

Chapter 7

Tracking Multiple Targets in Wireless Sensor Networks

This chapter addresses the problem of tracking multiple targets under measurement origin uncertainty in wireless sensor networks. By adopting the particle's representation of the probability density function of the target state, a multiple target tracking algorithm have been developed in this chapter. This algorithm is a hybrid of Particle filter (PF) and joint probabilistic data association filter (JPDAF), named as PF-JPDAF tracking algorithm. PF-JPDAF combines the advantage of PF being applicable to the general nonlinear systems with the ability of JPDAF that can effectively tackle the challenging data association problem when tracking multiple targets. Extensive simulations have been conducted to evaluate the performance of PF-JPDAF.

7.1 Introduction

Multiple target tracking is one of the typical applications of wireless sensor networks in which a large number of sensor nodes collaboratively sense, process and infer the states of multiple targets [1]. For centralized systems (*e.g.* radar, sonar...etc.), there are many multiple target tracking strategies have been proposed and the related techniques are well-established [27], [47]. However, wireless sensor networks' unique characteristics, especially their highly distributed nature and limited resources pose significant challenges in developing algorithms for the multiple target tracking applications in wireless sensor networks [1], [18]. The multiple target tracking algorithms developed for wireless sensor networks need to consider the interplay between information processing and sensor network architecture (networking), and efficiently coordinate sensor nodes to take sensing action, acquire the data, and process the information to achieve the distributive estimation of multiple targets states under the measurement origin uncertainty due to the presence of multiple targets and clutter.

In multiple target tracking, *data association* is a fundamental problem and it involves finding the correct correspondence between measurements and targets. This chapter develops PF-JPDAF algorithm to tackle the data association problem under the

measurement origin uncertainty. In PF-JPDAF, a *joint measurement-target association vector* which consists of all possible measurement and target pairs (components) is introduced. All components in this joint measurement-target association vector are then calculated according to their corresponding probabilities. These weighted components are used in the measurement update step in the recursive Bayesian estimation to update the target state estimate.

This chapter is organized as follows. Section 7.2 starts with a brief review of multiple target tracking techniques reported in the literature. It must be stated here that this review is focused on Particle filter based multiple target tracking algorithms for target tracking in wireless sensor networks. A full review of the multiple target tracking techniques is beyond the scope of this thesis. Section 7.3 formulates the multiple target tracking problem for the development of PF-JPDAF tracking algorithm in wireless sensor networks. Section 7.4 details the derivation and development of PF-JPDAF tracking algorithm. Section 7.5 presents the simulation results of PF-JPDAF tracking algorithm. Section 7.6 summarizes the whole chapter.

7.2 Multiple Target Tracking in Wireless Sensor Networks

This section provides a brief review of multiple target tracking techniques with the focus on the Particle filter (PF) based approaches. It includes several multiple target tracking strategies commonly adopted in the literature, as well as the related works of developing PF based multiple target tracking algorithms for wireless sensor networks.

7.2.1 Review of Multiple Target Tracking Techniques

In multiple target tracking, the aim is to recursively estimate, at each time step, the probability density function of the target state for each of the targets upon the receipt of new measurements. However, the measurements obtained at a sensing node may consist of both the measurements originating from targets and clutter alike. Therefore, the multiple target tracking algorithms need to identify which measurements should be associated with which targets. However, in a practical environment as the number of targets and the clutter rate both increase, the identification of the origins of the measurements quickly becomes more difficult and complex.

There are numerous strategies which have been proposed in the literature for solving multiple target tracking problem. In the Multiple Hypothesis Tracking (MHT) algorithm [27], each measurement to target hypothesis associates past measurement with a target and

a new set of hypotheses is formed from the previous hypotheses when new measurements arrive. However, such exhaustive MHT is not feasible in practical systems since it requires the evaluation of an exponentially increasing number of measurements to target association hypotheses and consequently demanding significant computational resources. To reduce the number of association hypotheses to a manageable level, techniques such as pruning, gating, and clustering have been proposed in the literature [27]. In the joint probabilistic data association filter (JPDAF) [47], the infeasible association hypotheses are pruned away by a gating procedure at each time step, and then the remaining hypotheses are computed and combined in proportion to the corresponding hypotheses' probabilities. The probabilistic Multiple Hypothesis Tracking (PMHT) algorithm [135]–[137] assumes that the individual measurement to target association hypothesis is statistically independent with respect to each other. By adopting this assumption, PMHT avoids enumerating all measurement to target association hypotheses and either pruning or gating procedure is needed.

Recently, PF based approaches have been applied to the multiple target tracking problem. These approaches are able to perform well in general nonlinear and non-Gaussian systems. Vermaak *et al.* proposed a group of efficient algorithms to address the data association problem that arises due to unlabelled measurements and the dimensionality problem that arises due to the increased size of the state-space associated with multiple targets [48]. Hue *et al.* also developed a sequential Monte Carlo algorithm for multiple target tracking and data fusion [49]. Their algorithm is in spirit quite similar to the PMHT. However, instead of using EM to calculate the probabilities of the measurement to target associations as in PMHT, they adopted Gibbs sampling algorithm to compute the associations probabilities. Several authors combined PF with the Finite Set Statistics (FISST) approach to solve the problem of joint target initiation and estimation [141]–[143]. While this combination is theoretically advantageous, it demands intense computation in that a huge number of particles may be needed to explore different dimensional state spaces for target initiation. Doucet *et al.* applied the jump Markov system, a general framework in multiple target initiation and tracking [144]. Kreucher *et al.* developed a joint multi-target probability density (JMPD) filter, which simultaneously captures the uncertainty about target number, target state, and target identification [145]. This algorithm is based on a strategy in which the surveillance region is divided into a number of small cells. Ng *et al.* developed a measurement clustering algorithm to estimate the number of targets and accordingly execute the new target initiation, the disappeared target removal, or the persistent target

state update in a recursive fashion [146]. They then combined the a Particle filter algorithm with an efficient 2-D data assignment algorithm to deal with the data association problem and estimate the state of the persistent target.

7.2.2 PF-Based Multiple Target Tracking in Wireless Sensor Networks

As detailed in Section 2.4.2 of Chapter 2, several PF based multiple target tracking algorithms have also been proposed for multiple target tracking in wireless sensor networks [19], [101], [103], [104]. Motivated by these works, this chapter develops PF-JPDAF multiple target tracking algorithm. Differ from the algorithm developed in [19], PF-JPDAF solves the data association problem explicitly, and thus can be applied into the occasions that multiple targets are closely spaced (however, not too closely spaced that the targets can be regarded as one “super target” as in [19]). Another deviation from the leader-based multiple target tracking algorithm proposed in [101], PF-JPDAF adopts hierarchical sensor network processing architecture. At each time step, the leader node activates several selected sensing nodes, collects their measurements and updates the estimate of target state. Target state estimation using the information from multiple sensing nodes should improve the tracking accuracy. Moreover, to reduce the computation burden of the leader node and facilitate distributive multiple target tracking in wireless sensor networks, the whole sensor field is partitioned into adjoining regions. When the targets move into different regions in which these targets are well separated, a single target tracking algorithm, such as the PF-PDAF developed in Chapter 5, is switched over for each of the targets; only when the targets move in the same region, the multiple target tracking algorithm, PF-JPDAF is used (see the simulations in Section 7.5.2). Unlike MCMCDA algorithm developed in [103], [104], PF-JPDAF algorithm only uses the measurements acquired at the current time step for the target state estimation. Thus, PF-JPDAF only requires modest resources utilization when it is applied for on-line multiple target tracking in wireless sensor networks.

7.3 Problem Formulation of Multiple Target Tracking in Wireless Sensor Networks for the Development of PF-JPDAF Algorithm

This section formulates the problem of tracking multiple targets for the development of PF-JPDAF algorithm for wireless sensor networks. It describes the state-space model specified for the multiple target tracking in wireless sensor networks, defines the joint measurement to target association vector in the presence of multiple targets and clutter, and establishes the measurement likelihood in the presence of multiple targets.

7.3.1 State Space Model for Multiple target tracking

Throughout this chapter, it is assumed that the number of targets is known and fixed during the whole period of the tracking task. The number of targets is denoted as T , and $t = 1, \dots, T$ denotes one of the T targets. The state vector of the t -th target at the k -th time step is designated as $\mathbf{x}_{t,k}$. The joint state of T targets is the concatenation of the individual target state and designated as $\mathbf{X}_k = (\mathbf{x}_{1,k}, \dots, \mathbf{x}_{t,k}, \dots, \mathbf{x}_{T,k})$, $t = 1, \dots, T$. It is further assumed that the individual target evolves independently; therefore, the system model is composed of T partial equations with each corresponding to a target:

$$\mathbf{x}_{t,k} = \mathbf{f}_{t,k}(\mathbf{x}_{t,k-1}, \mathbf{v}_{t,k}) \quad t = 1, \dots, T \quad (7.1)$$

where $\mathbf{f}_{t,k}$ can be nonlinear and non-Gaussian function. $\mathbf{v}_{t,k}$ is the process noise which is assumed to be Gaussian with zero-mean and the known covariance matrix $\mathbf{Q}_{t,k}$, i.e., $\mathbf{v}_{t,k} \sim N(0, \mathbf{Q}_{t,k})$. It is also assumed $\mathbf{v}_{t,k}$ is independent over different targets.

