

Chapter 1

Introduction

A wireless sensor network (WSN) consists of a large number of small, low-power, and inexpensive sensor nodes with on-board sensing, processing and wireless communication capabilities. These sensor nodes are networked and deployed “on the ground, in the air, under water, on bodies, in vehicles, and inside buildings” [1] to collaboratively monitor and perceive surrounding phenomena. Driven by advancements in semiconductor fabrication, system-on-a-chip (SoC) integration, wireless communication and signal processing techniques, the wireless sensor network has recently emerged as a new discipline [1]–[8]. Applications of wireless sensor networks span a broad range – military surveillance, condition-based equipment maintenance, environmental and habitats monitoring, health care, and disaster prevention and relief ... etc [2], [9]–[14]. It is envisioned that large-scale distributive wireless sensor networks will eventually instrument our entire planet and revolutionise the way in which we interact with the physical world [2]–[3].

The increasing popularity and deployment of wireless sensor networks have inspired the research community to explore various aspects of wireless sensor networks, including: sensor node hardware design; sensor network deployment and sensor nodes localisation; sensor network synchronisation; distributed estimation; medium access control (MAC); routing; sensor network tasking and querying; and modelling and inference ... etc [8]. Nevertheless, it remains to be a challenging and non-trivial task to design wireless sensor networks. Such challenges arise from the unique characteristics of wireless sensor networks, especially the highly dynamic topology and connectivity of sensor networks, and the constrained energy resource and communication bandwidth available for individual sensor nodes [2]–[3], [15]. To address these challenges, on the one hand, it is necessary to develop the *information processing* techniques that cater for representing, processing, storing and fusing data and information in a distributive manner in the wireless sensor networks; and on the other hand, there is also a need for developing the *networking techniques* that is concerned with sensor network discovery, control and information delivery (i.e., the routing protocol) as well as wireless channel management (i.e. the communication

protocol) to facilitate information processing. To conserve communication and computation resources and achieve scalability and adaptability, it is essential to blend information processing design with networking design in wireless sensor networks.

This thesis is concerned with the development of collaborative information processing techniques for estimating the state (i.e. position) of the moving ground vehicle (hereinafter referred to as “target tracking”) in the context of wireless sensor networks. The thesis aims to provide an understanding of the interaction between the information processing and the networking in wireless sensor networks. Based on such understanding, a collaborative information processing framework which integrates distributive estimation, hybrid communication protocol and hierarchical routing scheme is proposed in this thesis. A suite of estimation algorithms for tracking both single and multiple targets is developed. The emphases are put on coordinating sensor nodes for the target state estimation under measurement origin uncertainty and resources constraint in wireless sensor networks. Throughout this thesis, it is assumed that the target tracking is performed in a wireless sensor network which only consists of the stationary acoustic sensor nodes.

This chapter is organized as follow. Section 1.1 points out the motivation of this thesis. Section 1.2 overviews the target tracking problem in the context of wireless sensor networks. Section 1.3 summarizes the major contributions of this thesis. Section 1.4 presents the organization of the thesis.

1.1 Motivation

The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance. However, wireless sensor networks are now applied in many areas, including environment and habitat monitoring, healthcare, home automation, and traffic control. Among these applications, target tracking is a canonical application and has received increasing attention in the literature [1], [9], [16]–[26]. A typical single target tracking scenario is depicted in Figure 1.1 in which a large number of densely deployed sensor nodes co-operatively measure and estimate to determine the state of the target (*e.g.*, the position, velocity and heading of the target) over a sequence of time steps. In multiple target tracking, a wireless sensor network is tasked to infer the states of multiple targets from information gathered by the sensor nodes at each time step.

Unlike target tracking algorithms¹ developed for centralised platforms (*e.g.* radar, sonar...etc) [1], [27], the target tracking algorithms developed for wireless sensor networks need to be decentralised and energy efficient due to the highly dynamic network topology arising from the deployment of numerous volume of sensor nodes; and the severely constrained communication and computation resources of each individual sensor node.

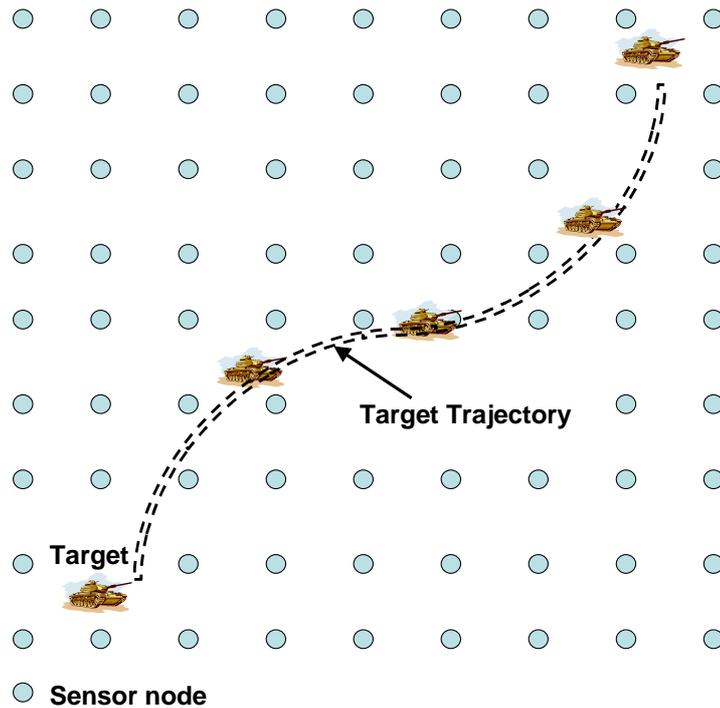


Figure 1.1 Illustrative target tracking in a wireless sensor network

In the past few years, a number of techniques have been developed for target tracking in wireless sensor networks, including the maximum likelihood estimation [16], the graph based method [89], the Kalman filter (KF) [106], and the sequential Monte Carlo (SMC) approach [18]–[20]. However, significant issues still remain: estimating target state under measurement origin uncertainty due to clutter which are arising from multi-path effects, sensor errors and spurious objects; managing the identities of multiple targets; approximating and propagating the estimation results amongst sensor nodes within the network; scheduling a group of sensor nodes based on the task requirement and resource

¹ In this thesis, the term “target tracking” includes both single and multiple target tracking.

availability; managing wireless channel for communication among sensor nodes; and routing information between sensor nodes.

The need to study the above issues motivates the research reported in this thesis. In the thesis, the term “*collaborative information processing*” refers to a unified approach that jointly addresses *information processing* and *networking* for the distributive estimation of target state in the highly dynamic and resources constrained wireless sensor networks. The collaborative information processing is the integration of three closely related components: the distributive estimation of target’s state under measurement origin uncertainty; the hierarchical routing in highly dynamic environment; and the self-organised hybrid communication on the dynamic basis. Each of the above components addresses one important aspect for target tracking application in wireless sensor networks; but on the other hand, these components are complementary to each other. The distributive estimation component is responsible for coordinating sensor nodes that participate in the tracking task, balancing the information contribution of individual sensor node against its resource consumption, acquiring necessary but non-redundant pieces of data and information, and making the estimation of the target state. The routing component and the communication component provide the distributive estimation component with an efficient and accurate information exchange backbone by forming the routes for information delivery and managing the access to the wireless channel.

1.2 Target Tracking in Wireless Sensor Networks

This section provides an overview of target tracking in wireless sensor networks. It consists of a description of the generic target tracking scenario in wireless sensor networks, a brief introduction of various target tracking approaches, and a highlight of the research works conducted in this thesis. However, the detail review of related target tracking techniques for wireless sensor network will be provided in Chapter 2.

Figure 1.2 depicts a generic scenario of target tracking in a wireless sensor network in which two vehicles traverse a sensor field simultaneously. In some regions of the sensor field (region I in Figure 1.2), two vehicles are spatially well separated and the tracking problem can be regarded as the tracking of two single targets. However, when two vehicles are closely-spaced (region II in Figure 1.2), it becomes a multiple target tracking problem that the states of the two targets must be estimated jointly. The target tracking problem, as well as the algorithms developed to tackle the target tracking problem will be discussed in detail in the following chapters.

Tracking Targets in a Wireless Sensor Network

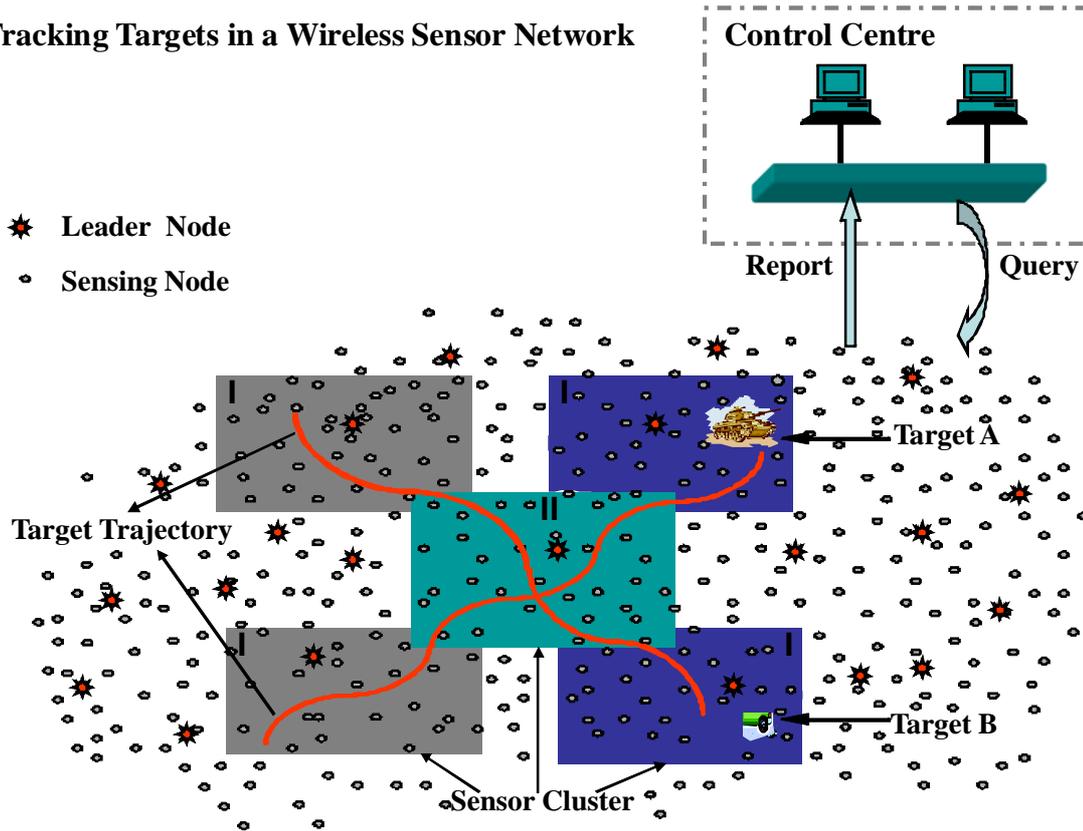


Figure 1.2 Target tracking in wireless sensor networks

During the past four decades, researchers have developed various tracking algorithms [27], [28]. However, these algorithms are mostly for centralised platforms (*e.g.*, radar, sonar etc) and their applicability to wireless sensor networks is unknown. Tracking algorithms developed for wireless sensor networks need to be *distributive* and, more importantly, *energy awareness* needs to be incorporated into every aspect of algorithm design in the wireless sensor networks.

The most frequently adopted framework for target tracking is the recursive Bayesian estimation. In recursive Bayesian estimation framework, target tracking is regarded as the process of estimating the target state (*i.e.*, position, velocity, and attitude ... *etc*) of a dynamic system (linear or non-linear) from noisy measurement data (Gaussian or non-Gaussian noise). It only requires a system model that defines the target dynamics, a measurement mode that relates the measurements to target state, and an initial probability distribution (prior knowledge) of the target state. The general recursions update the probability density function of the target state through two steps: a prediction step that uses the system model to propagate the probability density function of target state at the previous time step to form the prediction; and a correction step that incorporates the latest

measurement through Bayes' rule to form the new probability density function of the target state at the current time step.

For the linear and Gaussian problem, the above recursion can be computed analytically by the standard Kalman filter (KF) [29]–[31]. However, for non-linear problems, there are no closed form solutions and approximation methods need to be used. The well-known extended Kalman filter (EKF) linearises the state space model through the use of a first order truncated Taylor series expansion around the current estimate of the system state [29]. Instead of adopting analytical Taylor series linearisation as used in the EKF, there is also a number of Gaussian approximate derivative-free filters using deterministic sampling methods for the propagation of random variables through nonlinear systems [32]–[37]. Examples include the unscented Kalman filter (UKF) [33], the central difference filter (CDF) [34], and the sigma-point Kalman filters (SPKFs) [35]. Furthermore, there also exists another group of approximate method that is known as the sequential Monte Carlo Method (SMC) or Particle filter (PF) [38]–[45]. This approximation approach represents the distribution of the random variables by empirical point mass approximations and these are recursively updated using sequential importance sampling and resampling [38]. There also exist some hybrid approaches that integrate EKF, UKF or SPKF into PF to attain better trade-off between accuracy and efficiency [35], [46].

In most practical target tracking applications in wireless sensor networks, the sensor nodes may yield unlabelled measurements due to clutter and missed detections. Moreover, multiple targets, which are not sufficiently separated temporally and spatially in the sensor field, may also lead to unlabelled measurements at sensor nodes. Such measurement origin uncertainty leads to the challenging *data association problem*. In the target tracking literature, a large number of strategies are available to solve the data association problem on centralised platforms [27], [47]. The multiple hypotheses tracker (MHT) records all the possible association hypotheses over time. This approach, however, is at the expense of very heavy computation burden since the number of association hypotheses grows exponentially over time [27]. To control the number of association hypotheses, the joint probabilistic data association filter (JPDAF) applies a gating procedure to remove infeasible hypotheses. A filtering step is then conducted for each of the remaining hypotheses, which are combined in proportion to the corresponding posterior hypotheses probabilities [47]. The probabilistic multiple hypotheses tracker (PMHT) assumes the association variables to be independent to tackle the problem of gating [27]. Recently,

strategies have been proposed to combine the JPDAF and PMHT with sequential Monte Carlo method (SMC) for nonlinear applications [48], [49].

In this thesis, various techniques for target tracking are explored. To track a single target, a number of algorithms have been developed: the sequential extended Kalman filter (S-EKF), the sequential unscented Kalman filter (S-UKF), the Particle filter (PF), and the novel hybrid extended Kalman and Particle filter (EKPF). To tackle the data association problem due to measurement origin uncertainty, novel hybrid algorithms have also been developed which integrate Particle filter (PF) with probability density association filter (PDAF) for single target tracking, while integrating with joint probabilistic data association filter (JPDAF) for multiple target tracking in wireless sensor networks. By adopting the hierarchical sensor network architecture and utilising the Gaussian mixture model (GMM), distributive PF, EKPF and PF-PDAF algorithms are developed for tracking single target while distributive PF-JPDAF algorithm is developed for tracking multiple targets. Moreover, for single target tracking in wireless sensor networks, a recursive posterior Cramer-Rao lower bound (PCRLB) under measurement origin uncertainty is calculated to provide a theoretical lower bound on mean square error of the target state estimation to which the tracking algorithm can attain [50]. PCRLB will also facilitate sensing nodes selection in target tracking algorithm; the sensing nodes selection is based on the optimization of a composite objective function incorporates the information measure which is on the basis of PCRLB calculation and the energy consumption measure which is decided by the positions of the sensing nodes.

1.3 Thesis Contributions

This section summarizes the major contributions made in this thesis. Taking into account the interplay between information processing and networking, this thesis proposed a collaborative information processing framework to fulfil the tasks of distributive tracking both single and multiple target under measurement origin uncertainty in wireless sensor networks. The framework integrates the information processing which is responsible for the representation, fusion and processing of data and information with networking (i.e. communication protocol and routing protocol) which caters for the formation of network, the delivery of data and information, and the management of wireless channels.

The first key contribution of this thesis is to propose a collaborative information processing framework for target tracking in wireless sensor networks. This

framework provides a unified approach that jointly addresses information processing and networking for the distributive estimation of target state in the highly dynamic and resources constrained wireless sensor networks.

Within the collaborative information processing framework, this thesis developed a suite of algorithms for single target tracking in wireless sensor networks, including the S-EKF, the S-UKF, the PF, and the EKPF. The purpose is to develop appropriate tracking algorithms that are computationally accurate, efficient, consistent and suitable for distributed implementation for single target tracking in wireless sensor networks.

The second key contribution of this thesis is the development of PF and a novel EKPF tracking algorithms which use discrete samples (particles) to represent the probability density function of the target state and hence they can accommodate nonlinear measurement models. Moreover, the novel EKPF algorithm integrating EKF into PF is shown to further improve the tracking accuracy.

Given the limited energy and communication bandwidth of individual sensor nodes, a critical consideration in the design of wireless sensor networks is that most of the information processing and exchange must take place at a local level (*e.g.*, within a cluster of sensor nodes) to reduce the communication overhead. By adopting the hierarchical network architecture to achieve dynamic sensor nodes clustering and utilizing the Gaussian mixture model (GMM) to propagate estimation results amongst sensor clusters, this thesis developed distributive PF, EKPF and PF-PDAF for single target tracking and the distributive PF-JPDAF for multiple target tracking in wireless sensor networks.

The third key contribution of this thesis is the development of distributive tracking algorithms for wireless sensor networks. This helps to achieve scalability and adaptability as well as energy efficiency in highly dynamic and energy constrained wireless sensor networks.

In this thesis, the data association problem is tackled when tracking a single target in the cluttered environment or when tracking multiple targets that are closely spaced. The well-known PDAF and JPDAF are extended for target tracking under measurement origin uncertainty in wireless sensor networks. Especially, the particles are adopted to represent

the probability density function of the target state, thus the resulting PF-PDAF and PF-JPDAF algorithms are suitable for general nonlinear single target and multiple target tracking in wireless sensor networks.

The fourth key contribution of this thesis is the adoption of particles to represent the probability density function of the target state for the development of hybrid tracking algorithms including PF-PDAF and PF-JPDAF for both single and multiple target tracking under measurement origin uncertainties in wireless sensor networks.

In wireless sensor networks, one of the most important concerns is the maximising the tracking accuracy while operating within constraints on sensor network resources. In this thesis, a sensing nodes selection scheme is designed to allocate sensor network resources by optimising a composite objective function which incorporates the information measure on the basis of the calculation of posterior Cramer-Rao lower bound (PCRLB) and the resource consumption measure which is based on the locations of sensing nodes.

The fifth key contribution of this thesis is the development of a sensing nodes selection scheme in the distributive tracking algorithm that properly allocates sensor network resources to achieve desirable tracking performance while maintaining modest resources consumption in the target tracking task.

1.4 Thesis Organisation

This section presents the organisation of the thesis. Chapter 2 provides an overview of wireless sensor networks. It presents the concepts, applications and characteristics of wireless sensor networks. It identifies design challenges and surveys the approaches to these challenges with the emphases on two closely related design issues: information processing and networking. Since target tracking is one of the most important applications in wireless sensor networks and is also the central theme of this thesis, Chapter 2 also provides a detailed review of strategies and techniques reported in the literature for target tracking in the context of wireless sensor networks.

Chapter 3 proposes a collaborative information processing framework for target tracking in wireless sensor networks. This collaborative information processing framework includes

three major components – the distributive estimation of target state; the hierarchical routing; and the self-organised hybrid communication. Each of the components addresses one important aspect for target tracking application in wireless sensor networks; the combination of these components provides an integrated solution for target tracking in wireless sensor networks. Chapter 3 describes the functional blocks for each of the above three components, and especially, details the design of hierarchical routing protocol and hybrid communication scheme.

Chapter 4 develops techniques and algorithms for tracking a single target in wireless sensor networks. On the basis of the recursive Bayesian estimation, a number of tracking algorithms, namely S-EKF, S-UKF, PF and EKPF are developed in Chapter 4. Making use of the advantages of both Gaussian approximate filters (e.g., EKF) and sample-based approximation filters (e.g., PF), the novel hybrid EKPF algorithm is shown in Chapter 4 to consistently outperform S-EKF, S-UKF and PF algorithms. To determine the theoretical performance bound to which the tracking algorithms could attain, Chapter 4 computes the posterior Cramer-Rao lower bound (PCRLB) to benchmark the performance of the developed tracking algorithms. Moreover, extensive simulations are carried out in Chapter 4 to assess the performance of the developed tracking algorithms.

Chapter 5 addresses the problem of tracking a single target under measurement uncertainty due to clutter and missed detections in wireless sensor networks. By using particles to represent the probability density function of the target state, a particle filter (PF) and probabilistic data association filter (PDAF) hybrid tracking algorithm, referred to as the PF-PDAF is developed in Chapter 5. This PF-PDAF algorithm extends the well-known PDAF to the general nonlinear state-space model. Furthermore, the posterior Cramer-Rao lower bound (PCRLB) computed in Chapter 4 is also extended to accommodate clutter and missed detections in Chapter 5, and thus it can also provide a theoretical lower bound on the tracking performance under measurement origin uncertainty.

Chapter 6 develops the distributive tracking algorithms for tracking a single target in wireless sensor networks. On the basis of the hierarchical sensor network architecture and the dynamic sensor nodes clustering developed in Chapter 3 as well as the Gaussian mixture model (GMM), the PF and EKPF tracking algorithms developed in Chapter 4, and the PF-PDAF tracking algorithms developed in Chapter 5 are implemented in a distributive manner in Chapter 6. A composite objective function is also developed to facilitate the sensing nodes selection in the above distributive tracking algorithms. This composite objective function incorporates the information measure which is based on PCRLB and the

energy consumption measure which is decided by the positions of the sensing nodes. By adopting this composite object function, the distributive tracking algorithms develop in Chapter 6 can properly allocate sensing, computing and communication resources to attain desirable performance while still maintaining modest energy consumption when tracking a single target in wireless sensor networks. The distributive tracking algorithms developed in Chapter 6 will further enhance the applicability of the algorithms developed in Chapter 4 and Chapter 5.

Chapter 7 addresses the problem of tracking multiple targets under measurement origin uncertainty in the wireless sensor networks. By adopting the particles' representation of the probability density function of the target state, a particle filter (PF) and joint probabilistic data association filter (JPDAF) hybrid tracking algorithm, named as PF-JPDAF tracking algorithm is developed. PF-JPDAF combines the advantage of PF that can be applied to the general nonlinear problem with the merit of JPDAF that can effectively tackle the challenging data association problem. Chapter 7 also conducts extensive simulations on various multiple target tracking scenarios in wireless sensor works.

Chapter 8 concludes the thesis and suggests several directions for future research.

Chapter 2

Overview

This chapter is an overview of wireless sensor networks. It presents the concept, applications and characteristics of wireless sensor networks. It identifies design challenges and surveys the approaches to these design challenges with the emphases on two closely related design issues, the networking design and the information processing design. Since target tracking is one of the most important applications in wireless sensor networks and is also the central theme of this thesis, this chapter provides a detailed review of strategies and techniques reported in the literature for target tracking in the context of wireless sensor networks.

This chapter is organized as follows. Section 2.1 briefly describes the benefits and the applications of wireless sensor networks. Section 2.2 presents the characteristics of wireless sensor networks, and summarises the research challenges from three design perspectives: sensor node hardware design; networking design; and information processing design. Section 2.3 surveys the technical approaches to the above design challenges. Section 2.4 zooms to the current efforts in applying target tracking techniques to wireless sensor networks and supplies a survey on related research works reported in the literature.

2.1 Benefits and Applications of Wireless Sensor Networks

The deployment of wireless sensor networks offered significant benefits: the higher spatial resolutions and extended coverage; the greater fault-tolerance and robustness; and the reduced total cost because of the small size and lower price of sensor nodes [2], [3], [15]. These benefits lie in the fact that the sensor nodes are densely deployed in locations close to the observed phenomena to acquire the most relevant and reliable data and information, and through the data and information exchange between them, these sensor nodes are capable of building up an integrated understanding of the observed phenomena.

The significant benefits of wireless sensor networks attract a wide range of applications. To date, some wireless sensor networks applications have already been deployed into the real environment and even more applications have been envisioned [2], [3], [8], [15]. These applications can be roughly grouped into the following categories:

- The wireless sensor networks are applied in the detection, estimation, localisation, classification and tracking of moving objects. Examples can be found in [1], [5], [9].
- The sensor nodes are spatially distributed to convey the surrounding environment in a timely manner. This includes ecosystem monitoring, assembly line and machine monitoring, inventory control and logistics, and disaster detection ... etc. Examples include deploying wireless sensor network to monitor the parterres of petrels' nesting burrows [11], and using wireless sensor networks to investigate moisture contents in soil monitoring [51].
- The wireless sensor networks are equipped with mobility, for example, combining wireless sensor networks, robotics and/or unmanned aerial vehicles (UAVs) to provide new capabilities in various applications [52], [53].

It can be predicted that in the near future the wireless sensor networks technology will provide people with the capability of remote interaction with the physical world anywhere and at anytime, just as today's Internet allows people to access digital information anywhere and at anytime.

2.2 Characteristics and Research Challenges in Wireless Sensor Networks

Wireless sensor networks have a number of unique characteristics that make them inherently different from traditional data networks [2]–[5]. These characteristics may be divided into two major categories.

The first category is concerned with the pervasive nature of wireless sensor networks. In some applications, there may be several hundreds and even up to thousands of sensor nodes that are densely scattered over a geographical area. Each sensor node might only acquire a fragment of data and information, it is therefore vital to coordinate all sensor nodes or a cluster of sensor nodes to acquire the necessary pieces of data and information, process these data and information instantaneously, and subsequently transform them into meaningful knowledge regarding the observed physical phenomena and finally make this knowledge available to users (be they humans or machines). Moreover, sensor nodes are prone to failure due to harsh deployment environments. Hence, in contrast to traditional infrastructure-based communication networks, the connectivity and topology of wireless sensor networks are always evolving and highly dynamic. This requires scalability and adaptability to be built into wireless sensor networks and implies the needs of novel networking design and information processing design.

The second category deals with the significant resource constraints (e.g., the limited power supply, communication bandwidth and computing ability of the individual sensor node) on wireless sensor network operation. Energy efficiency is a critical issue and energy awareness needs to be incorporated into every aspects of wireless sensor networks development. However, developing energy efficient algorithms in a highly distributed environment is a non-trivial task and requires the consideration of multiple trade-offs between performances and resource utilisation in terms of communication and computing.

The above characteristics pose considerable research challenges to the design, development and deployment of wireless sensor networks: connecting and managing of sensor nodes within a communication network in scalable and energy efficient ways; and representing, fusing, processing and propagating data and information in a distributive manner while minimising the use of resources. Following sections will provide a short discussion of these challenges from three major design perspectives: sensor node design, information processing design, and networking design (i.e. routing protocol and communication protocol). Figure 2.1 depicts the relationship of these three design aspects within the integrated functions of sensing, processing, routing and communication in a wireless sensor network.

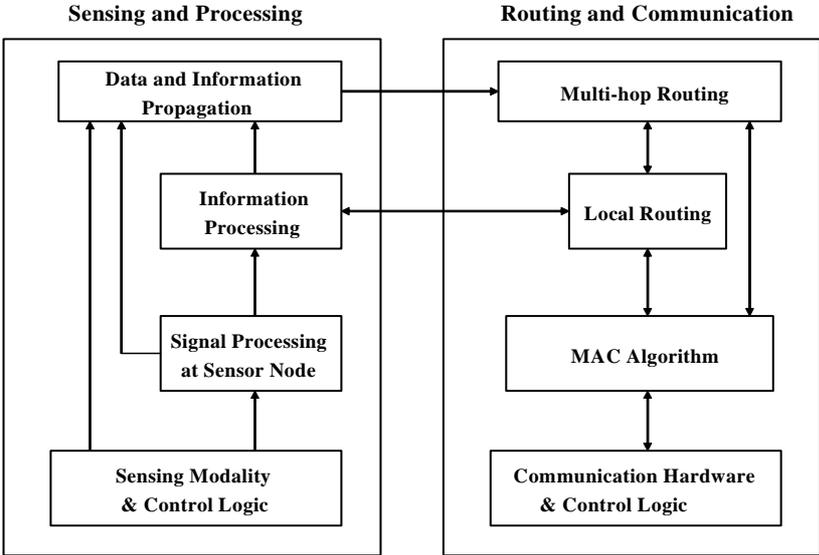


Figure 2.1 Sensing, processing, routing and communication in a wireless sensor network

2.2.1 Research Challenges in Sensor Node Design

Generally, a sensor node integrates following components: (1) a sensing unit which is equipped with one or more sensing modalities to perceive the physical world; (2) a processing unit for computing and management; (3) a communicating unit for data transmission and reception; (4) a power supply; and (5) an operating system (software) for the operation and control of whole sensor node [54]–[60]. Developed sensor nodes include Motes [54], WINS [4], Smart-Its [55], piconodes [56], NICTOR WSN Platform [57], FLECKs [154] and Zigbee Wireless Microcontroller [58] [59] ... etc.

In many applications of wireless sensor networks, the sensor nodes are un-attended in possibly hostile environments and hence it is generally impractical and even impossible to charge or replace the sensor nodes. Therefore, one critical requirement in sensor node design is the low power consumption. One common adopted approach to this requirement is based on a low duty cycle operation mechanism: for the majority of time, the sensor node is asleep; when the response is needed, the sensor node quickly wakes up and starts processing; and after carrying out its duty, the sensor nodes returns to sleep again. Various electronic circuit designs to achieve this low duty cycle operations in sensor nodes have been reported in the literature [54]–[59]. However, the full review of these works is out of the scope of this chapter.

2.2.2 Research Challenges in Networking Design

Networking design is concerned with wireless sensor network discovery, formation, control, communication and data and information routing. It develops protocols and algorithms for the data link layer and the network layer in the protocol stack [5], [61]. The networking design plays a pivotal role for a wireless sensor network to achieve self-organisation, scalability and adaptability.

In a wireless sensor network, the first task of networking design is to build up the network connectivity for wireless communication among the large number of densely scattered sensor nodes, and to pave the way for fair and efficient sharing of communication resources amongst sensor nodes. This task belongs to the medium access control (MAC) design in the data link layer [63]. In the wireless sensor networks, the chief design challenges of MAC are energy efficiency due to the limited energy resources of individual sensor nodes, and the scalability and adaptability arising from the frequently changing sensor network topologies.

The second task of networking design for wireless sensor networks is to develop routing protocols to carry out network discovery, formation and maintenance for data and information delivery within the network. However, designing routing protocols in wireless sensor networks is very challenging [5], [62]. First, traditional IP based protocols may not be applied in wireless sensor networks due to the relatively large number of sensor nodes. Second, sensor nodes are highly resource constrained and energy awareness needs to be incorporated into routing protocols. Third, position awareness of sensor nodes is important in the routing protocols design since information collection is normally based on the location. Finally, routing protocols need to exploit data aggregation to improve efficiency of resources utilisation.

2.2.3 Research Challenges in Information Processing Design

In a wireless sensor network, the information processing design caters for the extraction, representation, manipulation and propagation of data and information. It aims to provide a scalable and efficient way to process the high volume of spatially and temporally distributed data generated by a large number of sensor nodes while still maintaining the modest utilisation of resources consumption [5], [18]. The central theme of this thesis falls into this category of design.

The challenges of information processing design include the organisation of the participating sensor nodes into different clusters; the management of the identities of multiple concurrent physical phenomena; the estimation of the states of physical phenomena; and the approximation and propagation of the estimation results. Important research issues include information aggregation, storage and retrieval, information sharing between sensor nodes, estimation algorithms for the inference of physical phenomena, information querying and tasking methods, and information compression and replication.

2.2.4 The Interplay between Networking Design and Information Processing Design

The above networking and information processing are two closely related design issues in wireless sensor networks. Information processing algorithms require information dissemination and aggregation among sensor nodes. Quite clearly, this requires both the MAC and routing algorithms are in place. On the other hand, by exploiting information processing techniques, MAC and routing can be done in more energy-efficient ways. Some network parameters such as the residual energy of sensor nodes and the channel state of network can be conveyed and provided by information processing algorithms to enhance the performance of the MAC and routing protocols [64]. Effectively and seamlessly

integrating networking design and information processing design can lead to an efficient usage of limited resources and an improved network performance.

2.3 Literature Survey of Research Works in Wireless Sensor Networks

This section provides a literature survey focusing on the research works of networking and information processing design. Next section will survey the research works on target tracking in wireless sensor networks.

2.3.1 The Survey of MAC Protocol Design

MAC protocol manages the access to the shared communication medium by defining rules that allow sensor nodes to communicate with each other orderly and efficiently. It has been extensively studied in traditional wireless voice and data communications. There are two different types of MAC protocols [63]: the fixed-assignment channel access method which assign nodes onto different sub-channels to avoid collision, including time-division multiple access (TDMA), frequency-division multiple access (FDMA), code-division multiple access (CDMA), and space-division multiple access (SDMA); and the contention-based channel-access method in which nodes compete for the wireless communication channel, examples include carrier-sense multiple access (CSMA) and IEEE 802.11 standard.

Since wireless sensor networks differ from traditional wireless voice or data networks, MAC for wireless sensor networks also carries its own characteristics. The primary concerns of MAC in the context of wireless sensor networks is to attain energy efficient so as to prolong the lifetime of entire network; and also to achieve scalability and adaptability to suit for the highly dynamic network topology [63], [71].

In the past few years, many researchers have extended the MAC protocol for wireless sensor networks. Ye *et al.* [65] developed a Sensor-MAC (S-MAC) which is a hybrid of CSMA and TDMA. In S-MAC, the synchronized time sub-channels are maintained, however, unlike pure TDMA, these sub-channels can be much bigger than normal TDMA sub-channels and synchronization failures do not necessarily lead to communication failure because the handshakes mechanism (i.e. sender sends Request-to-send (RTS) message before transmitting message and receiver sends Clear-to-send (CTS) message if it is available) is employed. S-MAC adopts the periodic sleep-listen schedule based on a locally managed synchronisation. Neighbouring sensor nodes form virtual clusters to set up a common sleep schedules and this helps to reduce control overhead and enable traffic-

adaptive wake-up. Moreover, S-MAC adopts in-channel signalling to avoid overhearing unnecessary traffic. Enz *et al.* [66] proposed WiseMAC which is a single-channel contention MAC based on non-persistent CSMA (np-CSMA). To mitigate idle listening which is one of the major energy wastes in wireless channel, WiseMAC combines np-CSMA with preamble sampling techniques. The preamble sampling technique consists of regularly sampling the channel to check for activity. At the sender side, a wake-up preamble of size equal to the sampling period is transmitted ahead of every data packet to ensure that the receiver will be awake when data transmission begins. At the receiver side, if it finds the channel is busy, it continues to listen until it receives a data packet or the channel becomes idle again. By dynamically adjusting the length of the preamble, WiseMAC can achieve better performance under variable traffic conditions. Woo and Culler [67] investigated various configuration of CSMA. They proposed an adaptive rate control scheme aiming to achieve fair bandwidth allocation to all sensor nodes. Rajendran *et al.* [68] designed a traffic-adaptive MAC protocol that is based on the traditional TDMA, and the energy efficiency is achieved by ensuring that transmissions have no collisions, and by allowing sensor nodes to switch to sleep state whenever they are not transmitting or receiving. Recently, cross-layer design approaches by integrating MAC into routing protocols for wireless sensor networks are proposed by several researchers [69], [70]. For instance, the routing path could be chosen depending on the collision information available from MAC.

