%	95%
$\Delta t = 0.3s$	0%	1%	6%	3%	90%
$\Delta t = 0.4s$	1%	1%	7%	3%	88%
$\Delta t = 0.5s$	0%	1%	10%	4%	85%
Adaptive	0%	0%	1%	1%	98%

Table 4.1: Tracking sustaining time using multi-sensor scheduling schemes

Figure 4.15: Sampling interval varies at each time step

energy consumption. The performance of single sensor scheduling schemes is shown in Fig. 4.22. From this figure, we can see that the energy consumption is quite small when the adaptive single sensor scheduling scheme is adopted. However, both tracking schemes failed before 8s, due to the bad detection probability of a single sensor.

The performance of multi-sensor scheduling scheme with uniform and adaptive sampling interval is shown in Fig. 4.23. It is shown that only the schemes with adaptive sampling interval and short uniform sampling intervals ($\Delta t = 0.1s$ and $\Delta t = 0.2s$) can successfully track the target, while the other schemes failed to track (the scheme with $\Delta t = 0.3s$ failed at 17.7s, the scheme with $\Delta t = 0.4s$ failed at 14.4s, and the scheme with $\Delta t = 0.5s$ failed at 12s). Taking the successful tracking schemes for comparison, the accumulated energy consumption of the multi-sensor scheme with

Figure 4.16: Number of sensors used at each time step

$\Delta t = 0.1s$ (or with $\Delta t = 0.2s$) is $24.61mJ$ (or $13.45mJ$, respectively), while that for the proposed adaptive multi-sensor scheduling scheme is $8.006mJ$ only, representing a saving of 67.47% (or 40.48%, respectively) of the energy, which is a significant improvement.

Table 4.1 demonstrates the duration for tracking without losing the target in 100 runs of simulations by using multi-sensor scheduling schemes. It shows that the chance of completing the tracking task is reduced with the increase of the uniform sampling interval. The one with the sampling interval of $0.1s$ completes the tracking task for 99 times out of the 100 runs of the simulation, i.e., 99% of successful-tracking rate; the one with the sampling interval of $0.2s$ can complete the tracking task for 95 times, i.e., 95% successful-tracking rate. If the adaptive multi-sensor scheme is adopted, the tracking is successful for 98 times out of the 100 runs of simulation,

Figure 4.17: Sampling interval comparison for accurate and inaccurate q value

i.e., 98% successful-tracking rate, which is very near to the performance of the minimum uniform sampling interval case. Hence, the proposed scheme shows a superior reliability for target tracking.

4.5 Conclusion

In this chapter, a multi-sensor scheduling scheme with fixed sampling interval by selecting multiple tasking sensors at each time step according to the joint detection probability was first presented. A Monte Carlo method was used to generate random samples to approximate the random distribution of predicted target state. Furthermore a heuristic incremental sensor selection procedure has been proposed

Figure 4.18: Tracking accuracies performance with respect to the given threshold, for accurate and inaccurate q value

according to this approximation. Experimental results by simulations demonstrated that, compared to a single sensor scheduling approach, the proposed approach can significantly improve the reliability and the tracking accuracy.

A multi-hop configuration was then explored. The proposed scheme allows the tasking sensors to be selected from those sensors which are multi-hop away from the current tracking leader, just based on the prediction of the target movement. The topology of the sensor network plays a great role in this proposed scheme. Simulation results showed that, compared with a multi-sensor scheduling scheme with only single-hop configuration, the proposed algorithm can achieve better overall tracking performance, such as higher tracking accuracy, much lower energy consumption, and longer tracking sustaining distance for high speed target.

Figure 4.19: Estimated and real target trajectory in a random distributed WSN using VigilNet method.

Finally, an adaptive multi-sensor scheduling scheme was introduced. This scheme is divided into three steps. At first step, the sampling interval is determined based on a given tracking accuracy threshold. Next, a number of sensors are selected to form a temporary tasking group, based on their joint detection probability. A sensor within the tasking group is then selected as the leader according to the predicted energy consumption. A Monte Carlo method was used again to approximate the random distribution of the predicted target state by randomly generating a number of samples. Simulation results have shown that the proposed scheme reduces the energy consumption and improve the tracking reliability without degrading tracking accuracy, and it is robust to the uncertainty of the process noise.

Figure 4.20: Evolution of the traces of the position uncertainty using VigilNet method.

Figure 4.21: Number of sensors used at each time step using VigilNet method.

Figure 4.22: Energy consumptions in single sensor scheduling schemes: $\Delta t_s = 0.1s$ means an uniform sampling interval of 0.1s scheme is adopted; $\Delta t_s = \text{Adaptive}$ means the adaptive sampling interval scheme is adopted.

Figure 4.23: Energy consumption in the multi-sensor scheduling schemes with uniform sampling intervals of 0.1s, 0.2s, 0.3s, 0.4s, and 0.5s cases and the proposed adaptive multi-sensor scheduling scheme

Chapter 5

Quantized Kalman Filter

5.1 Introduction

A wireless sensor network generally consists of one or several fusion centers and a large number of sensors, which are spatially distributed to monitor activities of interest. Each sensor makes an observation of the signal and sends it to the fusion center. Due to a bandwidth constraint each sensor is only able to transmit a finite number of bits. One of the main objectives of current sensor network research is to design energy-efficient devices and algorithms to support all aspects of network operations.

Data to be quantized in most existing literature are the sensors observations. Larger quantization errors will be generated under the limited bandwidth if the values of the observations are larger, which brings larger information loss and further leads to the lower estimation accuracy. In this chapter, the quantized Kalman filter for a dynamic stochastic variable is studied in a sensor network. The quantized Kalman filter based on the quantized observations (QKFQO) is first presented, which is equivalent to the Kalman filter for the system with the observation noises including

the observation noises of the original system and quantization noises. Approximate solutions of two optimization problems for bandwidth scheduling are given, where one is to minimize the total energy consumption such that the total weighted sum of the upper bounds of variances of the quantization noises meets a threshold, the other is to minimize the total weighted sum of the upper bounds of variances of the quantization noises such that the total bandwidth satisfies a given threshold.

Quantizing observations, however, will bring large quantization errors if values of observations are large. Another quantized Kalman filter based on the quantized innovations (QKFQI) is developed subsequently. Similar to the SOI method [44], which severely quantizes the innovations where only a single bit is used to represent the quantization output, the proposed QKFQI also quantizes the innovations. Instead of using only one bit, the proposed QKFQI is trying to allocate the optimal number of bits for each sensor while achieving better balance between energy consumption and estimation accuracy. Compared with QKFQO, it is better in saving power and reducing information loss since the values of the innovations are usually smaller than those of the observations.

5.2 Problem Formulation

In this section we consider the state estimation problem by a sensor network consisting of N sensors as described in Section 2.4.1 and Section 2.4.2, where a fixed sampling interval model is used, i.e., the sampling interval $\Delta t_k = t_{k+1} - t_k$ is the same for each time step, which can be denoted as $\Delta t_k = T$. Our goal is to find the quantized Kalman filters under the limited bandwidth constraints. We make the following assumptions.

Assumption 5.1. $w(k) \in \mathbb{R}^n$ and $v_j(k) \in \mathbb{R}$, $j = 1, 2, \dots, N$ are uncorrelated white

noises with zero mean and variances $Q(k)$ and $\sigma_{v_j}^2(k)$. The initial value $x(0)$ with mean μ_0 and variance P_0 is independent of $w(k)$ and $v_j(k)$.

Assumption 5.2. The sensor measurement $z_j(k) \in [\underline{U}_j, \bar{U}_j]$, where $\underline{U}_j$ and $\bar{U}_j$, $j = 1, 2, \dots, N$ are known constants.

The estimation problem considered is shown in Fig. 5.1 where each sensor makes an observation $z_j(k)$, and then it is quantized to produce a quantized message $m_j(k)$ that is transmitted to the fusion center, and the fusion center will combine the received message $m_j(k)$ to give a final estimate for state $x(k)$ by Kalman filtering. We assume that the fusion center knows all the parameters of system as described in Section 2.4.1 and Section 2.4.2 and channel is perfect, i.e., no bit error during the transmission from sensors to the fusion center. If there is a sufficient bandwidth to be supplied, i.e., when $m_j(k) = z_j(k)$, the standard Kalman filter can be used [49]. Otherwise, the fusion center will generate the estimator based on the received symbols $\{m_j(k), j = 1, 2, \dots, N\}$.

Figure 5.1: Decentralized estimation scheme based on quantized observations

5.3 Quantized Observations Kalman Filter

5.3.1 Quantization Strategy

We adopt the quantization strategy in [55]. We quantize $z_j(k)$ in such a way that its resulting message $m_j(k)$ has $b_j(k)$ bits, where $b_j(k)$ is to be determined later. We have $L_j(k) = 2^{b_j(k)}$ quantization levels

$$\{a_l^{(j)} \in [\underline{U}_j, \bar{U}_j], l = 1, 2, \dots, L_j(k)\}.$$

These levels are uniformly spaced so that

$$\underline{U}_j = a_1^{(j)} < a_2^{(j)} < \dots < a_{L_j}^{(j)} = \bar{U}_j$$

and

$$a_{l+1}^{(j)} - a_l^{(j)} = \Delta_j(k) = (\bar{U}_j - \underline{U}_j)/(L_j(k) - 1)$$

for every time step k . Quantization is done in the following probabilistic manner. Suppose $z_j(k) \in [a_l^{(j)}, a_{l+1}^{(j)})$, and then $z_j(k)$ is quantized to $m_j(k)$ according to $P\{m_j(k) = a_l^{(j)}\} = 1 - p_j(k)$ and $P\{m_j(k) = a_{l+1}^{(j)}\} = p_j(k)$ with $p_j(k) = (z_j(k) - a_l^{(j)})/\Delta_j(k)$, $0 \leq p_j(k) \leq 1$. We readily verify that the quantized observation $m_j(k)$ is the unbiased estimation of observation $z_j(k)$. The quantization noise $n_j(k) = m_j(k) - z_j(k)$ is a Bernoulli random variable taking values at $-p_j(k)\Delta_j(k)$ and $(1 - p_j(k))\Delta_j(k)$, i.e.

$$P\{n_j(k) = -p_j(k)\Delta_j(k)\} = 1 - p_j(k)$$

and

$$P\{n_j(k) = (1 - p_j(k))\Delta_j(k)\} = p_j(k).$$

Variance of the quantization noise $n_j(k)$ is computed as

$$\sigma_{n_j}^2(k) = \Delta_j^2(k)p_j(k)(1 - p_j(k)) \leq \delta_j^2(k),$$

where

$$\delta_j^2(k) = (\bar{U}_j - \underline{U}_j)^2 / [4(2^{b_j(k)} - 1)^2].$$

Furthermore, $n_j(k)$, $v_j(k)$, $j = 1, 2, \dots, N$ and $w(k)$ are uncorrelated with each other.

Then we have the equivalent system in the fusion center

$$\begin{aligned} x(k+1) &= F(k)x(k) + w(k), \\ m(k) &= h(k)x(k) + \eta(k), \end{aligned} \tag{5.1}$$

where the augmented quantized observation $m(k) = [m_1(k), m_2(k), \dots, m_N(k)]'$, the augmented observation matrix $h(k) = [h'_1(k), h'_2(k), \dots, h'_N(k)]'$ and the augmented observation noise $\eta(k) = [\eta_1(k), \eta_2(k), \dots, \eta_N(k)]'$ with zero mean and variance matrix $Q_\eta(k) = \text{diag}(\sigma_{\eta_1}^2(k), \sigma_{\eta_2}^2(k), \dots, \sigma_{\eta_N}^2(k))$ where $\eta_j(k) = v_j(k) + n_j(k)$, $j = 1, 2, \dots, N$ are uncorrelated with each other with zero mean and the variance $\sigma_{\eta_j}^2(k) = \sigma_{v_j}^2(k) + \sigma_{n_j}^2(k) \leq \sigma_{v_j}^2(k) + \delta_j^2(k)$. We approximately consider the observation noise $\eta(k)$ to be a white noise. Then, the Kalman filtering can be used for system (5.1) where the upper bound of variance of the quantized noise is used. So we can obtain the quantized Kalman filter, more precisely, the QKFQO.

Comparing (2.1), (2.4) and (5.1), we see that their difference is the additional quantization noise term $n_j(k)$ in the observation equation of system (5.1). The precision of the QKFQO is better if the variance of the quantization noise is smaller. We can make its upper bound small, which means that more bandwidths need to be supplied. However, the sensor power and transmission bandwidth are limited in a

sensor network. In the following, we will design the optimal bandwidth scheduling under the required performance and limited bandwidth.

5.3.2 Optimal Bandwidth Scheduling

The basic energy model used here is described in Section 2.5. In Fig. 5.1, the main energy consumption comes from two aspects. One is the transmitting energy by which each sensor transmits the quantized observations to the fusion center and the other is the receiving energy by which the fusion center receives the quantized observations from sensors. Here we ignore the sensing and computation energy of the sensors and the fusion center. Then the total energy consumption of the whole sensor networks at each time step is described as follows:

$$E(k) = \sum_{j=1}^N [E_t(s_j, FC) + E_r(FC)] = \sum_{j=1}^N (e_t + e_d r_j^{\alpha_c} + e_r) b_j(k), \quad (5.2)$$

where r_j is the distance between the sensor s_j and the FC, and $b_j(k)$ is an integer denoting the number of quantized bits per sample of sensor s_j at k^{th} time step.