As in the previous chapters, it is also assumed that each target evolves according to the near constant velocity (CV) model. Equation 7.1 then becomes

$$\mathbf{x}_{t,k} = \mathbf{A}_{t,k} \mathbf{x}_{t,k-1} + \mathbf{v}_{t,k} \quad t = 1, \dots, T \quad (7.2)$$

where $\mathbf{A}_{t,k}$ is the state transition matrix for the t -th target at the k -th time step and it remains the same as in Chapter 4 (refer to Equation 4.4).

We assume that there are N_s sensing nodes within a sensor cluster participate in the tracking task at every time step. $n = 1, \dots, N_s$ is denoted one of the N_s sensing nodes. The full set of measurements obtained at the n -th sensing node is designated as $\mathbf{Z}_k^n = (\mathbf{z}_{1,k}^n, \dots, \mathbf{z}_{j,k}^n, \dots, \mathbf{z}_{l_k^n,k}^n)$, $j = 1, \dots, l_k^n$. l_k^n is the total number of measurements acquired by the n -th sensing node at the k -th time step. In general, l_k^n is a random variable itself. The measurements set \mathbf{Z}_k^n is comprised of the measurements generated by the targets (the number is represented by $l_{T,k}^n$) and the measurements generated by the clutter (the number is represented by $l_{C,k}^n$). It is apparent that $l_k^n = l_{T,k}^n + l_{C,k}^n$. The measurements in \mathbf{Z}_k^n are assumed to be independent of each other at the n -th sensing node, and independent of

those at the other sensing nodes. The cumulative measurements acquired at the n -th sensing node from the initial time step to the k -th time step is designated as $\mathbf{Z}_{0:k}^n = (\mathbf{Z}_0^n, \dots, \mathbf{Z}_k^n)$. At time step k , the concatenated measurements over all N_s sensing nodes that are involved in the tracking task is denoted as $\mathbf{Z}_k = (\mathbf{Z}_k^1, \dots, \mathbf{Z}_k^n, \dots, \mathbf{Z}_k^{N_s})$, $n = 1, \dots, N_s$, and accordingly, the cumulative measurement over all N_s sensing nodes from the initial time step to the k -th time step is denoted as $\mathbf{Z}_{0:k} = (\mathbf{Z}_0, \dots, \mathbf{Z}_k)$.

For the target originating measurements, they have been defined in the previous chapters. We rewrite them as follows:

$$\left| \mathbf{z}_{j,k}^n \right| = \frac{S_{t,k}}{\|\boldsymbol{\rho}_{t,k} - \mathbf{r}_k^n\|^2} + \mathbf{n}_k^n \quad t \in (1, \dots, T), \quad j \in (1, \dots, l_k^n) \quad (7.3)$$

where $S_{t,k}$ is the intensity of the acoustic signal generated by the t -th target at time step k . \mathbf{n}_k^n is the measurement noise at the n -th sensing node and it is assumed to be Gaussian with zero mean and known covariance matrix \mathbf{R}_k^n , i.e. $\mathbf{n}_k^n \sim N(0, \mathbf{R}_k^n)$. It is also assumed that \mathbf{n}_k^n is independent over different sensing nodes and also not correlated with the process noise $\mathbf{v}_{t,k}$. $\boldsymbol{\rho}_{t,k}$ and \mathbf{r}_k^n are the position coordinates of the t -th target and the n -th sensing node at time step k , respectively. It is assumed throughout this chapter that all sensing nodes are static.

The clutter originated measurements are assumed to be independent and uniformly distributed over the observation space V of a sensing node with the probability:

$$P_{0,k}^n(\mathbf{z}_{j,k}^n) = p(\mathbf{z}_{j,k}^n \mid \mathbf{z}_{j,k}^n \text{ is the clutter originated measurement}) = \frac{1}{V} \quad j \in (1, \dots, l_{C,k}^n) \quad (7.4)$$

where the first subscript 0 in $P_{0,k}^n$ refers to the clutter and the superscript n refers to the n -th sensing node. In the previous chapters, the assumption has already made that all sensing nodes have the same observation space V .

In target tracking, it is commonly assumed that the number of clutter originated measurements in the observation space V follows a Poisson probability mass function (pmf) given by [27]

$$P_F^n(l_{C,k}^n) = \exp(-\lambda V) \frac{(\lambda V)^{l_{C,k}^n}}{l_{C,k}^n!} \quad (7.5)$$

where λ is the clutter rate which is defined as the number of clutter originated measurements per unit volume of the observation space V of a sensing node.

The assumptions that have been made in both Section 4.2 of Chapter 4 and Section 5.2 of Chapter 5 are retained throughout this chapter. In addition, some assumptions are made here for the derivation of PF-JPDAF tracking algorithm: each of the targets can generate at most one measurement at a sensing node at a particular time step, but may go undetected by a sensing node; part or all of the measurements may be due to the clutter. The above assumptions are commonly made in the literature and adopting these assumptions will not lead to the losing of the generality of the PF-JPDAF algorithm.

7.3.2 Data Association and Measurement Likelihood

In the presence of multiple targets and clutter, the measurements obtained at an individual sensing node are normally unlabelled and it is necessary to assign the measurements to their originating targets. Therefore, we introduce a joint measurement to target association vector for each of the sensing nodes participating in the tracking task. It is assumed that this joint measurement to target association vector is independent over each sensing node.

At the k -th time step, for the n -th sensing node the joint measurement to target association vector is defined as

$$\theta_k^n = \bigcap_{j=1}^{l_k^n} \theta_{j,k}^n \quad (7.6)$$

where the individual component $\theta_{j,k}^n$ is a measurement and target pair, and it is defined as

$$\theta_{j,k}^n = \begin{cases} 0 & \text{if measurement } j \text{ is due to clutter} \\ t \in \{1, \dots, T\} & \text{if measurement } j \text{ is originated from target } t \end{cases} \quad (7.7)$$

Note that in the above equations, the superscript n refers to the n -th sensing node and the first subscript j refers to one of the l_k^n measurements.

For the n -th sensing node, the conditional probability density function of the above joint measurement to target association vector at time step k can be expressed as

$$\begin{aligned} p(\theta_k^n | \mathbf{Z}_{0:k}^n) &= p(\theta_k^n | \mathbf{Z}_k^n, \mathbf{Z}_{0:k-1}^n) \\ &= \frac{1}{c} p(\mathbf{Z}_k^n | \theta_k^n, \mathbf{Z}_{0:k-1}^n) P(\theta_k^n) \end{aligned} \quad (7.8)$$

where c is the normalization constant. The conditioning on the total measurement number l_k^n is implicit in the measurement to target association vector θ_k^n . $P(\theta_k^n)$ is the prior probability of the joint measurement to target association vector and will be defined later. The first term in Equation 7.8, $p(\mathbf{Z}_k^n | \theta_k^n, \mathbf{Z}_{0:k-1}^n)$ can be factorized as follows

$$\begin{aligned} p(\mathbf{Z}_k^n | \theta_k^n, \mathbf{Z}_{0:k-1}^n) &= \prod_{j=1}^{l_k^n} p(\mathbf{z}_{j,k}^n | \theta_{j,k}^n, \mathbf{Z}_{0:k-1}^n) \\ &= \prod_{j \in g_0^n} P_{0,k}^n(\mathbf{z}_{j,k}^n) \cdot \prod_{j \in g_t^n} p_t^n(\mathbf{z}_{j,k}^n | \mathbf{Z}_{0:k-1}^n) \end{aligned} \quad (7.9)$$

where $g_0^n = \{ \theta_{j,k}^n = 0, j \in \{1, \dots, l_k^n\} \}$ and $g_t^n = \{ \theta_{j,k}^n \neq 0, j \in \{1, \dots, l_k^n\} \}$ stand for the subsets of the measurement indices corresponding to the clutter originating measurements and the targets originating measurements acquired by the n -th sensing node, respectively. $P_{0,k}^n(\mathbf{z}_{j,k}^n)$ is the probability distribution of the clutter originating measurements and it has been defined in Equation 7.4. $p_t^n(\mathbf{z}_{j,k}^n | \mathbf{Z}_{0:k-1}^n)$ is the predictive likelihood and it can be computed by

$$p_t^n(\mathbf{z}_{j,k}^n | \mathbf{Z}_{0:k-1}^n) = \int p_t^n(\mathbf{z}_{j,k}^n | \mathbf{x}_{t,k}) p_t^n(\mathbf{x}_{t,k} | \mathbf{Z}_{0:k-1}^n) d\mathbf{x}_{t,k} \quad (7.10)$$

In the above equation, $p_t^n(\mathbf{x}_{t,k} | \mathbf{Z}_{0:k-1}^n)$ is the probability density function of the predicted state of the t -th target at the k -th time step and it is obtained from the prediction step of the Bayesian recursive equations (refer to Chapter 4 and repeated in Equation 7.15).