This thesis will adopt a hybrid of TDMA and CSMA MAC to build communication service for target tracking application in wireless sensor networks. Refer to Chapter 3 for the details.

2.3.2 The Survey of Routing Protocol Design

In a wireless sensor network, routing protocols is responsible for network formation and information delivery so as to facilitate the coordination and management among sensor nodes as well as the information gathering, processing and propagating across the entire sensor network. The routing protocols for wireless sensor networks can be grouped into three categories [62]: the flat-based routing protocol in which all sensor nodes are assigned equal roles; the hierarchical-based routing protocol in which sensor nodes play different roles; and the location-based routing protocol in which the sensor nodes' positions are exploited for the information relaying.

Some examples of flat-based routing protocol are Sensor Protocols for Information via Negotiation (SPIN) [72], Directed diffusion and its variations [73]–[75], Information-driven sensor querying (IDSQ) and constrained anisotropic diffusion routing (CADR) [76] ... etc. *SPIN* uses meta-data (high-level data descriptors) to eliminate the transmission of redundant data throughout the network. To overcome the problems of overlap and ensure that only useful information are transferred, SPIN adopts a negotiation scheme in which sensor nodes negotiate with each other before transmitting data and information. In addition, the sensor nodes in SPIN can base their communication decisions both upon the application-specific knowledge of the data and upon the knowledge of the energy resources that are available to them. This allows the sensor nodes to efficiently distribute data given a limited energy. *Directed diffusion* is data-centric in which data generated by sensor nodes is named by attribute-value pairs. A sensor node requests data by sending interests for named data, and data matching the interests are then drawn down towards that sensor node. Intermediate sensor nodes can cache, or transform data, and may direct interests based on previously cached data. Such process enables diffusion to achieve energy savings by selecting empirically good paths and by caching and processing data in-network. In *IDSQ* and *CADR*, an information utility measure is introduced to select sensor nodes to query and to dynamically guide data routing. This ensures that information gain is maximised while latency and bandwidth are minimised.

One example of location-based routing protocol is Geographic and Energy Aware Routing (GEAR) [78]. Rather than sending the interests to whole sensor networks as in Directed diffusion, GEAR only sends interests to a certain region of sensor network. GEAR uses energy aware neighbour selection to route data to the destination region and Recursive Geographic Forwarding or Restricted Flooding algorithm to disseminate the data inside the destination region. By doing so, GEAR can conserve more energy than Directed diffusion.

One representative hierarchical-based routing protocol is the Low Energy Adaptive Clustering Hierarchy (LEACH) protocol [77]. LEACH is a clustering based protocol and it outperforms classical clustering algorithms by using adaptive clusters and rotating cluster-heads, allowing the energy requirements of the system to be distributed among all the sensor nodes. In addition, LEACH is able to perform local computation in each cluster before transmitting the data to the base station. This greatly reduces the energy consumption in wireless sensor networks, as computation is much cheaper than communication.

In this thesis, the hierarchical sensor network architecture is adopted and sensor nodes are assigned different roles – the leader nodes which are responsible for information processing and the sensing nodes that make acoustic measurements and provide their measurements to the leader nodes. Accordingly, this thesis adopts a hierarchical routing protocol which is in spirit similar to LEACH but with some alteration and extension to suit for the target tracking application in wireless sensor networks. Refer to Chapter 3 for the details.

2.3.3 The Survey of Information Processing Design

Information processing techniques leverage the processes of transforming raw data and information collected from sensor nodes into meaningful knowledge about the observed physical phenomena. Information processing for wireless sensor network is an interdisciplinary research area with the focus on the representing, storing, processing and propagating spatially-temporarily distributed information under the constraints such as energy efficiency. Information processing in wireless sensor networks has increasingly drawn attention in research communities [9], [18], [20], [22], [79]–[88]. Current research efforts include system state estimation, sensor management, distributed compression, sensor querying and tasking, mobile-agent based information processing, and distributed inference and learning in wireless sensor networks ... etc.

Sayed *et al.* developed a collaborative signal processing (CSP) framework for the detection, classification and tracking of targets in wireless sensor networks [9], [22], [80]–[82]. In CSP framework, the sensor field is divided into smaller spatial coherence regions (SCRs) to facilitate distributed information processing. An important property of the SCRs is that the spatial signal field remains strongly correlated within a SCR and approximately uncorrelated in distinct SCRs. Accordingly, in each SCR, sensor nodes are assigned different roles – the normal sensor nodes for sensing and the leader nodes for coordinating information processing within SCR and communication between different SCRs. Two methods of information exchange amongst sensor nodes are developed – the data fusion method to exchange low dimensional feature vectors, and the decision fusion method to exchange high dimensional likelihood values. Furthermore, distributed classification algorithms are also incorporated into CSP framework for the classification of target type in wireless sensor networks.

Zhao *et al.* proposed a set of information-based approaches for information processing in wireless sensor networks [18], [79]. The main idea of these approaches is to determine

sensor nodes which will participate in information processing by optimizing an information utility function constrained by a given cost function in terms of communication and computing. Based on the similar idea of sensor node selection, Guo and Wang developed a dynamic sensor collaboration scheme [20]. In this scheme, at each time step the probability density function of the target state (called the belief state) is estimated and then used for optimal sensor selection for the next time step to maximise the information gain. Qi *et al.* developed the mobile agent based information processing for wireless sensor networks [84]. In their approach, a mobile agent, which is a special kind of “software” is adopted. It can migrate from one sensor node to another sensor node performing data and information processing. This approach differs from the commonly used client/server approaches where individual sensor nodes (clients) send measurements to a leader node (server) for information processing.

Other research works of information processing for wireless sensor networks include: distributed detection with the integration of wireless channel condition in the algorithmic design [85]; distributed compression-estimation under bandwidth constraints [86]; distributed inference using graphic models and message-passing algorithms [87]; and joint source-channel communication for distributed estimation [88] ... etc.

This thesis develops collaborative information processing techniques for target tracking applications in wireless sensor networks. It adopts a hierarchical sensor network architecture in which the sensor field is partitioned into a number of smaller regions like the above spatial coherence regions (SCRs). In each smaller region, a sensor cluster consisting of a leader node and a number of sensing nodes is formed. At each time step, the leader node selects a subset of sensing nodes to participate the tracking task; the sensing nodes selection is based on a composite function which is balancing the information utility and energy consumption of the sensing nodes. To attain desirable tracking performance while maintain efficient use of limited communication and computing resources, this thesis jointly optimises the interdependent operation of sensing, information processing, communication and routing in the wireless sensor network.

2.4 The Research Works of Target Tracking in Wireless Sensor Networks

Target tracking problem covers many aspects of wireless sensor networks, from target state estimation, communication, information routing, to estimation results propagation and inference ... etc. This thesis will focus on the task of tracking single and multiple target using stationary acoustic sensor nodes in wireless sensor networks. This section supplies a

survey of various strategies and approaches reported in the literature for target tracking in the context of wireless sensor networks with the special emphasis on the approaches that treat target tracking within the recursive Bayesian estimation framework.

2.4.1 Acoustic Energy Based Target Tracking

Target tracking using acoustic sensor nodes have been found in many applications. Examples include the speaker's position tracking in videoconference and multimedia human computer interface applications [43], [90], [91]; and the vehicle's location estimation in an open field [92] ... etc. Existing target tracking methods using acoustic sensor nodes are: direction of arrival (DOA), time delay of arrival (TDOA), and received signal strength or energy. DOA approaches estimates the target location by exploiting the phase difference measured at receiving sensors [93]–[95] and it is applied to the applications where the acoustic source emits a coherent, narrow band signal. TDOA estimates the target location by estimating time delays between different sensor nodes [96], [97]. It requires accurate measurements of the relative time delay between sensor nodes. To measure the relative time delay, acoustic signatures extracted from individual sensor node must be compared.

For wireless sensor network applications, acoustic energy based features is an appropriate choice since the acoustic power emitted by a moving target (*e.g.*, vehicle) usually varies slowly with respect to time. As such, the sampling rate for the acoustic energy time series can be set much lower than that for the raw acoustic time series [16]. Thus, little data will need to be transmitted over the wireless communication channel. This will not only reduce the energy consumption for data transmission at individual sensor nodes but also reduce the demand of communication bandwidth over shared wireless channels. It is well known that the energy of acoustic signal attenuates as a function of distance from the source [114]. Using this property, researchers proposed various acoustic energy based approaches for target tracking in wireless sensor networks. These approaches can be roughly classified into the following categories: graph based estimation [89], [111]–[113]; maximum likelihood estimation [16]; and recursive Bayesian estimation [19],[25]. Throughout the thesis, it is assumed that acoustic sensor nodes are employed to form the wireless sensor network and the tracking algorithms will be developed based on the acoustic energy measurements collected from these acoustic sensor nodes.

2.4.2 Recursive Bayesian Estimation for Target Tracking in Wireless Sensor Networks

The purpose of target tracking is to determine the target state (i.e. position, velocity and heading) using the sensor measurements. The most widely adopted approach for target tracking is the recursive Bayesian estimation in which the tracking problem is formulated and solved on the probabilistic basis. The fundamental Bayesian theorem is that the probability density function of the target state containing all information of the target and can be derived given the state space model [27], [28], [47], [99], [100]. The general recursive Bayesian estimation updates the probability density function of the target state through two stages: a prediction stage that propagates the probability density function at the previous time step through the target dynamics to form the one step ahead prediction; and a correction stage that incorporates the latest measurements through Bayes' rule to form the new probability density function at the current time step. The algorithms within recursive Bayesian estimation framework include Kalman filter, sequential Monte Carlo (e.g. Particle filter), along with numerous various variants and hybrids of these two algorithms. A survey of recursive Bayesian estimation for target tracking in wireless sensor networks is provided as follows.

Single Target or Well-Separated Multiple Target Tracking in Wireless Sensor Networks

Sayeed *et al.* developed a suite of algorithms for the detection, classification and tracking of targets in wireless sensor networks [9], [22], [80]–[82]. In these algorithms, the sensor field is divided into smaller regions and sensor nodes are assigned different roles. Such idea of sensor field partition is apparently attractive for distributive target tracking in wireless sensor networks and have also been adopted by other researchers [19]. However, the works reported in [9], [22], [80]–[82] mainly focused on the single target tracking problems. Although the authors also tackled the problems of multiple target tracking, their solutions were based on the assumption of sufficient separation in space and/or time of multiple targets, or tracking multiple targets with the use of target classification techniques. In contrast, in this thesis, we tackle the multiple target tracking problem by explicitly solving the challenging data association problem.

Zhao *et al.* proposed a leader-based tracking scheme for single target tracking in wireless sensor networks [8], [76], [79], [102]: at each time step, the sensor node which most reduces the uncertainty of target state estimation is selected as leader node; this leader node estimates the target state based on the estimation from previous leader node and its current measurement. For tracking multiple targets that well-separated in space, a leader

node is selected for each of the targets and the tracking task is performed in parallel within different sensor clusters located in the different smaller regions. Guo and Wang [20] adopted the similar leader-based scheme for single target tracking under measurement origin uncertainty introduced by clutter and missed detections. To improve the tracking performance and cater for nonlinear measurements, they used an Auxiliary Particle filter (APF) to implement the leader-based scheme. Sheng *et al.* [19] proposed distributed Particle filters for target tracking in wireless sensor networks. Instead of using the measurement of a single leader node, the estimation of target state is updated by making use of the measurements from a group of sensor nodes located in a cliques (the concept of sensor cliques is quite similar to that of SCR as described in Section 2.3.3). Each clique receives a partial estimate of target state from its preceding clique, and updates this partial estimate with local measurements made by the sensor nodes within the clique. Then, it forwards the updated partial estimate to the next clique. The process will repeat until the target moves out of the sensor field. By this way, the estimation of target state is in a distributive manner. However, the author did not consider the scenario of tracking target with clutter and missed detections (*e.g.* measurement origin uncertainty).

Without assuming different roles of sensor nodes, several researchers also proposed a number of distributive algorithms [17], [26], [106]–[108]. Coates proposed a distributive particle filter for target tracking in a wireless sensor network [17]. It is assumed that the measurements at each sensor node are independent so that the measurement likelihood function can be factored into products of partial likelihood functions. The procedure of Coates' method is as following: each sensor node maintains a separate particle filter; the partial likelihood function is updated at each sensor node using only the measurement acquired by this sensor node and the partial likelihood functions estimated in the preceding sensor node; the final probability density function of the target state will be back-propagated to all preceding sensor nodes and a new set of particles will be generated at each sensor node using this probability density function. In the above procedure, the partial likelihood function is represented by some kind of parametric model whose parameters require training. However, neither the parametric model nor the training method is detailed in [17]. The computation and communication resource required of this algorithm is quite high and so is not ideally suited to the resources constrained wireless sensor networks. Vercauteren and Wang developed a decentralised Gaussian mixture sigma-point information filter [26]. This decentralised filter makes use of the sigma-point filter for its better performance over extended Kalman filter for nonlinear tracking applications [35] and

the information filter (it is the inverse covariance form of the Kalman filter [100]) for its capability to be easily decentralised. The authors adopted the flat network architecture (peer-to-peer topology) with homogeneous sensor nodes. To handle the multimodal probability density, the probability density function of the target state is represented by using a Gaussian mixture model. At every time step, each sensor node runs a bank of parallel sigma-point information filter to compute the components for the above Gaussian-mixture model; sensor nodes communicate with each other to share the information of the cumulative quantities of these Gaussian components; and finally, each sensor node computes the updated quantities for the Gaussian components. In addition, each sensor node also performs a local probabilistic data association (PDA) to tackle the measurement origin uncertainty due to clutter and missed detections. Apparently, the above decentralised algorithms require extensive communication among sensor nodes, and the computation burden of each sensor node is also quite high. Therefore, the algorithm may not be suited to resources constrained wireless sensor networks either. Recently, Ribeiro *et al.* proposed a distributed Kalman filter jointly solving the compression and estimation problems for target tracking in wireless sensor networks [106]–[108]. The communication cost is conserved by quantifying the measurements at each sensor node and only feeding the quantified measurements into Kalman filter for target state estimation. The authors demonstrated that the complexity of this distributed Kalman filter was comparable to the equivalent Kalman filter based on the original measurements while the mean squared error (MSE) of the resultant estimate was close to the MSE based on the original measurements. However, as demonstrated in Chapter 4, when state-space model is non-linear (this thesis uses acoustic sensor nodes and the acoustic measurement model is nonlinear, refer to Chapter 4), the Kalman filter or extended Kalman filter based tracking algorithms may significantly deteriorate. Therefore, the distributed algorithms proposed in [106]–[108] have the limits when applied to nonlinear target tracking in wireless sensor networks.

Motivated by the above works, this thesis developed a suite of algorithms for single target tracking in wireless sensor networks, including sequential extended Kalman filter (S-EKF), sequential unscented Kalman filter (S-UKF), Particle filter (PF) and hybrid extended Kalman Particle filter (EKPF) (refer to Chapter 4). Among these algorithms, PF and EKPF can effectively combat the non-linearity introduced by the nonlinear acoustic measurement model. The novel EKPF developed in this thesis integrates extended Kalman filter as a core component to the Particle filter to achieve better tracking accuracy. In contrast to the above distributed tracking algorithms proposed in [17], [26], [106]–[108], this thesis adopts the

hierarchical network architecture (cluster-tree topology) to achieve distributive target tracking. Also differing from the algorithms developed in [8], [76], [79], [102] in which the estimation of target state is only based on the measurement from a single sensor node, in this thesis the estimation of target state is based on the measurements from a group of sensor nodes. As will be discussed in Chapter 3, such hierarchical network architecture is also beneficial to the development of communication protocol (MAC) and routing protocol for target tracking applications in wireless sensor network. Moreover, a Particle filter and probability density association filter (PDAF) hybrid algorithm, named as PF-PDAF algorithm is also developed for single target tracking under measurement origin uncertainty due to clutter and missed detections.

Multiple Target Tracking in Wireless Sensor Networks

In multiple target tracking, a sensor node may acquire more than one measurement; these measurements are not only generated by the targets but also possibly generated by the clutter. Hence multiple target tracking algorithm needs to solve the data association problem, i.e. correctly mapping the measurement and target pairs with the interference introduced by clutter. There are prolific strategies and methodologies have been proposed in the literature for multiple target tracking [27]. These strategies and methodologies will be reviewed in Chapter 7 and here we focus on the research works of multiple target tracking in the context of wireless sensor networks.

Sheng *et al.* extended the distributive Particle filter to multiple target tracking [19]. Instead of tackling the data association problem explicitly, the author proposed a clustering algorithm to separate and/or merge multiple targets into a number of target groups which are assumed to be well separated spatially. Within a target group, all targets are regarded as a “super target” and an independent Particle filter is maintained for estimating the state of this “super target” as tracking a single target. However, in many practical multiple target tracking scenarios [147], the targets may become neither well separated nor very closely spaced; therefore, the joint state estimation of multiple targets by explicitly solving the data association problem becomes necessary.

Vercauteren *et al.* extended the leader-based tracking scheme [18] to multiple target tracking [101]. In their solution, at each time step, one leader node performs a series of tasks including sensing, estimating target state, selecting next leader node, and handing over the estimation results. To facilitate implementing such leader-based multiple target tracking algorithm, the authors made several assumptions: no more than two targets can be

present in the measurement range of a leader node; one target is always within the range of its leader node while another target is considered as interference. These assumptions limit the number of measurements and targets, and accordingly, help reduce the complexity in solving the data association problem. During each time step in the tracking task, for each target, there is a leader node maintaining a Particle filter to update the target state. When two targets are moving close to each other (for example, two targets are crossing), the two leader nodes of both targets still encounter the data association problem and need to execute joint target state estimation algorithm. However, it may not be trivial and efficient to run the same joint target state estimation algorithm separately on these two leader nodes. Moreover, when two targets are crossing, it is also not a trivial task to select the leader node because one leader node may have chance to be selected for both targets. Furthermore, when there are more than two targets (assuming these targets at some time steps will be crossing or closely spaced), maintaining one leader node for each of the targets may not be realistic; thus the leader-based multiple target tracking algorithm proposed in [101] may not readily extendable to more than two targets tracking in wireless sensor networks.

Oh *et al.* proposed a Markov chain Monte Carlo data association (MCMCDA) algorithm for multiple target tracking in wireless sensor networks [103], [104]. This MCMCDA algorithm uses Metropolis-Hastings algorithms [105] to generate samples from a distribution on the problem space. The authors claimed MCMCDA can track unknown number of targets in the presence of clutter and missed detections. However, MCMCDA is a batch algorithm that is based on multiple scans; it requests not only the measurements obtained at current time step, but also the measurements from several previous time steps for the estimation of target state. This may increase the resources consumption in terms of memory, computation and communication. Although the authors also developed single-scan version of MCMCDA, it is not appealing because the implementation is not straightforward.

In contrast to the above works, this thesis develops the Particle filter based multiple target tracking algorithm – a Particle filter and joint probability density association filter hybrid algorithm (PF-JPDAF). Differing from the algorithm developed in [19], this PF-JPDAF algorithm tackles the data association problem explicitly, and thus can be applied into the occasions that multiple targets are closely spaced. Also differing from the leader-based multiple target tracking algorithm proposed in [101], PF-JPDAF adopts hierarchical sensor network architecture, at each time step, the leader node selects several sensing nodes, collecting their measurements, estimating the target state, and if necessary, propagating the

estimation results to the next leader node. Target state estimation using the information from multiple sensing nodes can improve the tracking accuracy. Unlike MCMCDA algorithm developed in [103], [104], PF-JPDAF only uses the measurements acquired at the current time step. Thus, in contrast to MCMCDA algorithm, PF-JPDAF only requires modest resources utilization when it is applied for on-line multiple target tracking in wireless sensor networks.

Chapter 3

Collaborative Information Processing Framework for Target Tracking in Wireless Sensor Networks

This chapter proposes a collaborative information processing framework for target tracking in wireless sensor networks. The framework includes three major components – the distributive estimation of target state under measurement origin uncertainty; the hierarchical routing in highly dynamic environment; and the self-organised hybrid communication on the dynamic basis. Each of the components addresses one important aspect for target tracking application in wireless sensor networks; on the other hand, these components are complementary to each other to provide an integrated solution for tracking targets in wireless sensor networks.

This chapter is organized as follows. Section 3.1 presents a hierarchical wireless sensor network architecture which is adopted in this thesis. This hierarchical wireless sensor network architecture is the foundation of the whole thesis and the algorithms developed in the following chapters are all based on it. Section 3.2 details the collaborative information processing framework for target tracking in wireless sensor networks. Both the hierarchical routing component and the self-organised hybrid communication component are detailed. The distributive estimation component is briefly introduced in this chapter; however, it is the focus of this thesis and will be developed in the following chapters. Section 3.3 concludes this chapter.

3.1 Hierarchical Sensor Network Architecture

Given the limited energy and communication bandwidth of individual sensor nodes, a critical consideration in wireless sensor networks is that most of data and information processing must take place at a local level, i.e. among a group of sensor nodes located in a relatively smaller region. This is also to comply with the observation that at each time instance, the data and information processing only require collaboration among a limited number of sensor nodes in a certain region and not among an arbitrary set of sensor nodes distributed across the whole sensor field.

Figure 3.1 shows a sensor field partition scheme proposed by Sayeed *et al.* [80]-[82]. Similar partition schemes are also adopted in [9], [19], [148]. In Figure 3.1, the sensor field of the size $D_x \times D_y$ is divided into a number of smaller regions with the same size of $D_{c,x} \times D_{c,y}$. Here, the evenly partition of the sensor field is only for illustration, however, it is not required in the collaborative information processing framework proposed in this chapter and the algorithms developed in the following chapters.

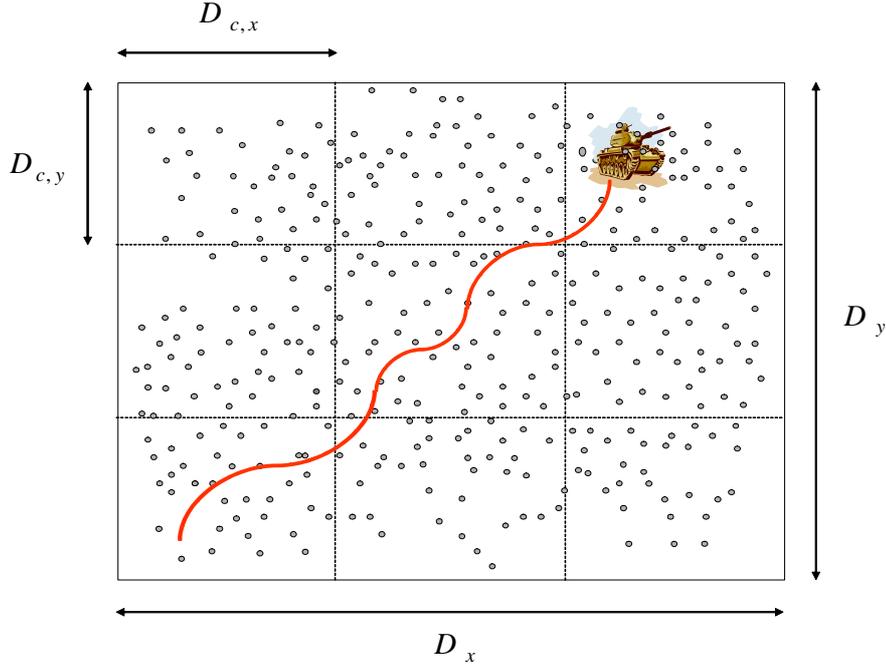


Figure 3.1 Sensor field partition

The sensor nodes used in this thesis are assumed to be equipped with only acoustic sensing modality. When a target moves in the sensor field, it generates a space-time signal $s(x, y, k)$. This signal is a function of the x - and y - coordinates and time k , and it can be represented as following [80]

$$s(x, y, k) = \int_{-B_x/2}^{B_x/2} \int_{-B_y/2}^{B_y/2} \int_{-B_k/2}^{B_k/2} \phi(v_x, v_y, f) e^{j2\pi v_x x} e^{j2\pi v_y y} e^{j2\pi f k} dv_x dv_y df \quad (3.1)$$

where B_x and B_y represent the spatial bandwidth in x - and y - coordinates and B_k represents the temporal bandwidth of the signal in new dimensional units, respectively. $\phi(v_x, v_y, f)$ is the underlying spectral representation which satisfies

$$E [\phi(v_x, v_y, f) \phi^T(v'_x, v'_y, f')] = \phi(v_x, v_y, f) \delta(v_x - v'_x) \delta(v_y - v'_y) \delta(f - f') \quad (3.2)$$

for some $\phi(v_x, v_y, f) \geq 0$ that represents the power spectral density (PSD) of the signal field, where T denotes matrix transposition and δ denotes Dirac delta function. The signal correlation function γ_s is related to the PSD via three-dimensional Fourier transformation:

$$\begin{aligned} \gamma_s(\Delta x, \Delta y, \Delta k) &= E \left[s(x + \Delta x, y + \Delta y, k + \Delta k) s^T(x, y, k) \right] \\ &= \int_{-B_x/2}^{B_x/2} \int_{-B_y/2}^{B_y/2} \int_{-B_k/2}^{B_k/2} \phi(v_x, v_y, f) e^{j2\pi(v_x \Delta x + v_y \Delta y + f \Delta k)} dv_x dv_y df \end{aligned} \quad (3.3)$$

Both power spectral density $\phi(v_x, v_y, f)$ and signal correlation function $\gamma_s(\Delta x, \Delta y, \Delta k)$ characteristic the statistics of the random target signal $s(x, y, k)$.

In the above sensor field partition scheme, the assumption is made that the target's signal $s(x, y, k)$ is perfectly correlated in each of the smaller regions in Figure 3.1, i.e. at any time step k , one has $s(x, y, k) = s(k)$; however, the temporal process $s(k)$ in different smaller regions are statistically independent. The above assumption facilitates the development of signal and information processing algorithms in wireless sensor networks. The individual sensor node samples the target signal $s(x, y, k)$, computes the intensity of this signal, and transmits the magnitude of signal's intensity to a fusion centre (leader node in this thesis, refer to Chapter 4) for the estimation of target state.

This thesis adopts a hierarchical sensor network architecture based on the above sensor field partition scheme: the whole sensor field is partitioned into a number of smaller regions, and in each smaller region, a group of sensor node forms a sensor cluster. During the target tracking task, sensor nodes are dynamically organised into different clusters (refer to Figure 1.2 in Chapter 1). Each sensor cluster is composed of two types of sensor nodes – the leader node and the sensing nodes:

- The leader nodes are sparsely deployed in the sensor field. Their number is less than the sensing nodes but still has sufficient coverage for the whole sensor field. Besides the acoustic sensing modality, each leader node is also equipped with an advanced digital signal processor (DSP) and a powerful wireless transmitter (i.e. the radio transmission range is larger than that of sensing node). Hence, the leader nodes have powerful information processing and communication capabilities;
- The sensing nodes are densely scattered over the sensor field. Sensing nodes are equipped with acoustic sensing modality. However, they are equipped with less powerful microprocessors and relatively short range wireless transmitters. The sensing

nodes provide leader nodes with the acoustic energy measurements, i.e. the intensity of the target's signal $s(x, y, k)$ upon request.

The formation of a sensor cluster is triggered by the detection of an approaching target, and it includes several phases and involves communication protocol and routing protocol. This will be discussed in the next section. Throughout this chapter, the following assumptions are made: a leader node has the knowledge of itself and its all neighbouring sensing nodes, including the positions, residual energy, and one-hop communication costs between itself and each neighbouring sensing node; the sensing nodes have the knowledge of the residual energy of themselves. Such knowledge can be established during network initialisation and discovery. Moreover, it is also assumed that the sensing node and leader node (i.e. cluster leader) exchange information via a single-hop communication within a sensor nodes cluster; there is no peer-to-peer communication among the sensing nodes within a sensor nodes cluster; and all sensing nodes and leader nodes are time-synchronized.

3.2 Collaborative Information Processing Framework

Unlike the centralised platform (*e.g.* radar, sonar...etc.), in a wireless sensor network the measurements and the tracking algorithms are physically distributed across sensor nodes in the network. This implies that along with the target state estimation, the communications amongst sensor nodes and information routing in network should also be viewed as an integral part of the target tracking problem in wireless sensor networks [22]–[24]. In particular, the following issues need to be considered together:

1. The estimation of the target state in a distributive manner;
2. The management of target identity in the presence of multiple targets and clutter;
3. The selection and organization of sensing nodes to collaboratively participate in the tracking task;
4. The routing protocol for data and information delivery within the sensor network;
5. The communication protocol for the management of wireless channel.

To address the above issues, this chapter proposes a collaborative information processing framework for tracking both single and multiple target in wireless sensor networks. The proposed framework emphasises the collaboration amongst sensor nodes, i.e., coordinating a group of sensing nodes to provide consistent, essential but non-redundant data and information for the accurate estimation of target state. Such

collaborative information processing will not only reduce the volume of data and information to be transported, but also the probability of collision and interference in the wireless channel. Thus, the power consumption in network is reduced and hence, the lifetime of the wireless sensor network can be prolonged.

The proposed collaborative information framework incorporates three major components: the distributive estimation of target state under measurement origin uncertainty; the hierarchical routing in highly dynamic environment; and the self-organised hybrid communication on dynamic basis. Figure 3.2 depicts the components and their building blocks of the proposed collaborative information processing framework. The first component, the distributive estimation of target state is the focus of this thesis, and a number of algorithms have been developed and will be presented in the following chapters. Other two components, the hierarchical routing and hybrid communication will be detailed in the following subsections. However, the full development and implementation of these two components are beyond the scope of this thesis.

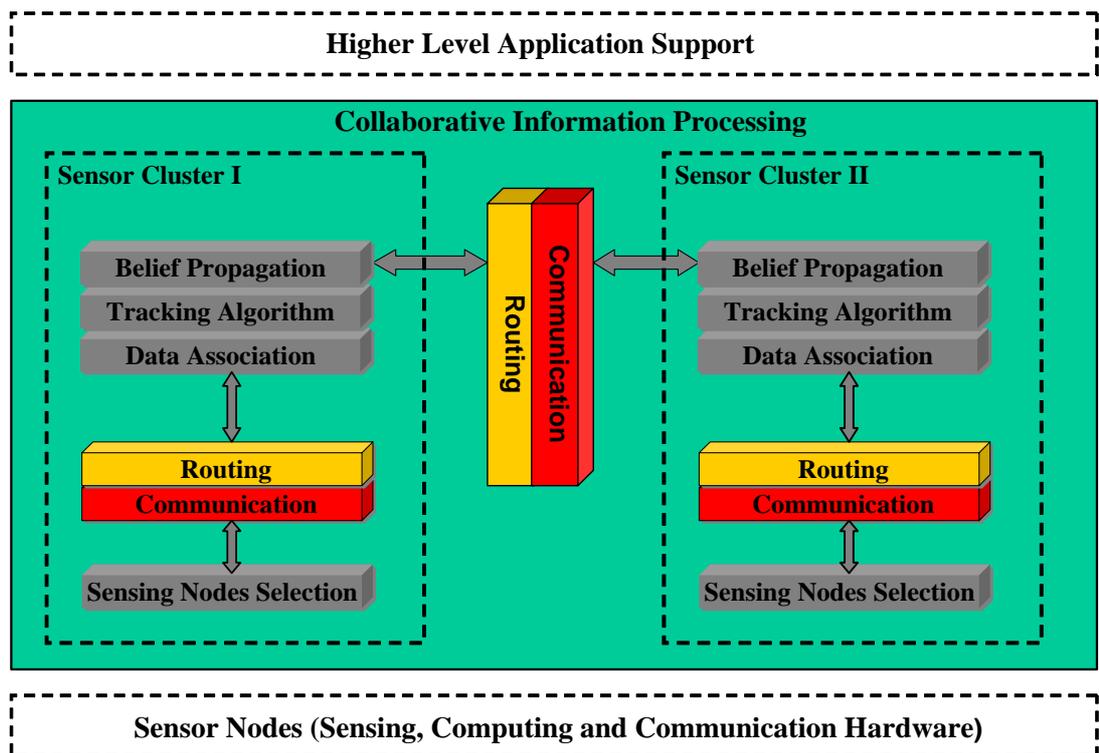


Figure 3.2 Collaborative information processing framework

3.2.1 Distributive Estimation of Target State under Measurement Origin Uncertainty

The ultimate purpose of the distributive estimation component is to coordinate sensor nodes to acquire necessary pieces of data and information, process these data and information, subsequently transform them into meaningful knowledge regarding the target state and finally makes this knowledge available to users (be they humans or machines). To fulfil this purpose, a number of algorithms have been developed in this thesis aiming to achieve desirable tracking accuracy while maintaining the modest consumptions of the system resources in terms of communication and computation. The building modules within the distributive estimation component are briefly described as follows. However, the details of these modules as well as the resulting algorithms will be presented in the following chapters of this thesis. Here, it also needs to emphasize that the distributive estimation component is highly correlated with the routing component and communication component in that several modules included in the distributive estimation component are also addressed in the latter two components.

Sensing nodes selection – Amongst the sensing nodes in the current active sensor cluster, not all of them provide useful information that effectively contributing to the estimation of the target state. Furthermore, if all of these sensing nodes take the measurements and transmit their measurements for target state estimation in the cluster leader, it will consume tremendous energy in the resources highly constrained wireless sensor networks. In this thesis, however, instead of activating all sensing nodes in a sensor cluster to take sensing action, the cluster leader selects an optimal subset of these sensing nodes, only activating the selected sensing nodes, and incorporating their measurements into the target state estimation. The sensing nodes selection will be detailed in Chapter 6, and the key idea is to balance the information contribution and the energy consumption at the sensing nodes within the current sensor cluster.

Data association algorithms – In most practical target tracking applications in wireless sensor networks, there may exist multiple targets and the clutter as well. In turn, the measurements obtained at a sensing node may consist of the measurements generated by the targets and the measurements generated by the clutter. This will make the assignment of measurements to the corresponding targets in the tracking algorithm difficult. Consequently, it is necessary to develop data association algorithm to solve this measurement origin ambiguity. In this thesis, two algorithms have been developed to

effectively tackle the data association problems. For the single target tracking in the presence of clutter, the hybrid of Particle filter and probability density association filter (PF-PDAF) has been developed (Chapter 5). For the multiple target tracking in the presence of clutter, the hybrid of Particle filter and joint probability density association filter (PF-JPDAF) has been developed (Chapter 7).

Distributed tracking algorithms – At each time step, the selected sensing nodes in a sensor cluster forward their acoustic energy measurements to the cluster leader, and in turn, the cluster leader runs the tracking algorithm to estimate the target state. This information processing process continues until the target moves out of this cluster, and then the cluster leader passes its belief (i.e. the probability distribution of the target state) to the cluster leader of next sensor cluster. In this way, the estimation of target state is updated distributively in the wireless sensor network. This thesis developed a number of distributive tracking algorithms in the following chapters, including the distributive PF algorithm, the distributive EKPF algorithm, the distributive PF-PDAF algorithm, and the distributive PF-JPDAF algorithm.