From the above sections we know that the estimation accuracy loss of the quantized Kalman filter comes from the quantization noises. We hope that the variances of the quantization noises are small to achieve a satisfactory precision, so we consider the following optimization problem:

$$\begin{aligned} \min \quad & \sum_{j=1}^N w_j b_j(k) \\ \text{subject to:} \quad & \sum_{j=1}^N \frac{f_j(k)}{\sigma_{v_j}^2(k)} \delta_j^2(k) \leq \gamma(k) \\ & b_j(k) \geq 0, \quad j = 1, 2, \dots, N \end{aligned} \quad (5.3)$$

where $w_j = e_t + e_d r_j^{\alpha_c} + e_r$, $f_j(k) = h_j(k)h_j'(k)$, $\gamma(k) > 0$ is a given performance constraint. In (5.3), $\delta_j^2(k)$ should be smaller if the weight $f_j(k)/\sigma_{v_j}^2(k)$ is larger, which means more bits are assigned for the sensor with better accuracy, since $f_j(k)/\sigma_{v_j}^2(k)$ in fact reflects the signal-to-noise ratio. To facilitate the subsequent analysis, we will relax the integer $b_j(k)$ to take real positive values, and drop the last constraints on $b_j(k)$ as they become inactive at the optimum point. Substituting $\delta_j^2(k) = (\bar{U}_j - \underline{U}_j)^2/[4(2^{b_j(k)} - 1)^2]$ into (5.3), so we can write the Lagrangian for the optimization problem as

$$L(b_j(k), \lambda(k)) = \sum_{j=1}^N w_j b_j(k) + \lambda(k) \left(\sum_{j=1}^N \frac{f_j(k)(\bar{U}_j - \underline{U}_j)^2}{4\sigma_{v_j}^2(k)(2^{b_j(k)} - 1)^2} - \gamma(k) \right). \quad (5.4)$$

Let $\partial L/\partial b_j = 0$ for all j , we have

$$\frac{2^{b_j(k)}}{(2^{b_j(k)} - 1)^3} = \frac{2\sigma_{v_j}^2(k)w_j}{(\ln 2)f_j(k)(\bar{U}_j - \underline{U}_j)^2\lambda(k)}. \quad (5.5)$$

Also, at the optimum point, we should have

$$\sum_{j=1}^N \frac{f_j(k)(\bar{U}_j - \underline{U}_j)^2}{4\sigma_{v_j}^2(k)(2^{b_j(k)} - 1)^2} = \gamma(k). \quad (5.6)$$

Unfortunately, the optimal solution cannot be solved in closed form from the nonlinear equations (5.5) and (5.6). To obtain a closed-form solution, we slightly modify (5.5) as

$$\frac{1}{(2^{b_j(k)} - 1)^2} \approx \frac{2^{b_j(k)}}{(2^{b_j(k)} - 1)^3} = \frac{2\sigma_{v_j}^2(k)w_j}{(\ln 2)f_j(k)(\bar{U}_j - \underline{U}_j)^2\lambda(k)}, \quad (5.7)$$

so we have

$$\frac{f_j(k)(\bar{U}_j - \underline{U}_j)^2}{4\sigma_{v_j}^2(k)(2^{b_j(k)} - 1)^2} = \frac{w_j}{2(\ln 2)\lambda(k)}. \quad (5.8)$$

From (5.6) and (5.8) we can obtain the approximate solution

$$\begin{aligned} b_j^*(k) &= \log_2 \left(1 + \frac{\sqrt{(\ln 2)\lambda^*(k)f_j(k)(\bar{U}_j - \underline{U}_j)}}{\sigma_{v_j}(k)\sqrt{2w_j}} \right), \\ \lambda^*(k) &= \left(\sum_{j=1}^N w_j \right) / (2(\ln 2)\gamma(k)). \end{aligned} \quad (5.9)$$

The approximate solution of the optimization problem above should be the smallest integer greater or equal to $b_j^*(k)$.

On the other hand, the optimization problem can also be formulated such that the estimation error is minimized, subject to a bandwidth constraint. Therefore, the following optimization problem is formulated:

$$\begin{aligned} \min \quad & \sum_{j=1}^N \frac{f_j(k)}{\sigma_{v_j}^2(k)} \delta_j^2(k) \\ \text{subject to:} \quad & \sum_{j=1}^N b_j(k) \leq B \\ & b_j(k) \geq 0, \quad j = 1, 2, \dots, N, \end{aligned} \quad (5.10)$$

where B is the total bit constraint. Similarly, we can write the Lagrangian for the optimization problem (5.10) as

$$L(b_j(k), \lambda(k)) = \sum_{j=1}^N \frac{f_j(k)(\bar{U}_j - \underline{U}_j)^2}{4\sigma_{v_j}^2(k)(2^{b_j(k)} - 1)^2} + \lambda(k) \left(\sum_{j=1}^N b_j(k) - B \right). \quad (5.11)$$

Let $\partial L / \partial b_j = 0$ for all j , we have

$$\frac{2^{b_j(k)}}{(2^{b_j(k)} - 1)^3} = \frac{2\sigma_{v_j}^2(k)\lambda(k)}{(\ln 2)f_j(k)(\bar{U}_j - \underline{U}_j)^2}. \quad (5.12)$$

Also, at the optimum point, we should have

$$\sum_{j=1}^N b_j(k) = B. \quad (5.13)$$

The closed optimal solution cannot be found from the nonlinear equations (5.12) and (5.13). To obtain a closed-form solution, we slightly modify the (5.12) as

$$\frac{1}{(2^{b_j(k)})^2} \approx \frac{2^{b_j(k)}}{(2^{b_j(k)} - 1)^3} = \frac{2\sigma_{v_j}^2(k)\lambda(k)}{(\ln 2)f_j(k)(\bar{U}_j - \underline{U}_j)^2}. \quad (5.14)$$

From (5.13) and (5.14) we can obtain the approximate solutions

$$\begin{aligned} b_j^*(k) &= \log_2 \left(\frac{\sqrt{(\ln 2)f_j(k)(\bar{U}_j - \underline{U}_j)}}{\sigma_{v_j}(k)\sqrt{2\lambda^*(k)}} \right), \\ \lambda^*(k) &= \frac{(\ln 2) \left(\prod_{j=1}^N f_j(k)(\bar{U}_j - \underline{U}_j)^2 / \sigma_{v_j}^2(k) \right)^{1/N}}{2^{1+(2B/N)}}. \end{aligned} \quad (5.15)$$

A suboptimal solution of the optimization problem (5.10) should be the largest integer smaller or equal to $b_j^*(k)$.

In the above we have obtained approximate solutions to the optimization problems (5.3) and (5.10). The following theorem states the errors of the approximate solutions.

Theorem 5.1. *Let $b_j(k)$ be the optimal solution to the optimization problems (5.3) or (5.10) such that $b_j(k) \geq 1$ for all j , and let $b_j^*(k)$ given by (5.9) or (5.15) be the approximate solution. Then the optimal solution $b_j(k)$ to the problem (5.3) satisfies*

$$b_j^*(k) - 1/2 \leq b_j(k) \leq b_j^*(k) + 1/2. \quad (5.16)$$

The optimal solution $b_j(k)$ to the problem (5.10) satisfies

$$b_j^*(k) - 3/2 \leq b_j(k) \leq b_j^*(k) + 3/2. \quad (5.17)$$

Proof. (5.16) can be proved similarly to [55]. The following is the proof of (5.17).

For optimization problem (5.10), since

$$\frac{1}{(2^{b_j(k)})^2} \leq \frac{2^{b_j(k)}}{(2^{b_j(k)} - 1)^3} \leq \frac{8}{(2^{b_j(k)})^2}, \quad (5.18)$$

from (5.12), (5.14) and (5.18) we have

$$\frac{\sigma_{v_j}^2(k) \lambda^*(k)}{4(\ln 2) f_j(k) (\bar{U}_j - \underline{U}_j)^2} \leq \frac{1}{(2^{b_j(k)})^2} \leq \frac{16 \sigma_{v_j}^2(k) \lambda^*(k)}{(\ln 2) f_j(k) (\bar{U}_j - \underline{U}_j)^2}. \quad (5.19)$$

Substituting $\lambda^*(k)$ in terms of $b_j^*(k)$ as stated in (5.15), we have

$$\frac{1}{8(2^{b_j^*(k)})^2} \leq \frac{1}{(2^{b_j(k)})^2} \leq \frac{8}{(2^{b_j^*(k)})^2}, \quad (5.20)$$

that is

$$\frac{1}{(2^{b_j^*(k)+3/2})^2} \leq \frac{1}{(2^{b_j(k)})^2} \leq \frac{1}{(2^{b_j^*(k)-3/2})^2},$$

which yields (5.17). $\square$

From Theorem 5.1, we can obtain the following corollary.

Corollary 5.1. *If the optimal solution $b_j(k) \geq 2$ for all j , $b_j^*(k)$ computed by (5.9) is just the optimal solution to the optimization problem (5.3) and $b_j^*(k)$ computed by (5.15) may have the error of one bit compared with the optimal solution to the optimization problem (5.10). Further, if the optimal solution $b_j(k) \geq 3$ for all j , $b_j^*(k)$ computed by (5.15) is just the optimal solution to the optimization problem (5.10).*

Theorem 5.1 shows that the approximate solutions (5.9) to problem (5.3) may have the errors of one bit and the approximate solutions (5.15) to problem (5.10) may have the errors of one to three bits. Corollary 5.1 shows that the errors are reduced as the bits of the optimal solutions increase.

5.4 Quantized Innovations Kalman Filter

In Section 5.3, we investigated the quantized Kalman filter based on quantized observations. However, if the values of observation data are large, quantizing observations will bring large information loss under the limited bandwidth, which means that the variances of quantization noises are large. So the quantized Kalman filter will have a low filtering accuracy. To reduce the information loss and improve the filtering performance, we will give another quantized Kalman filter based on the quantized innovations (QKFQI). Since the values of innovation data are smaller than those of observation data, quantizing innovations will bring smaller information loss than quantizing observations under the same bandwidth constraint.

We consider the QKFQI for system described in Section 2.4.1 and Section 2.4.2 with Assumption 5.1. The estimation scheme for QKFQI is shown in Fig. 5.2. Each sensor makes an observation, and computes the innovation $\epsilon_j(k) = z_j(k) - h_j(k)\hat{x}(k|k-1)$ where $\hat{x}(k|k-1)$ is the one-step predictor of state feedback from the fusion center, then the innovation $\epsilon_j(k)$ is quantized to produce a quantized innovation $\bar{\epsilon}_j(k)$ that is transmitted to the fusion center, and the fusion center will combine the received messages $\{\bar{\epsilon}_j(k), j = 1, 2, \dots, N\}$ for the Kalman filtering of state $x(k)$.

We still adopt the quantization strategy in the last section. Differences are that we quantize $\epsilon_j(k)$ to $\bar{\epsilon}_j(k)$ with $b_j(k)$ bits in the range $[-\sigma_{\epsilon_j}(k), \sigma_{\epsilon_j}(k)]$ where $\sigma_{\epsilon_j}(k)$

Figure 5.2: Decentralized estimation scheme based on quantized innovations

is the variance of the innovation $\epsilon_j(k)$ from the fusion center. Furthermore, $\epsilon_j(k)$ is quantized to $a_{L_j(k)}^{(j)}$ if $\epsilon_j(k) \geq \sigma_{\epsilon_j}(k)$ and to $a_1^{(j)}$ if $\epsilon_j(k) \leq -\sigma_{\epsilon_j}(k)$. The upper bound of the variance of the quantization noise $\tilde{\epsilon}_j(k) = \bar{\epsilon}_j(k) - \epsilon_j(k)$ is computed as $\delta_{\tilde{\epsilon}_j}^2(k) = \sigma_{\epsilon_j}^2(k)/(2^{b_j(k)} - 1)^2$. Also $\tilde{\epsilon}_j(k)$, $v_j(k)$, $j = 1, 2, \dots, N$ and $w(k)$ are uncorrelated with each other.

QKFQI is implemented in three stages:

The first stage is implemented in the fusion center as

$$\hat{x}(k|k-1) = F(k-1)\hat{x}(k-1|k-1), \quad (5.21)$$

$$P(k|k-1) = F(k-1)P(k-1|k-1)F'(k-1) + Q(k-1), \quad (5.22)$$

$$Q_\epsilon(k) = h(k)P(k|k-1)h'(k) + Q_v(k), \quad (5.23)$$

where $Q_v(k) = \text{diag}(\sigma_{v_1}^2(k), \sigma_{v_2}^2(k), \dots, \sigma_{v_N}^2(k))$ with the initial values $\hat{x}(0|0) = \mu_0$ and $P(0|0) = P_0$. One-step predictor $\hat{x}(k|k-1)$ of state $x(k)$ and the square root of the each diagonal element $\sigma_{\epsilon_j}^2(k)$ of $Q_\epsilon(k)$ are sent to the corresponding sensor

by feedback to compute the innovation $\epsilon_j(k)$ and determine the quantized range $[-\sigma_{\epsilon_j}(k), \sigma_{\epsilon_j}(k)]$ for the second stage computation.

The second stage is implemented in each sensor as

$$\epsilon_j(k) = z_j(k) - h_j(k)\hat{x}(k|k-1) \quad (5.24)$$

$$\bar{\epsilon}_j(k) = \begin{cases} a_1^{(j)} & \text{if } \epsilon_j(k) \leq -\sigma_{\epsilon_j}(k) \\ a_l^{(j)} \text{ or } a_{l+1}^{(j)} & \text{if } -\sigma_{\epsilon_j}(k) < a_l^{(j)} \leq \epsilon_j(k) < a_{l+1}^{(j)} < \sigma_{\epsilon_j}(k) \\ a_{L_j(k)}^{(j)} & \text{if } \epsilon_j(k) \geq \sigma_{\epsilon_j}(k) \end{cases} \quad (5.25)$$

Then the quantized innovations are sent to the fusion center for the third stage computation.

The third stage is implemented in the fusion center as

$$Q_{\bar{\epsilon}}(k) = Q_{\epsilon}(k) + \delta_{\bar{\epsilon}}(k), \quad (5.26)$$

$$K(k) = P(k|k-1)h'(k)Q_{\bar{\epsilon}}^{-1}(k), \quad (5.27)$$

$$\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)\bar{\epsilon}(k), \quad (5.28)$$

$$P(k|k) = (I_n - K(k)h(k))P(k|k-1), \quad (5.29)$$

where $\delta_{\bar{\epsilon}}(k) = \text{diag}(\sigma_{\bar{\epsilon}_1}^2(k), \sigma_{\bar{\epsilon}_2}^2(k), \dots, \sigma_{\bar{\epsilon}_N}^2(k))$. The above (5.21) - (5.29) are implemented at each time step.

Likewise, we can design the optimal bandwidth scheduling under the required performance and bandwidth constraint similarly to section 5.3. For vector observation case, the corresponding QKFQI with optimization problems can also be designed.