$p_t^n(\mathbf{z}_{j,k}^n | \mathbf{x}_{t,k})$ is the measurement likelihood of the j -th measurement at the n -th sensing node regarding the t -th target. It can be computed by

$$p_t^n(\mathbf{z}_{j,k}^n | \mathbf{x}_{t,k}) = \frac{1}{\sqrt{|2\pi\mathbf{R}_k^n|}} \exp\left[-\frac{1}{2}(\mathbf{z}_{j,k}^n - \mathbf{H}_k^n \mathbf{x}_{t,k})^T (\mathbf{R}_k^n)^{-1} (\mathbf{z}_{j,k}^n - \mathbf{H}_k^n \mathbf{x}_{t,k})\right] \quad (7.11)$$

where \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 of the measurement function. In PF-JPDAF, Equations 7.10 and 7.11 are calculated by particles. Details will be discussed in the next section.

Substituting Equations 7.4 and 7.9 into Equation 7.8, the probability density function of the joint measurement to target association vector at time step k becomes

$$p(\theta_k^n | \mathbf{Z}_{0:k}^n) = \frac{1}{c} (V)^{-l_{c,k}^n} \cdot \prod_{j \in g_t^n} p_{\theta_{j,k}^n}^n(\mathbf{z}_{j,k}^n | \mathbf{Z}_{0:k-1}^n) P(\theta_k^n) \quad (7.12)$$

where $P(\theta_k^n)$, the prior probability of the joint measurement to target association vector takes the following form [47]

$$P(\theta_k^n) = \frac{l_{c,k}^n!}{l_k^n!} P_F^n(l_{c,k}^n) \prod_{t=1}^T (P_d)^{\delta_t^n} (1 - P_d)^{1 - \delta_t^n} \quad (7.13)$$

where δ_t^n is the target detection indicator, and $\delta_t^n = 1$ indicates that the t -th target is being detected at the n -th sensing node. $P_F^n(l_{c,k}^n)$ is the probability distribution of the number of clutter originated measurements and it has been defined in Equation 7.5. P_d denotes the probability of the target being detected or the detection rate. As in Chapter 5, in this chapter we still assume P_d is time invariant and takes the same value across all sensing nodes.

Substituting Equations 7.5 and 7.13 into Equation 7.12, the conditional probability density function of the joint measurement to target association vector for the n -th sensing node at the k -th time step can now be written as

$$p(\theta_k^n | \mathbf{Z}_{0:k}^n) = \frac{1}{c} (\lambda)^{l_{c,k}^n} \prod_{t=1}^T (P_d)^{\delta_t^n} (1 - P_d)^{1 - \delta_t^n} \prod_{j \in g_t^n} p_t^n(\mathbf{z}_{j,k}^n | \mathbf{Z}_{0:k-1}^n) \quad (7.14)$$

where c is the normalization constant.

Based on the above derived probability density function of the joint measurement to target association vector, next section will present the general JPDAF methodology in the context of wireless sensor networks and then develop the PF-JPDAF algorithm by adopting particles to represent the probability density functions of the target state.

7.4 The Design of PF-JPDAF

The JPDAF is the well-know methodology for multiple target tracking. The original formulation of JPDAF assumes linear and Gaussian systems [47]. Recently, several authors use particles to represent the probability density function of the target state, and thus extend the original JPDAF to the general nonlinear and non-Gaussian systems [48], [129]. We adopt a similar approach and further extends it to multiple target tracking in wireless sensor networks. The PF-JPDAF algorithm developed in this section takes into account the unique characteristics of wireless sensor networks, and incorporates the hierarchical sensor network architecture and sensing nodes selection scheme to achieve distributive tracking of multiple targets under measurement origin uncertainty in wireless sensor networks. This section starts with an introduction of the methodology of general JPDAF and then details the development of PF-JPDAF for the multiple target tracking in wireless sensor networks.

7.4.1 General Methodology of JPDAF

JPDAF does not manipulate the probability density function of the joint target state $p(\mathbf{X}_k | \mathbf{Z}_{0:k})$, where $\mathbf{X}_k = (\mathbf{x}_{1,k}, \dots, \mathbf{x}_{t,k}, \dots, \mathbf{x}_{T,k})$, $t = 1, \dots, T$ directly. In contrast, it recursively updates the marginal probability density function of each individual target state, i.e. $p(\mathbf{x}_{t,k} | \mathbf{Z}_{0:k})$, $t = 1, \dots, T$ through the recursive Bayesian estimation as follows:

Prediction step:

$$p(\mathbf{x}_{t,k} | \mathbf{Z}_{0:k-1}) = \int p(\mathbf{x}_{t,k} | \mathbf{x}_{t,k-1}) p(\mathbf{x}_{t,k-1} | \mathbf{Z}_{0:k-1}) d\mathbf{x}_{t,k-1} \quad t = 1, \dots, T \quad (7.15)$$

Filtering step:

$$p(\mathbf{x}_{t,k} | \mathbf{Z}_{0:k}) \propto p(\mathbf{Z}_k | \mathbf{x}_{t,k}) p(\mathbf{x}_{t,k} | \mathbf{Z}_{0:k-1}) \quad t=1, \dots, T \quad (7.16)$$

In JPDAF, the prediction step is preformed independently for each of the T targets. However, the filtering step cannot be performed independently for each target. This is because the measurement likelihood for the t -th target, $p(\mathbf{Z}_k | \mathbf{x}_{t,k})$, $t=1, \dots, T$ in Equation 7.16 cannot be computed independently and explicitly for each target due to the data association ambiguity (i.e. we do not know which measurement originated from which target). JPDAF tackles this problem by performing a *soft assignment* for each component (i.e. each measurement and target pair) in the joint measurement to target association vector according to the corresponding probabilities of these components. The complete procedure of the above soft assignment is detailed as follows.

Similar to the single target tracking algorithms developed in the previous chapters, the multiple target tracking algorithm developed in this chapter also adopts the hierarchical sensor network architecture and sensor nodes clustering to achieve the distributive tracking of multiple targets. The following derivations assume the tracking task takes place in one sensor cluster; at every time step over the whole tracking period, the cluster leader activates a set of N_s sensing nodes (the number and individual sensing node of this set of N_s sensing nodes may vary from time step to time step), collects measurements from these sensing nodes, solves data association problem and updates the estimate of each target state. It needs to emphasize here that the PF-JPDAF algorithm developed in this section is readily extendable to its distributive counterpart (i.e. the distributive PF-JPDAF) to track multiple targets move over a series of sensor clusters just as the distributive PF-PDAF developed in Chapter 6.

At the k -th time step, the measurement likelihood for the t -th target, $p(\mathbf{Z}_k | \mathbf{x}_{t,k})$ in the filtering step (Equation 7.16) is factorized over N_s sensing nodes and can be written as follows:

$$p(\mathbf{Z}_k | \mathbf{x}_{t,k}) = \prod_{n=1}^{N_s} \left[\sum_j^{I_k^n} (\beta_{j,t}^n)_k p_t^n(\mathbf{z}_{j,k}^n | \mathbf{x}_{t,k}) \right] \quad t=1, \dots, T \quad (7.17)$$

where $(\beta_{j,t}^n)_k$ is the probability that the j -th measurement obtained at the n -th sensing node is generated by the t -th target at the k -th time step. It is necessary to emphasize that $p_t^n(\mathbf{z}_{j,k}^n | \mathbf{x}_{t,k})$ in Equation 7.17 is the measurement likelihood of the j -th measurement at the n -th sensing node regarding the t -th target; $p(\mathbf{Z}_k | \mathbf{x}_{t,k})$ is the mixture measurement likelihood for the t -th target that span over all sensing nodes and all measurements acquired by these sensing nodes.

Now the task is to compute the association probability $(\beta_{j,t}^n)_k$, where $n = 1, \dots, N_s$ ranges over all participating sensing nodes, $j = 1, \dots, l_k^n$ ranges over all measurements obtained at the n -th sensing node, $t = 1, \dots, T$ ranges over all targets. In JPDAF, the association probability $(\beta_{j,t}^n)_k$ is computed by summing over the marginal probabilities of the corresponding components (measurement and target pairs) as follows

$$(\beta_{j,t}^n)_k = \sum_{\theta_k^n \in (\tilde{\lambda}_{j,t}^n)_k} p(\theta_k^n | \mathbf{Z}_{0:k}^n) \quad (7.18)$$

where $(\tilde{\lambda}_{j,t}^n)_k$ is the set of all valid components (measurements and targets pairs) that are over all measurements $j = 1, \dots, l_k^n$ and all targets $t = 1, \dots, T$ for the n -th sensing node during the k -th time step. The probability density function of the joint measurement to target association vector $p(\theta_k^n | \mathbf{Z}_{0:k}^n)$ has been computed in Equation 7.14.

To illustrate the above JPDAF strategy, Table 7.1 exemplifies a tracking scenario at one particular time step. It consists of two targets (T1 and T2) and three measurements (1, 2 and 3). The first column of Table 7.1 is the index of all feasible components in the measurement to target association vector. In the second column, the numbers 1, 2 and 3 refer to the measurements that are assigned to the target T1 or T2 and the number 0 means that there is no measurement assigned to a given target. The third column calculates each probability density function of the measurement to target association vector for the given measurements and targets pairs based on Equation 7.14, where $g_{j,t} = p(\mathbf{z}_{j,k}^n | \mathbf{Z}_{0:k-1}^n)$ is the predictive measurement likelihood and will be derived in the next subsection.