Belief propagation algorithms – When the target moves out of the region occupied by a sensor cluster, the cluster leader needs to propagate its belief to the cluster leader of the next sensor cluster. Since this thesis adopts particles to represent the probability density function of target state, a high volume of particles data needed to be transferred from one cluster leader to another cluster leader. This will consume tremendous communication overheads and may even cause network congestion. To combat this problem, this thesis makes use of a Gaussian mixture model (GMM) to approximate the probability density function, and consequently, only the GMM parameters whose number is far less (two orders less as shown in the simulations in Chapter 6) than that of particles needs to be transmitted to the next cluster leader (refer to Chapter 6 for details). Hence, the communication overhead is greatly reduced and energy is saved.

3.2.2 Hierarchical Routing in Highly Dynamic Environment

Routing caters for network discovery, formation and data delivery. It is essential for properly transferring data and information to the right destination at the right time in a wireless sensor network. In the proposed collaborative information processing framework, the routing scheme provides a backbone for the distributive target state estimation

component, from sensing nodes selection and distributive target tracking to belief propagation. However, routing is one of very challenging design tasks in wireless sensor networks because:

1. It is not possible to adopt the traditional IP-based routing protocol since the relatively large number of sensor nodes are deployed in a wireless sensor network;
2. In target tracking applications, at each time step, the measurements obtained from numerous sensing nodes need to be transmitted to the cluster leader;
3. The sensor nodes are highly resource constrained;
4. There is a high probability that the data has some redundancy and such redundancy needs to be exploited by the routing protocols to improve energy and bandwidth utilisation.

Corresponding to the hierarchical wireless sensor architecture, a hierarchical type routing protocol similar to that proposed in [77] is adopted in this thesis. In this hierarchical routing protocol, the clusters are formed distributively across the sensor network without any centralized control. Moreover, the leader nodes perform data aggregation and integration, and consequently, the energy at an individual sensing node could be saved and the lifetime of a wireless sensor network can be prolonged. This section will details the design of such hierarchical routing protocol. However, the full development and implementation of the routing protocol is beyond the scope of this thesis.

Basically, the hierarchical routing is comprised of two stages: the first stage (cluster formation stage) is responsible for the cluster leader election and the sensor cluster formation; the second stage (steady stage) is used for sensing nodes selection within the sensor cluster, delivering the data (i.e. acoustic measurements) between the sensing nodes and the leader node within the sensor cluster, and forwarding the belief (i.e. estimation results) from the leader node of the current sensor cluster to the leader node of the next sensor cluster leader (belief propagation between cluster leaders will be detailed later in this section).

Figure 3.3 shows the time-line of this hierarchical routing protocol. Its process is described as follows.

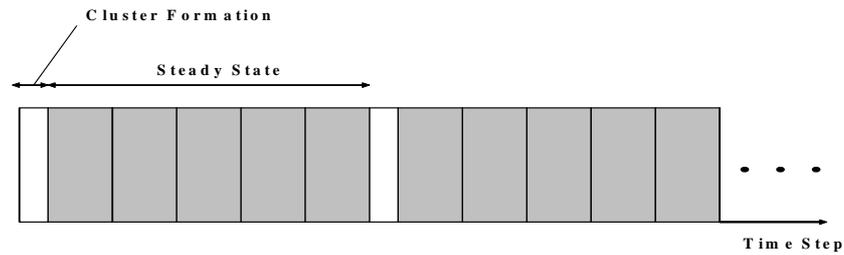


Figure 3.3 Time-line of the routing protocol operation

Cluster Leader Election and Sensor Cluster Formation

In single target tracking, when a leader node detects an approaching target (leader nodes are also equipped with sensing modality, refer to section 3.1), it volunteers itself to act as the cluster leader and recruits sensing nodes to form a sensor cluster. However, there is a critical design issue needs to be considered here: which leader node should be elected as the cluster leader and takes the responsibility to form the sensor cluster if several leader nodes detect the approaching target simultaneously. Ideally, the leader node that obtains the measurement with the largest magnitude of target's signal intensity should be elected because it is the closest leader node to the target. In multiple target tracking, two scenarios may exist: the targets are well separated in space; and the targets are closely spaced (but cannot be regarded as a "super target", refer to Chapter 2). In the first scenario, for each of the targets a unique sensor cluster is formed; and in each sensor cluster, the cluster leader election process will be the same as that of single target tracking. In the second scenario, the leader node may obtain the measurement which is the sum of the acoustic intensities of several signals with each corresponding to a target (in the following chapters, it is assumed that these acoustic signals can be separated by advanced signal processing techniques [21]). For this scenario, the leader node which obtains the measurement with the largest magnitude of signal intensity will still be elected as the cluster leader. Consequently, cluster leader node election process for this scenario will still be the same as that of the single target tracking. The cluster leader election process is described as follows.

Without a centralised facility existing for cluster leader election in a wireless sensor network, an effective method to determine the cluster leader is to adopt a bake-off timer whose values are determined by the magnitude of the intensity of target's signal obtained at

the leader nodes. A leader node whose acoustic signal intensity measurement intensity exceeds a predefined threshold sets a back-off timer and does not broadcast its intention of becoming the cluster leader until the timer expires. If by the time the back-off timer expires, this leader node receives an intention from another leader node and then it cancels the timer and terminates its volunteering process. Otherwise, this leader node becomes the cluster leader.

Nevertheless, there might exist a scenario where two leader nodes have the same value in their back-off timers since the distance from the target to these two leader nodes are the same. To solve this dilemma and also for the purpose of evenly distributing the load amongst the leader nodes, the value of the back-off timer for an individual leader node can be defined as a composite function consisting of two terms: a term decided by the acoustic signal intensity measurement obtained at this leader node; and a term defined by the residual energy in this leader node. Then, the set value of the back-off timer can be expressed as

$$T(m) = \alpha Z(m) + (1 - \alpha) E_{res}(m) \quad m \in \Omega \quad (3.4)$$

where $T(m)$ is the set value of the back-off timer for the m -th leader node. $Z(m)$ is the intensity of the acoustic signal acquired by the m -th leader node. $E_{res}(m)$ is the residual energy of the m -th leader node. α is the scaling factor which is used to balance the influence of the acoustic signal intensity measurement and the residual energy of the leader node. By adopting the above composite function, the confliction in the cluster leader election due to the same magnitude of the target's signal acquired by two leader nodes can be mitigated.

Once the cluster leader is elected, it broadcasts an advertised message over neighbouring sensing nodes using the carrier-sense multiple access with collision avoidance (CSMA/CA) MAC protocol (refer to next section for details). This message contains this cluster leader's ID and a header that distinguishes this message as an announcement message. Upon receiving the announcement message from the cluster leader, the neighbouring sensing nodes reply with a "Join Request" message. The message contains the sensing node's ID and its residual energy. Similarly, CSMA/CA MAC protocol is adopted for this message transmission. By receiving the "Join Request" messages, the cluster leader builds a cluster member list which includes the sensing nodes' IDs and their residual energies. Now the sensor nodes cluster is formed.

Sensing Nodes Selection and Measurements Transmission

After the formation of a sensor node cluster, then at each time step, the cluster leader selects a subset of sensing nodes from the cluster member list, activating these selected sensing nodes to sense and provide their measurements, and finally, updating the estimation of the target state. The selection criterion of sensing nodes is based on a composite objective function which consists of both information utility and energy consumption measures. More specifically, the information utility is calculated on the basis of the posterior Cramer-Rao lower bound (PCRLB). The energy consumption measure is based on the relative distance between the sensing nodes and the cluster leader. The PCRLB computation will be detailed in Chapters 4 and 5 while the sensing nodes selection scheme will be developed in Chapter 6.

From the computation results of the above composite function, the cluster leader will decide a subset of sensing nodes within the sensor cluster to participate in the tracking task. The cluster leader will send out an “Activation Request” message containing the ID of the selected sensing nodes. Upon hearing this message, the selected sensing nodes will wake up and return an “Activation Confirmed” message to the cluster leader to confirm their participation in the sensing task. Again, CSMA/CA MAC protocol is used for the above two messages transmission. Once receiving the “Activation Confirmed” message, the cluster leader assigns a Time Division Multiple Access (TDMA) schedule and sends it to the selected sensing nodes. After the TDMA schedule is known by all selected sensing nodes, the selected sensing nodes start to acquire the acoustic signal emitted by the target, processing it to get the intensity information, and then transmitting this acoustic intensity information along with the residual energy information of themselves to the cluster leader. In turn, the cluster leader uses the acoustic intensity information to estimate the target state and also updates the residual energy information of these selected sensing nodes in the cluster member table.

Belief Propagation between Cluster Leaders

In this thesis, “belief” refers to the probability density function of the target state. In the algorithms developed in the following chapters, the “belief” is represented by particles and their weights. As described in section 3.2.1, the belief propagation algorithm aggregates, approximates and propagates the estimation results from one cluster leader to another cluster leader. However, there still one issue remains: how the current cluster leader decides when to propagate its belief to which leader node. Basically, at each time step, after

the target state estimation, the cluster leader computes the distance between itself and the predicted position of the target of the next time step. If this distance becomes larger than a predefined threshold, then this cluster leader needs to propagate its belief to the next potential leader node of the cluster in its vicinity. To do so, this cluster leader broadcasts a message to its neighbouring leader nodes to initiate a new cluster leader election and sensor cluster formation process. It is assumed the communication range of this cluster leader is long enough that the message it broadcasted can reach all neighbouring leader nodes that have the potential to be the next cluster leader. Upon receiving this broadcasting message, the neighbouring leader nodes make the acoustic measurements. Then, the cluster leader election process as described earlier is started. Later on, when the new cluster leader is elected, the previous cluster leader forwards its belief to the new cluster leader. Then the new cluster leader starts to form a new sensor cluster to perform the tracking task. The above process will continue until the target moves out of the sensor field. It can be seen that the sensor cluster is formed dynamically, and at any time step, only a small portion of the sensing nodes is activated to take sensing and processing action while most sensing nodes are in “sleep” mode. This strategy will help conserve energy utilisation and in turn, prolong the lifespan of the wireless sensor network.

Figure 3.4 depicts the flow graph of the above process of cluster leader election, sensor cluster formation, sensing nodes selection and belief propagation. In some applications, the Users (i.e. higher level application software) that reside out of the wireless sensor network may query the target state. Hence, the current cluster leader needs to transmit its estimation results to the Users through multi-hop routing. However, multi-hop routing protocol is not considered in this thesis.

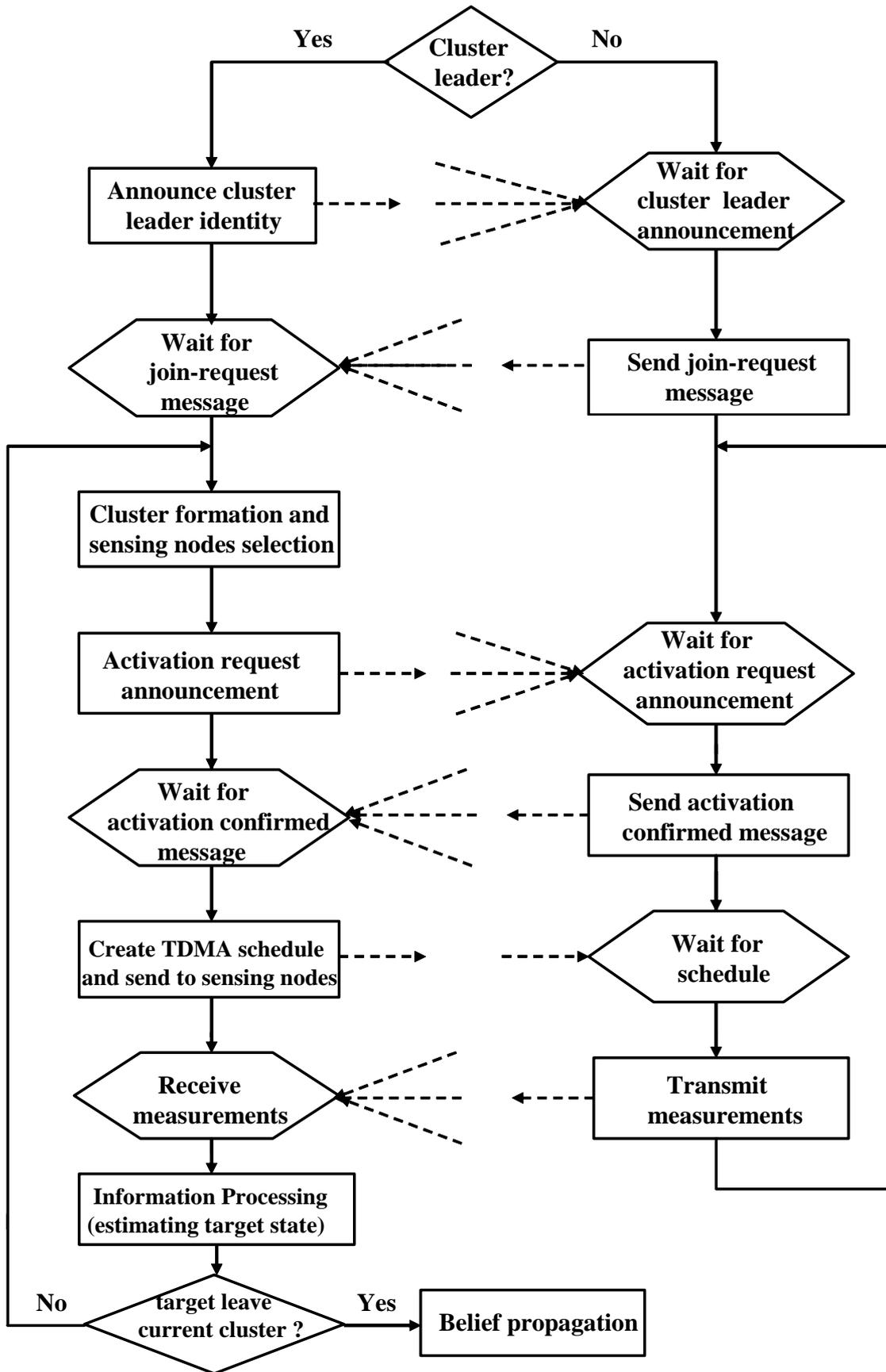


Figure 3.4 Flow graph of hierarchical routing protocol

3.2.3 Self-Organised Hybrid Communication on Dynamic Basis

In the above distributive estimation component and hierarchical routing component of the collaborative information processing framework, each task, from cluster leader election and sensor cluster formation, sensing nodes selection and measurements transmission, to belief propagation, requires communication over the wireless channel. The quality of the communication component of the collaborative information processing framework has a large impact on the overall performance of the target tracking in a wireless sensor network [67]. The major challenge of communication component design in wireless sensor networks is to avoid collision when two sensor nodes transmit data over the same wireless channel at the same time. The collision of the transmitted packets will require the re-transmission of the collided packets and this is at the cost of extra energy consumption.

The communication component in the proposed collaborative information processing framework is mainly the medium access control (MAC) protocols. MAC protocols have been extensively studied in traditional wireless communication systems [67]. It assists each sensor node to decide when and how to access the shared wireless channel and can be clarified into two categories: the scheduling based protocol such as time division multiple access (TDMA), frequency division multiple access (FDMA) and code division multiple access (CDMA); and the contention based protocol such as carrier sense multiple access (CSMA) and IEEE 802.11. In wireless sensor networks, the widely applied MAC protocols include the scheduling based protocol TDMA and the contention-based protocol CSMA. In this thesis, a hybrid of TDMA and CSMA/CA MAC protocol is adopted for the communication component of the collaborative information processing framework.

TDMA is adopted when the selected sensing nodes transmit their measurements to the cluster leader. In TDMA, the cluster leader allocates time slots for sensing nodes, i.e. dividing the wireless channel into many smaller time slots; and in each time slot, only one sensing node is allowed to transmit. The major advantage of TDMA is its energy efficiency since it effectively avoids the collision by pre-allocating the slots to individual sensing nodes. Moreover, low-duty-cycle operation of the sensing nodes can also be implemented under TDMA. A sensing node only turns on its radio when a time slot is assigned to it while turning off its radio when all time slots have been assigned to other sensing nodes. The disadvantage of TDMA is in that it has limited scalability and adaptability to the changes on the number of sensing nodes. Hence, TDMA is not suitable for either the sensor cluster formation or sensing nodes selection since the number of sensing nodes in both processes is time varying.

CSMCSMA/CA is adopted in two occasions in communication component of the proposed collaborative information processing framework. The first occasion is during the sensor cluster formation when the cluster leader announces its identity to recruit sensing nodes and the intended sensing nodes reply with the “Join Request” message. The second occasion is after the sensing nodes selection process when the cluster leader announces the list of selected sensing nodes to activate these sensing nodes, and the selected sensing nodes reply with the “Activation Confirmed” message. In CSMA, the communication channel is not divided into time slots and an individual sensing node is not assigned a time slot. Instead, the communication channel is shared by all sensor nodes and it is allocated on-demand by employing a contention mechanism to decide which node has the right to access the channel. The major advantage of CSMA is that it can scale easily to the change of the number of sensing nodes and flexible to the network topology changes. For example, in the above first occasion, the sensing nodes in the vicinity of the cluster leader need to send the “Join Request” messages to the cluster leader. Since the formation of sensor cluster has not completed yet at this stage, the cluster leader still doesn’t know which sensing node will join the sensor cluster. Hence, it is impossible for the cluster leader to pre-allocate the channel for the unknown sensing nodes to send the “Join Request” message. Consequently, TDMA is not appropriate here and CSMA is adopted. Similarly, in the above second occasion, the cluster leader doesn’t know whether the selected sensing nodes will confirm to take sensing action before these sensing nodes reply with the “Activation Confirmed” message. Hence, it is again impossible for the cluster leader allocating channel for these sensing nodes and we still need to resort to CSMA.

However, the pure CSMA has its own disadvantage in that the collisions can not be avoided. In sensor cluster formation and sensing nodes selection, if two sensing nodes want to send message to the cluster leader simultaneously, they need to contend for the channel and a collision may happen (as illustrated in Figure 3.5). In Figure 3.5, when sensing node *a* is sending a message to the cluster leader node, sensing node *c* is not aware of this

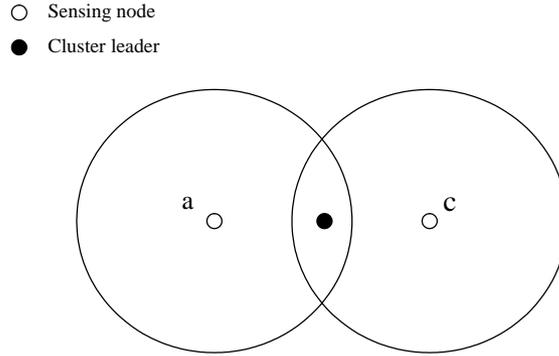


Figure 3.5 The hidden terminal problem

transmission due to the range of its radio. If sensing node c starts transmitting now, cluster leader will receive collided packets from both sensing nodes. Original CSMA could not solve this problem. In this chapter, an enhanced CSMA, called CSMA/CA (CA stands for collision avoidance) which uses a handshake mechanism is adopted. The handshake starts from the sender by sending a short Request-to-Send (RTS) packet to the intended receiver. Then the receiver replies with a Clear-to-Send (CTS) packet. The sender starts sending data after it receives the CTS packet. The purpose of RTS-CTS handshake is to let the neighbouring nodes know there is a data transmission over the channel. In Figure 3.5, although sensing node c cannot hear the RTS sent by sensing node a , it can hear the CTS from cluster leader. If a node hears an RTS or CTS destined to other nodes, it should back-off without sending its own packet. However, CSMA/CA does not completely eliminate collision problem, but now the collisions are mainly on RTS packets. Since the RTS packet is very short, the cost of collisions is greatly reduced. The details of CSMA/CA can be found in [65]. Other enhanced versions of CSMA have also been proposed in the literature [63], [68]. However, the detailed implementation of these CSMA protocols is beyond the scope of this thesis.

3.3 Summary

This chapter proposed a collaborative information processing framework for target tracking in wireless sensor networks. Three major components, namely the distributive estimation of target state, the hierarchical routing and the self-organised hybrid communication are detailed. On the one hand, these components and their associated algorithms are independent because they address different aspects of target tracking problems in wireless

sensor networks; on the other hand, they are highly correlated in the sense that they are complementary to each other in order to provide an integrated solution for tracking target in wireless sensor networks.

However, the focus of this thesis is to develop distributive estimation component. Developing the full set of routing component and communication component is beyond the scope of this thesis. The investigation of routing component and communication component is only in the context of backing up the distributed estimation component. The following chapters will develop various algorithms for the distributive estimation component of the proposed collaborative information processing framework.

Chapter 4

Tracking a Single Target in Wireless Sensor Networks

This chapter develops techniques and algorithms for tracking a single target in wireless sensor networks. On the basis of the recursive Bayesian estimation method, four tracking algorithms are developed, namely the sequential extended Kalman filter (S-EKF), the sequential unscented Kalman filter (S-UKF), the generic Particle filter (PF), and the hybrid extended Kalman Particle filter (EKPF). Extensive simulations are carried out to assess the performance of these algorithms. To determine the theoretical performance bound to which the tracking algorithms could attain, the posterior Cramer-Rao lower bound (PCRLB) is computed and compared for all four tracking algorithms.

4.1 Introduction

As discussed in the previous chapters, the task of target tracking in a wireless sensor network is to estimate the relevant state of the target (e.g. the position, velocity and heading of the target) in a timely manner, from information gathered by sensor nodes. The most frequently adopted paradigm for target tracking is the recursive Bayesian estimation. In this thesis, the acoustic sensor nodes are employed to form a wireless sensor network. The measurement obtained at an acoustic sensor node is a function of the distance between this sensor node and the target; and thus the measurement model is nonlinear. However, for such nonlinear target tracking problem which is the subject of this chapter, no analytical solutions exist for recursive Bayesian estimation and appropriate approximation methods are needed.

Numerous approximation methods to the recursive Bayesian estimation have been proposed in the literature [29]–[40]. The well-know extended Kalman filter (EKF) linearises the state space model through the use of a first order truncated Taylor series expansion around the system state estimation obtained from the previous time step [29], [30]. Without making use of the analytical Taylor series linearisation, the unscented Kalman filter (UKF) adopts a deterministic sampling approach to propagate the mean and covariance of the system state estimation obtained from the previous time step [32], [33].

In contrast to EKF and UKF, the Particle filter (PF) approach represents the probability density function of system state by empirical samples (particles) and then these samples are recursively updated using sequential importance sampling and resampling techniques [38]–[40].

Taking into account the unique characteristics of target tracking in wireless sensor networks, particularly the interplay between information processing and networking, this chapter develops tracking algorithms that aim to be accurate, robust, and energy efficient for estimating the state of a single target in wireless sensor networks².

For most practical target tracking in wireless sensor networks, clutter due to multi-path effects, spurious objects or sensor errors may yield unlabelled measurements (detector false alarms) at the sensor nodes. In addition, an individual sensor node may fail to detect a target (missed detections). Moreover, multiple targets, which are not sufficiently separated temporarily and spatially in the sensor field, may also lead to unlabelled measurements at sensor nodes. The above measurement origin uncertainty leads to the challenging data association problem [47]. However, this chapter only considers tracking a single target under the dual assumptions of no clutter and missed detection. The single target tracking algorithm under measurement origin uncertainty due to clutter and missed detections will be described in Chapter 5. For the sake of simplicity and without losing the generality, this chapter further assumes that the tracking task is performed within one sensor cluster which consists of a fixed set of sensing nodes (termed as active sensing nodes) and a cluster leader node; at every time step during the whole tracking task, the active sensing nodes provide the cluster leader node with their measurements and the cluster leader node executes the tracking algorithm to update the estimate of the target state. The distributive algorithms for tracking a single target over a series of sensor clusters within a sensor field will be described in Chapter 6. To quantify the theoretical performance that the algorithms developed in this chapter could attain, the posterior Cramer-Rao lower bound (PCRLB) is computed and compared for the developed tracking algorithms. PCRLB under measurement origin uncertainty will be derived and computed in Chapter 5. The multiple target tracking algorithm will be developed in Chapter 7.

The organisation of this chapter is as follows. Section 4.2 formulates the single target tracking problem in the context of wireless sensor networks. Section 4.3 reviews the

² In this chapter and the following chapters, we focus on the timely estimation of the target position from the measurements obtained at the sensing nodes.

general principle of the recursive Bayesian estimation. Section 4.4 through Section 4.7 propose a number of algorithms specified for single target tracking in wireless sensor networks, including the S-EKF, the S-UKF, the PF, and the EKPF. Extensive simulations are conducted to evaluate and compare the performance for these algorithms. The results from these simulations are detailed in Section 4.8. Section 4.9 computes the PCRLB which is the theoretical bound on the mean square error (MSE) of the target state estimate, and also carries out simulations to compare the $\sqrt{\text{PCRLB}}$ with root mean square error (RMSE) for the developed tracking algorithms on some synthetic tracking scenarios. Finally, Section 4.10 summarises this chapter.

4.2 Problem Formulation and System Description

This section formulates the single target tracking problem in the context of wireless sensor networks. The issues to be covered include the target motion model, the nonlinear measurement model, the clutter model, and the dynamic clustering based target tracking scheme.

4.2.1 Formulation of Single Target Tracking Problem in a Wireless Sensor Network

Considering the single target tracking scenario as depicted in Figure 1.1, the key elements of the single target tracking problem can be defined as a tuple, $\mathbf{T}_r = \langle \Psi, \mathbf{P}_\Psi, \mathbf{E}, \mathbf{P}_E, \mathbf{T}_g, \mathbf{S}_m, \mathbf{C}_m \rangle$ [18], [20]. Ψ is the collection of sensor nodes. \mathbf{P}_Ψ specifies the characteristics of each sensor node such as the location, type (sensing node or leader node), sensing modality (acoustic, seismic or magnetic) and residual energy. \mathbf{E} is the link connectivity among sensor nodes, and \mathbf{P}_E specifies the properties of each link such as link capacity and quality. \mathbf{T}_g defines the properties of target such as the target position, velocity and heading. \mathbf{S}_m is the signal model which describes how the target's signal propagates and attenuates in the physical medium. Throughout this thesis, acoustic sensor node is adopted and thus the signal model is the inverse squared distance model [16]. \mathbf{C}_m is used to define the clutter model when tracking target in cluttered environment.

In target tracking, it is common practice to adopt the dynamic state space model [27]. A state space model consists of a *system model* that describes the evolution of a target state and a *measurement model* that relates the measurement to the target state:

System Model (Process Model)

At time step k , the target state vector $\{\mathbf{x}_k; k \in \mathbb{N}\}$, $\mathbf{x}_k \in \mathfrak{R}^{n_x}$ is assumed to be unobserved (hidden) with initial probability density function $p(\mathbf{x}_0)$. The evolution of the state vector \mathbf{x}_k is given by

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_k) \quad (4.1)$$

where $\mathbf{f}_k: \mathfrak{R}^{n_x} \times \mathfrak{R}^{n_v} \rightarrow \mathfrak{R}^{n_x}$ can be linear or nonlinear function, $\{\mathbf{v}_k, k \in \mathbb{N}\}$ is an independent and identically distributed (i.i.d.) process noise vector, and n_x, n_v are dimensions of the state and process noise vectors, respectively.

2. Measurement Model (Observation Model)

At time step k , the measurement vector $\{\mathbf{z}_k; k \in \mathbb{N}\}$, $\mathbf{z}_k \in \mathfrak{R}^{n_z}$ is defined by

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) \quad (4.2)$$

where $\mathbf{h}_k: \mathfrak{R}^{n_x} \times \mathfrak{R}^{n_n} \rightarrow \mathfrak{R}^{n_z}$ can be linear or nonlinear function, $\{\mathbf{n}_k, k \in \mathbb{N}\}$ is an independent and identically distributed (i.i.d.) measurement noise vector, and n_z, n_n are dimensions of measurement and measurement noise vectors, respectively.

In this thesis, it is assumed that the process noise and measurement noise are uncorrelated with each other. The concatenated state vector and the measurement vector from the initial time step to the k -th time step are denoted as $\mathbf{x}_{0:k} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k\}$ and $\mathbf{z}_{0:k} = \{\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_k\}$, respectively.

4.2.2 Target Motion Model

Throughout this thesis, a target is considered as a slowly manoeuvring point object moving in a two-dimensional (2D) plane and thus can be adequately described by the nearly constant-velocity (CV) model [28], [116]. Accordingly, the system model defined in Equation 4.1 becomes,

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{v}_k \quad (4.3)$$

where the target state vector at time step k is given by $\mathbf{x}_k = [x, v_x, y, v_y]_k^T$. x and y are the target positions in x and y coordinate, respectively; v_x and v_y are the target velocities in x and y coordinate, respectively. The process noise vector \mathbf{v}_k is assumed to be Gaussian with zero mean and covariance matrix \mathbf{Q}_k , i.e. $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{Q}_k)$. In Equation 4.3, the system matrix \mathbf{A}_k and the process noise covariance matrix \mathbf{Q}_k are given as follows

$$\mathbf{A}_k = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

and

$$\mathbf{Q}_k = \begin{bmatrix} \frac{1}{3}T^3 & \frac{1}{2}T^2 & 0 & 0 \\ \frac{1}{2}T^2 & T & 0 & 0 \\ 0 & 0 & \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ 0 & 0 & \frac{1}{2}T^2 & T \end{bmatrix} \zeta_k \quad (4.5)$$

where T is the sampling time and ζ_k is the level of the power spectral density of the corresponding continuous process noise [28]. The system matrix \mathbf{A}_k is time invariant in the above nearly CV system model. However, in the following sections, the time index k of \mathbf{A}_k is still kept to account for the sake of generality.

4.2.3 Nonlinear Measurement Model

In this thesis, each individual sensing nodes is equipped with acoustic transducer and the measurement it acquires is a function of the distance between itself and the target. When a target passes the n -th sensing node at the k -th time step, the magnitude of the acoustic intensity measured at this node is [16]:

$$\|\mathbf{z}_k^n\| = \frac{S_k}{\|\mathbf{p}_k - \mathbf{r}_k^n\|^2 + c} + \varepsilon_k^n \quad (4.6)$$

where S_k is the intensity of the acoustic signal generated by the target at the k -th time step, and $\varepsilon_k^n \sim N(0, \mathbf{R}_k^n)$ is the additive Gaussian noise with zero mean and covariance matrix \mathbf{R}_k^n at the n -th sensing node during the k -th time step. \mathbf{p}_k and \mathbf{r}_k^n are the position coordinates of the target and the n -th sensing node, respectively. c is a very small constant related to the size of the target and it helps deal with the situation when the target moves very close to the sensing nodes.

For the sake of simplicity and without losing the generality, the following assumptions for the measurement model are made in this thesis [16], [23]:

1. The distance between a target and its neighbouring sensor nodes should be far enough (but not too far) that the sound source of the target (i.e. the engine of a vehicle) can be modelled as omni-directional; and that the signal propagation delay can be ignored.
2. The target does not undergo sudden manoeuvres relative to the sampling rates of the sensing nodes and thus can be modelled with a near constant velocity (CV) model.
3. All sensor nodes are synchronised.
4. The measurement noises at different sensing nodes are uncorrelated.

4.2.4 Clutter Model

In most target tracking applications in wireless sensor networks, the measurements obtained at individual sensing node may also originate from clutter and the target may even go to undetected [7], [26]. Therefore, at the k -th time step, the n -th sensing node may acquire a group of measurements which are denoted as $\mathbf{z}_k^n = (\mathbf{z}_{0,k}^n, \dots, \mathbf{z}_{l_k^n,k}^n)$, where l_k^n is the total number of measurements acquired by the n -th sensing node at the k -th time step. Among the l_k^n measurements, there will be two types of measurements:

(1) *The target originating measurement* which magnitude is defined in Equation 4.6. Note that Equation 4.6 is nonlinear.

(2) *The clutter originating measurements* that are assumed to be independent and uniformly distributed within the observation space V of a sensing node. In target tracking, it is commonly assumed that the number of the clutter originating measurements in the observation space follows a Poisson probability mass function (pmf) given by [27], [47]

$$P_F^k = \exp(-\lambda V) \frac{(\lambda V)^{l_{c,k}^n}}{l_{c,k}^n !} \quad (4.7)$$

where λ is the clutter rate which is defined as the number of clutter originating measurements per unit volume of the observation space. $l_{c,k}^n$ is the number of clutter originating measurements at the n -th sensing node during the k -th time step.

However, the algorithms developed in this chapter assume that there is one target and no clutter is present; consequently, there is only one measurement at each sensing node. The development of tracking algorithms in cluttered environment is deferred until Chapter 5.

4.2.5 Dynamic Clustering Based Target Tracking Scheme

As discussed in Chapter 3, the hierarchical sensor network architecture is adopted in this thesis. Sensor nodes are dynamically organised into different clusters according to their locations and the tracking tasks. When a leader node detects an approaching target, it recruits its neighbouring sensing nodes to form a sensor cluster. At every time step, this leader node selects a subset of the sensing nodes in the sensor cluster, collecting their measurements, and executing the tracking algorithm to update the estimate of the target state. This process continues until the target moves out of the current sensor cluster.

This chapter is focused on the development of fundamental techniques and algorithms for tracking a single target in a wireless sensor network with the assumptions that the tracking task takes place in one sensor cluster and a fixed set of sensing nodes (active sensing nodes) are activated to sense and provide cluster leader their measurements throughout the whole period of the tracking task. The extension of these algorithms for tracking a single target over a series of sensor clusters will be detailed in Chapter 6.

4.3 The Recursive Bayesian Estimation and Kalman Filter (KF)

Given the state space model (Equations 4.1 and 4.2), the purpose of the recursive Bayesian estimation is to compute the probability density function of the target state \mathbf{x}_k by using the available cumulative measurements $\mathbf{z}_{0:k} = \{ \mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_k \}$ received up to the k -th time step. In the recursive Bayesian estimation, the probability density function $p(\mathbf{x}_k | \mathbf{z}_{0:k})$ of the target state \mathbf{x}_k given the measurements $\mathbf{z}_{0:k}$ contains sufficient statistical information regarding the target state at the k -th time step [35], [39]. The target tracking task can then

be performed as recursively updating the above probability density function when the new measurement becomes available.

Assuming the probability density function $p(\mathbf{x}_{k-1} | \mathbf{z}_{0:k-1})$ at the $(k-1)$ -th time step is already known, the probability density function $p(\mathbf{x}_k | \mathbf{z}_{0:k})$ for the k -th time step can be computed through two steps: the *prediction step* and the *update step*.