Remark 5.1. *It is noted that more resources are required when implementing the QKFQI as compared with the QKFQO.*

There are two major differences between the QKFQI and the previous QKFQO.

One is the one-step predictor of output each sensor received from the fusion center. The extra resources needed here is mainly due to the computation of the one-step predictor of output (the fusion center's work), transmission of the computed state (the fusion center's work), and reception of the computed prediction at sensor side. The fusion center, however, is assumed of powerful computation capability with no energy constraint issue. This is a reasonable assumption since usually a fusion center is a powerful computer with solid power supply connected to the power plant. The transmission channel is also assumed to be perfect without error and sufficient bandwidth is provided. The reception energy consumption at sensor side is much less than the transmission energy consumption required at a later stage when the sensor transmits the computed innovation to the fusion center. Hence, this difference would not introduce a significant additional resources usage.

The second major difference is the additional computation at each sensor which calculates the respective innovation. However, the energy consumption for computation is much less than the transmission energy consumption, which is indeed ignored in the thesis as indicated in the first paragraph of Section 5.3.2, Chapter 5.

In conclusion, the extra resources needed for the innovation filter is not significant and will not affect much on the problem formulation.

5.5 Simulation

Here we consider a target tracking system in a sensor network. The single target is assumed to be a sound source moving with a constant velocity, starting from an arbitrary point (the base) toward another arbitrary point (the destination) in a

query area. The target motion in the Cartesian coordination system is modeled as

$$x(k+1) = F(T)x(k) + w(k), \quad (5.30)$$

where $x(k)$ is the state of the target, including x - and y -coordinates of the target and the velocities along x - and y -axes, at the k^{th} time step, $F(T)$ is the transition matrix, T is the sampling interval and $w(k)$ is the process noise with covariance matrix of $Q(k)$. $F(T)$ and $Q(k)$ are given by

$$F(T) = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, Q(k) = q \begin{bmatrix} T^3/3 & T^2/2 & 0 & 0 \\ T^2/2 & T & 0 & 0 \\ 0 & 0 & T^3/3 & T^2/2 \\ 0 & 0 & T^2/2 & T \end{bmatrix}, \quad (5.31)$$

where q is a known scalar that represents the intensity of the process noise [49].

The network is formed using acoustic amplitude sensors. They are uniformly distributed over the query area, as shown in Fig. 5.3 and Fig. 5.4, where the “*” stands for the location of a sensor node. Suppose there are N tasking sensors $s_1, s_2, \dots, s_N$ obtaining the measurements $z_1, z_2, \dots, z_N$ at the k^{th} time step. The measurement function of sensor s_j is given by

$$z_j(k) = \frac{a}{\|(x(k), y(k)) - (x_s(j), y_s(j))\|} + v_j(k), \quad (5.32)$$

where $a \in \Re$ is the assumed known amplitude of the sound source, $(x(k), y(k))$ is the location of the sound source to be estimated at the k^{th} time step, $(x_s(j), y_s(j))$ is the known position of sensor s_j and $v_j(k)$ is the zero-mean Gaussian measurement noise of sensor s_j with variance $\sigma_{v_j}^2$, and is independent of $w(k)$. We assume that all the sensors in this sensor network have the same characteristics, i.e., the same

sensing radius r_s and the same measurement noise covariance $\sigma_{v_j}^2$. The query area is $100m \times 100m$ with the coordinates from (0,0) to (100,100). A fusion center is located at (0,0) in this coordinate system. A multi-sensor scheduling scheme [56] with uniform sampling interval ($T = 0.1s$) is used to perform the tracking task.

The simulation is run for 100 time steps. QKFQO and QKFQI are implemented with optimization problem (5.3) where $\gamma = 0.1$ under two different noise variances $\sigma_{v_j}^2 = 0.1$ and $\sigma_{v_j}^2 = 0.01$ cases. We verify their average tracking errors by, up to time step k ,

$$\frac{1}{k} \sum_{t=1}^k [(x(t) - \hat{x}(t|t))'(x(t) - \hat{x}(t|t))]. \quad (5.33)$$

From Fig. 5.3 - Fig. 5.6, we can find that the performance of QKFQI is better than the one of QKFQO. In Fig. 5.3 and Fig. 5.4 where the blue curves denote the true paths and the red curves denote the tracking paths, when $\sigma_{v_j}^2 = 0.1$, the tracking of QKFQO actually failed at time step 86, however, the tracking of QKFQI is successful throughout the 100 time steps. From Fig. 5.5 and Fig. 5.6, we can also see that the tracking errors are smaller when the measurement noise variances are smaller.

When compared with the SOI algorithm [44], which severely quantizes the innovation to only one bit, it is found that the SOI method cannot finish the entire tracking course even for $\sigma_{v_j}^2 = 0.01$ case, i.e. failed at time step 22. The results of SOI method are shown in Fig. 5.7 and Fig. 5.8.

5.6 Conclusion

The quantized Kalman filtering problem was investigated in a sensor network. Two quantized Kalman filters have been developed based on the quantized observations

Figure 5.3: QKFQO: Tracking trajectory under $\sigma_{v_j}^2 = 0.1$

or the quantized innovations, respectively. Due to the bandwidth constraint, the optimization problems considering tradeoff between the energy consumption and performance have been investigated. The approximate solutions of the two optimization problems for bandwidth scheduling were given. Compared with the Kalman filter without quantization, they have lower performance due to loss of information in quantization. Between the two proposed quantization algorithms, it was found that QKFQI has better filtering performance than QKFQO under the same bit number constraint.

Figure 5.4: QKFQI: Tracking trajectory under $\sigma_{v_j}^2 = 0.1$

Figure 5.5: Tracking error comparison under $\sigma_{v_j}^2 = 0.1$

Figure 5.6: Tracking error comparison under $\sigma_{v_j}^2 = 0.01$

Figure 5.7: SOI: Tracking trajectory under $\sigma_{v_j}^2 = 0.01$

Figure 5.8: SOI: Tracking error comparison under $\sigma_{v_j}^2 = 0.01$

Chapter 6

Compressed Kalman Filter

6.1 Introduction

Energy limitation is a critical issue in WSN, due to the limited battery storage of each sensor node [57]. In the literature, many energy-efficient algorithms have been proposed to prolong the lifetime of a WSN. Sparse Topology and Energy Management (STEM) technique is proposed in [58], which trades off between the power savings and path setup latency in WSN. An optimal energy allocation policy is derived in [59], which minimizes the total energy cost of the network subject to certain level of performance requirement. In [60], an adaptive energy-efficient multi-sensor scheduling scheme is proposed. The sampling interval alters at each time step such that the energy consumption is less while maintaining the tracking performance. These algorithms focus on how to schedule the sensor nodes in WSN.

Another common way to reduce the energy consumption is to reduce the amount of information to be transmitted, since transmission is the major cost of energy consumption in WSNs. For example, in [44], the innovation is severely quantized to only one bit for transmission. A quantized Kalman filtering algorithm is proposed

in [61], which determines the optimal number of bits used by each sensor node such that the energy usage is minimum. A general multi-level quantized innovation Kalman filtering algorithm is studied in [62], which improves the performance as compared with the sign-of-innovation Kalman filter under 1-bit transmission case.

There are some other works based on dimension reduction method, which reduces the dimension of the information vector to a smaller one by a linear dimensionality reducing operators. In [63], an optimal linear estimator is developed for state dimensionality reduction, in the sense of minimizing the linear mean-square error. For correlated measurements, an iterative estimator is derived as well.

In this chapter, instead of quantizing the observations or innovations, we want to reduce the totality of transmitted data. It is noticed that the communications among sensor nodes consume the majority of the total energy consumed by a WSN, according to a common energy model [50]. In this chapter, for the sake of illustration of the proposed method, a simple energy model is used based on [50], in which the communication cost is directly proportional to the quantity of data transmitted.

In Kalman filtering (KF) based distributed target tracking in WSNs, the estimated state and the associated error covariance are necessary for the next time step estimation and must be passed around the sensor nodes. For example, in an n -dimensional system, there are $n(n+1)/2 + n$ elements needed to be transmitted at each time step. The first term $n(n+1)/2$ is the number of transmitted elements of the state error covariance matrix, due to its symmetry property, and the second term n is the number of transmitted elements of the estimated state. The number of transmitted elements increases dramatically with the increasing of the system dimension. In this chapter, a compressed Kalman filtering (CKF) algorithm is proposed. In CKF, a diagonal matrix is used to approximate the state error covariance matrix, provided that the diagonal matrix is an upper bound of the original covariance matrix to

prevent the divergence of the proposed filtering algorithm. The total number of transmitted elements is reduced to $2n$, including the state estimate and a diagonal matrix which is an upper bound of the covariance matrix, hence the energy cost is reduced.

6.2 Compressed Kalman Filter

6.2.1 Problem Formulation

Given the state estimation error covariance matrix $P(k|k) = [p_{ij}]_{n \times n}$ at time step t_k , our objective is to find a diagonal matrix $\bar{P}(k|k) = \text{diag}[z_i]_n, i = 1, 2, \dots, n$, such that it is an upper bound of the original matrix and minimizes the trace of the predicted state error covariance matrix $P(k+1|k)$, which is given as:

$$P(k+1|k) = F(\Delta t_k) \bar{P}(k|k) F'(\Delta t_k) + Q(\Delta t_k), \quad (6.1)$$

where $F(\Delta t_k)$ and $Q(\Delta t_k)$ are the state transition matrix and processing noise variance matrix, defined in (2.2) and (2.3), respectively, in Section 2.4.1. In the rest of this chapter, for simplicity, $F(\Delta t_k)$ and $Q(\Delta t_k)$ will be replaced by F and Q , respectively, as only fixed sampling interval is considered in this chapter.

Mathematically, the optimization problem is formulated as follows:

$$\begin{aligned} & \min_{\{z_i\}} (\text{tr}(P(k+1|k))) \\ & \text{subject to : } \bar{P}(k|k) \geq P(k|k) > 0, \end{aligned} \quad (6.2)$$

where $\text{tr}(A)$ means the trace of the matrix A .

6.2.2 General Suboptimal Solution

The above optimization problem can be solved by Linear Matrix Inequality (LMI) method. However, the computation load of carrying out the LMI method is too heavy for low cost sensor nodes. A suboptimal solution (*General Suboptimal solution*) can be obtained by simply setting $\bar{P}(k|k) = n \times \text{diag}[p_{ii}]_n$, where the original covariance matrix $P(k|k) = [p_{ij}]_{n \times n}$ and the dimension of $P(k|k)$ is n .

Theorem 6.1. *Consider a positive definite symmetric matrix $P_n = [p_{ij}]_{n \times n}$. The matrix $\bar{P}_n = n \times \text{diag}[p_{ii}]_n$ provides an upper bound of the matrix P_n , or in other words, we have $\bar{P}_n \geq P_n$.*

Proof. We prove this theorem by induction.

It is easy to verify that when $n = 2$, the argument holds. Here we suppose the argument still holds for $n = r - 1$, $n > r > 2$, that is, $\bar{P}_{r-1} \geq P_{r-1} > 0$, where $\bar{P}_{r-1} = (r - 1) \times \text{diag}[p_{ii}]_{r-1}$. Then for $n = r$, we have

$$P_r = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1r} \\ p_{12} & p_{22} & \cdots & p_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ p_{1r} & p_{2r} & \cdots & p_{rr} \end{bmatrix} = \begin{bmatrix} P_{r-1} & C \\ C' & p_{rr} \end{bmatrix}, \quad (6.3)$$

where $C = [p_{1r}, p_{2r}, \dots, p_{r-1,r}]'$, and

$$\bar{P}_r = \begin{bmatrix} rp_{11} & 0 & \cdots & 0 \\ 0 & rp_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & rp_{rr} \end{bmatrix}. \quad (6.4)$$

Thus,

$$\begin{aligned}
 \bar{P}_r - P_r &= \begin{bmatrix} (r-1)p_{11} & -p_{12} & \cdots & -p_{1r} \\ -p_{12} & (r-1)p_{22} & \cdots & -p_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ -p_{1r} & -p_{2r} & \cdots & (r-1)p_{rr} \end{bmatrix} \\
 &= \begin{bmatrix} A & -C \\ -C' & (r-1)p_{rr} \end{bmatrix}. \tag{6.5}
 \end{aligned}$$

Here we have $A = \bar{P}_{r-1} - P_{r-1} + \text{diag}[p_{ii}]_{r-1}$. Since we have the assumption that $\bar{P}_{r-1} \geq P_{r-1}$, so $\bar{P}_{r-1} - P_{r-1} \geq 0$. What is more, it is obvious that $\text{diag}[p_{ii}]_{r-1}$ is a positive definite diagonal matrix, since all the diagonal elements are positive. Therefore, we can have the conclusion that matrix A , which is the sum of a semi-positive definite matrix and another positive definite matrix, is positive definite. Hence, in order to prove the positive definiteness of $(\bar{P}_r - P_r)$, it is sufficient if $\det(\bar{P}_r - P_r) > 0$ can be proved. We prove this in the following.