Using Table 7.1, $(\beta_{j,t}^n)_k$ can be calculated as follows. For target T1 (note $\beta_{0,1}$ means there are no measurements assigned to target T1):

$$\begin{aligned}
\beta_{0,1} &= P_1 + P_2 + P_3 + P_4 \\
\beta_{1,1} &= P_5 + P_6 + P_7 \\
\beta_{2,1} &= P_8 + P_9 + P_{10} \\
\beta_{3,1} &= P_{11} + P_{12} + P_{13}
\end{aligned} \tag{7.19}$$

Similarly, for target T2:

$$\begin{aligned}
\beta_{0,2} &= P_1 + P_5 + P_8 + P_{11} \\
\beta_{1,2} &= P_2 + P_9 + P_{12} \\
\beta_{2,2} &= P_3 + P_6 + P_{13} \\
\beta_{3,2} &= P_4 + P_7 + P_{10}
\end{aligned} \tag{7.20}$$

Table 7.1 The calculation of each component in the joint measurement to target association vector

Index	Targets		$p(\theta_k^n \mathbf{Z}_{0:k}^n)$
	T1	T2	
1	0	0	$(1 - P_d)^2 \lambda^3$
2	0	1	$g_{1,2} P_d (1 - P_d) \lambda^2$
3	0	2	$g_{2,2} P_d (1 - P_d) \lambda^2$
4	0	3	$g_{3,2} P_d (1 - P_d) \lambda^2$
5	1	0	$g_{1,1} P_d (1 - P_d) \lambda^2$
6	1	2	$g_{1,1} g_{2,2} P_d^2 \lambda$
7	1	3	$g_{1,1} g_{3,2} P_d^2 \lambda$
8	2	0	$g_{2,1} P_d (1 - P_d) \lambda^2$
9	2	1	$g_{2,1} g_{1,2} P_d^2 \lambda$
10	2	3	$g_{2,1} g_{3,2} P_d^2 \lambda$
11	3	0	$g_{3,1} P_d (1 - P_d) \lambda^2$
12	3	1	$g_{3,1} g_{1,2} P_d^2 \lambda$
13	3	2	$g_{3,1} g_{2,2} P_d^2 \lambda$

The above process essentially enumerates all possible components (measurement and target pairs) in the joint measurement to target association vector for the purpose of computing the association probability $(\beta_{j,t}^n)_k$. However, the above process may lead to very heavy computation burden when the number of targets and measurements increases.

Hence, some pruning and gating techniques need to be applied. Refer to the next section for a discussion.

7.4.2 The Implementation of PF-JPDAF

In the original JPDAF, the Kalman filter is employed to obtain the one step in advance prediction (Equation 7.15) and to calculate the measurement likelihood (Equation 7.11). And the mixture likelihood as defined in Equation 7.17 is collapsed into a single Gaussian, so that the Kalman filter update is also obtained for the filtering step as defined in Equation 7.16. In contrast to the original JPDAF, the PF-JPDAF uses particles' representation instead of using Gaussian representation. More specifically, for each of the T targets, the probability density function of the target state $p(\mathbf{x}_{t,k} | \mathbf{Z}_{0:k})$, $t=1, \dots, T$ is represented by a set of N particles with their weights, i.e. $\{w_{t,k}^i, \mathbf{x}_{t,k}^i\}_{i=1}^N$, $t=1, \dots, T$.

Assuming the probability density function of each target state, $p(\mathbf{x}_{t,k-1} | \mathbf{Z}_{0:k-1})$, $t=1, \dots, T$ at the $(k-1)$ -th time step is already known and represented by a set of particles $\{w_{t,k-1}^i, \mathbf{x}_{t,k-1}^i\}_{i=1}^N$, $t=1, \dots, T$, now the task is to compute $p(\mathbf{x}_{t,k} | \mathbf{Z}_{0:k})$, $t=1, \dots, T$, the probability density function of each target state for the k -th time step given the measurements obtained at the N_s sensing nodes. Similar to the generic PF algorithm, the transition prior is taken as the state proposal distribution and the T sets of new particles for the states of the T targets are generated from it:

$$\mathbf{x}_{t,k}^i \sim p(\mathbf{x}_{t,k}^i | \mathbf{x}_{t,k-1}^i), \quad t=1, \dots, T; \quad i=1, \dots, N \quad (7.21)$$

Note that other forms of the state proposal distribution can be developed, for example, the optimal proposal distribution discussed in Chapter 4. However, due to the data association ambiguity, such proposal distribution is not easy to construct; and even such proposal distribution is constructed, it might not directly lead to an improvement in the overall performance of the tracking algorithm. This issue will be discussed in Section 7.4.3.

After the new particles are generated, the predictive likelihood in Equation 7.10 can be straightforwardly approximated by these particles:

$$p_t^n(\mathbf{z}_{j,k}^n | \mathbf{Z}_{0:k-1}^n) \approx \sum_{i=1}^N p_t^n(\mathbf{z}_{j,k}^n | \mathbf{x}_{t,k}^i), \quad t=1, \dots, T \quad (7.22)$$

where the measurement likelihood $p_t^n(\mathbf{z}_{j,k}^n | \mathbf{x}_{t,k}^i)$ can be expressed as

$$p_t^n(\mathbf{z}_{j,k}^n | \mathbf{x}_{t,k}^i) = \frac{1}{\sqrt{|2\pi\mathbf{R}_k^n|}} \exp\left[-\frac{1}{2}(\mathbf{z}_{j,k}^n - \mathbf{H}_k^n \mathbf{x}_{t,k}^i)^T (\mathbf{R}_k^n)^{-1} (\mathbf{z}_{j,k}^n - \mathbf{H}_k^n \mathbf{x}_{t,k}^i)\right] \quad (7.23)$$

By substituting Equations 7.22, 7.23 into Equation 7.14, the conditional probability density function of the joint measurement to target association vector $p(\theta_k^n | \mathbf{Z}_{0:k}^n)$ is obtained. In turn, by using Equation 7.18, the measurement to target association probabilities $(\beta_{j,t}^n)_k$ can be computed. And finally, the measurement likelihoods for all T targets can be calculated based on Equation 7.17; and according to these likelihoods, the new importance weights for each particles set (each particles set corresponds to one target) can then be obtained as follows

$$w_{t,k}^i \propto w_{t,k-1}^i p_t(\mathbf{Z}_k | \mathbf{x}_{t,k}^i), \quad t = 1, \dots, T \quad (7.24)$$

Finally, for each of the T targets, a new particle set $\{w_{t,k}^i, \mathbf{x}_{t,k}^i\}_{i=1}^N$, $t = 1, \dots, T$ is obtained and then used to approximate the probability density function of the target state, i.e. $p(\mathbf{x}_{t,k} | \mathbf{z}_{0:k})$, $t = 1, \dots, T$ at the k -th time step. The complete PF-JPDAF is listed in Algorithm 7.1.

Algorithm 7.1: PF-JPDAF for multiple target tracking in wireless sensor networks

1. At the initial time step $k = 0$, for $t = 1, \dots, T$, draw particles $i = 1, \dots, N$

from the target's prior state $\mathbf{x}_{t,0}^i \sim p_t(\mathbf{x}_{t,0})$.

2. For $k = 1, 2, \dots$, do the following:

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

2.2 For $n = 1, \dots, N_s$, do

- For $t = 1, \dots, T$, $j = 1, \dots, l_k^n$ and $i = 1, \dots, N$ compute the predictive measurement likelihood

$$p_t^n(\mathbf{z}_{j,k}^n | \mathbf{x}_{t,k}^i) = \frac{1}{\sqrt{|2\pi\mathbf{R}_k^n|}} \exp\left[-\frac{1}{2}(\mathbf{z}_{j,k}^n - \mathbf{H}_k^n \mathbf{x}_{t,k}^i)^T (\mathbf{R}_k^n)^{-1} (\mathbf{z}_{j,k}^n - \mathbf{H}_k^n \mathbf{x}_{t,k}^i)\right]$$

$$p_t^n(\mathbf{z}_{j,k}^n | \mathbf{Z}_{0:k-1}^n) \approx \sum_{i=1}^N p_t^n(\mathbf{z}_{j,k}^n | \mathbf{x}_{t,k}^i)$$

- For $t = 1, \dots, T$, $j = 1, \dots, l_k^n$, enumerate all valid measurement to target association pairs at the n th sensing node to form the set $(\tilde{\lambda}_{j,k}^n)_k$.
- For $\theta_k^n \in (\tilde{\lambda}_{j,t}^n)_k$, compute the conditional probabilities of the joint measurement to target association vector

$$p(\theta_k^n | \mathbf{Z}_{0:k}^n) = \frac{1}{c} (\lambda_k^c)^{l_k^c} \prod_{t=1}^T (P_d)^{\delta_t^n} (1 - P_d)^{1 - \delta_t^n} \prod_{j \in g_t^n} p_t^n(\mathbf{z}_{j,k}^n | \mathbf{Z}_{0:k-1}^n)$$

- For $t = 1, \dots, T$, $j = 1, \dots, l_k^n$, compute the association probability $(\beta_{j,t}^n)_k$ by summing over the marginal probabilities of the corresponding components (measurement and target pairs) in the joint measurement to target association vector

$$(\beta_{j,t}^n)_k = \sum_{\theta_k^n \in (\tilde{\lambda}_{j,t}^n)_k} p(\theta_k^n | \mathbf{Z}_{0:k}^n)$$

2.3 For $t = 1, \dots, T$, $i = 1, \dots, N$ compute the measurement likelihood

$$p(\mathbf{z}_k | \mathbf{x}_{t,k}) = \prod_{n=1}^{N_s} \left[\sum_j^{l_k^n} (\beta_{j,t}^n)_k p_t^n(\mathbf{z}_{j,k}^n | \mathbf{x}_{t,k}^i) \right]$$

2.4 For $t = 1, \dots, T$, $i = 1, \dots, N$ compute and normalize the particle weights

$$w_{t,k}^i \propto w_{t,k-1}^i p_t(\mathbf{z}_k | \mathbf{x}_{t,k}^i)$$

$$\tilde{w}_{t,k}^i = w_{t,k}^i / \sum_{j=1}^N w_{t,k}^j$$

2.5 Resampling: for $t = 1, \dots, T$, $i = 1, \dots, N$

Multiply/suppress samples $\mathbf{x}_{t,k}^i$ with high/low importance weights $\tilde{w}_{t,k}^i$ to obtain N random samples approximately distributed according to $p(\mathbf{x}_{t,k} | \mathbf{Z}_{0:k})$;

Set $w_{k,k}^i = \tilde{w}_{t,k}^i = N^{-1}$.