1. Prediction Step (time propagation)

Using the state transition probability density function $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ (can be computed via Equation 4.1) to calculate the prior probability density function $p(\mathbf{x}_k | \mathbf{z}_{0:k-1})$ by

$$p(\mathbf{x}_k | \mathbf{z}_{0:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{z}_{0:k-1}) d\mathbf{x}_{k-1} \quad (4.8)$$

2. Update Step (measurement update)

Incorporating the most recent measurement \mathbf{z}_k to update the prior probability density function $p(\mathbf{x}_k | \mathbf{z}_{0:k-1})$ through Bayes law

$$p(\mathbf{x}_k | \mathbf{z}_{0:k}) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{0:k-1})}{p(\mathbf{z}_k | \mathbf{z}_{0:k-1})} \quad (4.9)$$

where

$$p(\mathbf{z}_k | \mathbf{z}_{0:k-1}) = \int p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{0:k-1}) d\mathbf{x}_k \quad (4.10)$$

In the above computation, the initial probability density function of the target state is assumed available, i.e. $p(\mathbf{x}_0 | \mathbf{z}_0) \equiv p(\mathbf{x}_0)$ (\mathbf{z}_0 being the set of no measurements). In Equation 4.9, the probability density function $p(\mathbf{z}_k | \mathbf{x}_k)$ is termed as the *measurement likelihood* and can be obtained from the measurement model (Equation 4.2). The recurrence relations in Equations 4.8 and 4.9 form the basis of the recursive Bayesian estimation and they are general enough for any form of probability density functions (i.e. Gaussian or non-Gaussian) of the target state, all possible functions \mathbf{f}_k and \mathbf{h}_k (i.e. linear or nonlinear), and any noises distributions $p(\mathbf{v}_k)$ and $p(\mathbf{n}_k)$ in the state-space model.

In recursive Bayesian estimation, there exists a very few state estimation problems that satisfy the following linear and Gaussian conditions [29], [35]:

- 1) Both the functions $\mathbf{f}_k : \mathfrak{R}^{n_x} \times \mathfrak{R}^{n_v} \rightarrow \mathfrak{R}^{n_x}$ and $\mathbf{h}_k : \mathfrak{R}^{n_x} \times \mathfrak{R}^{n_n} \rightarrow \mathfrak{R}^{n_z}$ in the state-space model (Equations 4.1 and 4.2) are linear;
- 2) If the probability density function $p(\mathbf{x}_{k-1} | \mathbf{z}_{0:k-1})$ at the $(k-1)$ -th time step (previous time step) is Gaussian, the probability density function $p(\mathbf{x}_k | \mathbf{z}_{0:k})$ at the k -th time step (current time step) is also Gaussian provided that process noise \mathbf{v}_k and measurement noise \mathbf{n}_k are drawn from Gaussian distributions with known parameter.

For these estimation problems, the state space model (Equations 4.1 and 4.2) can be simplified as

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{v}_k \quad (4.11)$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{n}_k \quad (4.12)$$

where \mathbf{A}_k is the known linear state transition matrix (\mathbf{A}_k is defined in Equation 4.4 for nearly CV model), and \mathbf{H}_k is the known linear measurement matrix. The covariance matrices of the process noise vector \mathbf{v}_k and measurement noise vector \mathbf{n}_k satisfy the following equations

$$E[\mathbf{v}_i \mathbf{v}_j^T] = \begin{cases} \mathbf{Q}_k & i = j \\ 0 & i \neq j \end{cases} \quad (4.13)$$

$$E[\mathbf{n}_i \mathbf{n}_j^T] = \begin{cases} \mathbf{R}_k & i = j \\ 0 & i \neq j \end{cases} \quad (4.14)$$

$$E[\mathbf{v}_i \mathbf{n}_j^T] = 0 \quad \text{for all } i \text{ and } j \quad (4.15)$$

Note that in Equations 4.11~4.15 the state transition matrix \mathbf{A}_k , the measurement matrix \mathbf{H}_k , and the noises covariance matrices \mathbf{Q}_k and \mathbf{R}_k are allowed to be time variant.

Under the above linear and Gaussian conditions, the recursions in Equations 4.8~4.10 can be analytically computed by the standard Kalman filter (KF). Assuming the mean vector $\mathbf{m}_{k-1|k-1}$ and the covariance matrix $\mathbf{P}_{k-1|k-1}$ of the probability density function of the target state, $p(\mathbf{x}_{k-1} | \mathbf{z}_{0:k-1})$ at the $(k-1)$ -th time step are already known, we have

$$p(\mathbf{x}_{k-1} | \mathbf{z}_{0:k-1}) = N(\mathbf{x}_{k-1}; \mathbf{m}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}) \quad (4.16)$$

where $N(\mathbf{x}; \mathbf{m}, \mathbf{P})$ is the Gaussian probability density function with argument \mathbf{x} , mean vector \mathbf{m} , and covariance matrix \mathbf{P} as follows³

$$N(\mathbf{x}; \mathbf{m}, \mathbf{P}) = \frac{1}{\sqrt{(2\pi)^{n_x} \|\mathbf{P}\|}} \exp\left\{-\frac{1}{2} [\mathbf{x} - \mathbf{m}]^T \mathbf{P}^{-1} [\mathbf{x} - \mathbf{m}]\right\} \quad (4.17)$$

The KF then calculates the mean vector $\mathbf{m}_{k|k}$ and covariance matrix $\mathbf{P}_{k|k}$ for the k -th time step through the following recursive relationships [29]–[31]:

Prediction Step (Time propagation)

$$p(\mathbf{x}_k | \mathbf{z}_{0:k-1}) = N(\mathbf{x}_k; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) \quad (4.18)$$

where

$$\mathbf{m}_{k|k-1} = \mathbf{A}_k \mathbf{m}_{k-1|k-1} \quad (4.19)$$

$$\mathbf{P}_{k|k-1} = \mathbf{Q}_{k-1} + \mathbf{A}_k \mathbf{P}_{k-1|k-1} \mathbf{A}_k^T \quad (4.20)$$

Update Step (Measurement update)

$$p(\mathbf{x}_k | \mathbf{z}_{0:k}) = N(\mathbf{x}_k; \mathbf{m}_{k|k}, \mathbf{P}_{k|k}) \quad (4.21)$$

where

$$\mathbf{m}_{k|k} = \mathbf{m}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \mathbf{m}_{k|k-1}) \quad (4.22)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k|k-1} \quad (4.23)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \quad (4.24)$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \quad (4.25)$$

³ In all the equations of this chapter, the dimension of the vector \mathbf{x} is denoted by n_x , the transpose of a matrix \mathbf{M} is denoted by \mathbf{M}^T , and the inverse of a matrix \mathbf{M} is denoted by \mathbf{M}^{-1} .

In Equations 4.24 and 4.25, \mathbf{K}_k is the Kalman gain and \mathbf{S}_k is the covariance matrix of the innovation term $\mathbf{z}_k - \mathbf{H}_k \mathbf{m}_{k|k-1}$. Under the above purely linear and Gaussian assumption, the KF is the optimal solution to the state estimation problem in the sense that no other algorithms can outperform it [100].

However, most real-world state estimation problems consist of nonlinear systems (i.e. the measurement model adopted in this thesis is nonlinear); hence the integrals in the recursive Bayesian estimation (e.g. Equations 4.8~4.10) become intractable for these system. Therefore, approximation approaches must be employed. Following sections will develop approximation approaches to the recursive Bayesian estimation under nonlinear and non-Gaussian situations; and on the basis of these approaches, a number of tracking algorithms will be designed for tracking a single target in wireless sensor networks.

4.4 Sequential Extended Kalman Filter (S-EKF) for Tracking a Single

Target in Wireless Sensor Networks

This section starts with a description of the extended Kalman filter (EKF), and then proposes the sequential EKF (S-EKF) tracking algorithm which fits well in the hierarchical sensor network architecture for tracking a single target in wireless sensor networks.

4.4.1 Fundamentals of EKF

If \mathbf{f}_k and \mathbf{h}_k in Equations 4.1 and 4.2 are nonlinear functions, then the state space model cannot be explicitly transformed into the form of Equations 4.11 and 4.12 due to the non-linearity. The EKF approximates this non-linearity by a truncated Taylor expansion of nonlinear functions \mathbf{f}_k and \mathbf{h}_k evaluated around the target state estimation from the previous time step, and accordingly, the probability density function of the target state is approximated by Gaussians [29]–[31]. The process of EKF is detailed as follows.

Assuming the mean vector $\mathbf{m}_{k-1|k-1}$ and covariance matrix $\mathbf{P}_{k-1|k-1}$ of the probability density function of the target state, $p(\mathbf{x}_{k-1} | \mathbf{z}_{0:k-1})$ at the $(k-1)$ -th time step are already known, we have

$$p(\mathbf{x}_{k-1} | \mathbf{z}_{0:k-1}) \approx N(\mathbf{x}_{k-1}; \mathbf{m}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}) \quad (4.26)$$

Following the linearisation, the estimate of target state can be updated in a manner similar to the KF, the mean $\mathbf{m}_{k|k}$ and covariance $\mathbf{P}_{k|k}$ at the k -th time step are computed through the following recursive relationships

Prediction Step (Time propagation)

$$p(\mathbf{x}_k | \mathbf{z}_{0:k-1}) \approx N(\mathbf{x}_k; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) \quad (4.27)$$

where

$$\mathbf{m}_{k|k-1} = \mathbf{f}_k \mathbf{m}_{k-1|k-1} \quad (4.28)$$

$$\mathbf{P}_{k|k-1} = \mathbf{Q}_{k-1} + \hat{\mathbf{A}}_k \mathbf{P}_{k-1|k-1} \hat{\mathbf{A}}_k^T \quad (4.29)$$

Update Step (Measurement update)

$$p(\mathbf{x}_k | \mathbf{z}_{0:k}) \approx N(\mathbf{x}_k; \mathbf{m}_{k|k}, \mathbf{P}_{k|k}) \quad (4.30)$$

where

$$\mathbf{m}_{k|k} = \mathbf{m}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{h}_k \mathbf{m}_{k|k-1}) \quad (4.31)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \hat{\mathbf{H}}_k \mathbf{P}_{k|k-1} \quad (4.32)$$

In the above equations, $\hat{\mathbf{A}}_k$ and $\hat{\mathbf{H}}_k$ are local linearisations of nonlinear functions \mathbf{f}_k and \mathbf{h}_k , i.e. the Jacobian matrices of \mathbf{f}_k and \mathbf{h}_k :

$$\hat{\mathbf{A}}_k = \left. \frac{d\mathbf{f}_k(\mathbf{x})}{d\mathbf{x}} \right|_{\mathbf{x}=\mathbf{m}_{k-1|k-1}} \quad (4.33)$$

$$\hat{\mathbf{H}}_k = \left. \frac{d\mathbf{h}_k(\mathbf{x})}{d\mathbf{x}} \right|_{\mathbf{x}=\mathbf{m}_{k|k-1}} \quad (4.34)$$

The Kalman gain \mathbf{K}_k and the covariance matrix of the innovation term (i.e. $\mathbf{z}_k - \mathbf{H}_k \mathbf{m}_{k|k-1}$) \mathbf{S}_k are computed as follows

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \hat{\mathbf{H}}_k^T \mathbf{S}_k^{-1} \quad (4.35)$$

$$\mathbf{S}_k = \hat{\mathbf{H}}_k \mathbf{P}_{k|k-1} \hat{\mathbf{H}}_k^T + \mathbf{R}_k \quad (4.36)$$

The above EKF utilizes the first term in the Taylor expansion of the nonlinear function. However, the simple “first order Taylor series linearization” employed by the EKF affects the *accuracy* of the final target state estimation; and this may lead to *divergence* of the EKF itself. A higher order EKF that retains further terms in the Taylor expansion exists, however, it will be at the expense of additional complexity. The higher order EKF will not be discussed in this chapter and readers may refer to [29] for more details.

4.4.2 S-EKF for Single Target Tracking in Wireless Sensor Networks

Based on the above EKF, this subsection implements the sequential extended Kalman filter (S-EKF) for tracking a single target in wireless sensor networks. S-EKF runs a separate EKF for each of the measurements collected at a set of sensing nodes in a wireless sensor network. This is possible, if the measurement noise rendering the sensing node is statistically independent between the different sensing nodes; it should be noted that this statistically independent assumption has already been made earlier in this chapter.

In the following discussion we assume that a target is traversing in one sensor cluster which consists of a cluster leader and a fixed set of N_s sensing nodes; at every time step, all these N_s sensing nodes are activated to sense and transmit their measurements to the cluster leader for the target state estimate. The state-space model in Equations 4.1 and 4.2 can be expanded to include the measurements from these N_s sensing nodes as follows

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_k) \quad (4.37)$$

$$\mathbf{z}_k^c = \mathbf{h}_k^c(\mathbf{x}_k, \mathbf{n}_k^c) \quad c = 1, \dots, N_s \quad (4.38)$$

where the superscript c refers to the c -th sensing node, hence \mathbf{z}_k^c , \mathbf{h}_k^c and \mathbf{n}_k^c denote the measurement, the measurement function and the measurement noise at the c -th sensing node during the k -th time step, respectively.

For the measurement received from each sensing node, the cluster leader implements a separate EKF, termed as the sub-EKF; for N_s sensing nodes participating in the tracking task, there will be total N_s sub-EKFs. The sub-EKF for the measurement obtained from the c -th sensing node is indexed as the c -th sub-EKF. Accordingly, the last sub-EKF is

indexed as the N_s -th sub-EKF. In S-EKF, the estimation result from the previous sub-EKF becomes the prior estimation for the next sub-EKF.

Assuming the mean vector $\mathbf{m}_{k-1|k-1}$ and covariance matrix $\mathbf{P}_{k-1|k-1}$ of the probability density function $p(\mathbf{x}_{k-1}|\mathbf{z}_{0:k-1})$ at the $(k-1)$ -th time step is already known, the S-EKF computes the mean and covariance of the probability density function $p(\mathbf{x}_k|\mathbf{z}_{0:k})$ at the k -th time step is as follows.

At the beginning of the k -th time step, mean $\mathbf{m}_{k-1|k-1}$ and covariance matrix $\mathbf{P}_{k-1|k-1}$ become the input (i.e. prior estimate) of the first sub-EKF as in the non-sequential version of EKF:

$$\mathbf{m}_{k|k-1}^1 = \mathbf{f}_k \mathbf{m}_{k-1|k-1} \quad (4.39)$$

$$\mathbf{P}_{k|k-1}^1 = \mathbf{Q}_{k-1} + \hat{\mathbf{A}}_k \mathbf{P}_{k-1|k-1} \hat{\mathbf{A}}_k^T \quad (4.40)$$

In the above equations, the superscript “1” of mean $\mathbf{m}_{k|k-1}^1$ and covariance matrix $\mathbf{P}_{k|k-1}^1$ refers to the first sub-EKF. The computation of $\hat{\mathbf{A}}_k$ is the same as in EKF (Equation 4.33). The following sub-EKFs, for example, the c -th sub-EKF, uses the output from the $(c-1)$ -th sub-EKF as its prior target state estimate in the prediction step and the measurement obtained from the c -th sensing node in the update step; and subsequently, the output of the c -th sub-EKF is passed to the next sub-EKF, i.e. the $(c+1)$ -th sub-EKF as its prior target state estimate. Finally, the output of the last sub-EKF, the N_s -th sub-EKF is used as the target state estimate of the entire S-EKF. The above process is depicted in Figure 4.1 and can be described by the following equations:

$$\mathbf{m}_{k|k}^1 = \mathbf{m}_{k|k-1}^1 + \mathbf{K}_k^1 (\mathbf{z}_k^1 - \mathbf{h}_k^1 \mathbf{m}_{k|k-1}^1) \quad (4.41)$$

$$\mathbf{m}_{k|k}^c = (\mathbf{f}_k \mathbf{m}_{k|k}^{c-1}) + \mathbf{K}_k^c [\mathbf{z}_k^c - \mathbf{h}_k^c (\mathbf{f}_k \mathbf{m}_{k|k}^{c-1})] \quad c = 2, \dots, N_s \quad (4.42)$$

$$\mathbf{m}_{k|k} = \mathbf{m}_{k|k}^{N_s} \quad (4.43)$$

where the Kalman gains for the first sub-EKF and the c -th sub-EKF are:

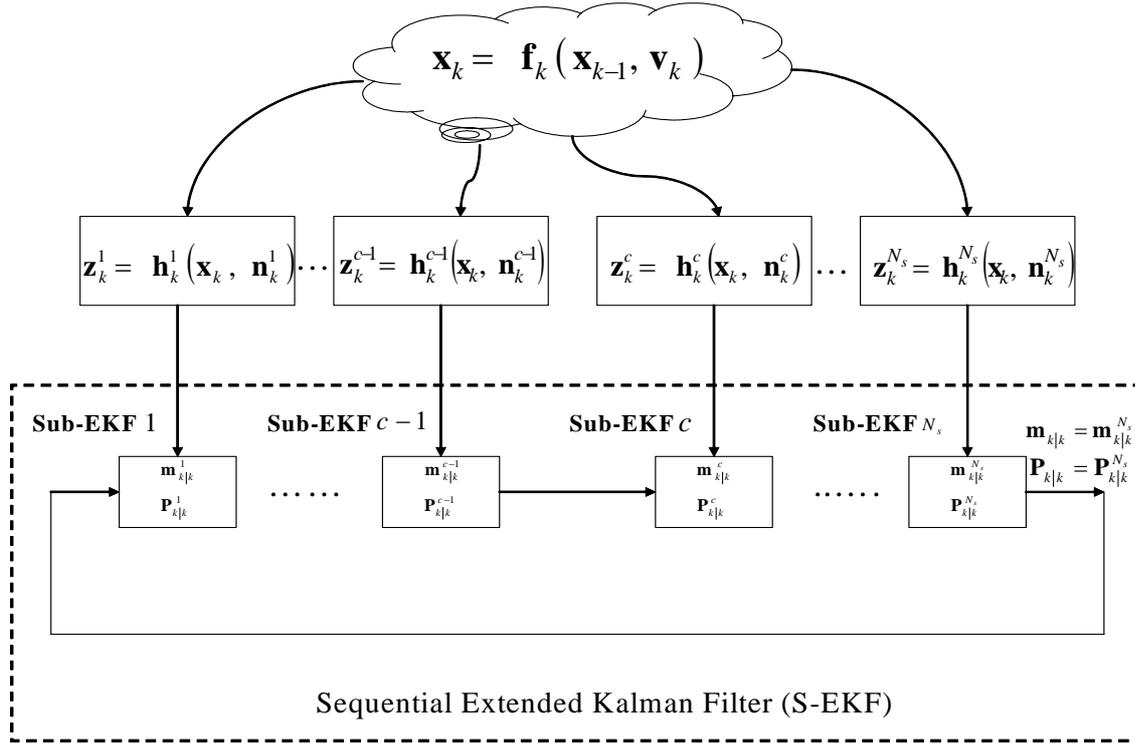


Figure 4.1 The illustrative flowchart of S-EKF algorithm

$$\mathbf{K}_k^1 = \mathbf{P}_{k|k-1}^1 (\hat{\mathbf{H}}_k^1)^T (\mathbf{S}_k^1)^{-1} \quad (4.44)$$

$$\begin{aligned} \mathbf{K}_k^c &= \mathbf{P}_{k|k-1}^c (\hat{\mathbf{H}}_k^c)^T (\mathbf{S}_k^c)^{-1} \\ &= (\mathbf{Q}_{k-1} + \hat{\mathbf{A}}_k \mathbf{P}_{k|k}^{c-1} \hat{\mathbf{A}}_k^T) (\hat{\mathbf{H}}_k^c)^T (\mathbf{S}_k^c)^{-1} \quad c = 2, \dots, N_s \end{aligned} \quad (4.45)$$

and the covariance matrices of the innovation terms are:

$$\mathbf{S}_k^1 = \hat{\mathbf{H}}_k^1 \mathbf{P}_{k|k-1}^1 (\hat{\mathbf{H}}_k^1)^T + \mathbf{R}_k^1 \quad (4.46)$$

$$\begin{aligned} \mathbf{S}_k^c &= \hat{\mathbf{H}}_k^c \mathbf{P}_{k|k-1}^c (\hat{\mathbf{H}}_k^c)^T + \mathbf{R}_k^c \\ &= \hat{\mathbf{H}}_k^c (\mathbf{Q}_{k-1} + \hat{\mathbf{A}}_k \mathbf{P}_{k|k}^{c-1} \hat{\mathbf{A}}_k^T) (\hat{\mathbf{H}}_k^c)^T + \mathbf{R}_k^c \quad c = 2, \dots, N_s \end{aligned} \quad (4.47)$$

and the Jacobian of the measurement function for sub-EKFs are:

$$\hat{\mathbf{H}}_k^1 = \left. \frac{d\mathbf{h}_k^1(\mathbf{x})}{d\mathbf{x}} \right|_{\mathbf{x}=\mathbf{m}_{k|k-1}^1}, \quad \hat{\mathbf{H}}_k^c = \left. \frac{d\mathbf{h}_k^c(\mathbf{x})}{d\mathbf{x}} \right|_{\mathbf{x}=\mathbf{f}_k \mathbf{m}_{k|k}^{c-1}}, \quad c = 2, \dots, N_s \quad (4.48)$$

The state covariance matrix is updated for each sub-EKF by

$$\mathbf{P}_{k|k}^1 = \mathbf{P}_{k|k-1}^1 - \mathbf{K}_k^1 \hat{\mathbf{H}}_k^1 \mathbf{P}_{k|k-1}^1 \quad (4.49)$$

$$\begin{aligned} \mathbf{P}_{k|k}^c &= \mathbf{P}_{k|k-1}^c - \mathbf{K}_k^c \hat{\mathbf{H}}_k^c \mathbf{P}_{k|k-1}^c \\ &= \left(\mathbf{Q}_{k-1} + \hat{\mathbf{A}}_k \mathbf{P}_{k|k}^{c-1} \hat{\mathbf{A}}_k^T \right) - \mathbf{K}_k^c \hat{\mathbf{H}}_k^c \left(\mathbf{Q}_{k-1} + \hat{\mathbf{A}}_k \mathbf{P}_{k|k}^{c-1} \hat{\mathbf{A}}_k^T \right) \end{aligned} \quad (4.50)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k}^{N_s} \quad (4.51)$$

The complete S-EKF tracking algorithm is listed in Algorithm 4.1.

The above S-EKF algorithm has several advantages: the architecture is easily extendable, and more sensing nodes can be added by simply putting them into the chain, instead of reconfiguring the entire algorithm; if a sensing node cannot provide a measurement at a particular time step, the corresponding sub-EKF can be simply skipped. The above advantages will enable S-EKF to achieve better scalability and adaptability when it is applied to track a single target in wireless sensor networks. However, S-EKF suffers from the same problem as EKF, the simple “first order Taylor series linearization” may also lead to the divergence of the entire algorithm, especially when there exists high nonlinearity in the tracking problem.

Algorithm 4.1 Sequential Extended Kalman Filter (S-EKF) for Single Target Tracking in Wireless Sensor Networks

- State-Space Model (assuming there is a fix set of N_s sensing nodes are activated to sense and transmit their measurements to the cluster leader at every time step)

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_k) \\ \mathbf{z}_k^c &= \mathbf{h}_k^c(\mathbf{x}_k, \mathbf{n}_k^c), \quad c = 1, \dots, N_s \end{aligned}$$

- Initialization (for time step $k = 0$)

$$\begin{aligned} \mathbf{m}_{0|0} &= E[\mathbf{x}_0] \\ \mathbf{P}_{0|0} &= E\left[(\mathbf{m}_{0|0} - \mathbf{x}_0) (\mathbf{m}_{0|0} - \mathbf{x}_0)^T \right] \\ E[\mathbf{v}_i \mathbf{v}_j^T] &= \begin{cases} \mathbf{Q}_k & i = j \\ 0 & i \neq j \end{cases} \\ E\left[\mathbf{n}_i^c (\mathbf{n}_j^c)^T \right] &= \begin{cases} \mathbf{R}_k^c & i = j \\ 0 & i \neq j \end{cases}, \quad c = 1, \dots, N_s \end{aligned}$$

$$E \left[\mathbf{n}_k^c (\mathbf{n}_k^\tau)^T \right] = \begin{cases} \mathbf{R}_k^c & \tau = c \\ 0 & \tau \neq c \end{cases}, \quad c, \tau = 1, \dots, N_s$$

- For time step $k = 1, 2, \dots$

1. Compute the Jacobians for system model

$$\hat{\mathbf{A}}_k = \left. \frac{d\mathbf{f}_k(\mathbf{x})}{d\mathbf{x}} \right|_{\mathbf{x}=\mathbf{m}_{k-1|k-1}}$$

2. Run the first Sub-EKF as follows

$$\mathbf{m}_{k|k-1}^1 = \mathbf{f}_k \mathbf{m}_{k-1|k-1}$$

$$\mathbf{P}_{k|k-1}^1 = \mathbf{Q}_{k-1} + \hat{\mathbf{A}}_k \mathbf{P}_{k-1|k-1} \hat{\mathbf{A}}_k^T$$

$$\hat{\mathbf{H}}_k^1 = \left. \frac{d\mathbf{h}_k^1(\mathbf{x})}{d\mathbf{x}} \right|_{\mathbf{x}=\mathbf{m}_{k|k-1}^1}$$

$$\mathbf{S}_k^1 = \hat{\mathbf{H}}_k^1 \mathbf{P}_{k|k-1}^1 (\hat{\mathbf{H}}_k^1)^T + \mathbf{R}_k^1$$

$$\mathbf{K}_k^1 = \mathbf{P}_{k|k-1}^1 (\hat{\mathbf{H}}_k^1)^T (\mathbf{S}_k^1)^{-1}$$

$$\mathbf{m}_{k|k}^1 = \mathbf{m}_{k|k-1}^1 + \mathbf{K}_k^1 (\mathbf{z}_k^1 - \mathbf{h}_k^1 \mathbf{m}_{k|k-1}^1)$$

$$\mathbf{P}_{k|k}^1 = \mathbf{P}_{k|k-1}^1 - \mathbf{K}_k^1 \hat{\mathbf{H}}_k^1 \mathbf{P}_{k|k-1}^1$$

3. Run Sub-EKFs for $c = 2, \dots, N_s$ in sequence as follows

$$\hat{\mathbf{H}}_k^c = \left. \frac{d\mathbf{h}_k^c(\mathbf{x})}{d\mathbf{x}} \right|_{\mathbf{x}=\mathbf{f}_k \mathbf{m}_{k|k}^{c-1}}$$

$$\mathbf{S}_k^c = \hat{\mathbf{H}}_k^c (\mathbf{Q}_{k-1} + \hat{\mathbf{A}}_k \mathbf{P}_{k|k}^{c-1} \hat{\mathbf{A}}_k^T) (\hat{\mathbf{H}}_k^c)^T + \mathbf{R}_k^c$$

$$\mathbf{K}_k^c = (\mathbf{Q}_{k-1} + \hat{\mathbf{A}}_k \mathbf{P}_{k|k}^{c-1} \hat{\mathbf{A}}_k^T) (\hat{\mathbf{H}}_k^c)^T (\mathbf{S}_k^c)^{-1}$$

$$\mathbf{m}_{k|k}^c = (\mathbf{f}_k \mathbf{m}_{k|k}^{c-1}) + \mathbf{K}_k^c [\mathbf{z}_k^c - \mathbf{h}_k^c (\mathbf{f}_k \mathbf{m}_{k|k}^{c-1})]$$

$$\mathbf{P}_{k|k}^c = (\mathbf{Q}_{k-1} + \hat{\mathbf{A}}_k \mathbf{P}_{k|k}^{c-1} \hat{\mathbf{A}}_k^T) - \mathbf{K}_k^c \hat{\mathbf{H}}_k^c (\mathbf{Q}_{k-1} + \hat{\mathbf{A}}_k \mathbf{P}_{k|k}^{c-1} \hat{\mathbf{A}}_k^T)$$

4. The output for time step k is then

$$\mathbf{m}_{k|k} = \mathbf{m}_{k|k}^{N_s} \quad \text{and} \quad \mathbf{P}_{k|k} = \mathbf{P}_{k|k}^{N_s}$$

4.5 Sequential Unscented Kalman Filter (S-UKF) for Tracking

a Single Target in Wireless Sensor Networks

This section starts with a description of the unscented Kalman filter (UKF), and then develops the sequential UKF (S-UKF) algorithm for tracking a single target in wireless sensor networks.

4.5.1 Fundamentals of UKF

Unlike the EKF, the UKF does not approximate the nonlinear state space model directly. In contrast, it uses a minimal set of deterministically chosen sample points to approximate the probability density function of a random vector (i.e. the target state in this thesis) [32], [33]. These sample points capture the mean and covariance of the random vector, and when the random vector undergoes a nonlinear transformation, they can capture the mean and covariance accurately to the second order with errors only in the third- and higher-order. Thus, the performance of UKF is generally more advanced than that of EKF when they are applied to the nonlinear systems.

The core of a UKF algorithm is a deterministic sampling approach called the *unscented transformation (UT)* for the calculation of the statistics of a random vector undergoing a nonlinear transformation. Supposing a n_x -dimension random vector \mathbf{x} with known mean \mathbf{m}_x and covariance \mathbf{P}_x is taking a nonlinear transformation $\mathbf{y} = \mathbf{g}(\mathbf{x})$, now the task is to calculate the mean \mathbf{m}_y and covariance \mathbf{P}_y of random vector \mathbf{y} . To do so, the UT deterministically chooses a set of $2n_x + 1$ weighted samples $\{\boldsymbol{\chi}_i, \omega_i\}_{i=0}^{2n_x}$ as follows [33]:

$$\begin{aligned}
 \boldsymbol{\chi}_0 &= \mathbf{m}_x, & \omega_0 &= \frac{\kappa}{n_x + \kappa}, & i &= 0 \\
 \boldsymbol{\chi}_i &= \mathbf{m}_x + \left(\sqrt{(n_x + \kappa) \mathbf{P}_x} \right)_i, & \omega_i &= \frac{1}{2(n_x + \kappa)}, & i &= 1, \dots, n_x \\
 \boldsymbol{\chi}_i &= \mathbf{m}_x - \left(\sqrt{(n_x + \kappa) \mathbf{P}_x} \right)_i, & \omega_i &= \frac{1}{2(n_x + \kappa)}, & i &= n_x + 1, \dots, 2n_x
 \end{aligned} \tag{4.52}$$

where κ is a scaling parameter and $\left(\sqrt{(n_x + \kappa) \mathbf{P}_x} \right)_i$ is the i th column (or row) of the matrix square root of $(n_x + \kappa) \mathbf{P}_x$. Each weight ω_i is associated with a sample point $\boldsymbol{\chi}_i$

and $\sum_{i=0}^{2n_x} \omega_i = 1$. The mean \mathbf{m}_y and covariance \mathbf{P}_y of random vector \mathbf{y} can be computed by the weighted summation of these sample points

$$\mathbf{m}_y \approx \sum_{i=0}^{2n_x} \omega_i \mathbf{g}(\boldsymbol{\chi}_i) \quad (4.53)$$

$$\mathbf{P}_y \approx \sum_{i=0}^{2n_x} \omega_i [\mathbf{g}(\boldsymbol{\chi}_i) - \mathbf{m}_y][\mathbf{g}(\boldsymbol{\chi}_i) - \mathbf{m}_y]^T \quad (4.54)$$

The above calculations are accurate to the second order of the Taylor series expansion of $\mathbf{y} = \mathbf{g}(\mathbf{x})$ for any nonlinear function. Errors are introduced in the third and higher order moments but can be scaled by properly choosing the parameter κ . However, if the nonlinearity is very severe, the above UT approximation may be inaccurate, and the *scaled unscented transformation (SUT)* is introduced [32], [33]. SUT employs another positive scaling parameter α to provide an extra degree of freedom to control the scaling of the sample points. Using SUT, the set of sample points $\{\boldsymbol{\chi}_i, \omega_i\}_{i=0}^{2n_x}$ is transformed into the scaled set $\{\boldsymbol{\chi}'_i, \omega'_i\}_{i=0}^{2n_x}$ by

$$\begin{aligned} \boldsymbol{\chi}'_i &= \boldsymbol{\chi}_0 + \alpha(\boldsymbol{\chi}_i - \boldsymbol{\chi}_0) \\ \omega'_i &= \begin{cases} \omega_0/\alpha^2 + (1-1/\alpha^2) & i=0 \\ \omega_i/\alpha^2 & i=1, \dots, 2n_x \end{cases} \end{aligned} \quad (4.55)$$

The unscented Kalman filter (UKF) is a straightforward application of SUT to Kalman filter (KF) framework. However, the state vector in UKF is redefined as the concatenation of the original state vector, the process noise vector, and the measurement noise vector, i.e.

$$\mathbf{x}_k^a = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{v}_k \\ \mathbf{n}_k \end{bmatrix} \quad k = 0, 1, 2, \dots \quad (4.56)$$

The effective dimension of this augmented state vector is now $n_a = n_x + n_v + n_n$, where n_x is the dimension of original state vector, n_v is the dimension of process noise vector and n_n is the dimension of measurement noise vector. Similarly, the augmented state covariance matrix is built up by concatenating the covariance matrices of the target state, the process noise and the measurement noise:

$$\mathbf{P}_k^a = \begin{bmatrix} \mathbf{P}_{k|k} & 0 & 0 \\ 0 & \mathbf{Q}_k & 0 \\ 0 & 0 & \mathbf{R}_k \end{bmatrix} \quad k = 0, 1, 2, \dots \quad (4.57)$$

where $\mathbf{P}_{k|k}$, \mathbf{Q}_k and \mathbf{R}_k are the covariance matrices of target state, process noise and measurement noise at the k -th time step, respectively. By augmenting the target state with the process and measurement noises, the uncertainties in the noises are taken into account. This allows for the effect of the noises on the system dynamic and measurement to be captured with the same level of accuracy as with which the target state is treated. The full UKF algorithm is listed as below. More details can be found in [32], [33], [35].