Firstly by comparing the matrices A and $\frac{1}{r-1}P_{r-1}$, we have

$$\begin{aligned}
 A - \frac{1}{r-1}P_{r-1} &= \frac{1}{r-1}[(r-1)A - P_{r-1}] \\
 &= \frac{1}{r-1}[(r-1)(\bar{P}_{r-1} - P_{r-1}) + (r-1) \times \text{diag}[p_{ii}]_{r-1} - P_{r-1}] \\
 &= \frac{r}{r-1}(\bar{P}_{r-1} - P_{r-1}) \\
 &\geq 0. \tag{6.6}
 \end{aligned}$$

Hence, $A \geq \frac{1}{r-1}P_{r-1}$, and $A^{-1} \leq (\frac{1}{r-1}P_{r-1})^{-1}$ (See Corollary 7.7.4, page 471, in [64]). Therefore,

$$C' A^{-1} C \leq C' \left(\frac{1}{r-1} P_{r-1} \right)^{-1} C.$$

On the other hand, since $P_r > 0$, we have

$$p_{rr} - C'P_{r-1}^{-1}C > 0,$$

or

$$(r-1)p_{rr} - C'\left(\frac{1}{r-1}P_{r-1}\right)^{-1}C > 0.$$

Hence,

$$(r-1)p_{rr} - C'A^{-1}C > 0.$$

Note that

$$\begin{bmatrix} I & 0 \\ C'A^{-1} & I \end{bmatrix} \begin{bmatrix} A & -C \\ -C' & (r-1)p_{rr} \end{bmatrix} = \begin{bmatrix} A & -C \\ 0 & (r-1)p_{rr} - C'A^{-1}C \end{bmatrix}.$$

Evaluate the determinant on the both sides,

$$\det \begin{bmatrix} I & 0 \\ C'A^{-1} & I \end{bmatrix} \det \begin{bmatrix} A & -C \\ -C' & (r-1)p_{rr} \end{bmatrix} = \det \begin{bmatrix} A & -C \\ 0 & (r-1)p_{rr} - C'A^{-1}C \end{bmatrix},$$

or

$$\det(\bar{P}_r - P_r) = ((r-1)p_{rr} - C'A^{-1}C)\det(A). \quad (6.7)$$

Since matrix A is positive definite, we have $\det(A) > 0$. Hence, $\det(\bar{P}_r - P_r) > 0$.

Together with that matrix A is positive definite, we know that the matrix $(\bar{P}_r - P_r)$ is positive definite, or $\bar{P}_r > P_r$. Therefore, by induction, we can conclude that for

any positive integer n , $\bar{P}_n = n \times \text{diag}[p_{ii}]_n$ is an upper bound of the original matrix $P_n = [p_{ij}]_{n \times n}$. This concludes the proof. $\square$

6.2.3 Boundedness of the Filtering Error Covariance of the CKF

The compressed Kalman filtering algorithm can be decomposed into the following three steps at each iteration:

$$\text{Predict Step : } P(k+1|k) = F\bar{P}(k|k)F' + Q;$$

$$\text{Update Step : } P^{-1}(k+1|k+1) = P^{-1}(k+1|k) + H'R^{-1}H;$$

$$\text{Collapse Step : } \bar{P}(k+1|k+1) = \phi(P(k+1|k+1)).$$

The operator $\phi(\cdot)$ is the collapse operator, which takes an arbitrary symmetric positive definite matrix and converts it to a diagonal symmetric positive definite matrix, as indicated in Theorem 6.1. The associated difference Riccati equation can be expressed as, by replacing $P(k+1|k)$ by X_{k+1} , $F(\Delta t_k)$ by F and $Q(\Delta t_k)$ by Q , respectively,

$$X_{k+1} = n \sum_{i=1}^n F J_i [X_k - X_k H' (H X_k H' + R)^{-1} H X_k] J_i' F' + Q, \quad (6.8)$$

where J_i is such a square matrix that only the $(i, i)^{th}$ element of the matrix is 1 while all the others are 0. For example,

$$J_2 = \begin{bmatrix} 0 & 0 & \mathbf{0}_{1 \times (n-2)} \\ 0 & 1 & \mathbf{0}_{1 \times (n-2)} \\ \mathbf{0}_{(n-2) \times 1} & \mathbf{0}_{(n-2) \times 1} & \mathbf{0}_{(n-2) \times (n-2)} \end{bmatrix},$$

where $\mathbf{0}_{(n-2) \times (n-2)}$ means a zero matrix of dimension $(n-2) \times (n-2)$, and n is the dimension of the system state. Denote $A_i = FJ_i$, then (6.8) can be written as

$$X_{k+1} = n \sum_{i=1}^n A_i [X_k - X_k H' (H X_k H' + R)^{-1} H X_k] A_i' + Q. \quad (6.9)$$

The following theorem provides sufficient conditions to guarantee that the sequence $\{X_k\}_{k=0}^{\infty}$ defined in (6.9) is bounded.

Theorem 6.2. *If there exists a square matrix $\bar{A}$ such that the following hold:*

1. $\min(\lambda_{\bar{A}'\bar{A}}) \geq \max(a_i' a_i, i = 1, 2, \dots, n)$, where $\lambda_{\bar{A}'\bar{A}}$ is an eigenvalue of the matrix $\bar{A}'\bar{A}$ and a_i is the i^{th} column vector of the state transition matrix F ;
2. $R > 0$, $Q \geq 0$, and $(H, n\bar{A})$ is a detectable pair,

then the sequence $\{X_k\}_{k=0}^{\infty}$ defined in (6.9) is bounded.

Proof. It is known that if $M > 0$ and $Y \geq 0$, then [64]

$$\min(\lambda_M) \text{tr}(Y) \leq \text{tr}(MY) \leq \max(\lambda_M) \text{tr}(Y).$$

Hence we have

$$\text{tr}(A_i Y A_i') = \text{tr}(A_i' A_i Y) \leq \max(\lambda_{A_i' A_i}) \text{tr}(Y),$$

where $A_i = FJ_i$. It is noticed that $\max(\lambda_{A_i' A_i}) = a_i' a_i$. Note that for any square matrix $\bar{A}$ such that $\min(\lambda_{\bar{A}'\bar{A}}) \geq \kappa_m$, where $\kappa_m = \max(a_i' a_i, i = 1, 2, \dots, n)$, we have

$$\begin{aligned} \kappa_m \text{tr}(Y) &\leq \min(\lambda_{\bar{A}'\bar{A}}) \text{tr}(Y) \\ &\leq \text{tr}(\bar{A}' \bar{A} Y) \\ &= \text{tr}(\bar{A} Y \bar{A}'), \end{aligned}$$

which means

$$\text{tr}(A_i Y A_i') \leq \text{tr}(\bar{A} Y \bar{A}').$$

Due to the fact

$$A_i[X_k - X_k H'(H X_k H' + R)^{-1} H X_k] A_i' = (A_i - K_i H) X_k (A_i - K_i H)' + K_i R K_i',$$

where $K_i = A_i H'(H X_k H' + R)^{-1}$ and $R > 0$, we know that $A_i[X_k - X_k H'(H X_k H' + R)^{-1} H X_k] A_i'$ is a positive semi-definite matrix. Therefore, we have

$$\begin{aligned} \text{tr}(X_{k+1}) &= \text{tr}(Q) + n \sum_{i=1}^n \text{tr}(A_i[X_k - X_k H'(H X_k H' + R)^{-1} H X_k] A_i') \\ &\leq \text{tr}(Q) + n \sum_{i=1}^n \text{tr}(\bar{A}[X_k - X_k H'(H X_k H' + R)^{-1} H X_k] \bar{A}) \\ &= \text{tr}\{Q + (n\bar{A})[X_k - X_k H' \times (H X_k H' + R)^{-1} H X_k](n\bar{A})\}. \end{aligned}$$

On the other hand, if $(H, n\bar{A})$ is a detectable pair and $R > 0$, $Q \geq 0$, then the sequence $\{Z_k\}_{k=0}^{\infty}$ defined in the following iteration,

$$Z_{k+1} = Q + n\bar{A}[Z_k - Z_k H'(H Z_k H' + R)^{-1} H Z_k] n\bar{A}',$$

is bounded [65], hence the sequence $\{\text{tr}(Z_k)\}_{k=0}^{\infty}$ is bounded as well. Since at each iteration, $\text{tr}(X_k) \leq \text{tr}(Z_k)$ and $X_k > 0$ and $Z_k > 0$, we have the conclusion that the sequence $\{X_k\}_{k=0}^{\infty}$ is bounded. This concludes the proof. $\square$

We use the following example to illustrate the existence of a matrix $\bar{A}$. Suppose the state transition matrix and the measurement function are

$$F = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}, H = [1 \ 0].$$

According to the above theorem, we have $\kappa_m = t^2 + 1$. It is easy to verify that if the required matrix is chosen as

$$\bar{A} = \begin{bmatrix} \sqrt{t^2 + 1} & 1 \\ 0 & \sqrt{t^2 + 1} \end{bmatrix},$$

then the pair $(H, n\bar{A})$ is detectable.

With regard to the target tracking application of WSNs, we give a closed-form solution of the optimization problem for a target moving in 1D space, and an approximate closed-form solution for a target moving in 2D space which is better than the *General Suboptimal solution* given above. The sampling interval $\Delta t = t_k - t_{k-1}$ is assumed to be fixed and known in both cases. For simplicity, we denote the sampling interval as t hereafter in this chapter.

6.3 Target Tracking Applications

In this section, we apply the CKF algorithm to target tracking applications in WSNs. An optimal solution to the problem (6.2) for a target moving in 1D space (e.g. moving along x -axis) with system dimension of 2, the target position and velocity, is derived in closed form. For a target moving in 2D space (e.g. moving in xy -plane) with system dimension of 4, a suboptimal solution is given, which is proved to be better than the *General Suboptimal solution* in terms of smaller $\text{tr}(P(k+1|k))$.

6.3.1 Case 1: Target Moving in 1D Space

In this section, we provide a closed-form solution to the optimization problem under the scenario that a WSN tracks a target moving in 1D space along the x -axis. The target motion is modeled as in Section 2.4.1, where the target state is $x(k) = [x_p(k), x_v(k)]'$, $x_p(k)$ is the x -coordinate of the target and $x_v(k)$ is the velocity of the target at time t_k . The state transition matrix F , the noise covariance matrices Q and R_j are given as follows:

$$F = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}, \quad Q = q \begin{bmatrix} t^4/4 & t^3/2 \\ t^3/2 & t^2 \end{bmatrix}, \quad R_j = \sigma_j^2, \quad (6.10)$$

where q is the intensity of the process noise [49]. The solution to the above optimization problem is given below.

Theorem 6.3. *Consider a system with 1D dynamic model as in Section 2.4.1 and (6.10), a diagonal matrix*

$$\bar{P}(k|k) = \text{diag}[p_{11} + |p_{12}|\sqrt{t^2 + 1}, p_{22} + |p_{12}|/\sqrt{t^2 + 1}]$$

is optimal in the sense of minimizing the trace of $P(k+1|k)$ as in (6.1), provided that $\bar{P}(k|k) \geq P(k|k) > 0$.

Proof. Denote the desired diagonal matrix and the state error covariance matrix, respectively, at time step k , as

$$\bar{P}(k|k) = \begin{bmatrix} z_1 & 0 \\ 0 & z_2 \end{bmatrix}, \quad P(k|k) = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix},$$

then according to (6.1), the trace of $P(k+1|k)$ is

$$\text{tr}(P(k+1|k)) = z_1 + z_2(1+t^2) + q(t^2 + t^4/4). \quad (6.11)$$

The relationship $\bar{P}(k|k) \geq P(k|k)$ leads to the following two inequalities:

$$z_1 - p_{11} \geq 0;$$

$$(z_1 - p_{11})(z_2 - p_{22}) - p_{12}^2 \geq 0.$$

Thus based on (6.11), we have

$$\begin{aligned} \text{tr}(P(k+1|k)) &\geq z_1 + (1+t^2)(p_{22} + p_{12}^2/(z_1 - p_{11})) + q(t^2 + t^4/4) \\ &\geq 2|p_{12}|\sqrt{1+t^2} + p_{11} + p_{22}(1+t^2) + q(t^2 + t^4/4), \end{aligned}$$

the minimum is achieved if and only if the following two equations hold,

$$(z_1 - p_{11})(z_2 - p_{22}) - p_{12}^2 = 0,$$

$$(1+t^2)p_{12}^2/(z_1 - p_{11}) = z_1 - p_{11},$$

or

$$z_1 = p_{11} + |p_{12}|\sqrt{t^2 + 1},$$

$$z_2 = p_{22} + |p_{12}|/\sqrt{t^2 + 1}.$$

Hence, the optimal solution to problem (6.2) in 1D case is (6.11). $\square$

Remark 6.1. *As compared to the General Suboptimal solution, the solution given by (6.11) achieves a smaller value of $\text{tr}(P(k+1|k))$. Applying the General Suboptimal*

solution to the system as described in Section 2.4.1 and (6.10), we have

$$\text{tr}(P(k+1|k))_G = 2p_{11} + 2p_{22}(1+t^2) + q(t^2+t^4/4),$$

where the subscript G stands for the result obtained by applying the General Suboptimal solution method; while we have obtained from (6.11) that

$$\begin{aligned} \text{tr}(P(k+1|k))_O &= p_{11} + |p_{12}|\sqrt{t^2+1} + q(t^2+t^4/4) \\ &\quad + (p_{22} + |p_{12}|/\sqrt{t^2+1}) \times (1+t^2), \end{aligned}$$

where the subscript O stands for the optimal solution. Since $P > 0$, we have $\det(P) > 0$, i.e., $p_{11}p_{22} - p_{12}^2 > 0$, or $\sqrt{p_{11}p_{22}} > |p_{12}|$. Hence,

$$\begin{aligned} \text{tr}(P(k+1|k))_G - \text{tr}(P(k+1|k))_O &= p_{11} + p_{22}(1+t^2) - 2|p_{12}|\sqrt{1+t^2} \\ &\geq 2\sqrt{1+t^2}(\sqrt{p_{11}p_{22}} - |p_{12}|) \\ &> 0, \end{aligned}$$

which means the solution given by (6.11) achieves a smaller value of $\text{tr}(P(k+1|k))$ as compared to the General Suboptimal solution.

6.3.2 Case 2: Target Moving in 2D Space

In this section, we provide an approximate solution to the optimization problem under the scenario that a WSN tracks a target moving in 2D space (the xy -plane). The target motion model and the measurement model are described in Section 2.4.1 and Section 2.4.2.

In order to reduce the computation complexity of the 4×4 matrix $P(k|k) = [p_{ij}]_{n \times n}$,

an approximate matrix $\tilde{P}(k|k)$ is first obtained, where

$$\tilde{P}(k|k) = \begin{bmatrix} 2p_{11} & 2p_{12} & 0 & 0 \\ 2p_{12} & 2p_{22} & 0 & 0 \\ 0 & 0 & 2p_{33} & 2p_{34} \\ 0 & 0 & 2p_{34} & 2p_{44} \end{bmatrix}, \quad (6.12)$$

then the solution to the problem (6.2) is given below:

Theorem 6.4. *Consider a system with 2D dynamic model as described in Section 2.4.1 and Section 2.4.2, a diagonal matrix*

$$\bar{P}(k|k) = \text{diag} \left[2p_{11} + 2|p_{12}|\sqrt{t^2 + 1}, 2p_{22} + 2|p_{12}|/\sqrt{t^2 + 1}, \right. \\ \left. 2p_{33} + 2|p_{34}|\sqrt{t^2 + 1}, 2p_{44} + 2|p_{34}|/\sqrt{t^2 + 1} \right] \quad (6.13)$$

is optimal in the sense of minimizing the trace of $P(k+1|k)$ as in (6.1), subject to the constraint that $\bar{P}(k|k) \geq \tilde{P}(k|k) \geq P(k|k) > 0$.