In the above PF-JPDAF tracking algorithm, it is required to enumerate all valid measurement and target pairs. However, the number of valid measurement and target pairs increases exponentially with the increase of the number of targets in the sensor field. Therefore, it is necessary to reduce the number of such measurement and target pairs to a

feasible level. Gating is a popular approach to achieve this reduction [47]. For each target a validation region is constructed and only the measurements that fall within this validation region are considered as possible candidates to be associated with the particular target. However, this chapter doesn't explicitly implement the gating procedure. Instead, a *soft-gating scheme* is implemented in this chapter, in which, at each time step, only a set of sensing nodes closest to the targets are selected to participate the tracking task (refer to the simulations in Section 7.5 for details).

Similar to the distributive PF-PDAF developed in Chapter 6, by adopting the hierarchical sensor network architecture and the GMM model, the distributive PF-JPDAF has also been developed. The simulation illustrates the distributive PF-JPDAF algorithm for distributively tracking multiple targets in wireless sensor networks (refer to the tracking scenario depicted in Figure 7.10).

7.4.3 Target State Proposal Distribution

In the above PF-JPDAF algorithm, the transition prior is taken as the target state proposal distribution:

$$\pi(\mathbf{x}_{t,k} | \mathbf{x}_{t,k-1}, \mathbf{Z}_k) = p(\mathbf{x}_{t,k} | \mathbf{x}_{t,k-1}) \quad \text{for } t = 1, \dots, T \quad (7.25)$$

It leads to a straightforward approach in which the new particles are generated from the target dynamics. However, as pointed out in the previous chapters, such proposal distribution may lead to divergence of the whole algorithm since the state space is explored without taking account of any knowledge of the measurements. It is proved in the literature that the optimal proposal distribution that minimize the variance of the importance weights is in the following form [38]:

$$\begin{aligned} \pi(\mathbf{x}_{t,k} | \mathbf{x}_{t,k-1}, \mathbf{Z}_k) &= p(\mathbf{x}_{t,k} | \mathbf{x}_{t,k-1}, \mathbf{Z}_k) \\ &\approx p(\mathbf{Z}_k | \mathbf{x}_{t,k}) p(\mathbf{x}_{t,k} | \mathbf{x}_{t,k-1}) \quad \text{for } t = 1, \dots, T \end{aligned} \quad (7.26)$$

where $p(\mathbf{Z}_k | \mathbf{x}_{t,k})$ is the measurement likelihood that only conditions on the target state, and $p(\mathbf{x}_{t,k} | \mathbf{x}_{t,k-1})$ is the probability density function of target dynamics. However, for single and multiple target tracking under measurement origin uncertainty, it is generally not possible to obtain a closed-form expression for the above optimal proposal distribution due to the data association uncertainty.

To construct the proposal distributions other than the transition prior, one approach is to approximate the optimal target proposal distribution by defining a mixture state proposal: part of the new particles is sampled from the target dynamics (e.g. transition prior); other part of the new particles is sampled from a mixture in which each component of the mixture is attributed to one measurement and target pair at one sensing node. This mixture state proposal distribution is defined as follows:

$$\pi(\mathbf{x}_{t,k} | \mathbf{x}_{t,k-1}, \mathbf{Z}_k) = \eta p(\mathbf{x}_{t,k} | \mathbf{x}_{t,k-1}) + (1-\eta) \sum_{n=1}^{N_s} \sum_{j=1}^{l_k^n} \gamma_{n,j} \pi(\mathbf{x}_{t,k} | \mathbf{x}_{t,k-1}, \mathbf{z}_{j,k}^n) \quad t=1, \dots, T \quad (7.27)$$

where $0 \leq \eta \leq 1$ is the factor that balances the contributions of the transition prior and the mixtures to the entire proposal distribution. $\gamma_{n,j}$ ($n=1, \dots, N_s$; $j=1, \dots, l_k^n$) are the mixture weights and $\sum_{n=1}^{N_s} \sum_{j=1}^{l_k^n} \gamma_{n,j} = 1$. It is further assumed that the mixture component in Equation 7.27 takes the following form:

$$\pi(\mathbf{x}_{t,k} | \mathbf{x}_{t,k-1}, \mathbf{z}_{j,k}^n) \propto p_t^n(\mathbf{z}_{j,k}^n | \mathbf{x}_{t,k}) p(\mathbf{x}_{t,k} | \mathbf{x}_{t,k-1}) \quad (7.28)$$

where $p_t^n(\mathbf{z}_{j,k}^n | \mathbf{x}_{t,k})$ is the measurement likelihood of the j -th measurement which is acquired at the n -th sensing node with respect to the t -th target (Equation 7.11). Equation 7.28 can be further approximated by a Gaussian as follows:

$$\pi(\mathbf{x}_{t,k} | \mathbf{x}_{t,k-1}, \mathbf{z}_{j,k}^n) \approx N(\mathbf{x}_{t,k} | \hat{\mathbf{m}}_{t,k}^n, \hat{\boldsymbol{\Sigma}}_{t,k}^n) \quad (7.29)$$

where

$$\hat{\boldsymbol{\Sigma}}_{t,k}^n = \left[(\mathbf{Q}_{t,k})^{-1} + (\mathbf{H}_{t,k}^n)^T (\mathbf{R}^n)^{-1} \mathbf{H}_{t,k}^n \right]^{-1} \quad (7.30)$$

$$\hat{\mathbf{m}}_{t,k}^n = \hat{\boldsymbol{\Sigma}}_{t,k}^n \left[(\mathbf{Q}_{t,k})^{-1} \mathbf{A}_{t,k} \mathbf{x}_{t,k-1} + (\mathbf{H}_{t,k}^n)^T (\mathbf{R}^n)^{-1} (\mathbf{z}_{j,k}^n - \hat{\mathbf{c}}_{t,k}) \right]^{-1} \quad (7.31)$$

$$\hat{\mathbf{c}}_{t,k} = \mathbf{g}(\hat{\mathbf{x}}_{t,k}, \mathbf{r}^n) - \mathbf{H}_{t,k}^n \mathbf{A}_{t,k} \mathbf{x}_{t,k-1} \quad (7.32)$$

$$\mathbf{g}(\hat{\mathbf{x}}_{t,k}, \mathbf{r}^n) = \frac{S_{t,k}}{\| \mathbf{A}_{t,k} \mathbf{x}_{t,k-1} - \mathbf{r}^n \|^2} \quad (7.33)$$

where $\mathbf{Q}_{t,k}$ is the covariance matrix of the process noise of the t -th target at the k -th time step, $\mathbf{H}_{t,k}^n$ is the Jacobian evaluated at the predicted target state $\hat{\mathbf{x}}_{t,k} = \mathbf{A}_{t,k} \mathbf{x}_{t,k-1}$ and it is given as follows:

$$\mathbf{H}_{t,k}^n = (\mathbf{R}_k^n)^{-1} \times \begin{bmatrix} \frac{-2S_{t,k}(\hat{x}_{t,k} - X^n)}{\left[(\hat{x}_{t,k} - X^n)^2 + (\hat{y}_{t,k} - Y^n)^2 \right]^2} & 0 & \frac{-2S_{t,k}(\hat{y}_{t,k} - Y^n)}{\left[(\hat{x}_{t,k} - X^n)^2 + (\hat{y}_{t,k} - Y^n)^2 \right]^2} & 0 \end{bmatrix} \quad (7.34)$$

where $\hat{\mathbf{x}}_{t,k} = \begin{bmatrix} \hat{x}_{t,k} & \hat{y}_{t,k} \end{bmatrix}$ is the predicted target position and $\mathbf{r}^n = \begin{bmatrix} X^n & Y^n \end{bmatrix}$ is the position of the n -th sensing node. $S_{t,k}$ is the acoustic signal energy of the t -th target.