Algorithm 4.2 Unscented Kalman Filter (UKF)

Definitions:

$$\mathbf{m}_{0|0} = E[\mathbf{x}_0], \quad \mathbf{P}_{0|0} = E \left[\left(\mathbf{m}_{0|0} - \mathbf{x}_0 \right) \left(\mathbf{m}_{0|0} - \mathbf{x}_0 \right)^T \right],$$

$$\mathbf{x}_k^a = \left[\mathbf{x}_k^T \quad \mathbf{v}_k^T \quad \mathbf{n}_k^T \right]^T, \quad \boldsymbol{\chi}_k^a = \left[\left(\boldsymbol{\chi}_k^x \right)^T \quad \left(\boldsymbol{\chi}_k^v \right)^T \quad \left(\boldsymbol{\chi}_k^n \right)^T \right]^T.$$

➤ For $k = 0$

$$\mathbf{m}_{0|0}^a = \left[\mathbf{m}_{0|0} \quad 0 \quad 0 \right]^T \quad (4.58)$$

$$\mathbf{P}_{0|0}^a = \begin{bmatrix} \mathbf{P}_{0|0} & 0 & 0 \\ 0 & \mathbf{Q}_0 & 0 \\ 0 & 0 & \mathbf{R}_0 \end{bmatrix} \quad (4.59)$$

➤ For $k = 1, 2, \dots$

1. Calculate sample points and the corresponding weights:

$$\boldsymbol{\chi}_{k-1|k-1}^a = \left[\mathbf{m}_{k-1|k-1}^a \quad \mathbf{m}_{k-1|k-1}^a + \sqrt{(n_a + \lambda) \mathbf{P}_{k-1|k-1}^a} \quad \mathbf{m}_{k-1|k-1}^a - \sqrt{(n_a + \lambda) \mathbf{P}_{k-1|k-1}^a} \right] \quad (4.60)$$

where

$$\left(\boldsymbol{\chi}_{k-1|k-1}^{\mathbf{a}}\right)_0 = \mathbf{m}_{k-1|k-1}^{\mathbf{a}}, \quad \omega_0^{(\mathbf{m})} = \frac{\lambda}{n_{\mathbf{a}} + \lambda}, \quad \omega_0^{(\mathbf{p})} = \frac{\lambda}{n_{\mathbf{a}} + \lambda} + (1 - \alpha^2 + \beta) \quad (4.61)$$

$$\left(\boldsymbol{\chi}_{k-1|k-1}^{\mathbf{a}}\right)_i = \mathbf{m}_{k-1|k-1}^{\mathbf{a}} \pm \left(\sqrt{(n_{\mathbf{a}} + \lambda) \mathbf{P}_{k-1|k-1}^{\mathbf{a}}}\right)_i, \quad \omega_i^{(\mathbf{m})} = \omega_i^{(\mathbf{p})} = \frac{1}{2(n_{\mathbf{a}} + \lambda)}, \quad i = 1, \dots, 2n_{\mathbf{a}} \quad (4.62)$$

in above equations, $\lambda = \alpha^2(n_{\mathbf{a}} + \kappa) - n_{\mathbf{a}}$ is the scaling parameter. Parameter α determines the spread of the sample points around \mathbf{m}_x and is usually set to a small positive value. Parameter κ is a secondary scaling parameter which is set to either 0 or $3 - n_{\mathbf{a}}$. Parameter β is a scalar parameter which provides an extra degree of freedom and used to incorporate any extra prior knowledge of the probability distribution of target state \mathbf{x}_k ($\beta = 2$ for Gaussian distribution) [33], [35].

2. Prediction Step (Time-update):

$$\boldsymbol{\chi}_{k|k-1}^{\mathbf{x}} = \mathbf{f}\left(\boldsymbol{\chi}_{k-1|k-1}^{\mathbf{x}}, \boldsymbol{\chi}_k^{\mathbf{v}}\right) \quad (4.63)$$

$$\mathbf{m}_{k|k-1} = \sum_{i=0}^{2n_{\mathbf{a}}} \omega_i^{(\mathbf{m})} \left(\boldsymbol{\chi}_{k-1|k-1}^{\mathbf{x}}\right)_i \quad (4.64)$$

$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{2n_{\mathbf{a}}} \omega_i^{(\mathbf{p})} \left[\left(\boldsymbol{\chi}_{k-1|k-1}^{\mathbf{x}}\right)_i - \mathbf{m}_{k|k-1}\right] \left[\left(\boldsymbol{\chi}_{k-1|k-1}^{\mathbf{x}}\right)_i - \mathbf{m}_{k|k-1}\right]^T \quad (4.65)$$

3. Update Step (Measurement update):

$$\left(\mathbf{z}_{k|k-1}\right)_i = \mathbf{h}\left(\boldsymbol{\chi}_{k|k-1}^{\mathbf{x}}, \boldsymbol{\chi}_k^{\mathbf{n}}\right) \quad (4.66)$$

$$\hat{\mathbf{z}}_{k|k-1} = \sum_{i=0}^{2n_{\mathbf{a}}} \omega_i^{(\mathbf{m})} \left(\mathbf{z}_{k|k-1}\right)_i \quad (4.67)$$

$$\mathbf{P}_{\mathbf{z}_k \mathbf{z}_k} = \sum_{i=0}^{2n_{\mathbf{a}}} \omega_i^{(\mathbf{p})} \left[\left(\mathbf{z}_{k|k-1}\right)_i - \hat{\mathbf{z}}_{k|k-1}\right] \left[\left(\mathbf{z}_{k|k-1}\right)_i - \hat{\mathbf{z}}_{k|k-1}\right]^T \quad (4.68)$$

$$\mathbf{P}_{\mathbf{x}_k \mathbf{z}_k} = \sum_{i=0}^{2n_{\mathbf{a}}} \omega_i^{(\mathbf{p})} \left[\left(\boldsymbol{\chi}_{k|k-1}^{\mathbf{x}}\right)_i - \mathbf{m}_{k|k-1}\right] \left[\left(\mathbf{z}_{k|k-1}\right)_i - \hat{\mathbf{z}}_{k|k-1}\right]^T \quad (4.69)$$

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{x}_k \mathbf{z}_k} \left(\mathbf{P}_{\mathbf{z}_k \mathbf{z}_k}\right)^{-1} \quad (4.70)$$

$$\mathbf{m}_{k|k} = \mathbf{m}_{k|k-1} + \mathbf{K}_k \left(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1} \right) \quad (4.71)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{P}_{\mathbf{z}_k \mathbf{z}_k} \mathbf{K}_k^T \quad (4.72)$$

4.5.2 S-UKF for Single Target Tracking in Wireless Sensor Networks

Analogous to the S-EKF as proposed in the Sub-section 4.4.2, a sequential UKF algorithm (S-UKF) employing a bank of UKF is also developed for tracking a single target in wireless sensor networks. The process of S-UKF is quite similar to that of S-EKF and is briefly presented as follows. At each time step, a fixed set of N_s sensing nodes transmit their measurements to the cluster leader. For each received measurement from these sensing nodes, the cluster leader implements a separate UKF, called the sub-UKF. The output (the target state estimate) from the previous sub-UKF becomes the input of the next sub-UKF. For example, the c -th sub-UKF uses the output from the $(c-1)$ -th sub-UKF as the prior target state estimate in the prediction step and the measurement from the c -th sensing node for the update step. The result of the c -th sub-UKF is then passed to the next sub-UKF, i.e. the $(c+1)$ -th sub-UKF. Finally, the output of the last sub-UKF N_s is used as the target state estimate of the entire S-UKF algorithm. For the complete S-UKF tracking algorithm, one only needs to substitute the equations of EKF in both prediction step and update step in Algorithm 4.1 with those of S-UKF (Algorithm 4.2). However, the full S-UKF is not listed here.

As S-EKF, the advantages of S-UKF also lie in its scalability and adaptability to the frequently topology changes in wireless sensor networks. In theory, the S-UKF consistently outperforms the S-EKF in terms of accuracy and robustness because UKF captures the mean and covariance of target state estimate accurately to the second order with errors only introduced in the third and higher orders. However, in practical applications, the performance of S-UKF and S-EKF will be problem dependent, and the S-UKF is not necessarily outperforms the S-EKF. As shown in the simulation, when the measurement model is nonlinear but the system model is linear (recalled that in this thesis we adopt the nearly constant velocity (CV) system model which is linear), the improvement of S-UKF over S-EKF is not very significant. In addition, the computation cost of S-UKF is higher than that of S-EKF.

In the following two sections, we use particles to represent the probability density function of the target state; and the resulting two algorithms – PF and EKPF can be applied to more general nonlinear and non-Gaussian systems. The simulation results show that EKPF consistently outperforms S-EKF and S-UKF in all synthetic tracking scenarios while PF outperforms S-EKF and S-UKF in most synthetic tracking scenarios.

4.6 Generic Particle Filter (PF) for Tracking a Single

Target in Wireless Sensor Networks

Although UKF and EKF adopt different approach to approximate the nonlinearity of the state-space model, they both assume a Gaussian probability density function of the target state to perform the recursive Bayesian estimation. In contrast, the Particle filter (PF) does not make any explicit assumptions on the form of the probability density function of the target state. Instead, PF approximates the probability density function of the target state by a set of weighted random samples (called particles); and hence it can be applied to more general nonlinear, non-Gaussian systems [38]–[45].

Starting with an introduction of Monte Carlo simulation and importance sampling, this section develops the generic PF algorithm which is based on the sequential importance sampling and resampling techniques for tracking a single target in wireless sensor networks.

4.6.1 Monte Carlo Simulation and Importance Sampling

In the Monte Carlo simulation, a set of weighted samples are randomly drawn from a given probability density function of the state vector $p(\mathbf{x}_{0:k} | \mathbf{z}_{0:k})$; and then these samples can be used to approximate this probability density function by the following empirical estimate:

$$p(\mathbf{x}_{0:k} | \mathbf{z}_{0:k}) \approx \tilde{p}(\mathbf{x}_{0:k} | \mathbf{z}_{0:k}) = \frac{1}{N} \sum_{i=1}^N \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i) \quad (4.73)$$

where $\delta(\cdot)$ is the Dirac delta function and $\{\mathbf{x}_{0:k}^i\}_{i=1}^N$ are the N weighted samples drawn from $p(\mathbf{x}_{0:k} | \mathbf{z}_{0:k})$. These samples are assumed to be independent and identically distributed (i.i.d.). For any expectation of the following form

$$E[\mathbf{g}(\mathbf{x}_{0:k})] = \int \mathbf{g}(\mathbf{x}_{0:k}) p(\mathbf{x}_{0:k} | \mathbf{z}_{0:k}) d\mathbf{x}_{0:k} \quad (4.74)$$

it can be approximated by the above weighted samples, i.e.

$$E[\mathbf{g}(\mathbf{x}_{0:k})] \approx \tilde{E}[\mathbf{g}(\mathbf{x}_{0:k})] = \frac{1}{N} \sum_{i=1}^N \mathbf{g}(\mathbf{x}_{0:k}^i) \quad (4.75)$$

According to the law of large numbers, as the sample numbers N increases, the above approximated expectation eventually converges to the true expectation:

$$\tilde{E}[\mathbf{g}(\mathbf{x}_{0:k})] \xrightarrow{N \rightarrow \infty} E[\mathbf{g}(\mathbf{x}_{0:k})] \quad (4.76)$$

However, in most target tracking applications, it is almost impossible to draw samples directly from the probability density function $p(\mathbf{x}_{0:k} | \mathbf{z}_{0:k})$ as in Equation 4.73. The PF circumvents this difficulty by adopting a technique called *importance sampling* in which the samples are instead drawn from a known, easy-to-sample, *proposal distribution* $\pi(\mathbf{x}_{0:k} | \mathbf{z}_{0:k})$. Therefore, the selection of such proposal distribution is a critical issue in the PF design.

Assuming a proposal distribution $\pi(\mathbf{x}_{0:k} | \mathbf{z}_{0:k})$ has already been obtained, Equation 4.74 can be changed to the following form by using Bayes law:

$$\begin{aligned} E[\mathbf{g}(\mathbf{x}_{0:k})] &= \int \mathbf{g}(\mathbf{x}_{0:k}) \frac{p(\mathbf{x}_{0:k} | \mathbf{z}_{0:k})}{\pi(\mathbf{x}_{0:k} | \mathbf{z}_{0:k})} \pi(\mathbf{x}_{0:k} | \mathbf{z}_{0:k}) d\mathbf{x}_{0:k} \\ &= \int \mathbf{g}(\mathbf{x}_{0:k}) \frac{p(\mathbf{z}_{0:k} | \mathbf{x}_{0:k}) p(\mathbf{x}_{0:k})}{p(\mathbf{z}_{0:k}) \pi(\mathbf{x}_{0:k} | \mathbf{z}_{0:k})} \pi(\mathbf{x}_{0:k} | \mathbf{z}_{0:k}) d\mathbf{x}_{0:k} \quad (4.77) \\ &= \int \mathbf{g}(\mathbf{x}_{0:k}) \frac{w_{0:k}}{p(\mathbf{z}_{0:k})} \pi(\mathbf{x}_{0:k} | \mathbf{z}_{0:k}) d\mathbf{x}_{0:k} \end{aligned}$$

where the variables $w_{0:k}$ are called the *un-normalized importance weights*, and given by

$$w_{0:k} = \frac{p(\mathbf{z}_{0:k} | \mathbf{x}_{0:k}) p(\mathbf{x}_{0:k})}{\pi(\mathbf{x}_{0:k} | \mathbf{z}_{0:k})} \quad (4.78)$$

$p(\mathbf{z}_{0:k})$ in Equation 4.77 is normally unknown and difficult to calculate, however, it can be eliminated by the following derivations:

$$\begin{aligned}
E[\mathbf{g}(\mathbf{x}_{0:k})] &= \frac{1}{p(\mathbf{z}_{0:k})} \int \mathbf{g}(\mathbf{x}_{0:k}) w_{0:k} \pi(\mathbf{x}_{0:k} | \mathbf{z}_{0:k}) d\mathbf{x}_{0:k} \\
&= \frac{\int \mathbf{g}(\mathbf{x}_{0:k}) w_{0:k} \pi(\mathbf{x}_{0:k} | \mathbf{z}_{0:k}) d\mathbf{x}_{0:k}}{\int p(\mathbf{z}_{0:k} | \mathbf{x}_{0:k}) p(\mathbf{x}_{0:k}) \frac{\pi(\mathbf{x}_{0:k} | \mathbf{z}_{0:k})}{\pi(\mathbf{x}_{0:k} | \mathbf{z}_{0:k})} d\mathbf{x}_{0:k}} \\
&= \frac{\int \mathbf{g}(\mathbf{x}_{0:k}) w_{0:k} \pi(\mathbf{x}_{0:k} | \mathbf{z}_{0:k}) d\mathbf{x}_{0:k}}{\int w_{0:k} \pi(\mathbf{x}_{0:k} | \mathbf{z}_{0:k}) d\mathbf{x}_{0:k}} \\
&= \frac{E_{\pi}[w_{0:k} \mathbf{g}(\mathbf{x}_{0:k})]}{E_{\pi}[w_{0:k}]}
\end{aligned} \tag{4.79}$$

where $E_{\pi}[\cdot]$ is denoted the expectation that is taken over the proposal distribution $\pi(\mathbf{x}_{0:k} | \mathbf{z}_{0:k})$. Hence, by drawing samples from $\pi(\mathbf{x}_{0:k} | \mathbf{z}_{0:k})$, the expectations of interest $E[\mathbf{g}(\mathbf{x}_{0:k})]$ can be approximated by the following estimate:

$$\begin{aligned}
E[\mathbf{g}(\mathbf{x}_{0:k})] &\approx \frac{\frac{1}{N} \sum_{i=1}^N \mathbf{g}(\mathbf{x}_{0:k}^i) w_{0:k}^i}{\frac{1}{N} \sum_{i=1}^N w_{0:k}^i} \\
&= \sum_{i=1}^N \tilde{w}_{0:k}^i \mathbf{g}(\mathbf{x}_{0:k}^i)
\end{aligned} \tag{4.80}$$

where $\tilde{w}_{0:k}^i$ are the *normalized importance weights* and they are given by

$$\tilde{w}_{0:k}^i = \frac{w_{0:k}^i}{\sum_{j=1}^N w_{0:k}^j} \tag{4.81}$$

Accordingly, by using the samples obtained from the above proposal distribution, the probability density function $p(\mathbf{x}_{0:k} | \mathbf{z}_{0:k})$ can be approximated as

$$p(\mathbf{x}_{0:k} | \mathbf{z}_{0:k}) \approx \tilde{p}(\mathbf{x}_{0:k} | \mathbf{z}_{0:k}) = \sum_{i=1}^N \tilde{w}_{0:k}^i \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i) \quad (4.82)$$

Note in Equation 4.82, $\{\mathbf{x}_{0:k}^i\}_{i=1}^N$ are samples drawn from the proposal distribution $\pi(\mathbf{x}_{0:k} | \mathbf{z}_{0:k})$ with their normalised importance weights $\{\tilde{w}_{0:k}^i\}_{i=1}^N$. The sufficient conditions for Equation 4.82 to hold can be found in [38], [42] and will not be discussed here.

4.6.2 Sequential Importance Sampling

To sequentially compute the probability density function of target state throughout all time steps in a tracking task, normally the following recursive form of the proposal distribution is adopted [38]:

$$\pi(\mathbf{x}_{0:k} | \mathbf{z}_{0:k}) = \pi(\mathbf{x}_{0:k-1} | \mathbf{z}_{0:k-1}) \pi(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{0:k}) \quad (4.83)$$

In the sequel, we refer to $\pi(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{0:k})$ as the *proposal distribution*. Under the Markovian assumptions that the current state is independent of all the previous measurements given the previous state; the current measurements are independent of all the previous measurements given the current state, $p(\mathbf{x}_{0:k})$ and $p(\mathbf{z}_{0:k} | \mathbf{x}_{0:k})$ in Equation 4.78 can be factorized over time steps as follows

$$p(\mathbf{x}_{0:k}) = p(\mathbf{x}_0) \prod_{j=1}^k p(\mathbf{x}_j | \mathbf{x}_{j-1}) \quad (4.84)$$

$$p(\mathbf{z}_{0:k} | \mathbf{x}_{0:k}) = \prod_{j=0}^k p(\mathbf{z}_j | \mathbf{x}_j) \quad (4.85)$$

Substituting Equations 4.83~4.85 into Equation 4.78, we can recursively estimate the importance weights $w_{0:k}$ as

$$w_{0:k} = w_{0:k-1} \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1})}{\pi(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{0:k})} \quad (4.86)$$

Equation 4.86 shows that for an appropriately chosen proposal distribution $\pi(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{0:k})$, the importance weights of particles can be sequentially updated. Moreover, in Equation 4.86, the measurement likelihood $p(\mathbf{z}_k | \mathbf{x}_k)$ and the state transition probability density function $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ can be calculated directly from the state space model (Equations 4.1 and 4.2). Therefore, to estimate the target state, one only needs to generate a prior set of particles and then recursively compute the importance weights of these particles. Normally, the initial particles are drawn from the initial distribution of target state $p(\mathbf{x}_0)$ and equally weighted as

$$\tilde{w}_0^i = \frac{1}{N} \quad i = 1, 2, \dots, N \quad (4.87)$$

The above process is known as *sequential importance sampling (SIS)* in the literature. However, as pointed out at the beginning of this chapter, the interests of target tracking applications is on the acquiring the probability density function $p(\mathbf{x}_k | \mathbf{z}_{0:k})$ at the current time step k rather than the probability density function $p(\mathbf{x}_{0:k} | \mathbf{z}_{0:k})$ that is over the whole period up to time step k . Hence, it is not necessary to keep the entire history of the particles trajectories; $\mathbf{x}_{0:k}^i$ and $g(\mathbf{x}_{0:k})$ in the above equations can be replaced by \mathbf{x}_k^i and $g(\mathbf{x}_k)$. Hence, Equations 4.80 and 4.82 can be rewritten as follows:

$$E[g(\mathbf{x}_k)] \approx \sum_{i=1}^N \tilde{w}_k^i g(\mathbf{x}_k^i) \quad (4.88)$$

$$p(\mathbf{x}_k | \mathbf{z}_{0:k}) \approx \tilde{p}(\mathbf{x}_k | \mathbf{z}_{0:k}) = \sum_{i=1}^N \tilde{w}_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \quad (4.89)$$

Choosing the proposal distribution $\pi(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{0:k})$ is one of the most critical design issues in PF. In [42], the authors proved that the proposal distribution in the following form

$$\pi(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{0:k}) = p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k) \quad (4.90)$$

minimizing the variance of the importance weights. $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k)$ is referred to as the *optimal proposal distribution*. However, the proposal distribution in the form of

$$\pi(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{0:k}) = p(\mathbf{x}_k | \mathbf{x}_{k-1}) \quad (4.91)$$

is the most popular choice of proposal distribution and has been applied in many state estimation problems including target tracking. In the literature, this state transition probability density function $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ is also named as the *transition prior*. As mentioned earlier in this section, the transition prior can be computed by using the system model (Equation 4.1). For example, if the process noise is Gaussian with zero mean and covariance \mathbf{R}_k , the transition prior is simply as

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}) = N(\mathbf{x}_k; \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{0}), \mathbf{R}_k) \quad (4.92)$$

Substituting Equation 4.91 into Equation 4.86, the importance weights of the particles become:

$$\begin{aligned} w_k &= w_{k-1} \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1})}{\pi(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{0:k})} \\ &= w_{k-1} p(\mathbf{z}_k | \mathbf{x}_k) \end{aligned} \quad (4.93)$$

It can be seen that, by adopting transition prior as the proposal distribution, the importance weights of particles are easily updated by using the measurement likelihood $p(\mathbf{z}_k | \mathbf{x}_k)$. The measurement likelihood can be computed through the measurement model (Refer to Equation 4.95).

Nevertheless, the transition prior is not conditioned on any measurements, especially the most recent measurement. When it is adopted in PF for the practical target tracking, the transition prior can possibly lead to the degeneracy problem of the entire PF algorithm. This is because the particles cannot be moved toward the region of high measurement likelihood without the information contained in the most recent measurements; especially, it is found that after a limited number of iterations, only a few particles retain significant importance weights while most particles have negligible weights. The above side-effects of adopting the transition prior as proposal distribution is more obvious in the cases where the

measurement likelihood has a very sharp peak and/or have very little “overlap” with the transition prior.

The straightforward approach to overcome the above degeneracy problem is to use a very large number of particles. However, this carries the cost of heavy computational requirements and hence unattractive for practical applications in highly resources constrained wireless sensor networks. Those feasible solutions to the degeneracy problem include a strategy for resampling and constructing a good proposal distribution [38], [39]. The resampling method will be presented in the next subsection and the construction of good proposal distribution will be discussed in the next section. Especially, we propose a novel extended Kalman filter and Particle filter hybrid algorithm, named as extended Kalman Particle filter (EKPF) tracking algorithm. EKPF constructs a better proposal distribution by making use of the EKF component to incorporate the most recent measurement to propagate the particles to high measurement likelihood region in the state-space.

4.6.3. Resampling of the Particles

To address the rapid degeneracy problem in the above sequential importance sampling (SIS) method, a resampling (i.e. selection) stage can be adopted to eliminate the particles with low importance weights and multiply the particles with high importance weights [39]. The resampling scheme involves generating a new set of particles $\{\mathbf{x}_k^{i*}, \tilde{w}_k^{i*}\}_{i=1}^N$ by resampling (with replacement) N times from the approximated discrete representation of the probability density function $p(\mathbf{x}_k | \mathbf{z}_{0:k}) \approx \sum_{i=1}^N \tilde{w}_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i)$ such that $p(\mathbf{x}_k^{i*} = \mathbf{x}_k^j) = \tilde{w}_k^j$.

Finally, the above resampling process will produce N new particles with equal weighting $\frac{1}{N}$.

4.6.4 Generic PF for Target Tracking in Wireless Sensor Networks

As mentioned earlier in this chapter, it is assumed that a fixed set of N_s sensing nodes are activated to sense and provide cluster leader with their measurements. Upon receiving these measurements, the cluster leader runs the PF tracking algorithm which is on the basis of the sequential importance sampling and resampling (SIR) to update the target state

estimate. This approach fuses the measurements from different sensing nodes into a single multiple sensing nodes measurement likelihood (recall the assumption has already been made in this chapter that the measurement obtained from a particular sensing node is independent of the measurements obtained from other sensing nodes). This leads to the factorized measurement likelihood over all N_s sensing nodes

$$p(\mathbf{z}_k | \mathbf{x}_k^i) = \prod_n^{N_s} p(\mathbf{z}_k^n | \mathbf{x}_k^i) \quad (4.94)$$

where

$$p(\mathbf{z}_k^n | \mathbf{x}_k^i) = \frac{1}{\sqrt{|2\pi\mathbf{R}_k^n|}} \exp \left[-\frac{1}{2} (\mathbf{z}_k^n - \hat{\mathbf{H}}_k^n \mathbf{x}_k^i)^T (\mathbf{R}_k^n)^{-1} (\mathbf{z}_k^n - \hat{\mathbf{H}}_k^n \mathbf{x}_k^i) \right] \quad (4.95)$$

is the probability density function of the measurement likelihood regarding the measurement obtained by the n -th sensing node at the k -th time step. In Equations 4.94 and 4.95, $\mathbf{z}_k = (\mathbf{z}_k^1, \dots, \mathbf{z}_k^n, \dots, \mathbf{z}_k^{N_s})$ is the concatenated measurement over all N_s sensing nodes that involve in the tracking task at the k -th time step, \mathbf{z}_k^n is the measurement acquired by the n -th sensing node, \mathbf{R}_k^n is the covariance matrix of the measurement noise at the n -th sensing node, and \mathbf{H}_k^n is the Jacobian matrix for the measurement model (refer to Section 4.4). Finally, for the PF adopting the transition prior $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ as the proposal distribution, the un-normalized importance weights of particles are computed as

$$\begin{aligned} w_k^i &= w_{k-1}^i p(\mathbf{z}_k | \mathbf{x}_k^i) \\ &= w_{k-1}^i \left[\prod_n^{N_s} p(\mathbf{z}_k^n | \mathbf{x}_k^i) \right] \end{aligned} \quad (4.96)$$

The above generic Particle filter (PF) using the sequential importance sampling and resampling (SIR) techniques and adopting the transition prior as the proposal distribution is listed below.

**Algorithm 4.3 The Generic Particle Filter (PF) for Single Target Tracking
in Wireless Sensor Networks**

- Initialization: $k = 0$

For $i = 1, 2, \dots, N$ draw particle \mathbf{x}_k^i from the prior of the target state $p(\mathbf{x}_0)$.

- For $k = 1, 2, \dots$

1. Importance sampling step

- For $i = 1, 2, \dots, N$, sample $\mathbf{x}_k^i \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}^i)$.

- For $i = 1, 2, \dots, N$, evaluate the importance weights

$$w_k^i = w_{k-1}^i \left[\prod_n p(\mathbf{z}_k^n | \mathbf{x}_k^i) \right]$$

$$p(\mathbf{z}_k^n | \mathbf{x}_k^i) = \frac{1}{\sqrt{|2\pi\mathbf{R}_k^n|}} \exp \left[-\frac{1}{2} (\mathbf{z}_k^n - \hat{\mathbf{H}}_k^n \mathbf{x}_k^i)^T (\mathbf{R}_k^n)^{-1} (\mathbf{z}_k^n - \hat{\mathbf{H}}_k^n \mathbf{x}_k^i) \right]$$

- For $i = 1, 2, \dots, N$, normalize the importance weights: $\tilde{w}_k^i = w_k^i / \sum_{j=1}^N w_k^j$

2. Selection step (resampling)

- Multiply (suppress) samples \mathbf{x}_k^i with high (low) importance weights \tilde{w}_k^i to obtain N random samples \mathbf{x}_k^{i*} approximately distributed according to $p(\mathbf{x}_k | \mathbf{z}_{1:k})$.

- For $i = 1, 2, \dots, N$, set $w_k^i = \tilde{w}_k^i = N^{-1}$.

- Output: The output of the algorithm is a set of particles that can be used to approximate the probability density function of target state as follows $\hat{p}(\mathbf{x}_k | \mathbf{z}_{0:k}) = \frac{1}{N} \sum_{i=1}^N \delta(\mathbf{x}_k - \mathbf{x}_k^{i*})$.

4.7 PF and EKF Hybrid Algorithm (EKPF) for Tracking a Single Target in Wireless Sensor Networks

On the basis of the SIR techniques and adopting transition prior as the proposal distribution, the generic PF algorithm developed in the previous section is conceptually simple, and easily implemented with moderate computation cost for tracking a single target in wireless sensor networks. However, in this generic PF, the propagation of the particles are solely decided by the target dynamic (i.e. system model); hence, the positions of particles at current time step are decided by the positions of the particles at the previous time step without taking into account the information contained in the most recent measurements.

Consequently, the generic PF might not properly relocate the particles in the state-space. This may lead to the algorithm ignoring some important regions of the state-space when searching for a potential target.

There are several strategies have been proposed in the literature to properly relocate particles in the state-space by designing a better proposal distribution [35], [121]–[123]. Motivated by these strategies, in this section we develop a novel hybrid extended Kalman Particle filter (EKPF) tracking algorithm which provides the resulting algorithm with the important property of better proposal distribution.

The key idea behind the EKPF is to choose a proposal distribution that is conditioned on the most recent measurement \mathbf{z}_k for the target state estimate at the k -th time step. Such proposal distribution not only helps to effectively propagate particles to the high measurement likelihood area, but also allows for easily sampling and computing the importance weights of the particles. As pointed out in Section 4.6, one such proposal distribution is the optimal proposal distribution $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k)$. However, it is normally not easy to directly generate particles from this optimal proposal distribution. Consequently, it is often needed to resort to Gaussian distribution to approximate this optimal proposal distribution and then draw samples from the approximation, i.e.

$$\pi(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{0:k}) = p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k) \approx q_N(\mathbf{x}_k | \mathbf{z}_{0:k}) \quad (4.97)$$

where q_N denotes a Gaussian distribution. Generally, this approximated proposal distribution $q_N(\mathbf{x}_k | \mathbf{z}_{0:k})$ is a better choice than the transition prior $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ since it takes into account the information contained in the most recent measurement.

A tractable way of generating above Gaussian approximated proposal distribution is to use a separate EKF to generate a Gaussian proposal distribution for each of the N particles:

$$q_N(\mathbf{x}_k^i | \mathbf{z}_{0:k}) = N(\mathbf{x}_k^i; \mathbf{m}_{k|k}^i, \mathbf{P}_{k|k}^i) \quad i = 1, \dots, N \quad (4.98)$$

At the k -th time step, EKPF algorithm uses the EKF equations, with the most recent measurement, to compute the mean and covariance of the proposal distribution for each of the N particles which are propagated from the $(k-1)$ -th time step. Subsequently, the N particles for the k -th time step can be obtained. The process of EKPF is described as follows.

It is assumed that the target state estimation $p(\mathbf{x}_{k-1} | \mathbf{z}_{0:k-1})$ at the $(k-1)$ -th time step is already known and represented by N equally weighted particles $\{\mathbf{x}_k^i\}_{i=1}^N$ (recall the resampling step of generic PF algorithm in Section 4.6). The assumption is also made that the mean $\mathbf{m}_{k-1|k-1}^i$, $i=1, \dots, N$ and covariance matrix $\mathbf{P}_{k-1|k-1}^i$, $i=1, \dots, N$ for each of these N particles are already computed at the $(k-1)$ -th time step. Then the mean $\mathbf{m}_{k|k}^i$, $i=1, \dots, N$ and covariance matrix $\mathbf{P}_{k|k}^i$, $i=1, \dots, N$ of the N particles at the k -th time step can be obtained through the standard EKF steps:

$$\mathbf{m}_{k|k-1}^i = \mathbf{f}_k \mathbf{m}_{k-1|k-1}^i \quad (4.99)$$

$$\mathbf{P}_{k|k-1}^i = \mathbf{Q}_{k-1} + \hat{\mathbf{A}}_k^i \mathbf{P}_{k-1|k-1}^i (\hat{\mathbf{A}}_k^i)^T \quad (4.100)$$

$$\mathbf{m}_{k|k}^i = \mathbf{m}_{k|k-1}^i + \mathbf{K}_k^i (\mathbf{z}_k - \mathbf{h}_k \mathbf{m}_{k|k-1}^i) \quad (4.101)$$

$$\mathbf{P}_{k|k}^i = \mathbf{P}_{k|k-1}^i - \mathbf{K}_k^i \hat{\mathbf{H}}_k^i \mathbf{P}_{k|k-1}^i \quad (4.102)$$

where

$$\hat{\mathbf{A}}_k^i = \left. \frac{d\mathbf{f}_k(\mathbf{x}^i)}{d\mathbf{x}^i} \right|_{\mathbf{x}^i = \mathbf{m}_{k-1|k-1}^i} \quad (4.103)$$

$$\hat{\mathbf{H}}_k^i = \left. \frac{d\mathbf{h}_k(\mathbf{x}^i)}{d\mathbf{x}^i} \right|_{\mathbf{x}^i = \mathbf{m}_{k|k-1}^i} \quad (4.104)$$

$$\mathbf{K}_k^i = \mathbf{P}_{k|k-1}^i (\hat{\mathbf{H}}_k^i)^T (\mathbf{S}_k^i)^{-1} \quad (4.105)$$

$$\mathbf{S}_k^i = \hat{\mathbf{H}}_k^i \mathbf{P}_{k|k-1}^i (\hat{\mathbf{H}}_k^i)^T + \mathbf{R}_k \quad (4.106)$$

Similar to the S-EKF, S-UKF and generic PF tracking algorithms developed in the previous sections, here we still assume that a fixed set of N_s sensing nodes are activated and provide the cluster leader with their measurements for target state estimate. Since a separate EKF is implemented for each of the N particles, there will be total $N \times N_s$ EKFs in EKPF for N_s measurements received from the N_s sensing nodes. However, the EKPF

algorithm developed in this section only makes use of the measurement from one *most informative sensing node* to propagate particles, i.e. only implements one EKF instead of the S-EKF (consisting of N_s EKFs) for each of the N particles. Normally, the criteria for selecting the most informative sensing node includes information utility based and geometry based measures [76]. The EKPF algorithm developed in this chapter uses the geometry based measure which selects the sensing node that lies closest to the target as the most informative sensing node. Since the sensing node adopted in this chapter is equipped with acoustic modality, the measurement acquired at an individual sensing node is the inverse of the squared distance between the target and this sensing node. Thus, at each time step, upon receiving the measurements from the N_s sensing nodes, the cluster leader selects the sensing node with largest measurement as the most informative sensing node and uses its measurement to implement a separate EKF for each of the N particles. Also as in the generic PF, the importance weights of the particles in EKPF are updated by the factorized measurement likelihood over all N_s measurements from N_s sensing nodes, i.e.

$$w_k^i = w_{k-1}^i \left[\prod_n^{N_s} p(\mathbf{z}_k^n | \mathbf{x}_k^i) \right].$$

The complete EKPF algorithm is listed in the Algorithm 4.4.

The EKPF algorithm is a hybrid of PF and EKF. The PF part provides the general probabilistic framework to handle nonlinear systems while the EKF part generates a better proposal distribution by taking into account the most recent measurements. Compared to the generic PF, EKPF introduces more computation burden. However, EKPF makes use of the most recent measurement and consequently, sampling at current time step is more efficient and the number of particles needed in the algorithm might be reduced considerably. This will be demonstrated through simulations in the next section. Simulation results shows that the EKPF outperforms the generic PF, S-EKF and S-UKF in the context of tracking accuracy and robustness when these algorithms are applied to track a single target in wireless sensor networks.

**Algorithm 4.4 The Extended Kalman Particle Filter (EKPF) for Single Target Tracking
in Wireless Sensor Networks**

- Initialization: $k = 0$

For $i = 1, 2, \dots, N$ draw particle $\mathbf{x}_k^{(i)}$ from the initial distribution $p(\mathbf{x}_0)$.