Proof. It is not difficult to prove $\tilde{P}(k|k) \geq P(k|k)$, provided $P(k|k) > 0$. The rest is similar to the proof of Theorem 6.3. $\square$

Remark 6.2. *As compared to the General Suboptimal solution, the solution given by (6.13) achieves a smaller value of $\text{tr}(P(k+1|k))$. Applying the General Suboptimal solution, we have*

$$\text{tr}(P(k+1|k))_G = 4p_{11} + 4p_{33} + 2q(t + t^3/3) + 4(p_{22} + p_{44})(1 + t^2);$$

while applying the solution given by (6.13), we have

$$\begin{aligned} \text{tr}(P(k+1|k))_S &= 2(p_{11} + p_{33}) + 2q(t + t^3/3) + 4(|p_{12}| + |p_{34}|)\sqrt{t^2 + 1} \\ &\quad + 2(p_{22} + p_{44})(1 + t^2), \end{aligned}$$

where the subscript S stands for the suboptimal solution given by (6.13). Hence,

$$\begin{aligned} &\text{tr}(P(k+1|k))_G - \text{tr}(P(k+1|k))_S \\ &= 2[p_{11} + p_{33} + (p_{22} + p_{44})(1 + t^2)] - 4(|p_{12}| + |p_{34}|)\sqrt{1 + t^2} \\ &\geq 4\sqrt{1 + t^2}(\sqrt{p_{11}p_{22}} - |p_{12}| + \sqrt{p_{33}p_{44}} - |p_{34}|). \end{aligned}$$

Since $\tilde{P} > 0$, we have $p_{11}p_{22} - p_{12}^2 > 0$ and $p_{33}p_{44} - p_{34}^2 > 0$, which means

$$(\sqrt{p_{11}p_{22}} - |p_{12}| + \sqrt{p_{33}p_{44}} - |p_{34}|) > 0.$$

Thus, the solution given by (6.13) achieves a smaller value of $\text{tr}(P(k+1|k))$ as compared to the General Suboptimal solution.

We should notice that the operations in CKF is similar to the original KF, with an additional compression operation which replaces the original state error covariance matrix $P(k|k)$ with the diagonal one $\bar{P}(k|k)$.

6.4 Simulation Results

6.4.1 Energy Model

Our energy model used in this chapter is based on [50], as described in Section 2.5. Energy is consumed mainly during data communication, therefore in the simulations

we only consider such energy. The communication energy consumption, both for transmitting (2.18) and receiving data (2.19),

$$E(s_j, s_k) = (e_t + e_d r_{jk}^{\alpha_c})b + e_r b, \quad (6.14)$$

is the focus, which in fact also dominates the energy consumption of WSNs. We present this communication energy consumption comparisons only between our proposed CKF algorithm and the standard KF algorithm in the simulation results shown in the following sections.

6.4.2 Case 1: Target Moving in 1D Space

We consider a target moving along the x -axis in a chain-deployed sensor network as an example in this section, as shown in Fig. 6.1. The cone shape illustrates the detection region of each sensor node [66], and the terminal point is the sensor location. Sensor nodes' coordinates are $(x_{s_j}, 0), \forall j = 1, 2, \dots, N$. For illustration purpose, we assume the sensors are located at $(0, 0), (10, 0), \dots, (190, 0)$ along the x -axis, and each sensor monitors its right hand side area, for example, sensor s_5 monitors the segmentation from $(40, 0)$ to $(50, 0)$, or the interval $[40, 50]$. Hence,

Figure 6.1: Target moving in 1D space

this sensor network consists of 20 sensor nodes. The measurement function for sensor s_j is

$$z_j(k) = \|x_p(k) - x_{s_j}\| + v_j(k),$$

where $\|x_p(k) - x_{s_j}\|$ means the distance between the target and the sensor s_j , $v_j(k)$ is the measurement noise with zero mean and known covariance σ_j^2 . The measurement noise covariance is assumed to be $\sigma_j^2 = 0.01$. The linearized measurement function is $H = [1, 0]$. A sensor s_{j_2} is triggered to perform the tracking task, i.e., sensing the target and performing the CKF algorithm, by the previous tasking sensor s_{j_1} , if the estimated target location is within the monitoring area of s_{j_2} . Based on (6.14), the communication energy consumption of sensor s_{j_1} at time t_k , which only considers the transmission energy usage, is modeled as

$$E_c(k) = (e_t + e_d(x_{s_{j_1}} - x_{s_{j_2}})^2)b,$$

where e_t and e_d are decided by the specifications of the transmitter of s_{j_1} , and b is the totality of transmitted data (for example in this case, the system dimension is $n = 2$, by using the KF, we have $b = n + n \times (n + 1)/2 = 5$, where the first n is the number of entries in the state vector $\hat{x}(k|k)$ and the second term $n \times (n + 1)/2$ is the number of different entries in the associated symmetric covariance matrix $P(k|k)$; while by using the CKF or the CKF with the *General Suboptimal solution*, $b = n + n = 4$, where the first n is the number of entries in the state vector $\hat{x}(k|k)$ and the second n is the number of none zero entries in the diagonal matrix $\bar{P}(k|k)$, as discussed in the second last paragraph in Section 6.1). The sensor network is operated for 100 time steps with the sampling interval $t = 0.1s$. The initial state of the target is $x_0 = [2, 5]'$, and the initial estimate is $\hat{x}_0 = [0, 0]'$ with the error covariance matrix $P_0 = 0.1I_2$, where I_2 is the identity matrix of dimension 2.

A comparison of tracking performance of the CKF, under both the optimal solution and the General Suboptimal solution, with the KF, in one particular run of simulation is shown in Fig. 6.2 (Position tracking) and Fig. 6.3 (Position estimation error), Fig. 6.4 (Velocity tracking) and Fig. 6.5 (Velocity estimation error), and

Figure 6.2: Position tracking in one dimension space

Fig. 6.8 (Energy consumption). The position estimation error $p_e(k)$ at time step k is obtained by calculating the difference between the estimated target position and the real target position, i.e., $p_e(k) = \hat{x}_p(k) - x_p(k)$, and the velocity estimation error $v_e(k)$ at time step k is obtained by calculating the difference between the estimated target velocity and the real target velocity, i.e., $v_e(k) = \hat{x}_v(k) - x_v(k)$. It is shown in these figures that the CKF has comparable performance with the original KF algorithm.

By using the Monte Carlo method, 1000 times of simulation are carried out to compare the statistical properties of the proposed CKF algorithm and the original KF algorithm. The results are illustrated in Fig. 6.6 and Fig. 6.7. The average energy consumptions for both CKF and KF algorithms are shown in Fig. 6.8. It is clearly shown in these figures that the proposed CKF algorithm has comparable performance as compared to the original optimal KF algorithm, but the energy consumption is less, as the KF consumed 17.93mJ energy for communication, while the CKF algorithm consumed 14.35mJ, i.e., the energy saving is about 19.97%. The CKF with *General Suboptimal solution* consumed the same amount of energy, i.e.,

Figure 6.3: Position estimation error in one dimension space

Figure 6.4: Velocity tracking in one dimension space

Figure 6.5: Velocity estimation error in one dimension space

Figure 6.6: Computed filtering error variances for position of target moving in one dimension space case

Figure 6.7: Computed filtering error variances for velocity of target moving in one dimension space case

Figure 6.8: Energy consumption evaluation in one dimension space case

14.35mJ, but its estimation error is a little bit higher, as indicated in Fig. 6.6 and Fig. 6.7, which is consistent as discussed in the remark of Theorem 6.3.

6.4.3 Case 2: Target Moving in 2D Space

In this section, we consider a target moving in the xy -plane which consists of grid-deployed sensor nodes as an example. For illustration purpose, we assume each sensor monitors a circular area around it, as in [60]. In this case, the sensor network consists of 225 sensor nodes, which are uniformly distributed in the square area. Sensor nodes' coordinates are $(x_{s_j}, y_{s_j}), \forall j = 1, 2, \dots, N$. A sensor is scheduled to perform the tracking task, i.e., sensing the target and performing the CKF algorithm, if it is the nearest one to the predicted the target position. The measurement function for sensor s_j is

$$z_j(k) = \sqrt{(x_p(k) - x_{s_j})^2 + (y_p(k) - y_{s_j})^2} + v_j(k).$$

The measurement Jacobian matrix is

$$H' = \begin{bmatrix} (x_p(k) - x_{s_j}) / (\sqrt{(x_p(k) - x_{s_j})^2 + (y_p(k) - y_{s_j})^2}) & 0 \\ 0 & (y_p(k) - y_{s_j}) / (\sqrt{(x_p(k) - x_{s_j})^2 + (y_p(k) - y_{s_j})^2}) \end{bmatrix}.$$

Similarly, based on (6.14), the communication energy consumption of sensor s_{j_1} at time k is modeled as

$$E_c(k) = (e_t + e_d((x_{s_{j_1}} - x_{s_{j_2}})^2 + (y_{s_{j_1}} - y_{s_{j_2}})^2))b,$$

Figure 6.9: Tracking trajectories in two dimension space case

where e_t and e_d are decided by the specifications of the transmitter of s_{j1} , and b is the totality of transmitted data (for example in this case, the system dimension is 4, by using the KF, $b = n + n \times (n + 1)/2 = 14$, while by using the CKF, $b = n + n = 8$). The sensor network is operated for 50 time steps with the sampling interval $t = 0.1s$. The initial state is $x_0 = [80, 3, 5, 3]'$, and the initial estimate is $\hat{x}_0 = [76.67, 0, 3.333, 0]'$, which is the location of the nearest sensor to the target, with the error covariance matrix $P_0 = 10I_4$, where I_4 is the identity matrix of dimension 4. In this example, the target is traveling from the right lower corner to the left upper corner, as shown in Fig. 6.9 (the sensor nodes of the wireless sensor are not shown in this figure for better illustration of the target trajectory).

The performance of the CKF is evaluated in Fig. 6.9 - Fig. 6.15. Fig. 6.9 shows

Figure 6.10: Position estimation error in two dimension space case

Figure 6.11: Velocity estimation in two dimension space case: x -direction

Figure 6.12: Velocity estimation error in two dimension space case: x -direction

Figure 6.13: Velocity estimation in two dimension space case: y -direction

Figure 6.14: Velocity estimation error in two dimension space case: y -direction

Figure 6.15: Energy consumption evaluation in two dimensional space case

the real, KF estimated, and CKF estimated (compressed according to both the solution of (6.13) and the *General Suboptimal solution*, shown as CKF_G in the figures) trajectories of the target. The position estimation error is shown in Fig. 6.10, which is the distance between the estimated target position and the real target position at each time step. It is shown in both figures that both filtering algorithms converge to the real trajectory of the target. The estimated target velocities along x -axis and y -axis with the comparison to the real values are shown in Fig. 6.11 and Fig. 6.13. The velocity estimation errors along x -axis and y -axis are shown in Fig. 6.12 and Fig. 6.14. All the figures show that the CKF has comparable performance with the original KF algorithm. What is more, the CKF consumes much less energy than the original KF algorithm, as shown in Fig. 6.15, in which the KF algorithm accumulatively consumed 28.61mJ energy for communication, while the CKF algorithm (the curve by applying the CKF_G algorithm coincides with the curve by applying the solution of (6.13)) accumulatively consumed only 16.35mJ communication energy, i.e., saved 42.85% energy.

6.5 Conclusion

This chapter presents a compressed Kalman filtering algorithm for target tracking in WSNs. Based on the well-known KF algorithm, an additional compression operation is introduced in the CKF algorithm, which compresses the symmetric covariance matrix into a diagonal matrix to reduce the totality of transmitted data, hence the energy consumption is less. A general suboptimal compression solution is presented. The boundedness of the filtering error covariance of the proposed CKF was also discussed in the chapter. Better solutions were derived for target tracking applications in WSN for the cases that a single target is traveling in either 1D space or 2D space, which have better performance over the general suboptimal solu-

tion. The simulation results demonstrated that the proposed CKF algorithm offers significant energy savings, with comparable tracking performance.

Chapter 7

L-Banded Approximation and Drop Rate

7.1 Introduction

Due to the unreliable communication channel, the phenomenon of data loss across WSNs has been studied extensively [67] [68] [69] [70] [71] [72] [73]. These works mainly focus on the dropping rate of transmitting observations.

Transmitting local processed estimates is another way to communicate in WSNs, instead of transmitting observations. However, the bandwidth requirement is higher for transmitting estimates than that for transmitting raw observations. In order to reduce the totality of transmitted data hence reducing the bandwidth usage and energy consumption, we may approximate the covariance matrix by a banded matrix [10], which is defined in the following way: a matrix is a **l -banded matrix** ($l \geq 0$) if the elements outside a band defined by the l^{th} lower diagonal and the l^{th} upper diagonal are zero, and the **width** of this matrix is defined to be l . In Chapter 6, a compressed Kalman filter was presented which approximates the estimation

error covariance by a diagonal matrix. This approximation can be considered as a 0-banded matrix.

In this chapter, we will first give the estimation error covariance for a centralized filter, which collects the raw measurements from each sensor node. Then a decentralized filter, which collects the estimates generated by local sensor nodes, is discussed. The l -banded method is also presented, which balances the tracking accuracy and bandwidth usage, as well as the power consumption. A numerical example using 0-banded approximation is presented before concluding this chapter.

7.2 Problem Formulation

We consider a wireless sensor network consisting of N sensor nodes ($s_1, s_2, \dots, s_N$) and a fusion center (FC), where sensor nodes obtain measurements and send them to the FC, as shown in Fig. 7.1.