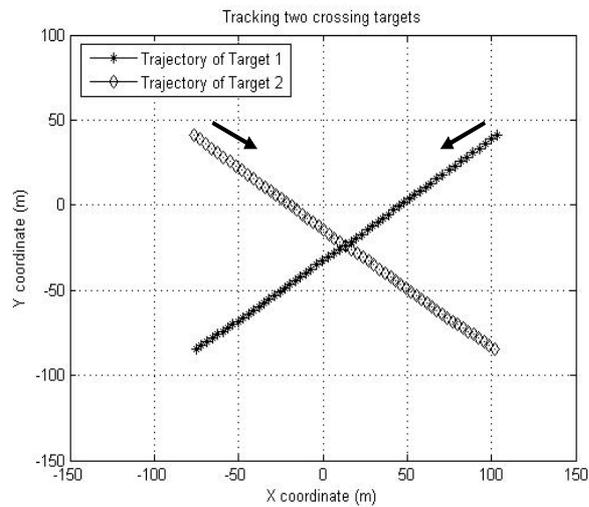
However, the performance of the above mixture state proposal greatly depends on the problem to be solved at hand. We found that the PF-JPDAF adopting the above mixture proposal distribution does not outperform the PF-JPDAF adopting the transition prior as the proposal distribution. This is because the measurements used to calculate the mixture components include both the target and clutter originating measurements; although parts of the new particles which are generated by the target originating measurements can lead PF-JPDAF to explore the areas in the state-space closing to the positions of the targets, the other parts of new particles which are generated by the clutter originating measurements may mislead PF-JPDAF to explore the areas in the state space that may be even far away from the positions of targets. Moreover, the PF-JPDAF adopts the mixture proposal distribution introduces extra computation burdens in the wireless sensor networks.

7.5 Simulations

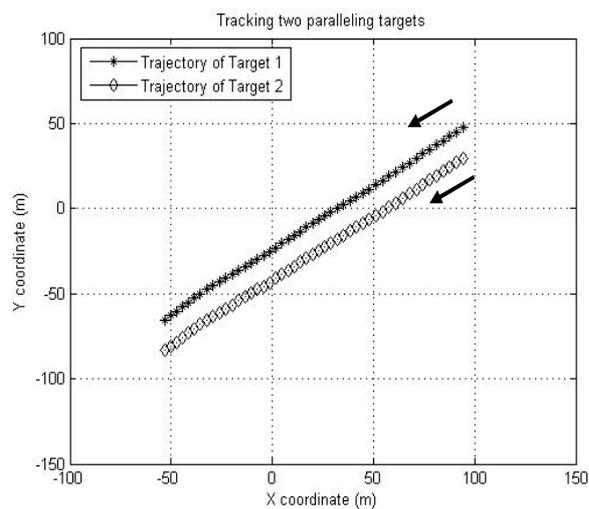
To evaluate the performance of the PF-JPDAF algorithm developed in this chapter, extensive simulations have been conducted on two synthesized tracking scenarios as depicted in Figure 7.1: the first scenario is two crossing targets tracking; and the second scenario is two close-spaced paralleling targets tracking.

For the first tracking scenario, two different tracking schemes have been implemented and evaluated. The first scheme assumes that all sensor nodes form one single sensor cluster and a single PF-JPDAF algorithm is executed for target state estimation. The second scheme is a distributive tracking scheme, it partitions the whole sensor field into several smaller regions with each region occupied by a sensor cluster; and in each sensor cluster, PF-PDAF algorithm or PF-JPDAF algorithm is performed for target state estimation. The simulation setup for the above multiple target tracking scenarios is summarized as follows.

PF-JPDAF algorithm is tested with the different settings of clutter and detection rates for both crossing targets tracking and close-spaced paralleling targets tracking. In the simulations, 50 independent Monte Carlo runs are conducted for each setting; and for an individual Monte Carlo run, the ground truth of the target remains unchanged while the target originated measurements are regenerated according to Equation 7.3 at each time step. The magnitude of target signal is assumed to be time-invariant and takes the same value for each target, i.e. $S_{t,k} = 5000 \quad t = 1, 2$. The measurement noise is set to $\mathbf{n}_k^n \sim N(0, 1)$ for all sensing nodes at each time step.



(a)



(b)

Figure 7.1 Two synthesized multiple target tracking scenarios
 (a) Two crossing targets (b) Two close-spaced paralleling targets

Throughout this chapter, the assumption has been made that the measurement range of each sensing node is $200 m$. For an individual sensing node, the clutter originating measurements are simulated to be uniformly distributed in a square with the size of $400 m \times 400 m$ centered at the location of this sensing node. The number of clutter originating measurements follows the Poisson distribution as defined by Equation 7.5. The magnitudes of the clutter generated signal are set as the same as that of the target (i.e. 5000) and also corrupted by the Gaussian noise $N(0,1)$. By assuming the same magnitude for the signal generated by the target and clutter, the difficulty of recognizing the origins of the measurements is greatly increased; hence we can assess the PF-JPDAF algorithm's ability in effectively solving the data association problem.

In the simulation, the prior estimate of each target state is assumed to be Gaussian with the mean vector $\mathbf{x}_{0|0}$ and covariance matrix $\mathbf{P}_{0|0}$:

$$\mathbf{x}_{0|0} = \mathbf{x}_{truth} + \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{P}_{0|0} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (7.35)$$

where \mathbf{x}_{truth} is the ground truth of the target at time step $k=0$ (i.e. the initial position of the target). The particles number used in the PF-JPDAF algorithm is 1000.

As in the previous chapters, two types of RMSEs are used to assess the performance of PF-JPDAF algorithm: the RMSEⁿ which is computed by averaging over all time steps for each individual Monte Carlo run, and the RMSE_k which is computed by averaging over all Monte Carlo runs (i.e. 50 runs) for each time step. The mathematical definition of these two RMSEs can be found in Chapter 4 and are not repeated here.

7.5.1 Simulation Results of Tracking Two Crossing Targets

To evaluate the performance of PF-JPDAF algorithm in effectively tracking two crossing targets, two different sensing nodes deployment strategies are adopted. In the first deployment strategy (hereinafter named as *Layout 1*, Figure 7.2), total 24 sensing nodes are distributed covering the area in which the two targets will traverse. The distances between two sensing nodes are $40 m$ and $20 m$ in X- and Y- coordinates, respectively. During the whole period of the tracking task, at each time step, a set of eight sensing nodes are

selected to make acoustic measurements and transmit their measurement to the cluster leader (the cluster leader is not drawn in Figures 7.2). In the second deployment strategy (hereinafter named as *Layout 2*, Figure 7.3), in the area where two targets are far away, the sensing nodes are distributed along the roadside of each target; in the area where two targets will move closely, the sensing nodes are distributed to cover the area. During the time steps $k = 1 \sim 18$, a set of 20 sensing nodes (enclosed in the two black ellipses in Figure 7.3) are selected to make measurements and transmit their measurement to the cluster leader, the distance between two sensing nodes is 10 m ; during the time steps $k = 19 \sim 37$, a set of 16 sensing nodes (enclosed in a rectangle in Figure 7.3) are selected to make measurements and transmit their measurement to the cluster leader, the distances between two sensing nodes are 21 m and 13 m in X- and Y- coordinates, respectively; and during the time steps $k = 38 \sim 56$, a set of 20 sensing nodes (enclosed in the two blue ellipses in Figure 7.3) are selected to make measurements and transmit their measurement to the cluster leader, the distance between two sensing nodes is 10 m .

The sensing nodes selections in the above Layouts 1 and 2 are empirically decided for the PF-JPDAF algorithm to attain desirable tracking accuracy at the reasonable computation cost. However, the sensing nodes selection schemes developed in Chapter 6 can be extended and adopted in PF-JPDAF algorithm for multiple target tracking. It will involve extensive mathematical derivation and computation arising from the presence of multiple targets [152], [153]. Sensing nodes selection scheme for multiple target tracking will not be discussed in this thesis.

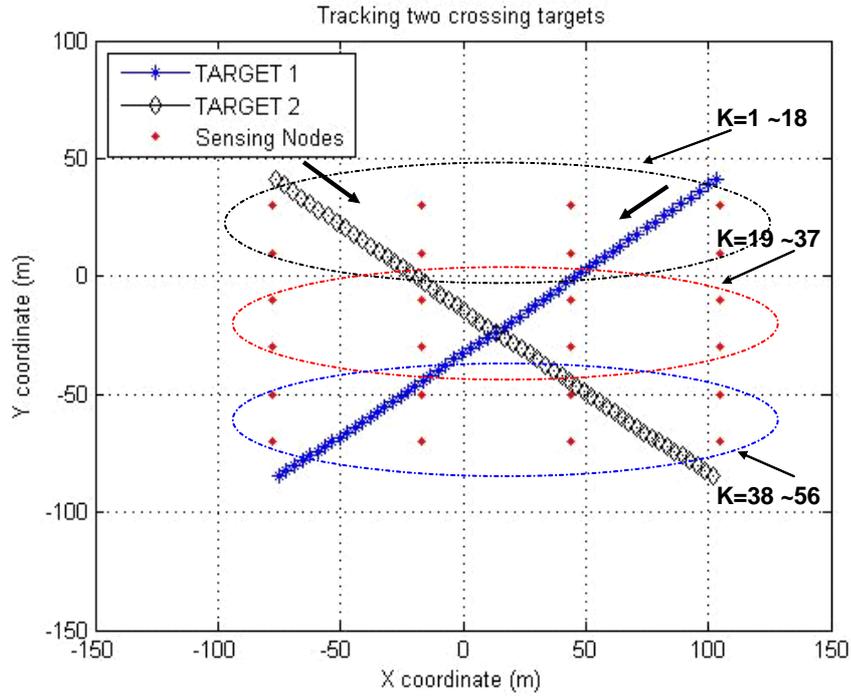


Figure 7.2 The sensing nodes deployment for tracking two crossing targets (**Layout 1**) (k is the time step and the sensing nodes enclosed in the ellipses denote the selected sensing nodes at each time step.)