- For $k = 1, 2, \dots$

1. Particles propagation: for particles $i = 1, 2, \dots, N$

1.1 Prediction step:

Compute the process model Jacobians:

$$\hat{\mathbf{A}}_k^i = \left. \frac{d\mathbf{f}_k(\mathbf{x}^i)}{d\mathbf{x}^i} \right|_{\mathbf{x}^i = \mathbf{m}_{k-1|k-1}^i}$$

Compute the predicted state mean and covariance

$$\mathbf{m}_{k|k-1}^i = \mathbf{f}_k \mathbf{m}_{k-1|k-1}^i$$

$$\mathbf{P}_{k|k-1}^i = \mathbf{Q}_{k-1} + \hat{\mathbf{A}}_k^i \mathbf{P}_{k-1|k-1}^i (\hat{\mathbf{A}}_k^i)^T$$

1.2 Update step:

Compute the measurement model Jacobians:

$$\hat{\mathbf{H}}_k^i = \left. \frac{d\mathbf{h}_k(\mathbf{x}^i)}{d\mathbf{x}^i} \right|_{\mathbf{x}^i = \mathbf{m}_{k|k-1}^i}$$

Update estimate with the most recent measurement

$$\mathbf{S}_k^i = \hat{\mathbf{H}}_k^i \mathbf{P}_{k|k-1}^i (\hat{\mathbf{H}}_k^i)^T + \mathbf{R}_k$$

$$\mathbf{K}_k^i = \mathbf{P}_{k|k-1}^i (\hat{\mathbf{H}}_k^i)^T (\mathbf{S}_k^i)^{-1}$$

$$\mathbf{m}_{k|k}^i = \mathbf{m}_{k|k-1}^i + \mathbf{K}_k^i (\mathbf{z}_k - \mathbf{h}_k \mathbf{m}_{k|k-1}^i)$$

$$\mathbf{P}_{k|k}^i = \mathbf{P}_{k|k-1}^i - \mathbf{K}_k^i \hat{\mathbf{H}}_k^i \mathbf{P}_{k|k-1}^i$$

2. Particles' weight update: for $i = 1, 2, \dots, N$

$$w_k^i = w_{k-1}^i p(\mathbf{z}_k | \mathbf{x}_k^i)$$

$$= w_{k-1}^i \left[\prod_n^{N_s} p(\mathbf{z}_k^n | \mathbf{x}_k^i) \right]$$

$$p(\mathbf{z}_k^n | \mathbf{x}_k^i) = \frac{1}{\sqrt{|2\pi\mathbf{R}_k^n|}} \exp \left[-\frac{1}{2} (\mathbf{z}_k^n - \hat{\mathbf{H}}_k^n \mathbf{x}_k^i)^T (\mathbf{R}_k^n)^{-1} (\mathbf{z}_k^n - \hat{\mathbf{H}}_k^n \mathbf{x}_k^i) \right]$$

$$\tilde{w}_k^i = w_k^i / \sum_{j=1}^N w_k^j \quad (\text{Normalisation})$$

3. Resampling

Multiply (suppress) particles \mathbf{x}_k^i with high (low) importance weights \tilde{w}_k^i to

obtain N random samples \mathbf{x}_k^{i*} approximately distributed according to $p(\mathbf{x}_k | \mathbf{z}_{0:k})$.

- Output: a set of particles that can be used to approximate the probability density function

of the target state, i.e. $\hat{p}(\mathbf{x}_k | \mathbf{z}_{0:k}) = \frac{1}{N} \sum_{i=1}^N \delta(\mathbf{x}_k - \mathbf{x}_k^{i*})$.

Finally, it should be noted that the UKF can also be used to construct the proposal distribution, i.e. integrating UKF into PF to propagate particles to the areas of high measurement likelihood. However, since this chapter adopts the nearly constant velocity (CV) model which is a linear system dynamic model, UKF is not expected to exhibit consistent superiority over EKF (refer to [128] and also evidenced by the simulation results in the next section). Moreover, since the computation cost of UKF is normally heavier than that of EKF, the resulting UKF and PF hybrid algorithm will demand considerably higher computation resources and may not be applicable in the resource constrained wireless sensor networks. Therefore, the UKF and PF hybrid algorithm is not considered in this chapter.

4.8 Simulations

Extensive simulations have been conducted to evaluate the tracking algorithms developed in this chapter. This section presents the simulation results of S-EKF, S-UKF, PF and EKPF and compares the performance of these algorithms by computing the root mean square error (RMSE) values. In the next section, RMSE values of S-EKF, S-UKF, PF and EKPF will also be compared with the posterior Cramer-Rao lower bound (PCRLB) which is a lower bound on the tracking accuracy to which the tracking algorithms can attain. In the following simulations, all algorithms are implemented in Matlab and run on a Pentium 4, 2.8 GHz computer.

4.8.1 Simulation Setup

Figure 4.2 shows a typical simulation setup. The sensor field is two dimensional with the size of $300\text{ m} \times 250\text{ m}$. A ground vehicle traverses this sensor field from north-east to

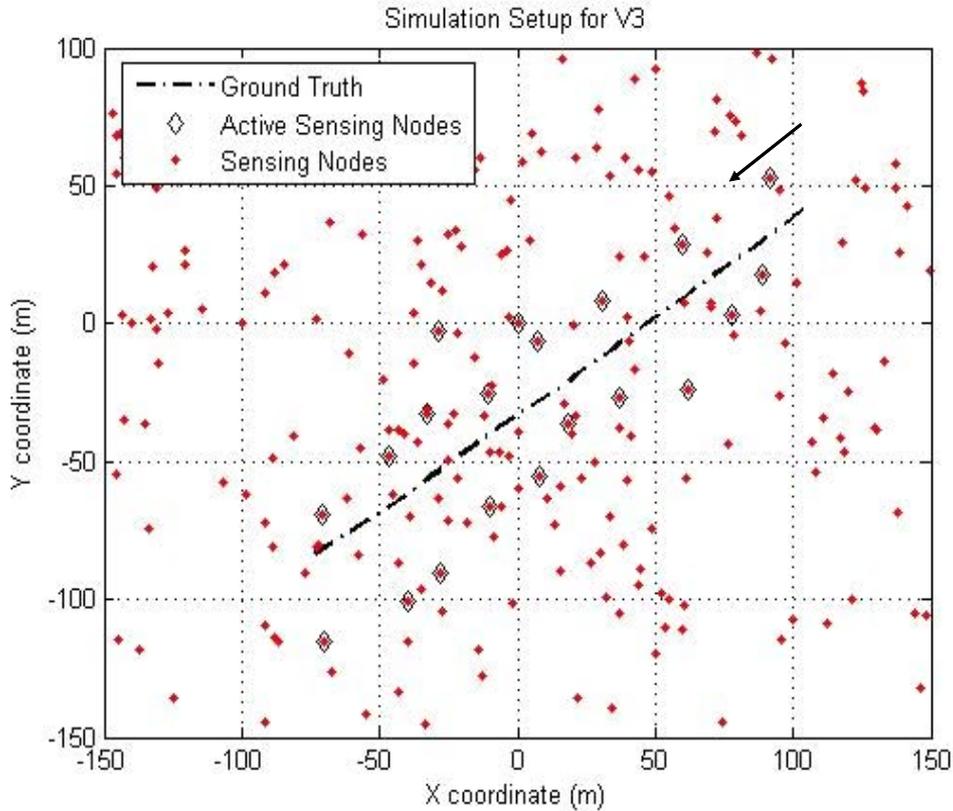


Figure 4.2 The typical simulation setup of tracking a single target in a wireless sensor network

south-west, its trajectory (also called “ground truth”) is depicted in the figure. There is a set of 200 sensing nodes randomly deployed in this sensor field. In the simulations throughout this chapter, it is assumed that the tracking task is performed within one sensor cluster which consists of one leader node (e.g., the cluster leader, was not drawn in Figure 4.2) and 20 sensing nodes (named as active sensing nodes) that are selected from the 200 deployed sensing nodes. At every time step during the whole period of the tracking task, these 20 active sensing nodes are activated to sense and provide the cluster leader with their measurements; and upon receiving these measurements, the cluster leader executes one of the four algorithms developed in this chapter to update the target state estimate. However, it should be noted that in practical target tracking applications in wireless sensor networks, the active sensing nodes are not fixed. Instead, they are selected by some criteria and their number may also vary from time step to time step. In this thesis, we use the geometry based criterion for the selection of active sensing nodes: at each time step, the active sensing nodes at the current time step are selected based on their distances to the predicted position of the target; the sensing nodes that lie closest to the predicted target position becomes the active sensing nodes (refer to Chapter 6 for the details). This active sensing

nodes selection scheme explains why some of the active sensing nodes are favourably placed in Figure 4.2. However, in order to assess the robustness of the developed tracking algorithms, not all active sensing nodes are deployed very close to the target position.

Figure 4.3 depicts six different tracking scenarios that used in the simulations. These tracking scenarios are set up with different target trajectories, different target dynamics and different sets of active sensing nodes. The target trajectories in Figure 4.3 are digested from a real on-site experiment (details can be found in [23]). In Figure 4.3, the label V12, V4, V10, V3, V6 and V1 correspond to the vehicles DW12, DW4, DW10, DW3, DW6 and DW1 in [23]; however, these labels are referred to the above six different tracking scenarios throughout this thesis.

In the simulations, the measurement at individual sensing node is synthesized according to the measurement model as defined in Equation 4.6, i.e. at the k -th time step, the magnitude of the measurement acquired by the n -th sensing node is synthesized as follows.

$$\| \mathbf{z}_k^n \| = \frac{S}{(x_k - x_n)^2 + (y_k - y_n)^2} + \varepsilon_k^n \quad n = 1, 2, \dots, N_s \quad (4.107)$$

where (x_k, y_k) and (x_n, y_n) are the positions of the target and the n -th sensing node, respectively. S is the source energy, defined to be the acoustic intensity measured at 1 m away from the target. In the simulations, S is set to 5000 and the background noise is set as $\varepsilon_k^n \sim N(0, 1)$, $n = 1, 2, \dots, N_s$ for all active sensing nodes. It needs to be pointed out that although the signal to noise ratio (SNR) is 37 dB at the target position, the actual SNR at individual sensing node depends linearly on the distance between this sensing node and the target. For example, for a sensing node that is 50 meters away from the target, its SNR is merely 10 dB. In the following simulations, it is also assumed that there is no clutter and missed detection.

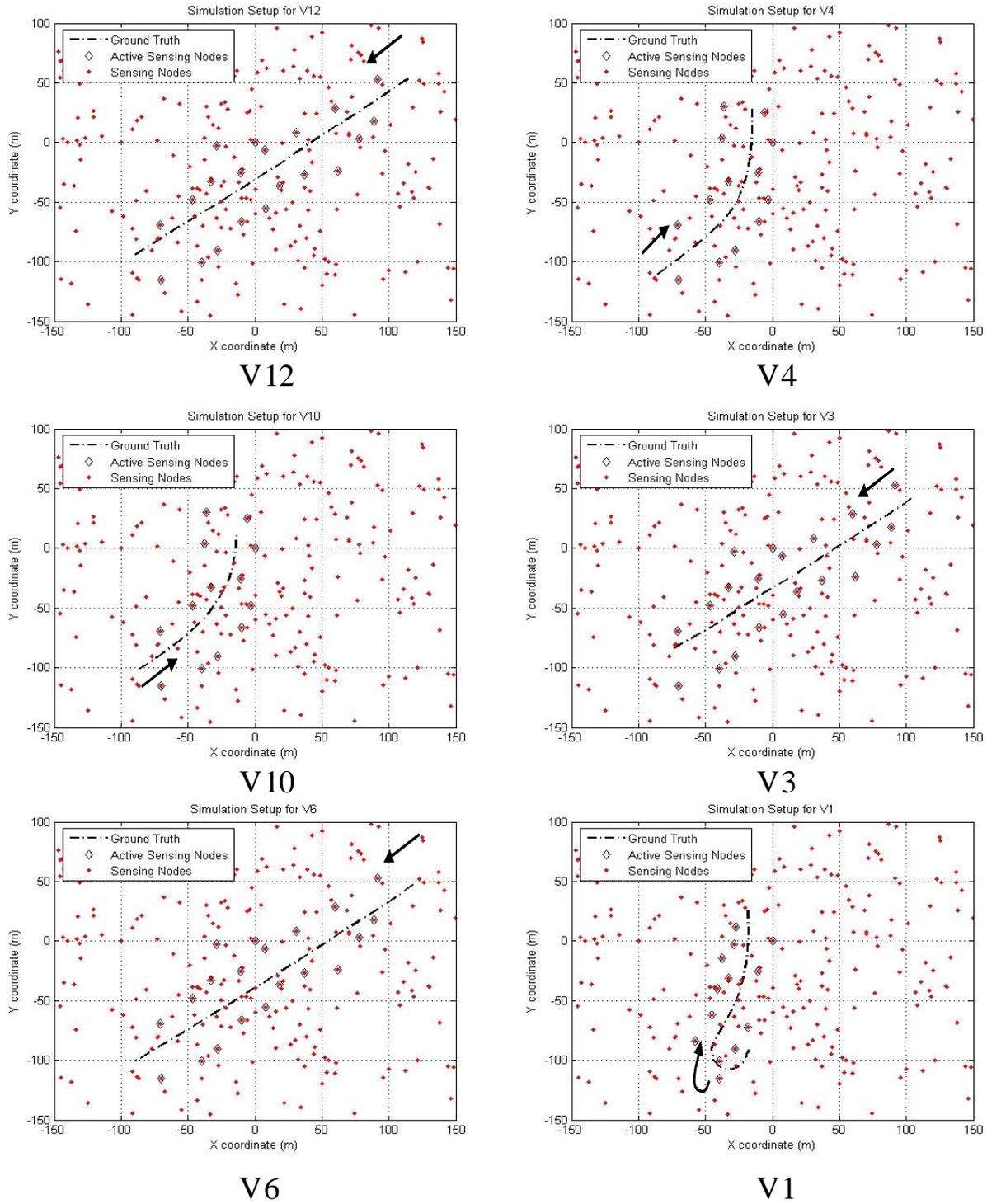


Figure 4.3 Six tracking scenarios with different target trajectories, target dynamics and active sensing nodes (V12, V4, V10, V3, V6 and V1 denote six different tracking scenarios throughout this thesis)

In the simulations, four target tracking algorithms, S-EKF, S-UKF, PF and EKPF are tested on each of the six tracking scenarios as depicted in Figure 4.3. For each algorithm, the prior estimate of the target state is assumed Gaussian with the mean $\mathbf{x}_{0|0}$ and

covariance matrix $\mathbf{P}_{0|0}$. To describe the different level of the uncertainty in the prior estimate of target state, the mean $\mathbf{x}_{0|0}$ and covariance $\mathbf{P}_{0|0}$ are categorized into four groups:

$$\mathbf{x}_{0|0}^1 = \mathbf{x}_{truth} + \begin{pmatrix} 10 \\ 0 \\ 10 \\ 0 \end{pmatrix}, \quad \mathbf{P}_{0|0}^1 = \begin{pmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{pmatrix} \quad (4.108)$$

$$\mathbf{x}_{0|0}^2 = \mathbf{x}_{truth} + \begin{pmatrix} 5 \\ 0 \\ 5 \\ 0 \end{pmatrix}, \quad \mathbf{P}_{0|0}^2 = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix} \quad (4.109)$$

$$\mathbf{x}_{0|0}^3 = \mathbf{x}_{truth} + \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{P}_{0|0}^3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.110)$$

$$\mathbf{x}_{0|0}^4 = \mathbf{x}_{truth} + \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{P}_{0|0}^4 = 0 \quad (4.111)$$

where \mathbf{x}_{truth} is the initial position of the target (i.e. the ground truth at time step $k = 0$).

In the simulations, 200 independent Monte Carlo runs have been conducted for each of the four tracking algorithms and the root mean square error (RMSE) is used to compare the performance of these algorithms. Two different types of RMSE are computed in this chapter: one is averaged over all time steps for each individual Monte Carlo run, and another is averaged over all 200 Monte Carlo runs for each time step. These two types of RMSE are defined as follows.

$$RMSE^n = \sqrt{\frac{1}{K} \sum_{k=1}^K \|\mathbf{r}_k^n - \hat{\mathbf{r}}_k^n\|^2} \quad (4.112)$$

$$RMSE_k = \sqrt{\frac{1}{\Psi} \sum_{n=1}^{\Psi} \|\mathbf{r}_k^n - \hat{\mathbf{r}}_k^n\|^2} \quad (4.113)$$

In Equation 4.112, $RMSE^n$ is referred to as the RMSE value of the n -th Monte Carlo run that is averaged over all time steps of the target tracking task. K is the total number of time steps in the tracking task. $\mathbf{r}_k^n = [x_k^n \ y_k^n]^T$ and $\hat{\mathbf{r}}_k^n = [\hat{x}_k^n \ \hat{y}_k^n]^T$ correspond to the true target position and the estimated target position at the k -th time step during the n -th Monte Carlo run, respectively. In Equation 4.113, $RMSE_k$ is referred to as the RMSE value of the k -th time step that is averaged over all 200 Monte Carlo runs. Ψ is the total number of Monte Carlo runs (i.e. $\Psi = 200$ in the simulation). $\mathbf{r}_k^n = [x_k^n \ y_k^n]^T$ and $\hat{\mathbf{r}}_k^n = [\hat{x}_k^n \ \hat{y}_k^n]^T$ correspond to the true target position and the estimated target position in the n -th Monte Carlo run at the k -th time step, respectively. The above two RMSE parameters indicate how much the target position estimation obtained from the tracking algorithms deviates from the true target position and will be used to assess the performance of the algorithms developed in this chapter.

4.8.2 The Simulation Results of S-EKF and S-UKF Tracking Algorithms

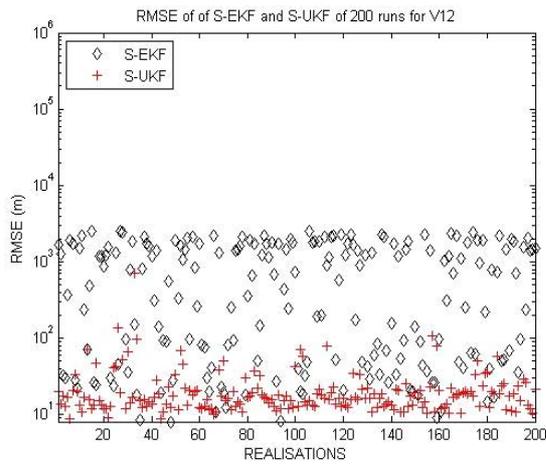
This section conducted simulations for both S-EKF and S-UKF tracking algorithms with varying settings including the mean and covariance of the prior estimate of target state and the signal to noise ratio (SNR). The simulation results are obtained by performing 200 Monte Carlo runs of S-EKF and S-UKF under these settings for each of the six tracking scenarios as depicted in Figure 4.3.

Figure 4.4 depicts $RMSE^n$ values of S-EKF and S-UKF of 200 Monte Carlo runs. Figure 4.5 depicts the $RMSE_k$ values of S-EKF and S-UKF of time steps averaged over these 200 Monte Carlo runs. Figure 4.6 also shows the $RMSE_k$ values of S-EKF and S-UKF of time steps; however, instead of being averaged over all 200 Monte Carlo runs, the $RMSE_k$ values in Figure 4.6 is computed by excluding the bottom 50 runs with the lowest $RMSE^n$ values and the top 50 runs with the largest $RMSE^n$ values (it is referred to as the processed data in the figure). The purpose of this data exclusion is to reduce the bias imposed by very large RMSE values in some Monte Carlo runs. In the simulations depicted in Figures 4.4, 4.5 and 4.6, for each Monte Carlo run of a tracking scenario, the true target trajectory and the set of active sensing nodes are kept unchanged, but the simulated

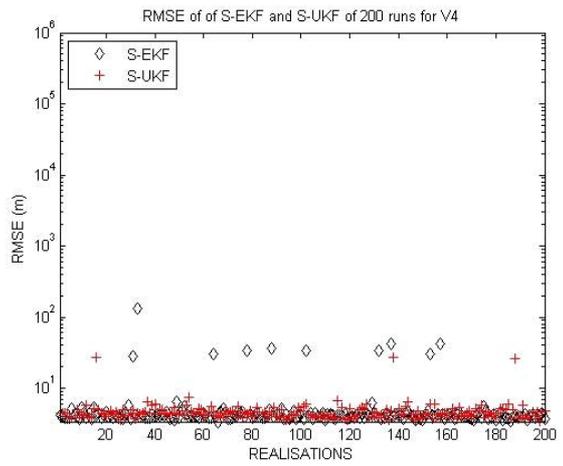
measurements at each sensing node are regenerated according to Equation 4.107 in which the target energy is set as $S=5000$ and the background noise is set as $\varepsilon_k^n \sim N(0,1)$ $n=1, 2, \dots, N_s$. In addition, both S-EKF and S-UKF adopt the same prior estimate of target state in which the mean and covariance are set as $\mathbf{x}_{0|0}^3$ and $\mathbf{P}_{0|0}^3$ (defined in Equation 4.110), respectively.

From Figures 4.4, 4.5 and 4.6, it can be seen that the performance of both S-EKF and S-UKF are not very robust and they even become divergent (an individual Monte Carlo run is considered to be divergent when the magnitude of its $RMSE^n$ value exceeds $50 m$). The worst result of S-EKF is the tracking scenario V12 in which there are 146 runs out of 200 runs are divergent. The poor performance of S-EKF is not unexpected because of the first order Taylor expansion of nonlinear measurement model in S-EKF. From Figures 4.4, 4.5 and 4.6, it can also be seen that the performance of S-UKF is not always better than that of S-EKF. In some tracking scenarios, S-UKF is even outperformed by S-EKF. For example, in tracking scenario V6, 86 runs of S-UKF algorithm are divergent while only 8 runs of S-EKF algorithm are divergent. In theory, the UKF (S-UKF) can improve the tracking accuracy over the EKF (S-EKF) [32]; however, in this simulation the improvement of S-UKF over S-EKF is not very significant. This is because the system model (Equation 4.3) chosen in this chapter is the nearly constant-velocity (CV) model which is linear [128]. Simulations also show that the computation cost of S-UKF is much higher than that of S-EKF. In a typical run of these two algorithms, S-UKF needs 0.0245 second for one time step target state estimation while the S-EKF tracking algorithm only needs 0.0041 second⁴.

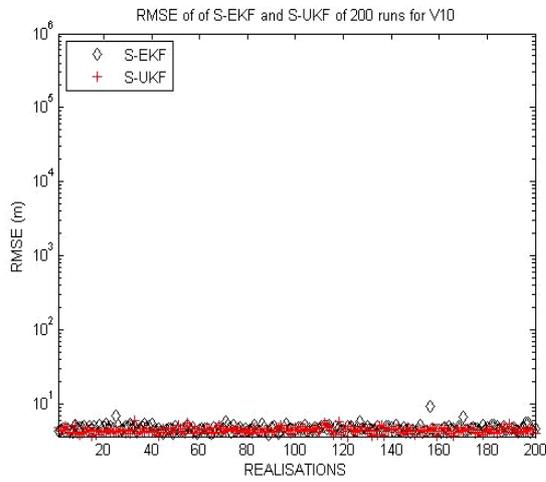
⁴ Both S-EKF and S-UKF are implemented in Matlab and run on a Pentium 4, 2.8 GHz laptop.



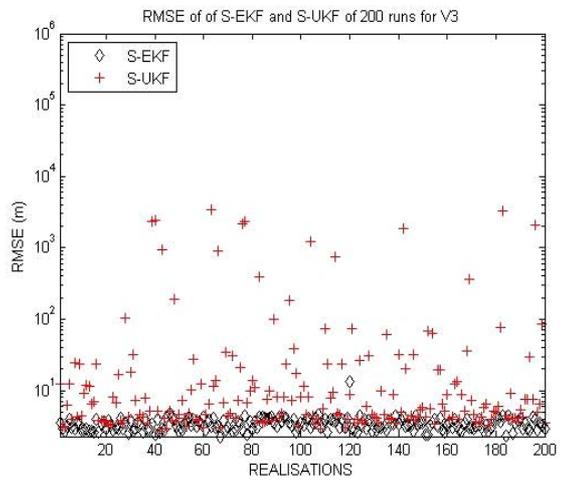
V12



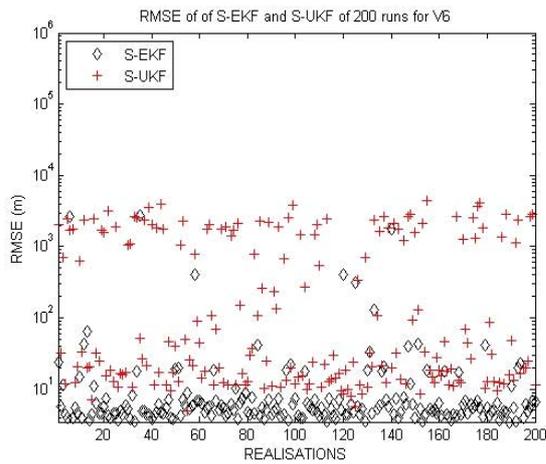
V4



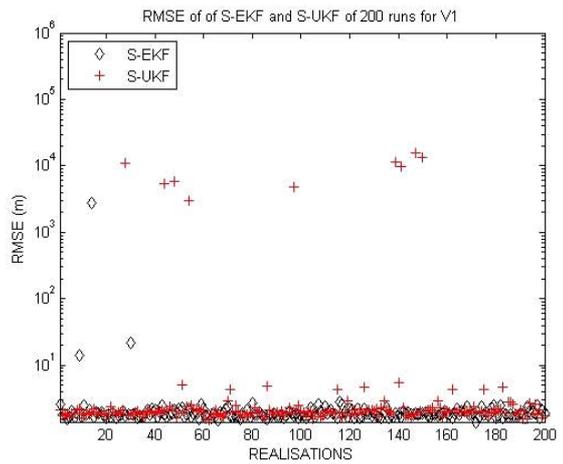
V10



V3

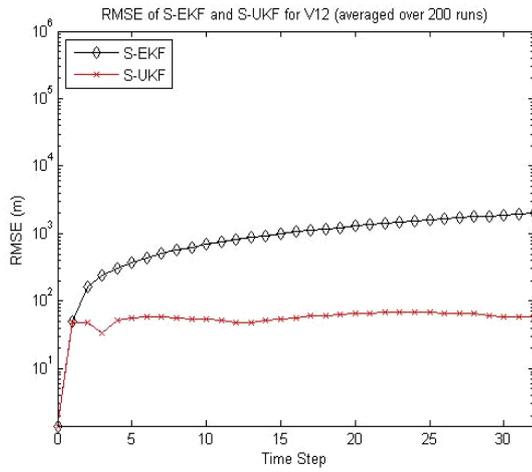


V6

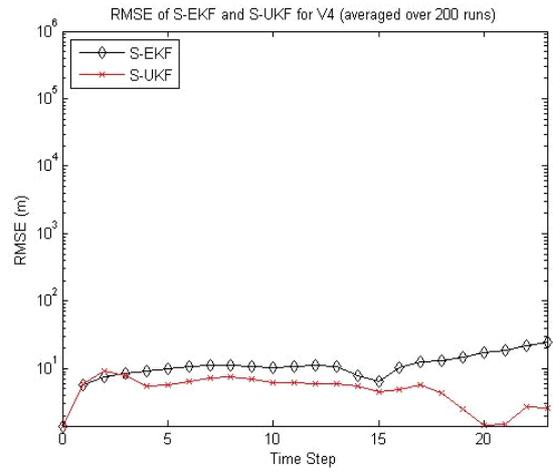


V1

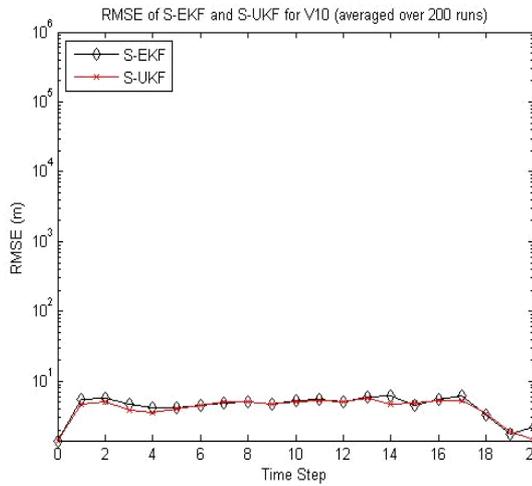
Figure 4.4 $RMSE^n$ values of S-EKF and S-UKF algorithms of 200 Monte Carlo runs for six tracking scenarios



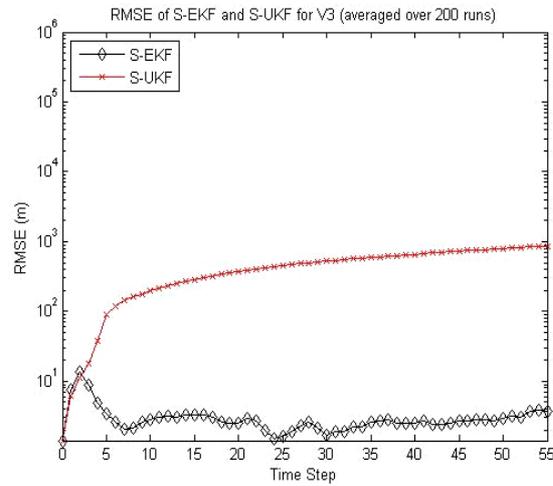
V12



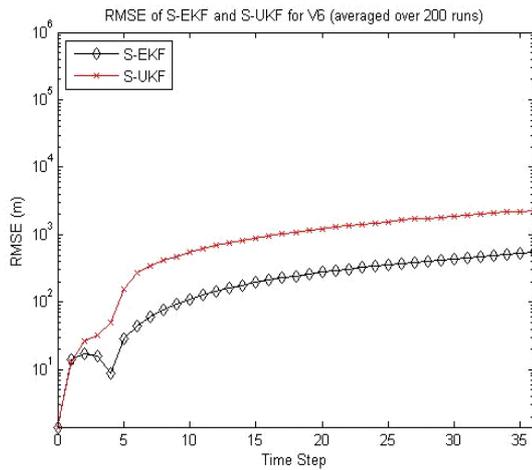
V4



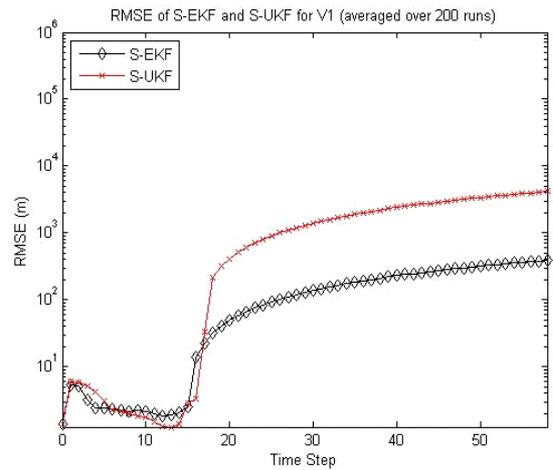
V10



V3

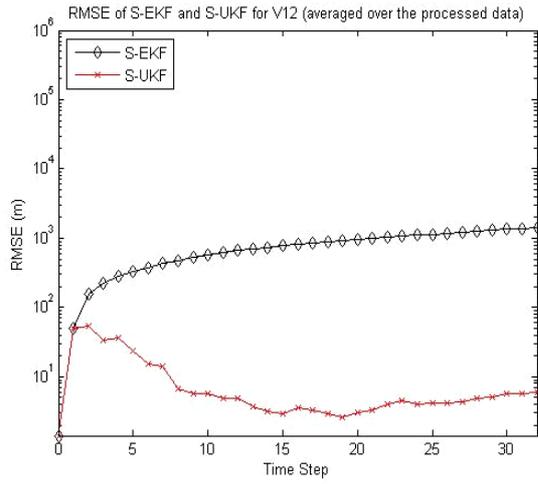


V6

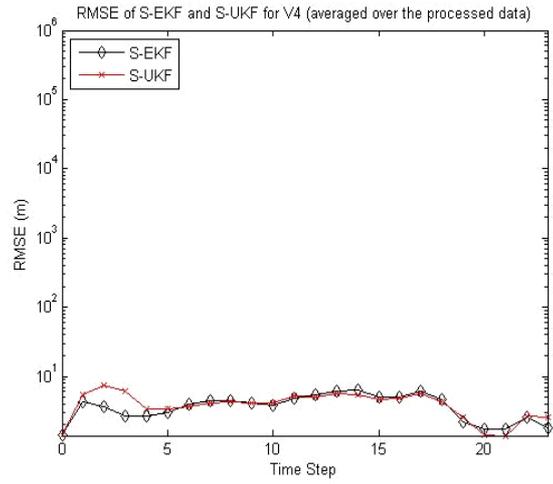


V1

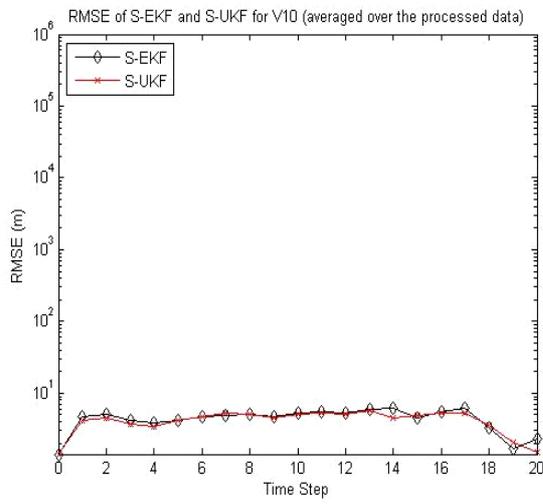
Figure 4.5 $RMSE_k$ values of S-EKF and S-UKF algorithms of each time step for six tracking scenarios (averaged over 200 runs)



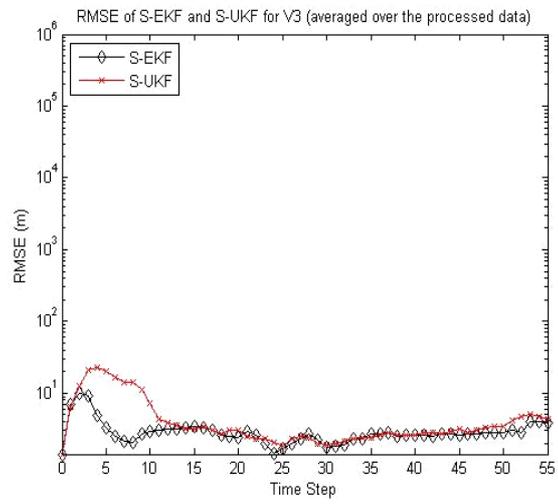
V12



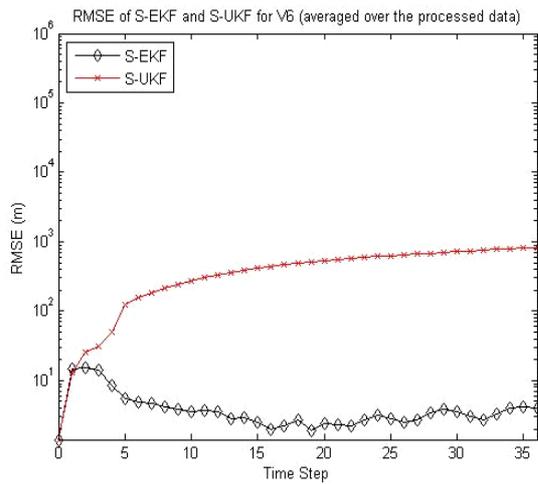
V4



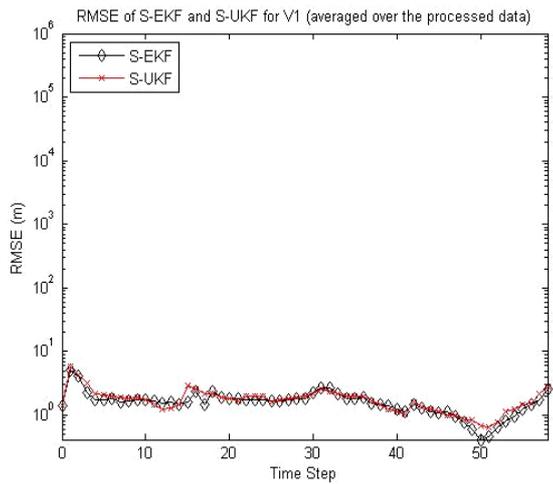
V10



V3



V6



V1

Figure 4.6 $RMSE_k$ values of S-EKF and S-UKF algorithms of each time step for six tracking scenarios (averaged over the processed data)

Figures 4.7 and 4.8 depict the performance of S-EKF and S-UKF under different SNR settings for the tracking scenario V12, respectively. The setting of the prior estimate of target state in both figures is $\mathbf{x}_{0|0}^4$ and $\mathbf{P}_{0|0}^4$ (Equation 4.111), i.e. the uncertainty in prior estimate of the target state is very low. From Figures 4.7 and 4.8, it can be seen that when the magnitude of SNR drops, the performance of S-EKF degrades faster than that of S-UKF. At the SNR setting of 23 dB, the mean of $RMSE_k$ (i.e. the RMSE is averaged over both time steps and 200 Monte Carlo runs) are 11.05 m for S-UKF and 30.66 m for S-EKF.