Figure 7.1: A wireless sensor network

We model the dynamic of the system to be estimated and the measurement function for sensor node as in equations (2.1)-(2.4), with uniform sampling interval. Denote $w(\Delta t_k) = w(k)$, $F(\Delta t_k) = F$, and $Q(\Delta t_k) = Q$ for simplicity. $w(k)$ and $v_j(k)$ are

assumed to have the following correlations:

$$\mathbf{E} \left\{ \begin{bmatrix} w(k) \\ v_j(k) \end{bmatrix} [w(t)' \ (v_j(t))'] \right\} = \begin{bmatrix} Q & S_j \\ (S_j)' & R_j \end{bmatrix} \delta_{tk}, \quad (7.1)$$

$$\mathbf{E}\{v(k)^i(v(t)^j)'\} = S_{ij}\delta_{tk}, \ i \neq j, \quad (7.2)$$

where the symbol $\mathbf{E}$ denotes the mathematical expectation, the superscript $'$ denotes the transpose, and δ_{tk} is the Kronecker delta function. We also denote all the measurements at sensor node s_j up to time instant k as $Z_j(k) \triangleq \{z_j(0), z_j(1), \dots, z_j(k)\}$.

We assume that the sensor nodes and the fusion center are linked by an unreliable wireless communication channel, which means the packages transmitted from sensor nodes to the fusion center may be dropped. The drop out rate is denoted by $(1 - \lambda)$, or alternatively speaking, the rate of data arriving at the fusion center is λ .

There are two methods for a sensor s_j to transmit the data to the fusion center:

1. Directly transmit the raw measurement $z_j(k)$. In this case, only one data is transmitted.
2. Transmit the local processed estimate $\hat{x}^j(k|k)$ and its associated error covariance matrix $P^j(k|k)$, where the superscript j indicates the sensor s_j . In this case, the estimations are obtained by applying the well known Kalman filtering (KF) algorithm for linear case or the extended Kalman filtering (EKF) algorithm for nonlinear case, based on the local measurements $Z_j(k)$.

The totality of transmitted data is $n + n(n - 1)/2$ for the latter communication method while for the former only 1 data is transmitted, i.e. the measurement $z_j(k)$. Due to the package drop characteristic of the channel, intuitively we can say that the second communication method should perform better than the first one.

However, the second one requires larger communication bandwidth, hence more energy consumption.

We denote the approximation process at sensor node s_j as

$$L^j(k) = f_l(P^j(k|k)), \quad (7.3)$$

where $f_l(\cdot)$ is the function converting the covariance matrix $P^j(k|k)$ into an l -banded matrix $L^j(k)$. The fusion center receives the estimates $\hat{X}^s(k)$ and their associated banded error covariance matrices $L^s(k)$ from all the sensor nodes, which are denoted by $\hat{X}^s(k) \triangleq \{\hat{x}^1(k), \hat{x}^2(k), \dots, \hat{x}^N(k)\}$ and $L^s(k) \triangleq \{L^1(k), L^2(k), \dots, L^N(k)\}$.

7.3 Centralized filter

In this section we consider a sensor network consisting of two wireless sensor nodes and a fusion center. The measurements at the sensor nodes are denoted as $z_1(k)$ and $z_2(k)$, as indicated in (2.4). All measured data are communicated to a central fusion center for processing.

Based on [67] and [73], let us define

$$\begin{aligned} g_{\lambda_1 \lambda_2}(X) &= F X F' + Q - \lambda_1 \lambda_2 F X H' (H X H' + R)^{-1} H X F' \\ &\quad - \lambda_1 (1 - \lambda_2) F X h_1 (h_1' X h_1 + R_1)^{-1} h_1' X F' \\ &\quad - \lambda_2 (1 - \lambda_1) F X h_2 (h_2' X h_2 + R_2)^{-1} h_2' X F', \end{aligned} \quad (7.4)$$

where $H = [h_1', h_2']$, λ_1 and λ_2 are the defined as $\Pr(\gamma_{1,k} = 1) = \lambda_1$ and $\Pr(\gamma_{2,k} = 1) = \lambda_2$ respectively, where $\gamma_{1,k}$ and $\gamma_{2,k}$ are mutually uncorrelated i.i.d. Bernoulli

random variables. Let $P_k = P(k|k-1)$, then

$$\mathbf{E}[P_{k+1}|P_k] = g_{\lambda_1\lambda_2}(P_k), \quad (7.5)$$

and

$$\mathbf{E}[P_{k+1}] = \mathbf{E}[g_{\lambda_1\lambda_2}(P_k)], \quad (7.6)$$

where the expectation is taken with respect to both $\gamma_{1,k}$ and $\gamma_{2,k}$. If the percentage of the sensor measurement packets that are correctly received is sufficiently large, then the error covariance P_{k+1} is bounded for any k and we have the following theorem [73].

Theorem 7.1. *First define*

$$\begin{aligned} \phi(\bar{K}, \bar{K}_1, \bar{K}_2, \bar{P}) &= (1 - \lambda_1)(1 - \lambda_2)(F X F' + Q) + \lambda_1 \lambda_2 (A X A' + V) \\ &\quad + \lambda_1(1 - \lambda_2)(A_1 X A'_1 + V_1) + \lambda_2(1 - \lambda_1)(A_2 X A'_2 + V_2), \end{aligned}$$

where $A = F + K[h'_1, h'_2]'$, $A_1 = F + K_1 h_1$, $A_2 = F + K_2 h_2$, $V = Q + K R K'$, $V_1 = Q + K_1 R_{11} K'_1$, $V_2 = Q + K_2 R_{22} K'_2$, and $X \geq 0$. Suppose there exist matrices $\bar{K}$, $\bar{K}_1$, $\bar{K}_2$ and $\bar{P} > 0$ such that $\bar{P} > \phi(\bar{K}, \bar{K}_1, \bar{K}_2, \bar{P})$,

then

- $\forall P_0 \geq 0$, the iteration $\bar{P}_{k+1} = g_{\lambda_1\lambda_2}(\bar{P}_k)$ converges and

$$\lim_{k \rightarrow \infty} \bar{P}_k = \lim_{k \rightarrow \infty} g_{\lambda_1\lambda_2}^k(\bar{P}_0) = \bar{P} \quad (7.7)$$

is independent of initial condition $\bar{P}_0$.

- $\bar{P}$ is the unique positive semidefinite solution of $\bar{P}_{k+1} = g_{\lambda_1\lambda_2}(\bar{P}_k)$, which can

be obtained by solving the following equation:

$$\begin{aligned}
P = & FPF' + Q - \lambda_1\lambda_2 FPH'(HPH' + R)^{-1}HPF' \\
& - \lambda_1(1 - \lambda_2)FPh_1(h_1'Ph_1 + R_1)^{-1}h_1'PF' \\
& - \lambda_2(1 - \lambda_1)FPh_2(h_2'Ph_2 + R_2)^{-1}h_2'PF'.
\end{aligned} \tag{7.8}$$

If we denote the solution of above equation (7.8) as $P_{\lambda_1\lambda_2}$, then we have the following lemma.

Lemma 7.1. *If $0 \leq \lambda_1 \leq 1$ is fixed and $0 \leq \lambda_2^{(1)} \leq \lambda_2^{(2)} \leq 1$, then $P_{\lambda_1\lambda_2^{(1)}} \geq P_{\lambda_1\lambda_2^{(2)}}$. Similarly, for fixed $0 \leq \lambda_2 \leq 1$ and $0 \leq \lambda_1^{(1)} \leq \lambda_1^{(2)} \leq 1$, $P_{\lambda_1^{(1)}\lambda_2} \geq P_{\lambda_1^{(2)}\lambda_2}$.*

7.4 Decentralized filter

By using the decentralized filter, each local sensor subsystem of system (2.1) and (2.4) will perform the local optimal KF/EKF. The information from local estimators are then transmitted to a fusion center to achieve a global optimal or suboptimal state estimate. The local optimal KF is performed as described in equations (2.6)-(2.14). An optimal information fusion at the fusion center in the linear minimum variance sense was obtained in Theorem 1 in [74]. However, in this paper, they assumed there is no package dropped out during the communication. In this section, we take the package drop into consideration. The optimal fusion in terms of linear minimum variance is given as

$$\hat{x}^o(k+1|k+1) = \sum_{j=1}^N \bar{A}_j \hat{x}^j(k+1|k+1), \tag{7.9}$$

where the optimal matrix weights $\bar{A}_j$, $j = 1, 2, \dots, N$ are computed by

$$\bar{A} = \Phi^{-1} e (e' \Phi^{-1} e)^{-1}, \quad (7.10)$$

where $\Phi = (P^{ij}(k+1|k+1))$, $i, j = 1, 2, \dots, N$ is an $nN \times nN$ symmetric positive definite matrix, $P^{ii}(k|k)$ denotes the covariance matrix $P^{ii}(k|k)$ of sensor s_i if $i = j$, or the cross covariance matrix $P^{ij}(k|k)$ between sensors s_i and s_j if $i \neq j$, $\bar{A} = [\bar{A}_1, \bar{A}_2, \dots, \bar{A}_N]'$ and $e = [I_n, \dots, I_n]'$ are both $nN \times n$ matrices. The corresponding variance of the optimal information estimator is computed by

$$P^o(k+1|k+1) = (e' \Phi^{-1} e)^{-1} \quad (7.11)$$

and we have the conclusion that $P^o(k+1|k+1) \leq P^j(k+1|k+1)$, $j = 1, 2, \dots, N$, where the superscript o means the optimal solution.

In our work, we consider a sensor network consisting of two wireless sensor nodes and a fusion center and an imperfect communication channel with possibility of package drop out. For simplicity, we write $\hat{x}_{FC}(k|k)$ at FC as $\hat{x}(k)$, and the estimates sent to FC from j^{th} sensor node as $\hat{x}^j(k|k) = \hat{x}^j(k)$, then

$$\hat{x}(k) = \gamma_{1,k} \gamma_{2,k} A_1 \hat{x}^1(k) + \gamma_{1,k} \gamma_{2,k} A_2 \hat{x}^2(k) + \gamma_{1,k} (1 - \gamma_{2,k}) \hat{x}^1(k) + \gamma_{2,k} (1 - \gamma_{1,k}) \hat{x}^2(k), \quad (7.12)$$

or

$$\hat{x}(k) = (\gamma_{1,k} \gamma_{2,k} A_1 + \gamma_{1,k} (1 - \gamma_{2,k})) \hat{x}^1(k) + (\gamma_{1,k} \gamma_{2,k} A_2 + \gamma_{2,k} (1 - \gamma_{1,k})) \hat{x}^2(k). \quad (7.13)$$

To achieve the unbiasedness of the estimate $\hat{x}(k)$, we require $\mathbf{E}(\hat{x}(k)) = \mathbf{E}(x(k))$.

Since $\mathbf{E}(\hat{x}^j(k)) = \mathbf{E}(x(k))$, $j = 1, 2$, thus, we require

$$I_n = \gamma_{1,k}\gamma_{2,k}A_1 + \gamma_{1,k}(1 - \gamma_{2,k})I_n + \gamma_{1,k}\gamma_{2,k}A_2 + \gamma_{2,k}(1 - \gamma_{1,k})I_n. \quad (7.14)$$

From (7.13) and (7.14) we have the fusion estimation error

$$\begin{aligned} \tilde{x}(k) &= x(k) - \hat{x}(k) \\ &= (\gamma_{1,k}\gamma_{2,k}A_1 + \gamma_{1,k}(1 - \gamma_{2,k})I_n)(x(k) - \hat{x}^1(k)) \\ &\quad + (\gamma_{1,k}\gamma_{2,k}A_2 + \gamma_{2,k}(1 - \gamma_{1,k})I_n)(x(k) - \hat{x}^2(k)) \\ &= (\gamma_{1,k}\gamma_{2,k}A_1 + \gamma_{1,k}(1 - \gamma_{2,k})I_n)\tilde{x}^1(k) + (\gamma_{1,k}\gamma_{2,k}A_2 + \gamma_{2,k}(1 - \gamma_{1,k})I_n)\tilde{x}^2(k). \end{aligned}$$

Let $A = [\gamma_{1,k}\gamma_{2,k}A_1 + \gamma_{1,k}(1 - \gamma_{2,k})I_n, \gamma_{1,k}\gamma_{2,k}A_2 + \gamma_{2,k}(1 - \gamma_{1,k})I_n]'$, and the error covariance matrix of the fusion estimator, $P_{FC}(k|k) = P(k)$, is

$$P(k) = \mathbf{E}(\tilde{x}\tilde{x}') = A'\Sigma A. \quad (7.15)$$

where $\Sigma = (P^{ij}(k|k))$, $i, j = 1, 2$ is an $2n \times 2n$ symmetric positive definite matrix, $P^{ij}(k|k)$ is the covariance matrix if $i = j$ (i.e., $P^{ii}(k|k) = P^i(k|k)$) and the cross covariance matrix if $i \neq j$. We are interested in the expected error covariance matrix $\mathbf{E}(P(k))$, where the expectation is taken of both $\gamma_{1,k}$ and $\gamma_{2,k}$. Hence we have

$$\mathbf{E}(P(k)) = \mathbf{E}[\mathbf{E}(\tilde{x}\tilde{x}')] = \mathbf{E}(A'\Sigma A) = A'_\lambda \Sigma A_\lambda, \quad (7.16)$$

where $A_\lambda = \mathbf{E}(A) = [\lambda_1\lambda_2A_1 + \lambda_1(1 - \lambda_2)I_n, \lambda_1\lambda_2A_2 + \lambda_2(1 - \lambda_1)I_n]'$, λ_1 and λ_2 are defined as $\Pr(\gamma_{1,k} = 1) = \lambda_1$ and $\Pr(\gamma_{2,k} = 1) = \lambda_2$, respectively. And the equation (7.14) becomes

$$I_n = \lambda_1\lambda_2A_1 + \lambda_1(1 - \lambda_2)I_n + \lambda_1\lambda_2A_2 + \lambda_2(1 - \lambda_1)I_n = A'_\lambda e. \quad (7.17)$$

The performance index is chosen to be the trace of the above covariance matrix, i.e. $J = \text{tr}(\mathbf{E}(P(k)))$, which becomes

$$J = \text{tr}(A'_\lambda \Sigma A_\lambda). \quad (7.18)$$

Hence the problem becomes to find the optimal fusion matrix weights $\bar{A}_i$, $i = 1, 2$ under the condition (7.17) to minimize the performance index (7.18). Applying the Lagrange multiplier method, we have

$$F = J + 2\text{tr}[\Theta(A'_\lambda e - I_n)], \quad (7.19)$$

where $\Theta = (\theta_{ij})$ is an $n \times n$ matrix. Set $\partial F / \partial A_\lambda|_{A_\lambda = \bar{A}_\lambda} = 0$, we have

$$\Sigma \bar{A}_\lambda + e\Theta = 0. \quad (7.20)$$

Therefore we have the following matrix equation, which combines (7.17) and (7.20),

$$\begin{bmatrix} \Sigma & e \\ e' & 0 \end{bmatrix} \begin{bmatrix} \bar{A}_\lambda \\ \Theta \end{bmatrix} = \begin{bmatrix} 0 \\ I_n \end{bmatrix}. \quad (7.21)$$

By using the formula of the inverse matrix [75], we have

$$\begin{bmatrix} \bar{A}_\lambda \\ \Theta \end{bmatrix} = \begin{bmatrix} \Sigma & e \\ e' & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ I_n \end{bmatrix} = \begin{bmatrix} \Sigma^{-1}e(e'\Sigma^{-1}e)^{-1} \\ -(e'\Sigma^{-1}e)^{-1} \end{bmatrix}. \quad (7.22)$$

Hence the optimal matrix weights $\bar{A}_i$, $i = 1, 2$ are computed by

$$\bar{A}_\lambda = \Sigma^{-1}e(e'\Sigma^{-1}e)^{-1}, \quad (7.23)$$

and the minimal error covariance matrix of the fusion estimator is, by substituting

(7.23) into (7.16),

$$P_o = (e'\Sigma^{-1}e)^{-1}. \quad (7.24)$$

The results are summarized in the following theorem.