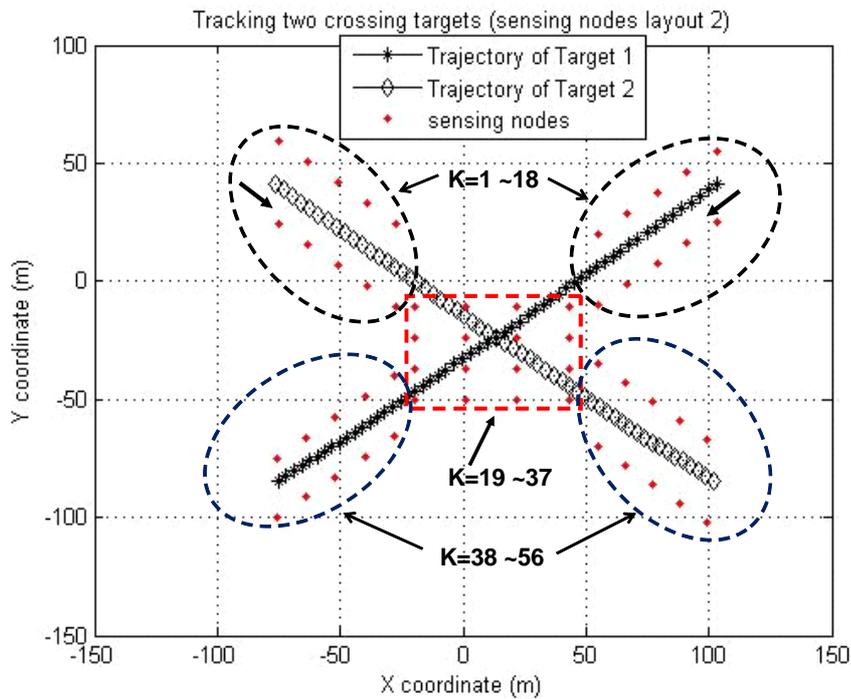


Figure 7.3 The sensing nodes deployment for tracking two crossing targets (**Layout 2**) (k is the time step and the sensing nodes enclosed in the ellipses denote the selected sensing nodes at each time step.)

Simulation results of PF-JPDAF using Layout 1

Figure 7.4 depicts the estimated trajectories of two crossing targets obtained by the PF-JPDAF algorithm using Layout 1 under four different sets of clutter and detection rates: $P_d = 1, C_d = 0$; $P_d = 1, C_d = 0.5$; $P_d = 0.9, C_d = 0$; and $P_d = 0.9, C_d = 0.5$. Note that the above results are averaged over 50 Monte Carlo runs. Figures 7.5 and 7.6 show the RMSE values of two crossing targets using Layout 1 under the settings of $P_d = 1, C_d = 0$ and $P_d = 0.9, C_d = 0.5$, respectively.

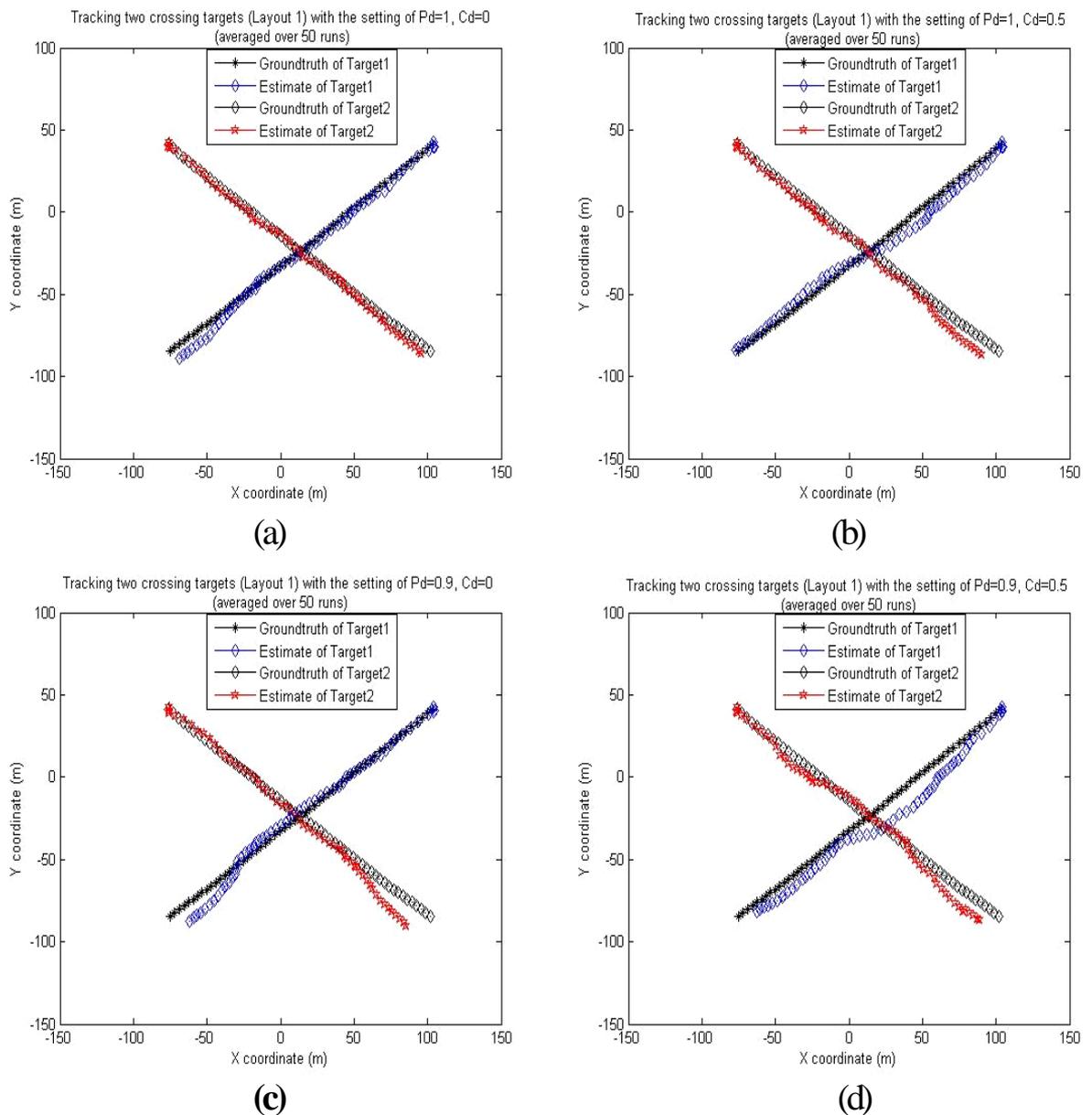


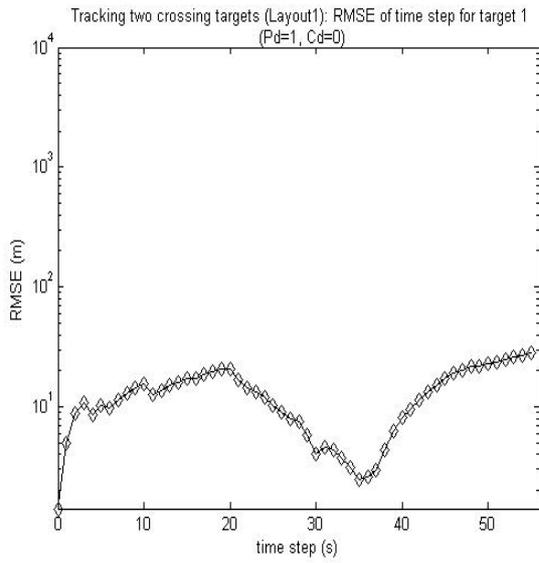
Figure 7.4 Estimated trajectories of two crossing targets under different settings of detection and clutter rates (**Layout 1**)