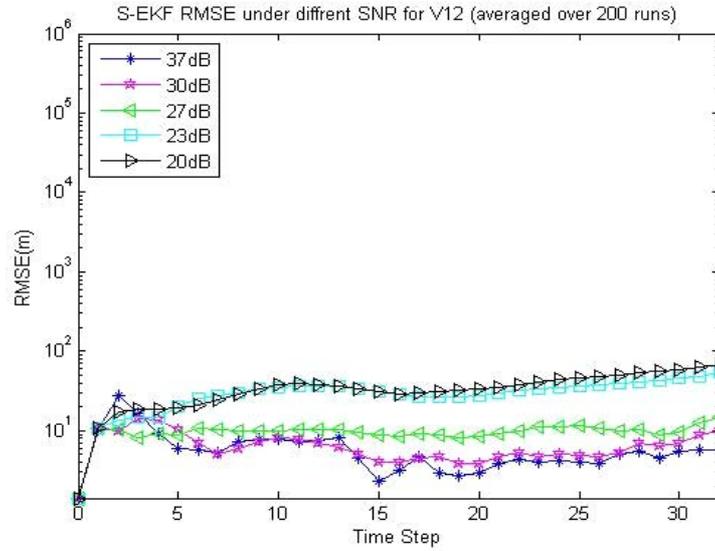


Figure 4.7 S-EKF $RMSE_k$ values under different SNR for tracking scenario V12 (prior estimate of the target state is $\mathbf{x}_{0|0}^4$ and $\mathbf{P}_{0|0}^4$)

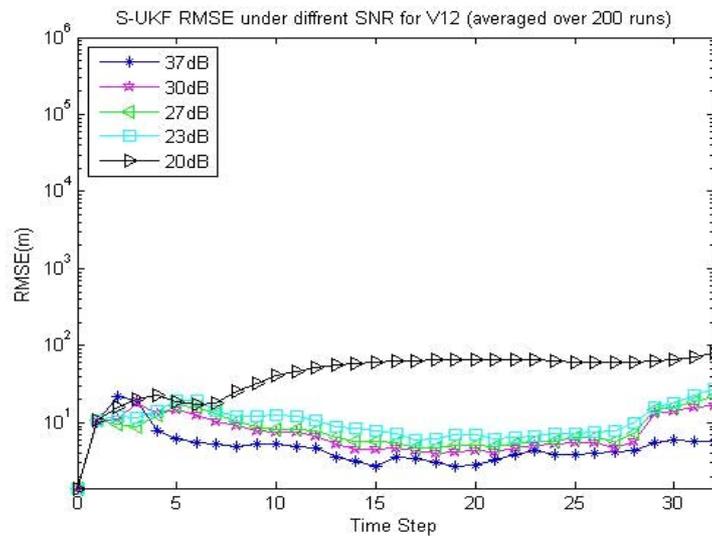


Figure 4.8 S-UKF $RMSE_k$ values under different SNR for tracking scenario V12 (prior estimate of the target state is $\mathbf{x}_{0|0}^4$ and $\mathbf{P}_{0|0}^4$)

Figure 4.9 shows the performance of S-EKF and S-UKF under different initialisation conditions – different prior estimate of the target state for tracking scenarios V10 and V12. Three sets of different initialisation conditions as defined in Equations 4.108~4.110, corresponding to high, medium and low level of uncertainty are adopted. It can be seen that generally S-UKF is more robust than S-EKF. In tracking scenario V10, when the prior uncertainty turns to high, S-UKF performs better than S-EKF. In tracking scenario V12, S-EKF outperforms S-UKF under low prior uncertainty; however, when the prior uncertainty turns to medium, the performance of S-EKF degrades much faster than that of S-UKF.

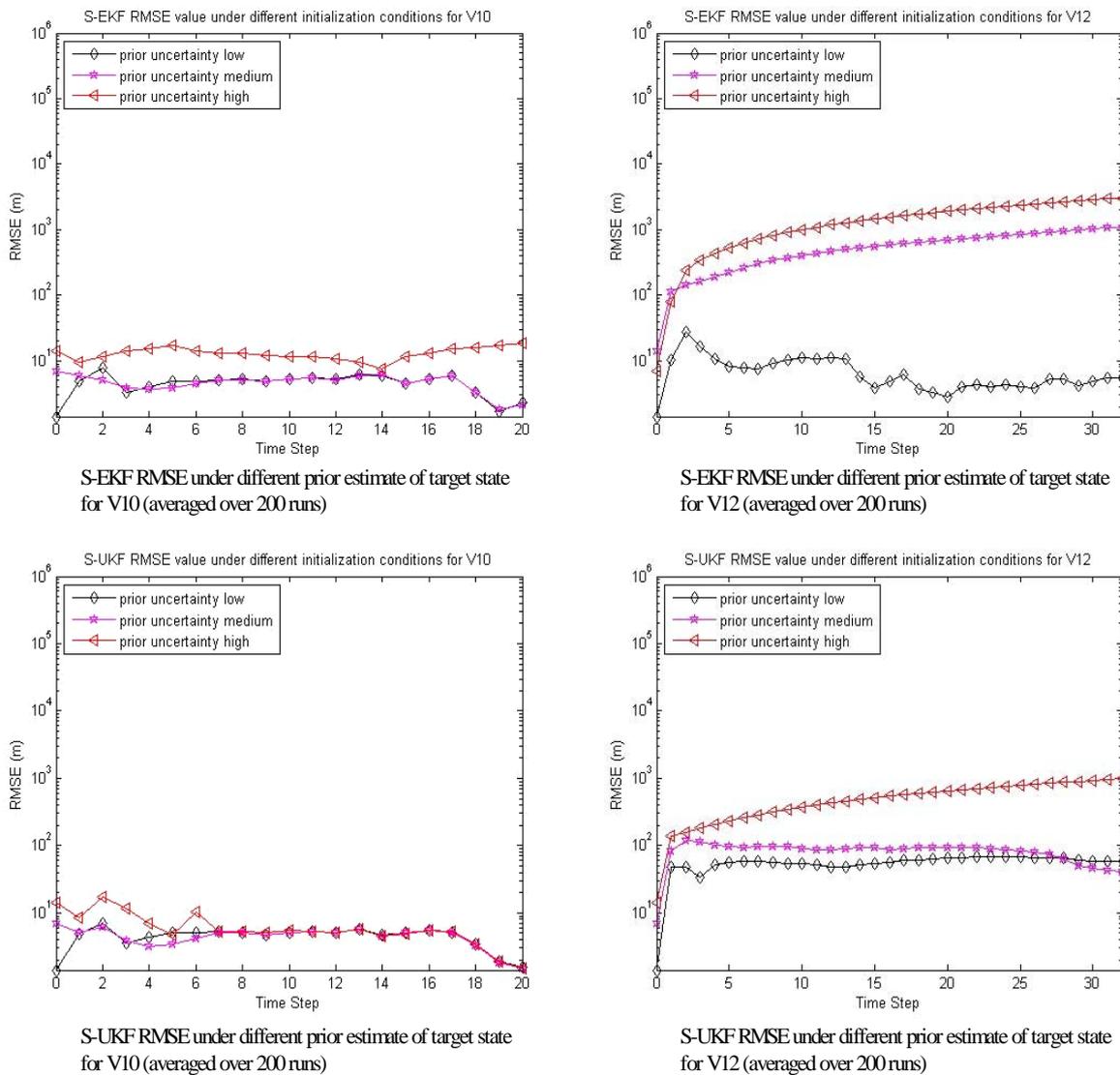


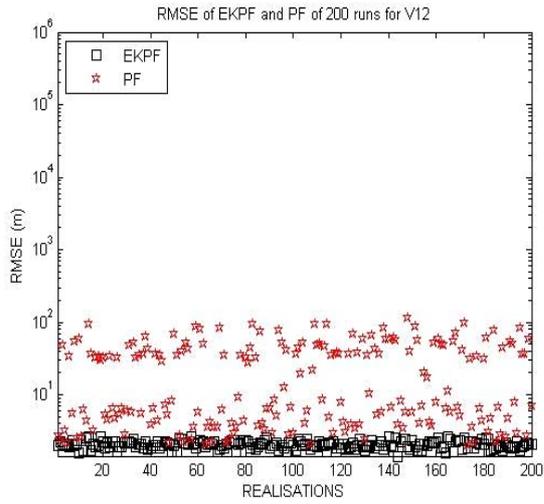
Figure 4.9 S-EKF and S-UKF algorithms $RMSE_k$ values under different prior estimate of target state for tracking scenarios V12 and V10

4.8.3 The Simulation Results of PF and EKPF Tracking Algorithms

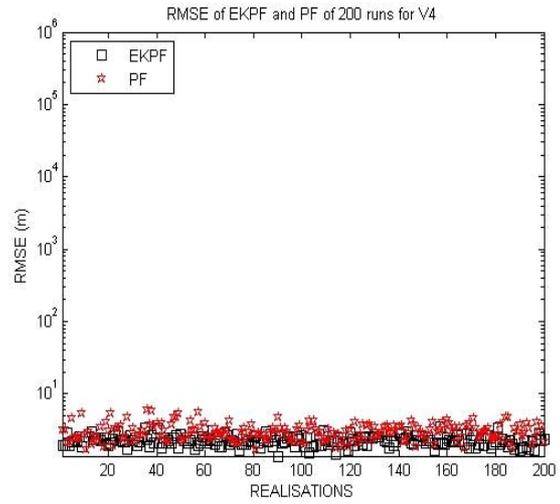
This section conducts simulations for the PF and EKPF tracking algorithms with varying settings including the prior estimate of the target state, SNR and the particle numbers used in the algorithms. The simulation results are obtained by performing 200 Monte Carlo runs of PF and EKPF under these settings for each of the six tracking scenarios as depicted in Figure 4.3.

Figure 4.10 plots the $RMSE^n$ values of PF (using 1000 particle) and EKPF (using 200 particles) of 200 Monte Carlo runs. Figure 4.11 plots the $RMSE_k$ values averaged over 200 Monte Carlo runs. Figure 4.12 also shows the $RMSE_k$ values of PF and EKPF; however, these $RMSE_k$ values are computed by the exclusion of bottom 50 runs with the lowest RSME values and top 50 runs with the largest RMSE values (it is referred to as the processed data in the figures). To compare the performance of PF and EKPF with that of S-EKF and S-UKF, the simulation results from Figures 4.5 and 4.6 are also included in Figures 4.11 and 4.12, respectively. For each Monte Carlo run of a given tracking scenario, the true target trajectory and the set of active sensing nodes are kept unchanged but the simulated measurements at each sensing node are regenerated according to Equation 4.107, setting the target energy as $S = 5000$ and the background noise as $\varepsilon_k^n \sim N(0,1)$. In the simulations, all four tracking algorithms, S-EKF, S-UKF, PF and EKPF adopt the same prior estimate of the target state in which the mean and covariance are set to $\mathbf{x}_{0|0}^3$ and $\mathbf{P}_{0|0}^3$ (defined in Equation 4.110), respectively.

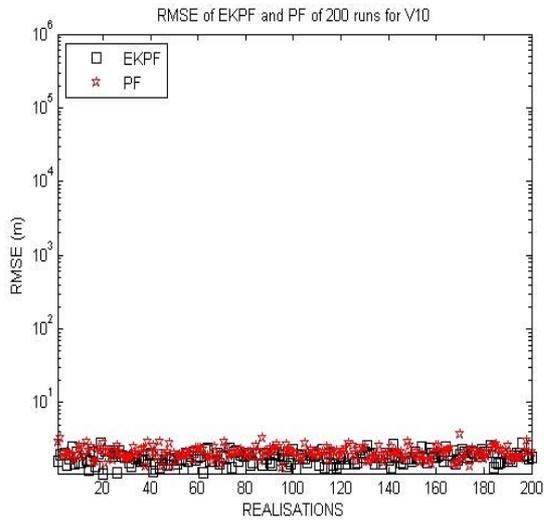
From Figures 4.10, 4.11 and 4.12, it could be seen that the overall performance of PF and EKPF is superior to the performance of S-EKF and S-UKF; and EKPF outperforms other three algorithms in term of tracking accuracy and robustness. Figure 4.10 shows that, in 200 independent Monte Carlo runs for each of the six tracking scenarios, very few magnitudes of $RMSE^n$ values of EKPF exceed $10\ m$, with most $RMSE^n$ being less than $5\ m$. In contrast, in many runs of S-EKF and S-UKF, the magnitude of RMSE exceeds $50\ m$ (refer to Figure 4.4). It is also shown in Figures 4.11 and 4.12 that PF in general could attain better tracking accuracy than both S-EKF and S-UKF.



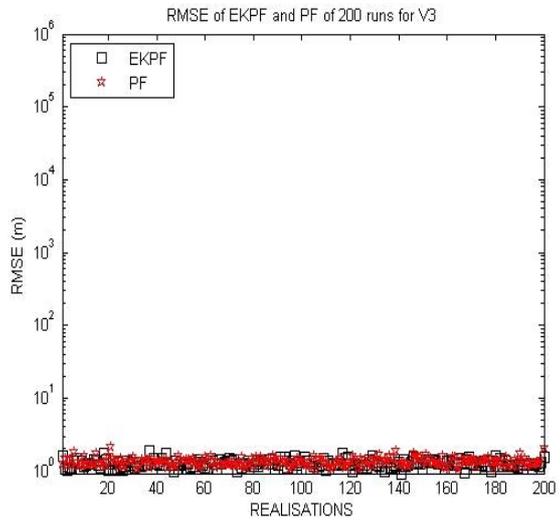
V12



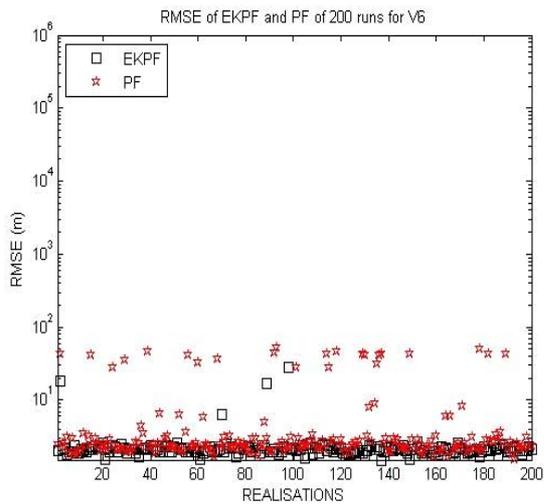
V4



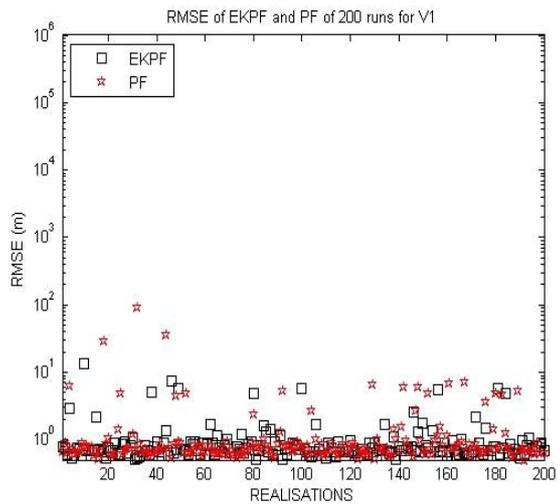
V10



V3

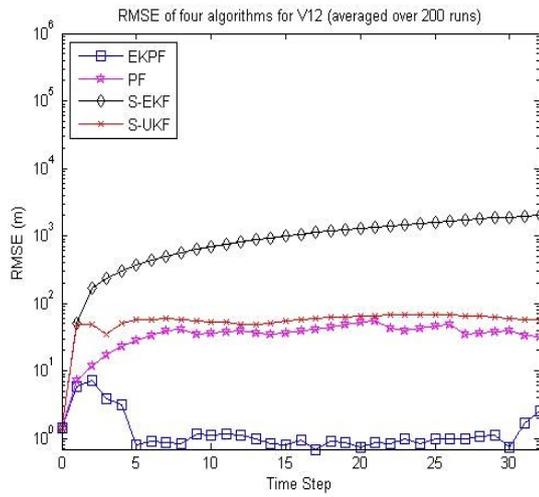


V6

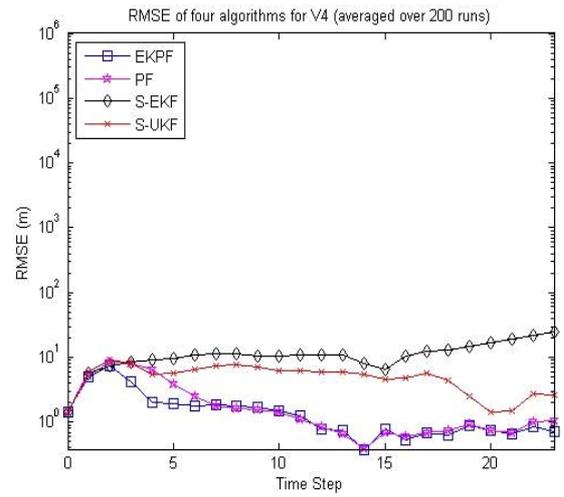


V1

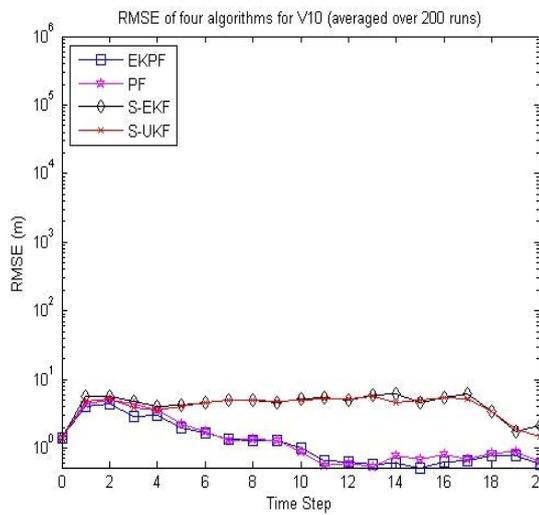
Figure 4.10 $RMSE^n$ values of PF and EKPF algorithms of 200 Monte Carlo runs for six tracking scenarios



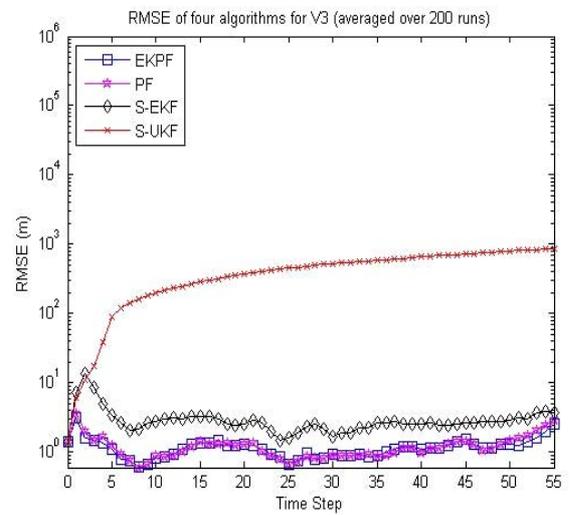
V12



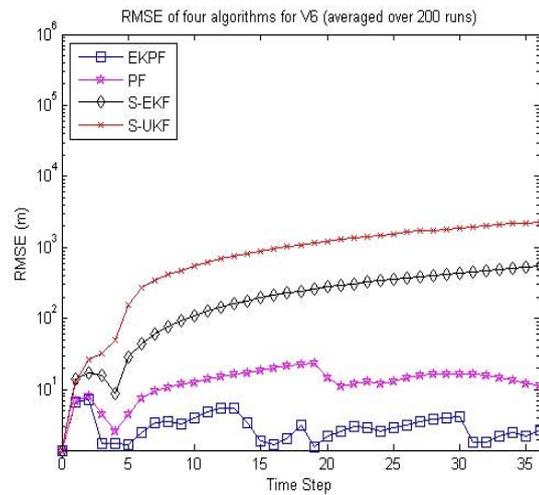
V4



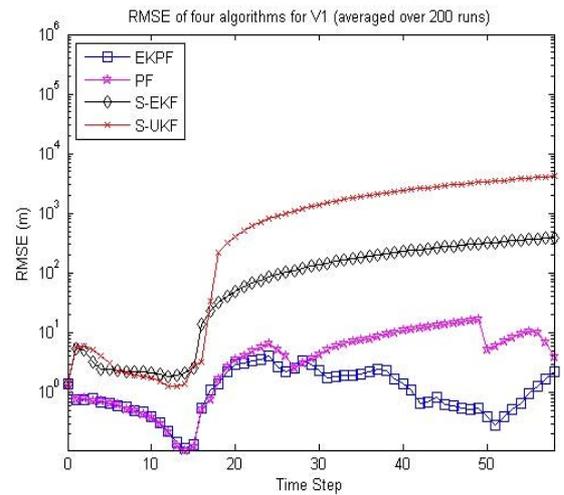
V10



V3

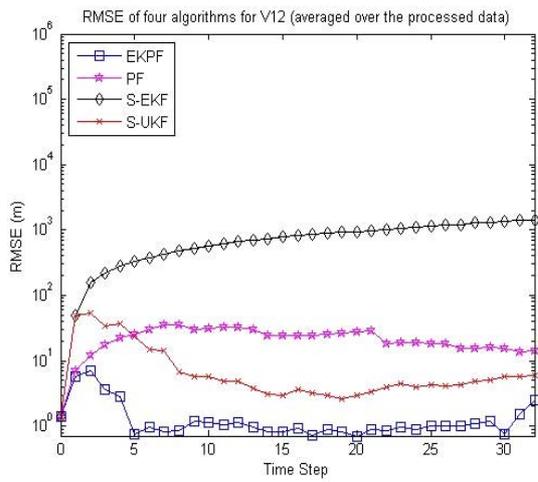


V6

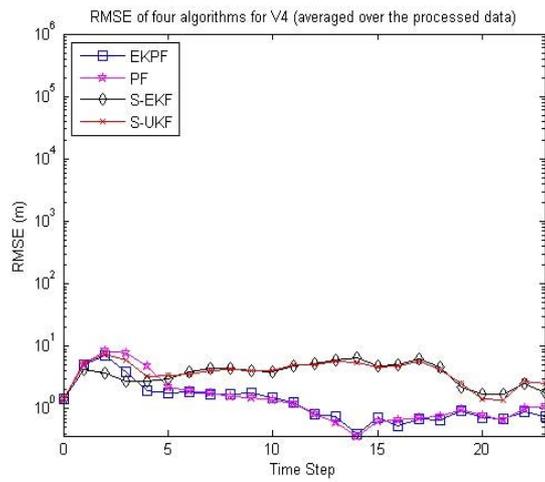


V1

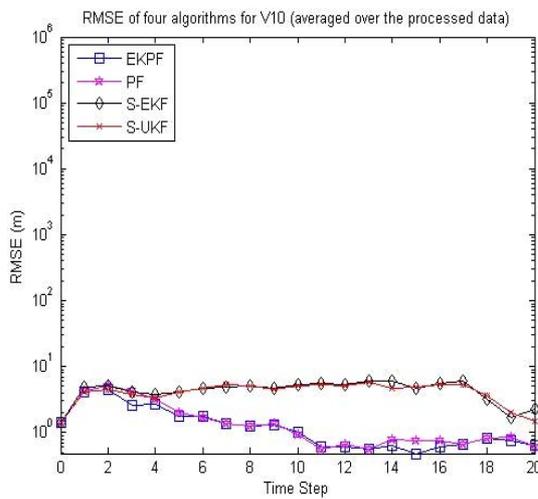
Figure 4.11 $RMSE_k$ values of S-EKF, S-UKF, PF and EKPF algorithms of each time step for six tracking scenarios (averaged over 200 runs)



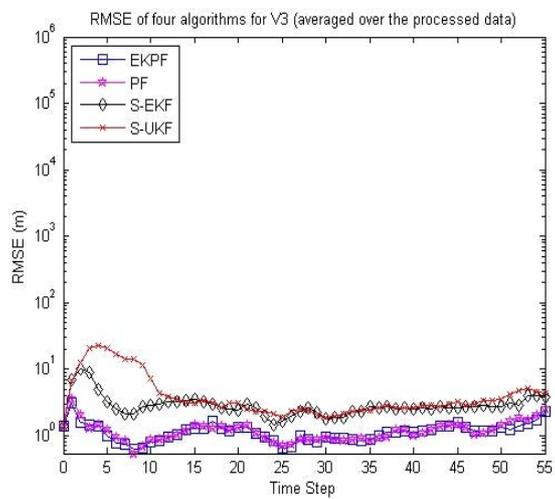
V12



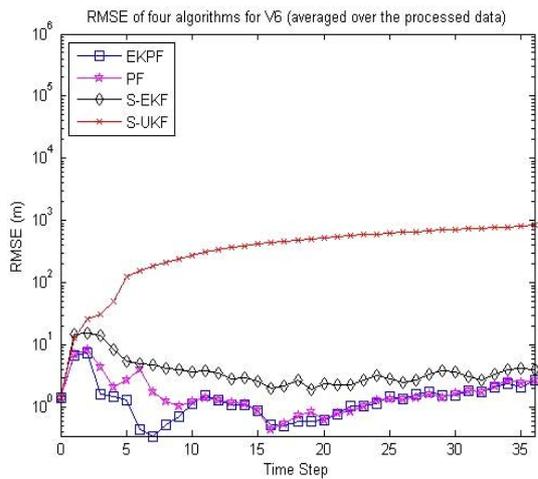
V4



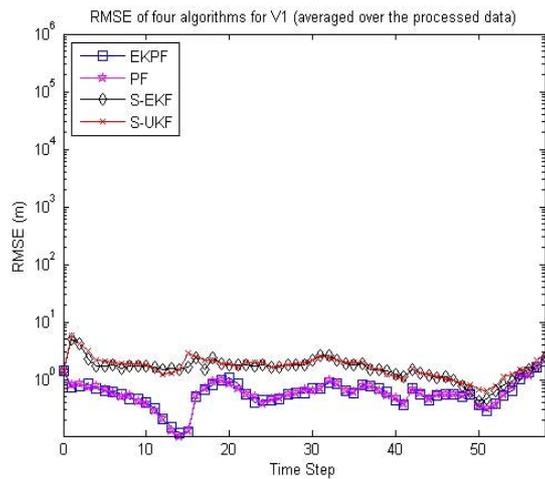
V10



V3



V6



V1

Figure 4.12 $RMSE_k$ values of S-EKF, S-UKF, PF and EKPF algorithms of each time step for six tracking scenarios (averaged over the processed data)

Figures 4.13 and 4.14 depict the performance of PF and EKPF algorithms for tracking scenario V12 under different SNR settings, respectively. In both figures, the mean and covariance of the prior estimate of the target state is still set to $\mathbf{x}_{0|0}^3$ and $\mathbf{P}_{0|0}^3$; the particle numbers adopted in PF is 1000 and in EKPF is 200. From Figures 4.13 and 4.14, it can be seen that the $RMSE_k$ magnitude of both EKPF and PF turns to divergent when SNR was decreased to 20dB; however, in other four occasions with different SNR settings, EKPF outperforms PF. As shown in Figure 4.14, the tracking accuracy of EKPF decreases when SNR decreased. However, as shown in Figure 4.13, PF does not exhibit similar performance characteristics: the magnitude of $RMSE_k$ under SNR 37 dB is larger than the magnitudes of $RMSE_k$ under SNR 30 dB and SNR 27 dB; and the magnitude of $RMSE_k$ under SNR 30 dB is yet again larger than the magnitude of $RMSE_k$ under SNR 27 dB. This curious behaviour is due to the PF particle propagation details. In the PF tracking algorithm, the particles in the state space are driven by the process noise in the system model to move from one time step to the next; very low noise levels (i.e. high SNR) might cause the particles not to move to the high measurement likelihood area in the state space. In turn, PF could not attain high accuracy and the magnitude of RMSE tends to be large. Especially, in cases where the likelihood is too narrow (highly peaked value due to low measurement error), the RMSE values of Monte Carlo runs are not bounded. Further evidence of this behaviour can be found in Figures 4.15 and 4.16. Figure 4.15 shows the $RMSE^n$ values of 200 Monte Carlo runs under SNR 37 dB and 27 dB, from which it can be observed that the RMSE values under SNR 37 dB are not well-bounded compared to RMSE values under SNR 27 dB. Figure 4.16 depicts the measurement likelihood function under SNR 37 dB and 27 dB at two time instances. It can be found that the likelihood under SNR 37 dB has a much narrower peak than that under SNR 27 dB. This explains why the PF performance under SNR 37 dB is worse than the performance under SNR 27 dB. In contrast to PF tracking algorithm, EKPF tracking algorithm takes into account the most recent measurements into the proposal distribution and use this to move the particles to the area of high measurement likelihood, which allows it to achieve higher tracking accuracy in a more consistent way.