Theorem 7.2. *Let $\hat{x}^j(k)$, $j = 1, 2$ be the estimators of the system state $x(k)$ of each sensor node at time instant k , then the individual estimation error is $\tilde{x}^i(k) = x(k) - \hat{x}^i(k)$, $j = 1, 2$. Assume that $\tilde{x}^i(k)$ and $\tilde{x}^j(k)$ are correlated, and the covariance and cross covariance matrices are denoted by $P^{ii}(k)$ and $P^{ij}(k)$, respectively. Then the optimal fusion estimator is given as, in the sense of linear minimum covariance,*

$$\hat{x}(k) = (\lambda_1\lambda_2\bar{A}_1 + \lambda_1(1 - \lambda_2)I_n)\hat{x}^1(k) + (\lambda_1\lambda_2\bar{A}_2 + \lambda_2(1 - \lambda_1)I_n)\hat{x}^2(k),$$

where the optimal matrix weights $\bar{A}_i$, $i = 1, 2$ are computed by

$$\bar{A} = \Sigma_k^{-1}e(e'\Sigma_k^{-1}e)^{-1},$$

where $\Sigma_k = (P^{ij}(k))$, $i, j = 1, 2$ is an $2n \times 2n$ symmetric positive definite matrix, $\bar{A} = [\lambda_1\lambda_2\bar{A}_1 + \lambda_1(1 - \lambda_2)I_n, \lambda_1\lambda_2\bar{A}_2 + \lambda_2(1 - \lambda_1)I_n]'$ and $e = [I_n, I_n]'$ are both $2n \times n$ matrices. The corresponding variance of the optimal information fusion estimator is computed by

$$P^o(k) = (e'\Sigma_k^{-1}e)^{-1},$$

and we have the conclusion $P^o(k) \leq P^i(k)$, $i = 1, 2$.

The last conclusion can be easily derived by using the Schwartz's matrix inequality,

which gives

$$\begin{aligned}
P^o(k) &= (e' \Sigma_k^{-1} e)^{-1} \\
&= (e' \Sigma_k^{-1/2} \Sigma_k^{1/2} e_i)' (e' \Sigma_k^{-1/2} e)^{-1} (e' \Sigma_k^{-1/2} \Sigma_k^{1/2} e_i) \\
&\leq (\Sigma_k^{1/2} e_i)' (\Sigma_k^{1/2} e_i) \\
&= P^i(k),
\end{aligned}$$

where $e_1 = [I_n, 0_{n \times n}]'$, and $e_2 = [0_{n \times n}, I_n]'$ are $2n \times n$ matrices.

The local Kalman filtering error cross covariance, $P^{ij}(k)$, between the i^{th} and the j^{th} sensor nodes can be obtained directly applying Theorem 2 in [74], as stated in the following theorem.

Theorem 7.3. *With the system and measurement models given in (2.1) and (2.4), and the correlation assumptions (7.1) and (7.2), the local Kalman filtering cross covariance between the i^{th} and the j^{th} sensor nodes can be obtained by the following recursion:*

$$\begin{aligned}
P^{ij}(k+1|k+1) &= [I_n - K_k^i h_i] [F P^{ij}(k|k) F' + Q - F K_{k-1}^i (S_i)' \\
&\quad - S_j (K_{k-1}^j)' F'] [I_n - K_k^j h_j]' + K_k^i S_{ij} (K_k^j)', \quad (7.25)
\end{aligned}$$

where $P^{ij}(k|k)$, $i, j = 1, 2, \dots, N$ ($i \neq j$) are the filtering error cross covariance matrices between the i^{th} and j^{th} sensors with an initial values of $P^{ij}(0|0) = P_0$.

7.5 L-Banded Approximation

Instead of transmitting the whole package of the state $\hat{x}^j(k|k)$ and its associated error covariance $P^j(k|k)$, as well as the cross covariance matrices $P^{ij}(k|k)$, a com-

pressed version of these quantities can reduce the bandwidth usage and save the communication energy.

In [10], the authors derived algorithms to approximate a matrix with banded inverse. In Chapter 6, the compressed Kalman filter is a specified 0-banded approximation compression algorithm to reduce the totality of transmitted data. By applying CKF to the decentralized filter discussed above, we can achieve a balance between the tracking accuracy and the bandwidth usage.

7.6 A Numerical Example

We use a numerical example to illustrate the balance of tracking accuracy and the bandwidth usage of l -banded approximation.

We consider a simple target model with system dimension of 2, i.e.

$$\begin{bmatrix} x(k+1) \\ x_v(k+1) \end{bmatrix} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ x_v(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(k), \quad (7.26)$$

and the measurement model is

$$z_j(k) = [1 \ 0] \begin{bmatrix} x(k) \\ x_v(k) \end{bmatrix} + v_j(k), \quad j = 1, 2. \quad (7.27)$$

The process noise is assumed to have the covariance $Q = 0.1$, the measurement noise covariance is assumed to be $\sigma_j^2 = 0.001$, the sampling interval is $t = 0.1s$ and the system is operated for 100 samples. The initial state of the target is $x_0 = [0, 0]'$, and the initial estimate is $\hat{x}_0 = [0, 0]'$ with the error covariance matrix $P_0 = 0.01I_2$, where I_2 is the identity matrix of dimension 2. The rate of data arriving at the FC

is $\lambda = 0.8$.

Three transmission methods are simulated: first one is transmitting the raw measurements (Transmit measurement), second one is transmitting the full version of the local processed estimates (Transmit estimates), and the third one is transmitting the estimates when CKF (Banded) is used. The comparison of the tracking accuracy, defined as the trace of the state error covariance matrix, is shown in Fig. 7.2. The connectivity of communication channel for each sensor is shown in Fig. 7.3 and Fig. 7.4, respectively, in which 1 means the package is successfully transmitted from sensor node to FC, while 0 means the package is dropped out.

Figure 7.2: Tracking accuracy comparison

By using the Monte Carlo method, 500 times of simulation are carried out to illustrate the statistical properties of the three algorithms. The result is illustrated in Fig. 7.5. It is clearly shown in the figure that the l -banded decentralized algorithm has comparable performance as compared to the original decentralized algorithm,

Figure 7.3: Communication channel of sensor 1

but the energy consumption is less, since it transmits less number of data. For a smaller value of λ , the rate of data arriving at the FC, we can draw a similar conclusion, as shown in Fig. 7.6.

7.7 Conclusion

In this chapter, the uncertainty of the wireless communication channel was taken into consideration. Three types of transmission data were discussed. Beside the conventional raw observations transmission, transmitting local processed estimates has a large advantage that it includes all the historical information even if the communication channel is unreliable. The shortcoming of transmitting local processed estimates is the huge requirement on the bandwidth, especially when the state dimension is large. L -banded matrix approximation balances the bandwidth usage

Figure 7.4: Communication channel of sensor 2

and the information loss. A numerical example was used to illustrate the different performance among the three methods, where CKF was used as a 0-banded approximation method.

Figure 7.5: Tracking accuracy comparison, $\lambda = 0.8$ Figure 7.6: Tracking accuracy comparison, $\lambda = 0.2$

Chapter 8

Conclusion and Future Work

8.1 Conclusion

Wireless sensor network (WSN) has received increasing interest in the research community. Target tracking is considered as a canonical problem for the WSN applications. Minimization of sensor power consumption and optimal bandwidth usage are of critical importance in WSNs. This thesis has been mainly concerned with the study and improvements of power management and bandwidth allocation, as well as the tracking reliability, by developing several sensor scheduling schemes and filtering algorithms.

The effects of varying the sampling interval was explored to reduce the energy consumption in WSNs. Two sensor operation modes were introduced in adaptive sensor scheduling schemes: Fast tracking mode and Tracking maintenance mode. Two sensor scheduling schemes with adaptive sampling interval were developed and proved to be more power efficient, while not destroying the tracking accuracy requirement.

To overcome the problem caused by the limited detection range of sensor nodes,

scheduling schemes using multiple sensors were proposed. It is proved that these schemes can improve the tracking reliability as well as the tracking accuracy, which is another important performance merit in target tracking application. A multi-hop configuration was introduced to achieve higher tracking accuracy and longer tracking sustaining distance for high speed target.

In Chapter 5, two types of quantized Kalman filter were introduced, one is based on the quantized observations (QKFQO) and the other one is based on the quantized innovations (QKFQI). The approximate solutions of two optimization problems for bandwidth allocation were given in this chapter. The robustness of these two quantized filters is guaranteed since they are implemented by using the upper bounds of variances of the quantized noises. Between these two quantized filters, QKFQI has been proved to have better precision than QKFQO under the same bandwidth constraint.

In Chapter 6, a compressed Kalman filter has been proposed for target tracking in a WSN. This filter reduces the totality of transmitted data while maintaining the tracking performance. The approximation algorithm of a given square symmetric matrix by a diagonal matrix which is also the upper bound of the given matrix has also been developed. The boundedness of the proposed filter has been proved in the chapter as well. It was shown that the proposed filter offers significant energy savings, with comparable tracking performance.

In Chapter 7, we further studied the compressed Kalman filter when the wireless communication channel is unreliable. The package dropping rate affects the performance of CKF. Two types of transmission methods were discussed: one is transmitting the raw measurements and the other one is transmitting the local processed estimate and its associated covariance. A banded matrix [10] was introduced to approximate the filtering error covariance matrix $P(k|k)$. A numerical example was

finally used to illustrate the performance of these different methods.

In summary, the thesis has provided several sensor scheduling schemes and filtering algorithms aimed at achieving better energy efficiency and better bandwidth utilization, with comparable target tracking performance. Simulations were given to demonstrate the effectiveness of the proposed theoretical results.

8.2 Future Work

Some possible topics for future research are listed here.

1. When calculating the detection probability, a Monte Carlo method is proposed in those sensor scheduling schemes in Chapter 3 and Chapter 4. This imposes a heavy computation load on sensor nodes. In future, a less computational intense algorithm for computing detection probability should be developed.
2. Based on the filter proposed in Chapter 6, it is worth to consider an improved compressed Kalman filter in which the transmitted data is quantized. It is noted that for a given bandwidth, more quantization levels can be used for a diagonal (0-banded) covariance matrix. While increasing the number of transmitted data, that is, using a wider L -banded covariance matrix, the quantization will be coarser. It is worth to consider the trade off between compression and quantization. An optimal width L for the banded covariance matrix is likely to be obtained.

Author's Publications

1. **Lin Jianyong**, Xie L., & Xiao W. (2009). Target tracking in wireless sensor networks using compressed Kalman filter. *International Journal of Sensor Networks*, 6(3/4), 251-262.
2. **Lin Jianyong**, Xiao W., Lewis F. L., & Xie L. (2009). Energy-efficient distributed adaptive multisensor scheduling for target tracking in wireless sensor networks. *IEEE Transactions on Instrumentation and Measurement*, 58(6), 1886-1896.
3. Xiao W., Zhang S., **Lin Jianyong**, & Tham C. K. (2010). Energy-efficient adaptive sensor scheduling for target tracking in wireless sensor networks. *Journal of Control Theory and Applications, Special Issue on Networked Sensing, Decision Making and Control*, 8(1), 86-92.
4. **Lin Jianyong**, Xie L., & Xiao W. (2007). State-centric multi-sensor scheduling for target tracking in wireless sensor networks. In: *Proceedings of the 6th IEEE International Conference on Information, Communications and Signal*, 1-4.
5. **Lin Jianyong**, Lewis F. L., Xiao W., & Xie L. (2006). Accuracy based adaptive sampling and multi-sensor scheduling for collaborative target tracking. In: *Proceedings of the 9th IEEE International Conference on Control, Automation, Robotics and Vision*, 7-12.
6. Sun S., **Lin Jianyong**, Xie L., & Xiao W. (2007). Quantized Kalman filtering. In: *Proceedings of the 22nd IEEE International Symposium on Intelligent Control, Part of IEEE Multi-Conference on Systems and Control*, 7-12.
7. Xiao W., Xie L., **Lin Jianyong**, & Li J. (2006). Multi-sensor scheduling for reliable target tracking in wireless sensor networks. In: *Proceedings of 6th International Conference on ITS Telecommunications*, 996-1000.