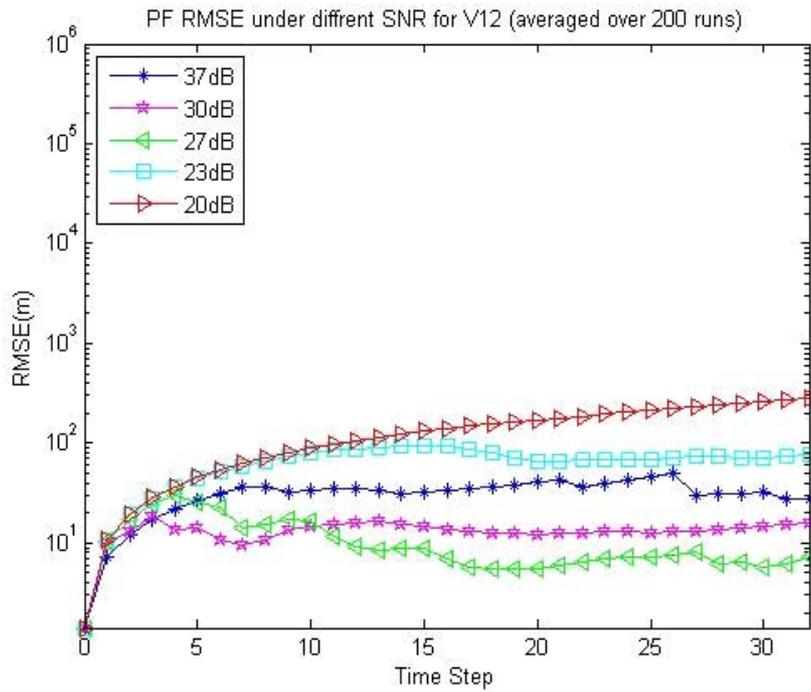


Figure 4.13 PF $RMSE_k$ values under different SNR settings for tracking scenario V12 (averaged over 200 runs)

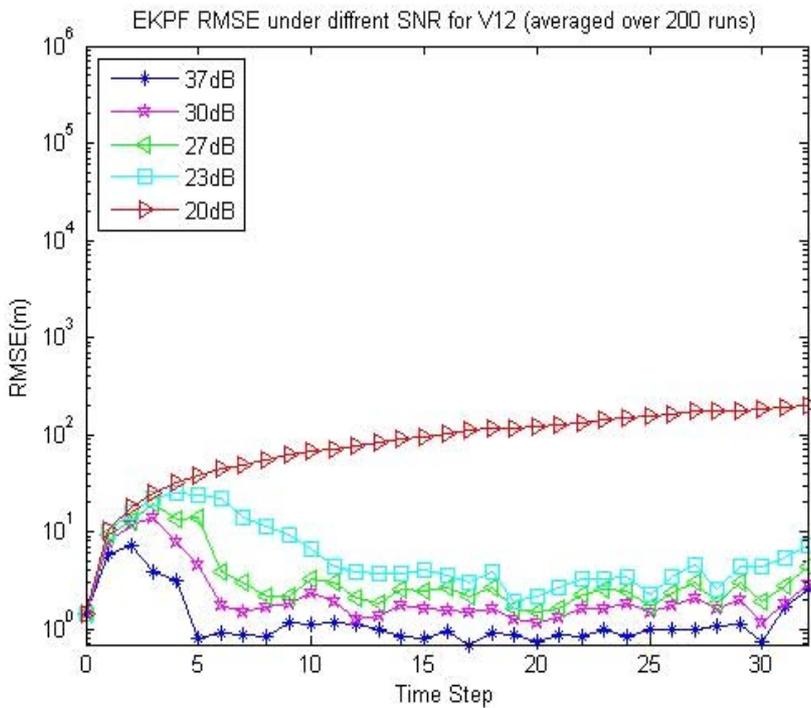


Figure 4.14 EKPF $RMSE_k$ values under different SNR settings for tracking scenario V12 (averaged over 200 runs)

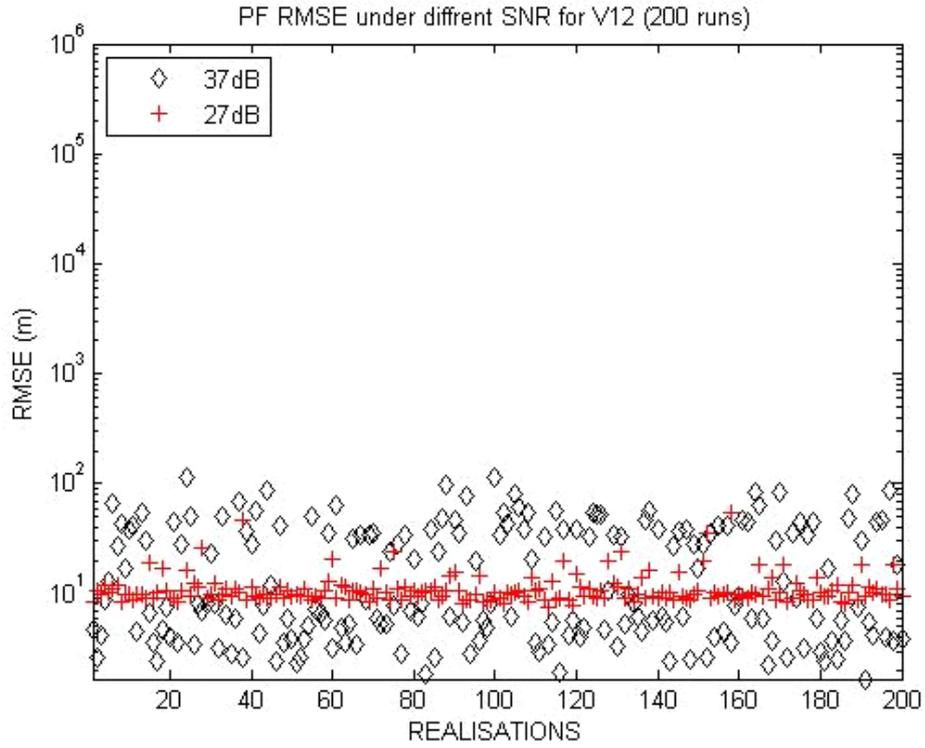


Figure 4.15 PF $RMSE^n$ values under SNRs 37 dB and 27 dB for tracking scenario V12

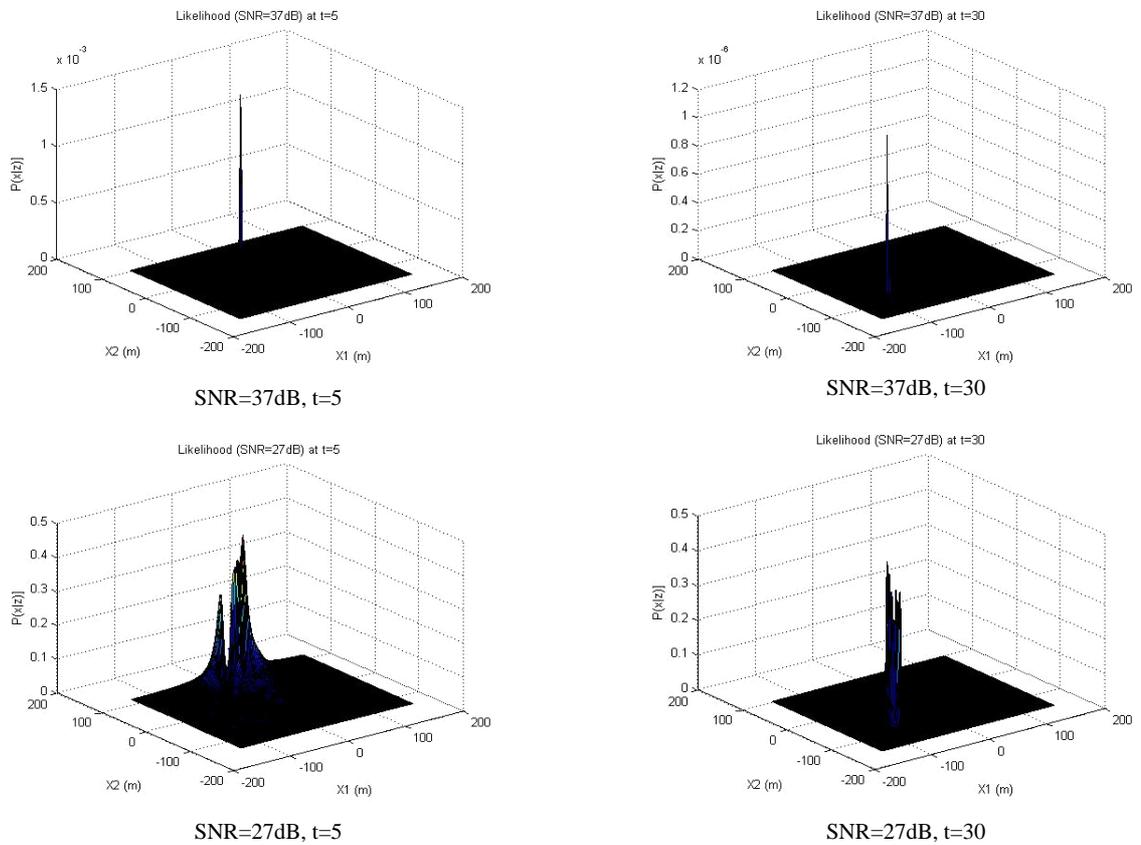
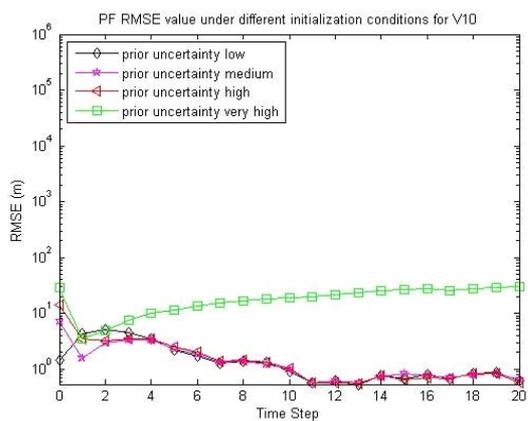
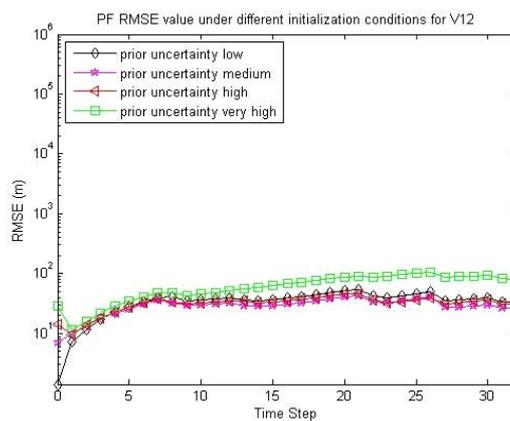


Figure 4.16 Snapshots of the likelihood with SNRs 37 dB and 27 dB for tracking scenario V12

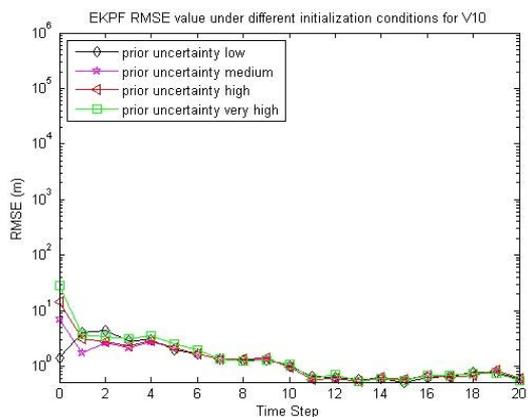
Figure 4.17 shows the performance of PF and EKPF under different initialisation conditions, i.e. the different prior estimate of the target state for tracking scenarios V10 and V12. Four sets of different initialisation conditions are: uncertainty high (defined by Equations 4.108); uncertainty medium (defined by Equations 4.109); uncertainty low (defined by Equations 4.110); and uncertainty very high which is defined as $\mathbf{x}_{0|0}^5 = \mathbf{x}_{truth} + [20 \ 0 \ 20 \ 0]^T$, $\mathbf{P}_{0|0}^5 = \text{diag}(10, 10, 10, 10)$. Table 4.1 lists the mean of the $RMSE_k$ values (i.e. $RMSE$ is averaged over both time steps and 200 runs) for tracking scenarios V12 and V10. From both Figure 4.17 and Table 4.1, it can be seen in both tracking scenarios, EKPF performs very well even when the uncertainty of the prior estimate of target state becomes very large. In contrast, the performance of PF degrades dramatically when the uncertainty turns to very large. Therefore, Figure 4.17 and Table 4.1 again show that EKPF outperforms PF in terms of tracking accuracy and robustness.



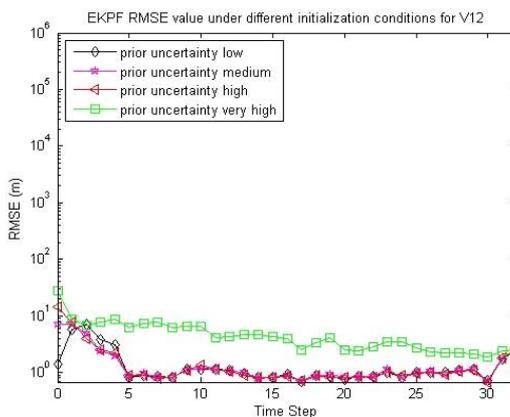
PF RMSE under different prior estimate of target state (V10) (averaged over 200 runs)



PF RMSE under different prior estimate of target state (V12) (averaged over 200 runs)



EKPF RMSE under different prior estimate of target state (V10) (averaged over 200 runs)



EKPF RMSE under different prior estimate of target state (V12) (averaged over 200 runs)

Figure 4.17 PF and EKPF $RMSE_k$ under different prior estimate of target state for tracking scenarios V12 and V10 (averaged over 200 runs)

Table 4.1 Mean of $RMSE_k$ (m) over 200 runs for tracking scenarios V12 and V10

Tracking Scenario	Tracking Algorithm	Uncertainty Small	Uncertainty Medium	Uncertainty Large	Uncertainty Very Large
V12	EKPF	2.058	2.245	3.092	6.250
V12	PF	26.340	20.753	22.895	40.333
V10	EKPF	1.828	2.054	3.446	6.247
V10	PF	2.117	2.191	3.544	12.848

Figure 4.18 and 4.19 show the RMSE values of PF and EKPF using different numbers of particles. SNR is set to 37 dB and the prior estimate of the target state is set with mean $\mathbf{x}_{0|0}^1$ and covariance $\mathbf{P}_{0|0}^1$ (Equation 4.108). Both figures show increases in the number of particles lead to better performance of both PF and EKPF algorithms. However, increasing particle number will introduce extra computation burden and may not be favoured in real wireless sensor networks environments. Figure 4.19 also reveals that EKPF can still attain reasonable accuracy when it uses a relatively small number of particles. Recalled from the simulation results as depicted in Figures 4.10, 4.11 and 4.12, the performance of EKPF employing 200 particles is equivalent to or even better than the performance of PF employing 1000 particles. Therefore, the particle numbers in EKPF can be significantly reduced without decreasing the accuracy of the tracking results. This helps to mitigate the computation cost introduced by employing N EKFs (N is the number of particles adopted in the algorithm) in EKPF.

Table 4.2 lists the representative run time of one time step target state estimate with S-EKF, S-UKF, PF and EKPF tracking algorithms.

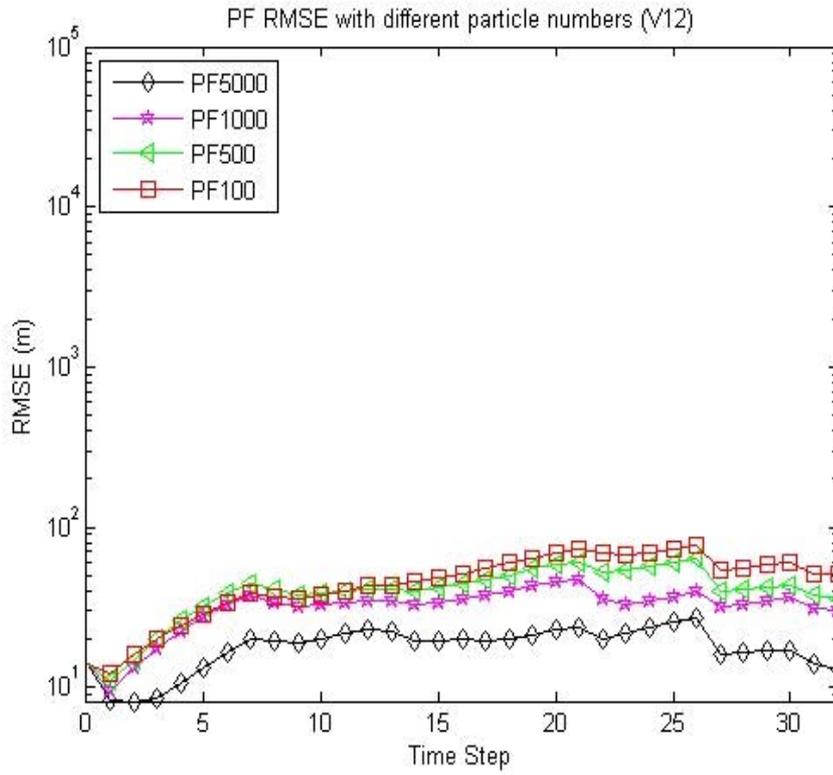


Figure 4.18 PF $RMSE_k$ values under different particle numbers for tracking scenario V12 (averaged over 200 runs)

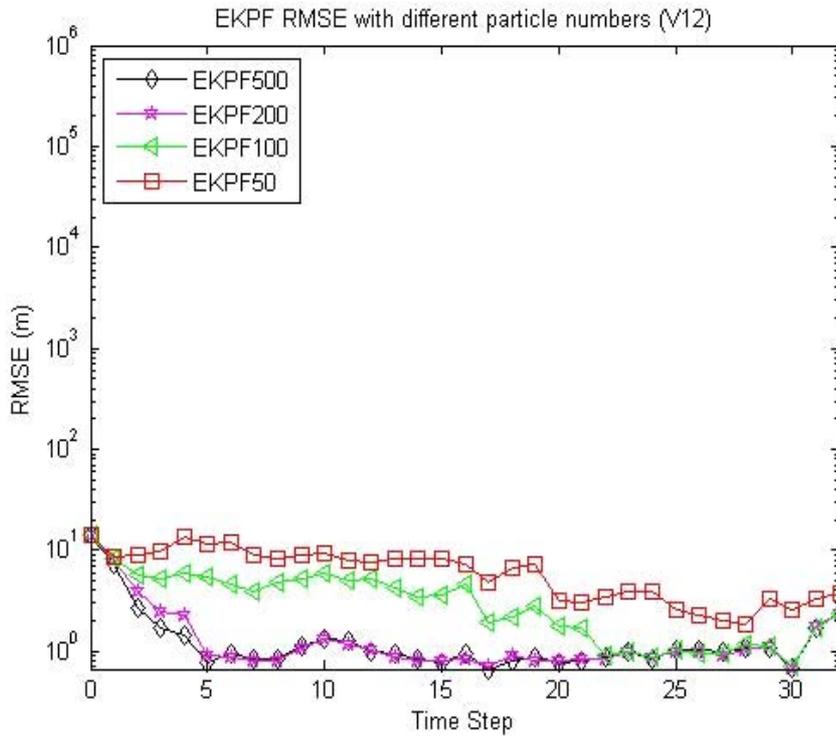


Figure 4.19 EKPF $RMSE_k$ values under different particle numbers for tracking scenario V12 (averaged over 200 runs)

Table 4.2 Run time of four tracking algorithms for one time step target state estimate

Algorithm	Particle Numbers	Run Time (s)
S-EKF		0.0041
S-UKF		0.0245
PF	5000	0.1767
	1000	0.0253
	500	0.0138
	100	0.0056
EKPF	500	0.4243
	200	0.1532
	100	0.0804
	50	0.0440

4.8.4 Conclusion Remarks of Simulation Results

From the above simulation results, it can be concluded that EKPF tracking algorithm outperforms S-EKF, S-UKF and PF tracking algorithms in terms of tracking accuracy and robustness. However, compared to S-EKF, S-UKF and PF tracking algorithms, EKPF tracking algorithm introduces a higher computation burden (Table 4.2). Therefore, for practical target tracking in wireless sensor networks, the selection of an appropriate algorithm will involve carefully trading off the tracking performance (accuracy and robustness) with the computation cost and the tracking conditions (the knowledge regarding the sensor field such as sensor nodes deployment, SNR settings, and the initial estimate of the target state... etc). It is expected that the exact trade-off will be problem dependent, and is beyond the scope of our investigation.

4.9 Posterior Cramer-Rao Lower Bound (PCRLB) for Tracking a Single Target in Wireless Sensor Networks

This section computes the posterior Cramer-Rao lower bound (PCRLB) for tracking a single target in wireless sensor networks. The PCRLB is defined to be the inverse of the Fisher information matrix (FIM) for a random vector. It provides a lower bound on the mean square error (MSE) of the target state estimate to which the tracking algorithms can attain. The PCRLB was first introduced by Van Trees [50]. Tichavsky derived the recursive PCRLB facilitating the update of FIM from one time step to the next time step

[117]. Recently, several authors proposed a number of approaches to calculate the PCRLB under various conditions in target tracking applications [118], [119], [124]–[126].

In this section, the PCRLB under the assumptions of no clutter and missed detections is calculated and compared for the tracking algorithms developed in this chapter. In Chapter 5, the PCRLB that accommodates the measurement origin uncertainty due to clutter and missed detections will be derived and computed. In Chapter 6, the PCRLB will be used as the information utility measure in implementing the sensing nodes selection scheme for the distributive target tracking in wireless sensor networks.

4.9.1 PCRLB for Single Target Tracking without Clutter and Missed Detections

For the state space model as defined in Equations 4.1 and 4.2, the relationship between the PCRLB and the error covariance matrix of the target state estimate is:

$$E \left\{ [\hat{\mathbf{x}}_k - \mathbf{x}_k][\hat{\mathbf{x}}_k - \mathbf{x}_k]^T \right\} \geq \mathbf{J}_k^{-1} \quad (4.114)$$

where $\hat{\mathbf{x}}_k$ is the estimate of the true target state \mathbf{x}_k . \mathbf{J}_k is the FIM (i.e. the inverse of the PCRLB) at the k -th time step and its (i, j) -th element is defined as

$$\mathbf{J}_k(i, j) = E \left\{ \left[-\frac{\partial^2 \ln p(\mathbf{z}_k, \mathbf{x}_k)}{\partial \mathbf{x}_k(i) \partial \mathbf{x}_k(j)} \right] \right\} \quad (4.115)$$

where $p(\mathbf{z}_k, \mathbf{x}_k)$ denotes the joint probability density function of the target state \mathbf{x}_k and the measurement \mathbf{z}_k , $\mathbf{x}_k(i)$ denotes the i -th component of \mathbf{x}_k , and the expectation $E(\cdot)$ is taken with respect to both \mathbf{z}_k and \mathbf{x}_k .

Assuming we have already obtained the FIM \mathbf{J}_k of the k -th time step, we can compute the FIM \mathbf{J}_{k+1} for the $(k+1)$ -th time step as follows [117]:

$$\mathbf{J}_{k+1} = D_k^{22} - D_k^{21} \left(\mathbf{J}_k + D_k^{11} \right)^{-1} D_k^{12} \quad (4.116)$$

where

$$D_k^{11} = E \left\{ \left[-\nabla_{\mathbf{x}_k} \nabla_{\mathbf{x}_k}^T \ln p(\mathbf{x}_{k+1} | \mathbf{x}_k) \right] \right\} \quad (4.117)$$

$$D_k^{12} = E \left\{ \left[-\nabla_{\mathbf{x}_{k+1}} \nabla_{\mathbf{x}_k}^T \ln p(\mathbf{x}_{k+1} | \mathbf{x}_k) \right] \right\} \quad (4.118)$$

$$D_k^{21} = E \left\{ \left[-\nabla_{\mathbf{x}_k} \nabla_{\mathbf{x}_{k+1}}^T \ln p(\mathbf{x}_{k+1} | \mathbf{x}_k) \right] \right\} = \left[D_k^{12} \right]^T \quad (4.119)$$

$$D_k^{22} = E \left\{ \left[-\nabla_{\mathbf{x}_{k+1}} \nabla_{\mathbf{x}_{k+1}}^T \ln p(\mathbf{x}_{k+1} | \mathbf{x}_k) \right] \right\} \\ + E \left\{ \left[-\nabla_{\mathbf{x}_{k+1}} \nabla_{\mathbf{x}_{k+1}}^T \ln p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}) \right] \right\} \quad (4.120)$$

Since the system model adopted in this chapter is the nearly CV model which is linear (Equation 4.3), Equations 4.117, 4.118, 4.119 and the first part of Equation 4.120 are independent of \mathbf{x}_k :

$$D_k^{11} = \mathbf{A}_k^T \mathbf{Q}_k^{-1} \mathbf{A}_k \quad (4.121)$$

$$D_k^{12} = -\mathbf{A}_k^T \mathbf{Q}_k^{-1} \quad (4.122)$$

$$D_k^{21} = -\mathbf{Q}_k^{-1} \mathbf{A}_k^T \quad (4.123)$$

$$D_k^{22} = \mathbf{Q}_k^{-1} + E \left\{ \left[-\nabla_{\mathbf{x}_{k+1}} \nabla_{\mathbf{x}_{k+1}}^T \ln p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}) \right] \right\} \quad (4.124)$$

where \mathbf{A}_k is the system matrix and \mathbf{Q}_k is the process noise covariance matrix, both have been defined earlier in this chapter.

In Equation 4.124, the expectation $E \left\{ \left[-\nabla_{\mathbf{x}_{k+1}} \nabla_{\mathbf{x}_{k+1}}^T \ln p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}) \right] \right\}$ gives the dependency of PCRLB on the measurement. As mentioned in Section 4.2, in most practical target tracking applications in wireless sensor networks, the measurements obtained at the n -th sensor node consists of two types of measurements: the target originating measurement with the assumption that measurement noise is Gaussian; and the clutter originating measurements which are assumed to be uniformly distributed within the observation space of the n -th sensing node. Thus, the measurement likelihood $p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1})$ is a mixture of Gaussian and uniform probability densities and this lead to the difficulties in calculating the expectation in Equation 4.124. However, as derived in the next chapter, the expectation in Equation 4.124 can be reduced to

$$E \left\{ \left[-\nabla_{\mathbf{x}_{k+1}} \nabla_{\mathbf{x}_{k+1}}^T \ln p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}) \right] \right\} \\ = q_{k+1} \hat{\mathbf{H}}_{k+1}^T \mathbf{R}_{k+1}^{-1} \hat{\mathbf{H}}_{k+1} \quad (4.125)$$

where the matrix $\hat{\mathbf{H}}_{k+1}$ is the Jacobian of the measurement function \mathbf{h}_{k+1} in the state space model (Equations 4.1 and 4.2) and it is given by

$$\hat{\mathbf{H}}_{k+1} = \left. \frac{d\mathbf{h}_{k+1}(\mathbf{x})}{d\mathbf{x}} \right|_{\mathbf{x}=\mathbf{f}_k, \mathbf{m}_{k|k}} \quad (4.126)$$

where $\mathbf{m}_{k|k}$ is the mean of the target state estimate at the k -th time step (refer to Section 4.3). In Equation 4.125, the scalar variable q_{k+1} is called the *information reduction factor* (IRF) which represents the effect of the measurement origin uncertainty due to clutter and missed detections on the PCRLB [118], [119]. IRF depends on several factors such as the covariance of measurement noise, the probability of detection, the density of clutter originating measurements, and the volume of the observation space of the sensing node. In this chapter, however, it has already assumed that there is no clutter and missed detections, thus the IRF is simply set to unit, i.e. $q_{k+1} = 1$. The calculation of IRF that takes account of the measurement origin uncertainty due to clutter and missed detections is deferred to Chapter 5.

By substituting Equations 4.121~4.126 (setting $q_{k+1} = 1$) into 4.116, the FIM, \mathbf{J}_{k+1} of the $(k+1)$ -th time step, becomes

$$\mathbf{J}_{k+1} = \left(\mathbf{Q}_k + \mathbf{A}_k^T \mathbf{J}_k^{-1} \mathbf{A}_k \right)^{-1} + \mathbf{\Gamma}_{k+1} \quad (4.127)$$

where

$$\mathbf{\Gamma}_{k+1} = \hat{\mathbf{H}}_{k+1}^T \mathbf{R}_{k+1}^{-1} \hat{\mathbf{H}}_{k+1} \quad (4.128)$$

is the measurement contribution to \mathbf{J}_{k+1} . Note that the initial FIM (\mathbf{J}_k at $k=0$) is the inverse of the covariance of the prior target state estimate, i.e. $\mathbf{J}_0 = \mathbf{P}_{0|0}^{-1}$.

4.9.2 PCRLB Calculation

From Equations 4.126~4.128, it can be seen that the target state at the k -th time step is needed for the computation of FIM \mathbf{J}_{k+1} for the $(k+1)$ -th time step. Similar to the generic PF tracking algorithm, we adopt particles' representation of the target state to compute the FIM \mathbf{J}_{k+1} (PCRLB). In the computation, the particles are propagated from the k -th time

step to the $(k+1)$ -th time which is almost the same as the importance sampling step in the PF tracking algorithm (refer to Section 4.6):

$$\mathbf{x}_{k+1}^i = \mathbf{A}_{k+1} \mathbf{x}_k^i \quad i=1,2,\dots,N \quad (4.129)$$

where \mathbf{x}_{k+1}^i denotes the i -th particle at the k -th time step and \mathbf{A}_{k+1} is the system matrix and has been defined in Equation 4.4 for the nearly CV model.

In order to compare the PCRLB with the MSE of the tracking algorithms developed in this chapter, the same assumption as in the previous sections is also made here: a fixed set of N_s active sensing nodes participated in the tracking task. Consequently, Equation 4.128 becomes:

$$\mathbf{\Gamma}_{k+1} = \sum_{n=1}^{N_s} \left(\hat{\mathbf{H}}_{k+1}^n \right)^T \left(\mathbf{R}_{k+1}^n \right)^{-1} \left(\hat{\mathbf{H}}_{k+1}^n \right) \quad (4.130)$$

With the measurement model defined in Equation 4.6 and the particles' representation of the target state, the Jacobian $\hat{\mathbf{H}}_{k+1}^n$ for the n -th sensing node at the $(k+1)$ -th time step can be computed as follows:

$$\hat{\mathbf{H}}_{k+1}^n = \begin{pmatrix} \frac{-2S \left(x_{k+1}^i - x_{k+1}^n \right)}{\left(x_{k+1}^i - x_{k+1}^n \right)^2 + \left(y_{k+1}^i - y_{k+1}^n \right)^2} \\ 0 \\ \frac{-2S \left(y_{k+1}^i - y_{k+1}^n \right)}{\left(x_{k+1}^i - x_{k+1}^n \right)^2 + \left(y_{k+1}^i - y_{k+1}^n \right)^2} \\ 0 \end{pmatrix}^T \quad (4.131)$$

where S is the target energy and set to $S = 5000$ as in the previous sections. $\left(x_{k+1}^n, y_{k+1}^n \right)$ is the n -th sensing node's position. $\left(x_{k+1}^i, y_{k+1}^i \right)$ is the i -th particle's position and given by Equation 4.129.

Substituting Equation 4.131 into Equation 4.130, $\mathbf{\Gamma}_{k+1}$ can be computed as follows

$$\mathbf{\Gamma}_{k+1} = \begin{pmatrix} \tau_{k+1}^{11} & 0 & \tau_{k+1}^{13} & 0 \\ 0 & 0 & 0 & 0 \\ \tau_{k+1}^{31} & 0 & \tau_{k+1}^{33} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.132)$$

where τ_{k+1}^{11} , $\tau_{k+1}^{13} = \tau_{k+1}^{31}$ and τ_{k+1}^{33} are given as below

$$\tau_{k+1}^{11} = \frac{1}{N\mathbf{R}_{k+1}} \left\{ \sum_{i=1}^N \sum_{n=1}^{N_s} \frac{4S^2 (x_{k+1}^i - x_{k+1}^n)^2}{\left[(x_{k+1}^i - x_{k+1}^n)^2 + (y_{k+1}^i - y_{k+1}^n)^2 \right]^2} \right\} \quad (4.133)$$

$$\tau_{k+1}^{13} = \frac{1}{N\mathbf{R}_{k+1}} \left\{ \sum_{i=1}^N \sum_{n=1}^{N_s} \frac{4S^2 (x_{k+1}^i - x_{k+1}^n)(y_{k+1}^i - y_{k+1}^n)}{\left[(x_{k+1}^i - x_{k+1}^n)^2 + (y_{k+1}^i - y_{k+1}^n)^2 \right]^2} \right\} \quad (4.134)$$

$$\tau_{k+1}^{33} = \frac{1}{N\mathbf{R}_{k+1}} \left\{ \sum_{i=1}^N \sum_{n=1}^{N_s} \frac{4S^2 (y_{k+1}^i - y_{k+1}^n)^2}{\left[(x_{k+1}^i - x_{k+1}^n)^2 + (y_{k+1}^i - y_{k+1}^n)^2 \right]^2} \right\} \quad (4.135)$$

In the above equations, it is assumed that the measurement noise at each sensing node takes the same value with zero mean and covariance \mathbf{R}_{k+1} . Combining Equations 4.132~4.135 with Equations 4.127~4.128, we can finally obtain the FIM \mathbf{J}_{k+1} for the $(k+1)$ -th time step. Using the above procedure, the PCRLB can be recursively computed for single target tracking in wireless sensor networks.

Note that in the above computation of FIM \mathbf{J}_{k+1} for the $(k+1)$ -th time, we didn't make use of the measurements obtained at the $(k+1)$ -th time step, i.e. \mathbf{z}_{k+1}^n , $n = 1, \dots, N_s$. The implication is that we can predict PCRLB: if we have already known the target state estimate and the PCRLB at the k -th time step, then we can compute the PCRLB for the $(k+1)$ -th time step. Based on this property, in chapter 6 we adopt PCRLB as the information utility measure in a composite function facilitating the sensing nodes selection in distributive target tracking in wireless sensor networks.

Figure 4.20 depicts the $\sqrt{\text{PCRLB}}$ and $RMSE_k$ values of S-EKF, S-UKF, PF and EKPF algorithms for each of the six tracking scenarios (Figure 4.3) under SNR 37 dB and with the prior estimate of the target state of mean $\mathbf{x}_{0|0}^3$ and covariance $\mathbf{P}_{0|0}^3$ (Equation 4.110). It

can be seen from Figure 4.20 that in all six tracking scenarios, the $RMSE_k$ magnitudes of EKPF are very close to the magnitudes of \sqrt{PCRLB} . This again proves the superiority of EKPF over PF, S-EKF and S-UKF for tracking a single target in wireless sensor networks.

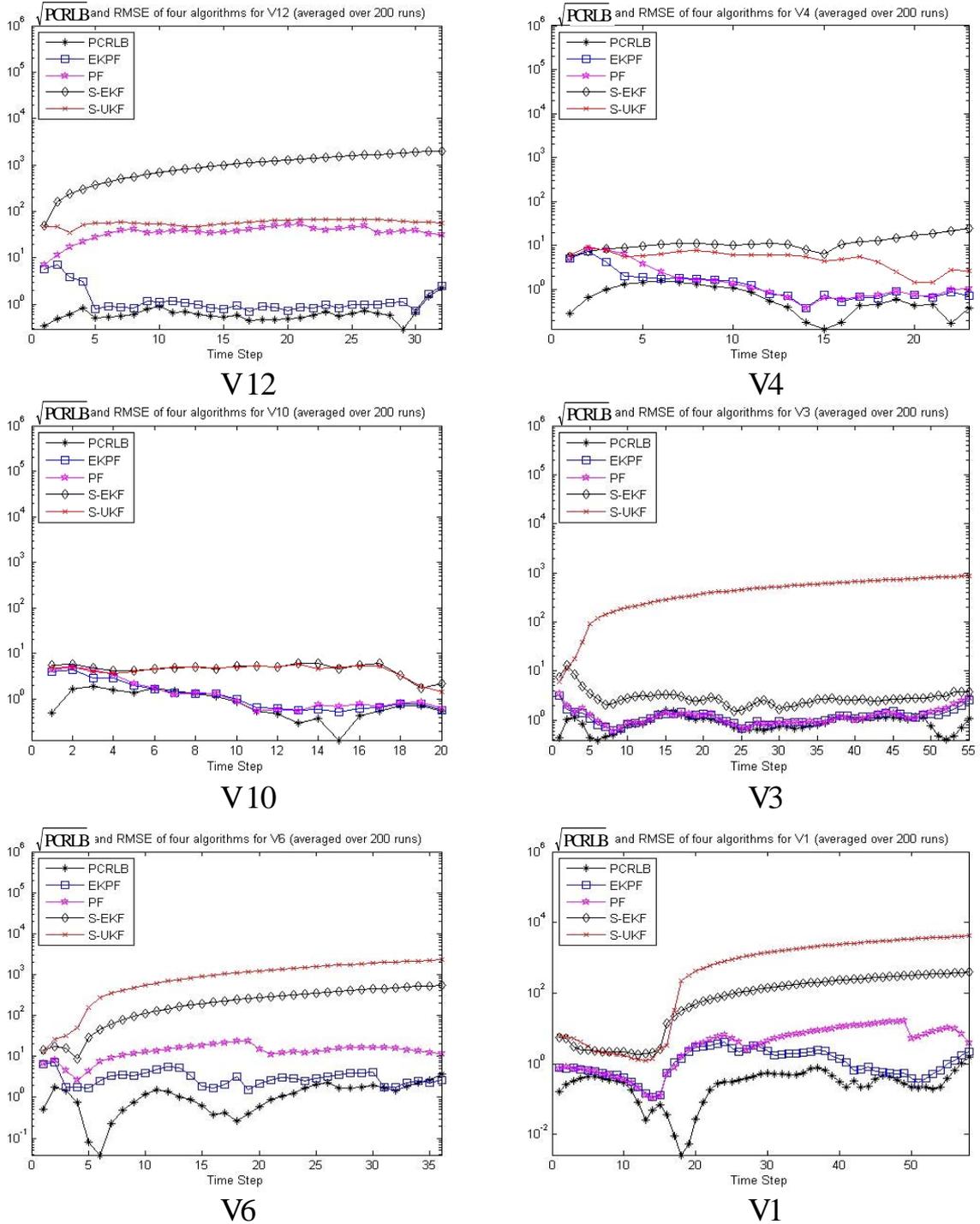


Figure 4.20 \sqrt{PCRLB} and $RMSE_k$ of different tracking algorithms for the six tracking scenarios

4.10 Summary

On the basis of the recursive Bayesian estimation method, this chapter develops a number of tracking algorithms including S-EKF, S-UKF and PF for tracking a single target in wireless sensor networks. Especially, a novel hybrid EKPF is also developed and the simulation results show that the EKPF outperforms other three algorithms in terms of tracking accuracy and robustness. Despite the EKPF requiring greater computational efforts than PF, it is possible to reduce the particles used in the EKPF to mitigate this without incurring any loss in performance. To help evaluate the performance of the developed tracking algorithms, the PCRLB which is the theoretical lower bound on the mean square error of the target state estimation is also computed in this chapter.

The practical target tracking task in wireless sensor networks is normally performed in cluttered environment. The measurements acquired at the individual sensing node may contain clutter generating measurements and the target may even go to undetected. To deal with such measurement origin uncertainty, Chapter 5 will develop a Particle filter (PF) and probabilistic data association filter (PDAF) hybrid algorithm, named as PF-PDAF. Due to the unique characteristics of wireless sensor networks, the tracking algorithms for wireless sensor networks need to be distributive. To fulfil this requirement, Chapter 6 will develop distributive PF, EKPF and PF-PDAF tracking algorithms.