Bibliography

- [1] Willsky, A. S., Bello M., Castanon D. A., Levy B. C., & Verghese G. (1982). Combining and updating of local estimates and regional maps along sets of one-dimensional tracks. *IEEE Trans. Automatic Control*, 27, 799-813.
- [2] Castanon, D. A. & Teneketzis D. (1985). Distributed estimation algorithms for nonlinear systems. *IEEE Trans. Automatic Control*, 30(5), 418-425.
- [3] Xiao J. J. & Luo Z. Q. (2005). Universal decentralized detection in a bandwidth-constrained sensor network. *IEEE Trans. Signal Processing*, 53(8), 2617-2624.
- [4] Cardei M. & Wu J. (2006). Energy-efficient coverage problems in wireless ad hoc sensor networks. *Computer Commun.*, 29, 413-420.
- [5] Chair Z. & Varshney P. K. (1988). Distributed Bayesian hypothesis testing with distributed data fusion", *IEEE Trans. Syst. Man, Cybern.*, 18, 695-699.
- [6] Luo Z. Q. (2005). An isotropic universal decentralized estimation scheme for a bandwidth constrained ad hoc sensor network. *IEEE Journal on Selected Areas in Communications*, 23(4), 735-744.
- [7] Luo Z. Q. (2005). Universal decentralized estimation in a bandwidth constrained sensor network. *IEEE Trans. Information Theory*, 51(6), 2210-2219.
- [8] Luo Z. Q. & Xiao J. J. (2004). Decentralized estimation in inhomogeneous environment. In: *Proc. IEEE International Symposium on Information Theory*, 517-520.
- [9] Luo Z. Q. & Xiao J. J. (2005). Universal decentralized estimation in inhomogeneous sensing environment", *IEEE Trans. Information Theory*, 51(10), 3564-3575.
- [10] Kavcic A. & Moura J. M. F. (2000). Matrices with banded inverses: inversion algorithms and factorization of Gauss-Markov processes. *IEEE Tran. on Information Theory*, 46(4), 1495-1509.
- [11] www.xbow.com

- [12] Xiao J., Cui S., Luo Z., & Goldsmith A. J. (2004). Joint estimation in sensor networks under energy constraints. *Sensor and Ad Hoc Communications and Networks*, 264 - 271.
- [13] Mainwaring A., Polastre J., Szewczyk R., Culler D., & Anderson J. (2002). Wireless sensor networks for habitat monitoring. In: *Proceedings of the 1st ACM Int. Workshop Wireless Sensor Networks and Applications*, 88-97.
- [14] Elson J., Girod L., & Estrin D. (2002). Fine-grained network time synchronization using reference broadcasts. In: *Proceedings of the 5th Symp. Operating Systems Design and Implementation (OSDI), Boston, MA.*, 147-163.
- [15] Klemmer S., Waterson S., & Whitehouse K. (2000). *Toward a location-based context-aware sensor infrastructure* [On-line]. Available: <http://guir.berkeley.edu/projects/location/Location.pdf>
- [16] Estrin D., Govindan R., Heidenmann J. S., & Kumar S. (1999). Next century challenges: scalable coordination in sensor networks. In: *Proceedings of the 5th Annu. ACM/IEEE Int. Conf. Mobile Computing and Networking*, 263-270.
- [17] Royer E. M. & Toh C.-K. (1999). A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Pers. Commun.*, 6, 46-55.
- [18] Zhao F., Shin J., & Reich J. (2002). Information-driven dynamic sensor collaboration. *IEEE Signal Processing Mag.*, 19, 61-72.
- [19] Guibas L. (2002). Sensing, tracking and reasoning with relations. *IEEE Signal Processing Mag.*, 19(2), 73-85.
- [20] Hintz K. & McVey E. (1991). A measure of the information gain attributable to cueing", *IEEE Trans. Systems Man and Cybernetics*, 21(2), 434-442.
- [21] Wang H., Yao K., Pottie G., & Estrin D. (2004). Entropy-based sensor selection heuristic for target localization. In: *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, 36-45.
- [22] Geman D. & Jedynak B. (1996). An active testing model for tracking roads from satellite images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(1), 1-14.
- [23] Lin C.-Y., Peng W.-C., & Tseng Y.-C. (2006). Efficient in-network moving object tracking in wireless sensor networks. *IEEE Trans. Mobile Computing*, 5(8), 1044-1056.
- [24] Owen M. W., Klammer D. M., & Dean B. (2000). Evolutionary control of an autonomous field. In: *Proceedings of the 3rd International Conference on Information Fusion*, 1, MOD1/3-MOD1/9.

- [25] Manyika J. & Durrant-Whyte H. (1994). *Data Fusion and Sensor Management: a Decentralized Information-Theoretic Approach*. New York: Ellis Horwood.
- [26] Zhao F., Liu J., Liu J., Guibas L., & Reich J. (2003). Collaborative signal and information processing: an information directed approach. In: *Proceedings of IEEE*, 91, 1199-1209.
- [27] Liu J., Reich J., & Zhao F. (2003). Collaborative in-network processing for target tracking. *Journal on Applied Signal Processing*, 378-391.
- [28] Chu M., Haussecker H., & Zhao F. (2002). Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *International Journal of High Performance Computing Applications*, 16(3), 293-313.
- [29] Xiao W., Wu J. K., & Xie L. (2005). Adaptive sensor scheduling for target tracking in wireless sensor network. In: *Proceedings of SPIE*, 104-112.
- [30] Isler V. & Bajcsy R. (2005). The sensor selection problem for bounded uncertainty sensing models. In: *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, 151-158.
- [31] Chen W.-P., Hou J. C., & Sha L. (2004). Dynamic clustering for acoustic target tracking in wireless sensor networks. *IEEE Trans. Mobile Computing*, 3(3), 258-271.
- [32] He T., Krishnamurthy S., Stankovic J. A., Abdelzaher T., Luo L., Stoleru R., Yan T., Gu L., Zhou G., Hui J. & Krogh B. (2004). VigilNet: an integrated sensor network system for energy-efficient surveillance. *ACM Trans. Sensor Networks*, 2(1), 1-38.
- [33] Xiao W., Xie L., Lin J., & Li J. (2006). Multi-sensor scheduling for reliable target tracking in wireless sensor networks. In: *Proceedings of ITS Telecommunications*, 996-1000.
- [34] Kalandros M., & Pao L. Y. (2002). Covariance control for multisensor systems. *IEEE Trans. Aerospace and Electronic Systems*, 38, 1138-1157.
- [35] Savkin A. V. (2005). Analysis and synthesis of networked control systems: topological entropy, observability, robustness and optimal control. *Automatica*, 42, 51-62.
- [36] Stilwell D. J. & Bishop B. E. (2000). Platoons of underwater vehicles. *IEEE Control Systems Magazine*, 20(6), 45-52.
- [37] Nair G. N. & Evans R. J. (2003). Exponential stabilisability of finite-dimensional linear systems with limited data rates. *Automatica*, 39(4), 585-593.
- [38] Nair G. N. & Evans R. J. (2002). Mean square stabilisability of stochastic linear systems with data rate constraints. In: *Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, Nevada*, 2, 1632-1637.

- [39] Nair G. N. & Evans R. J. (2000). Stabilization with data-rate-limited feedback: tightest attainable bounds. *Systems and Control Letters*, 41(1), 49-56.
- [40] Tatikonda S. C. (2000). *Control under communication constraints*. Ph.D. thesis, Massachusetts Institute of Technology.
- [41] Tatikonda S. & Mitter S. (2004). Control over noisy channels. *IEEE Transactions on Automatic Control*, 49(7), 1196C1201.
- [42] Kim Y. H. & Ortega A. (2005). Quantizer design and distributed encoding algorithm for source localization in sensor networks. In: *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, 231-238.
- [43] Kim Y. H. & Ortega A. (2005). Quantizer design for source localization in sensor networks. In: *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 4, 857-860.
- [44] Ribeiro A., Giannakis G. B., & Roulmeliotis S. I. (2006). SOI-KF: distributed Kalman filtering with low-cost communications using the sign of innovations. *IEEE Trans. Signal Processing*, 54(1), 2617-2624.
- [45] Cui S., Goldsmith A. J., & Bahai A. (2004). Energy-constrained modulation optimization. *IEEE Transactions on Wireless Communications*, 4(5), 2349-2360.
- [46] Cui S., Goldsmith A. J., & Bahai A. (2004). Joint modulation and multiple access optimization under energy constraints. In: *Proceedings of IEEE Global Telecommunications Conference 2004*, 1, 151-155.
- [47] Xiao J.-J., Cui S., Goldsmith A. J., & Luo Z.-Q. (2004). Joint estimation in sensor networks under energy constraints. In: *Proceedings of IEEE 1st Annual Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, 264-271.
- [48] Krasnopeev A., Xiao J.-J., & Luo Z.-Q. (2005). Minimum energy decentralized estimation in sensor network with correlated sensor noise. In: *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing 2005*, 3, 18-23.
- [49] Bar-Shalom Y., Li X. R., & Kirubarajan T. (2001). *Estimation with applications to tracking and navigation*. New York: John Wiley & Sons.
- [50] Bhardwaj M. & Chandrakasan A. P. (2002). Bounding the lifetime of sensor networks via optimal pole assignments. In: *Proceedings of INFOCOM*, 1587-1596.
- [51] Ye X. Y. (2007). *Power Monitoring for Wireless Sensor Networks*. Final Year Project Report, School of Electrical & Electronic Engineering, Nanyang Technological University.

- [52] Miller M. J. & Vaidya N. H. (2005). A MAC protocol to reduce sensor network energy consumption using a wakeup radio. *IEEE Trans. Mobile Computing*, 4(3), 228-242.
- [53] Zou Y. & Chakraharty K. (2005). A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks. *IEEE Transactions On Computers*, 54(8), 978-991.
- [54] Liu J., Liu J., Reich J., Cheung P., & Zhao F. (2003). Distributed group management for track initiation and maintenance in target localization application. In: *Proceedings of 2nd Int. Workshop Information Processing in Sensor Networks*, 113-128.
- [55] Krasnopeev A., Xiao J. J., & Luo Z. Q. (2005). Minimum energy decentralized estimation in a wireless sensor network with correlated sensor noises. *Journal on Wireless Communication and Networking*, 4, 473-482.
- [56] Lin J., Lewis F., Xiao W., & Xie L. (2006). Accuracy based adaptive sampling and multi-sensor scheduling for collaborative target tracking. In: *Proceedings of 9th International Conference on Control, Automation, Robotics and Vision*, 1-6.
- [57] Xiao J.-J., Ribeiro A., Luo Z.-Q., & Giannakis G. B. (2006). Distributed compression-estimation using wireless sensor network. *IEEE Signal Processing Magazine*, 23(4), 27-41.
- [58] Schurgers C., Tsiatsis V., & Srivastava M. B. (2002). STEM: topology management for energy efficient sensor networks. In: *Proceedings of IEEE Aerospace Conference 2002*, 3, 1099-1108.
- [59] Hong Y.-W. & Scaglione A. (2006). Energy-efficient broadcasting with cooperative transmissions in wireless sensor networks. *IEEE Trans. Wireless Commun.*, 5(10), 2844-2855.
- [60] Lin J., Xiao W., Lewis F., & Xie L. (2009). Energy efficient distributed adaptive multi-sensor scheduling for target tracking in wireless sensor networks. *IEEE Transactions on Instrumentation and Measurement*, 58(6), 1886-1896.
- [61] Sun S., Lin J., Xie L., & Xiao W. (2007). Quantized Kalman filtering. In: *Proceedings of 22nd IEEE International Symposium on Intelligent Control*, 7-12.
- [62] You K., Xie L., Sun S., & Xiao W. (2008). Multiple-level quantized innovation Kalman filter. In: *Proceedings of the 17th International Federation of Automatic Control*, 1420-1425.

- [63] Schizas I. D., Giannakis G. B., & Luo Z.-Q. (2006). Optimal dimensionality reduction for multi-sensor fusion in the presence of fading and noise. In: *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 2006*, 4, IV-IV.
- [64] Horn R. A. & Johnson C. A. (1985). *Matrix Analysis*. Cambridge University Press.
- [65] Kailath T., Sayed A., & Hassibi B. (2000). *Linear Estimation*. NJ: Prentice Hall.
- [66] Xiao W., Wu J. K., Shue L., Li Y., & Xie L. (2006). A prototype ultrasonic sensor network for tracking of moving targets. In: *Proceedings of the 1st IEEE Conference on Industrial Electronics and Applications*, 1-6.
- [67] Sinopoli B., Schenato L., Franceschetti M., Poolla K., Jordan M. I., & Sastry S. S. (2004). Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9), 1453-1464.
- [68] Sinopoli B., Schenato L., Franceschetti M., Poolla K., & Sastry S. S. (2005). An LQG optimal linear controller for control systems with packet losses. In: *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005, Seville, Spain*, 458-463.
- [69] Shi L., Epstein M., Tiwari A., & Murray R. M. (2005). Estimation with information loss: asymptotic analysis and error bounds. In: *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005, Seville, Spain*, 1215-1221.
- [70] Micheli M. (2001). *Random sampling of a continuous-time stochastic dynamical system: analysis, state estimation, applications*. M.S. thesis, Dept. Elect. Eng., Univ. California, Berkeley.
- [71] Micheli M. & Jordan M. I. (2002). Random sampling of a continuous-time stochastic dynamical system. In: *Proceedings of the 15th Int. Symp. Mathematical Theory of Networks and Systems (MTNS), South Bend, IN*.
- [72] Gupta V., Hassibi B., & Murray R. M. (2005). On sensor fusion in the presence of packet-dropping communication channels. In: *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005, Seville, Spain*, 3547-3552.
- [73] Liu X. & Goldsmith A. (2004). Kalman filtering with partial observation losses. In: *Proceedings of the 43th IEEE Conference on Decision and Control*, 4, 4180-4186.
- [74] Sun S.-L. & Deng Z.-L. (2004). Multi-sensor optimal information fusion Kalman filter. *Automatica*, 40(6), 1017-1023.

-
- [75] Xu N. S. (2001). *Stochastic Signal Estimation and System Control*. Beijing: Beijing Industry University Press.