(a) $P_d = 1, C_d = 0$

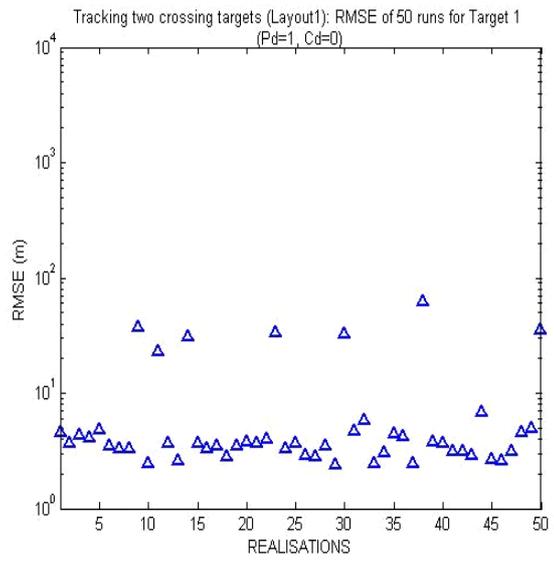
(b) $P_d = 1, C_d = 0.5$

(c) $P_d = 0.9, C_d = 0$

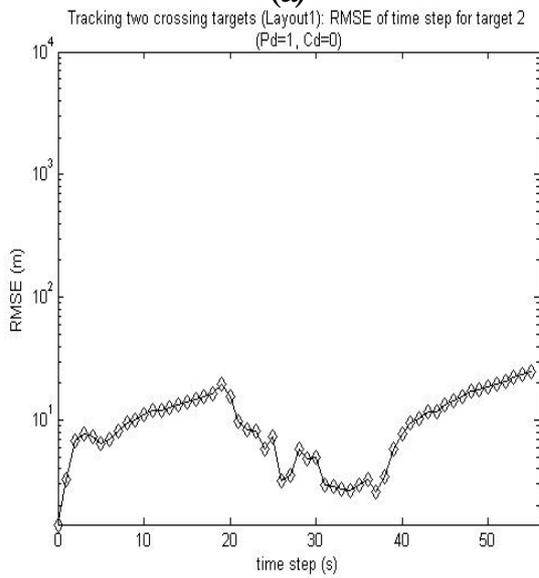
(d) $P_d = 0.9, C_d = 0.5$



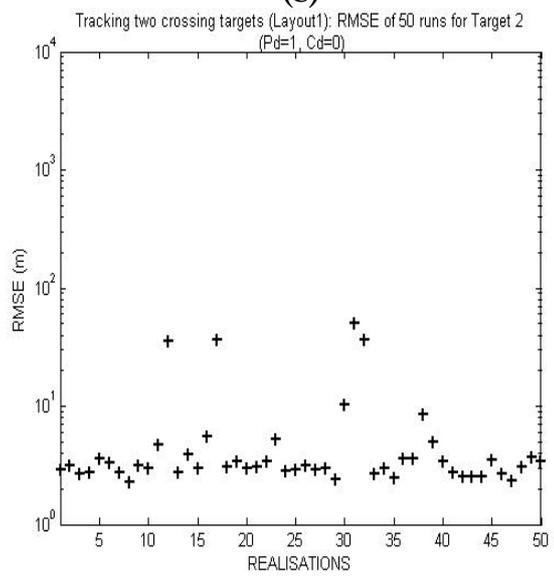
(a)



(b)



(c)



(d)

Figure 7.5 RMSE values of PF-JPDAF algorithm for tracking two crossing targets with the setting of $P_d = 1$, $C_d = 0$ (**Layout 1**)

(a) $RMSE_k$ of Target 1

(b) $RMSE^n$ of Target 1

(c) $RMSE_k$ of Target 2

(d) $RMSE^n$ of Target 2

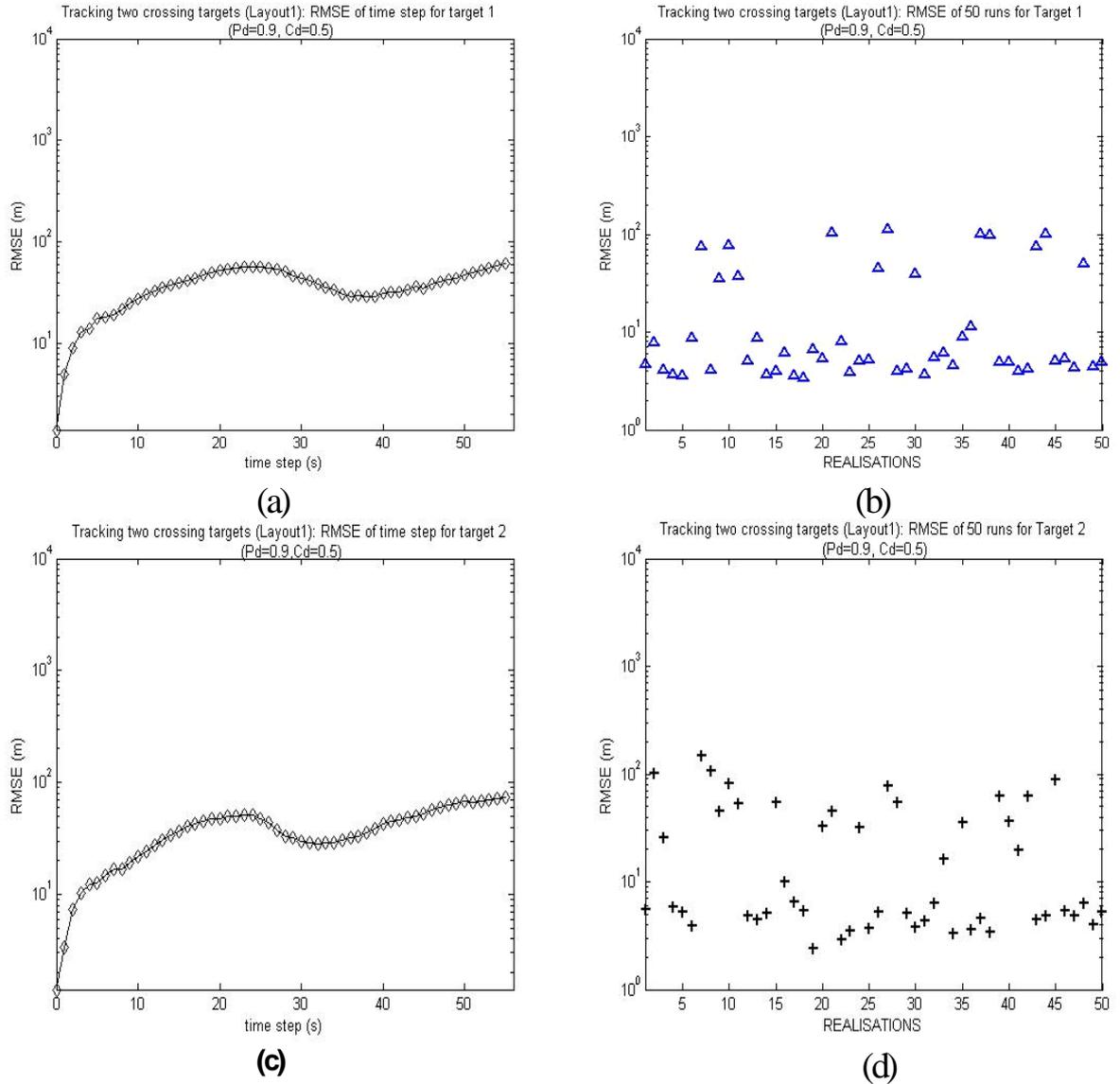


Figure 7.6 RMSE values of PF-JPDAF algorithm for tracking two crossing targets with the setting of $P_d = 0.9, C_d = 0.5$ (**Layout 1**)

- (a) $RMSE_k$ of Target 1
- (b) $RMSE^n$ of Target 1
- (c) $RMSE_k$ of Target 2
- (d) $RMSE^n$ of Target 2

From Figures 7.4, 7.5 and 7.6, it can be seen that when measurement origin uncertainty is low, the PF-JPDAF algorithm can track two crossing targets well; especially when there is no clutter and missed detections, i.e. $P_d = 1, C_d = 0$, the estimated trajectories of two targets almost fit the true trajectories (Figure 7.4 (a)) and the magnitudes of $RMSE^n$ values of two targets in most runs are less than 10 m (43 runs out of total 50 runs for target 1 and 45 runs out of total 50 runs for target 2. Refer to Figures 7.5 (b) and 7.5 (d)). However, the performance of PF-JPDAF algorithm is deteriorated when the clutter rate is

increased or the detection rate is decreased; when the measurement origin uncertainty changes to $P_d = 0.9, C_d = 0.5$, the estimated trajectories deviates from the true trajectories (Figure 7.4 (d)) and the magnitudes of RMSE values of two targets in many runs are larger than 50 m (13 runs out of total 50 runs for target 1 and 17 runs out of total 50 runs for target 2. Refer to Figures 7.6 (b) and 7.6 (d)). The relatively poor performance under higher clutter rates is because the sensing nodes in Layout 1 are sparsely deployed: some sensing nodes are laid far away from the both targets; and at these sensing nodes, the magnitude of clutter originated measurements might be larger than that of target originated measurements. This will increase the difficulty in solving the data association problem; and as a consequence, the deterioration on the overall performance of the PF-JPDAF algorithm can be expected.

Simulation results of PF-JPDAF using Layout 2

Figure 7.7 depicts the estimated trajectories of two crossing targets obtained by the PF-JPDAF algorithm using Layout 2 under four different sets of clutter and detection rates: $P_d = 1, C_d = 0$; $P_d = 1, C_d = 0.5$; $P_d = 0.9, C_d = 0$; and $P_d = 0.9, C_d = 0.5$. Note that these results are also averaged over 50 Monte Carlo runs. Figures 7.8 and 7.9 show the RMSE values of two crossing targets using Layout 2 under the settings of $P_d = 1, C_d = 0$ and $P_d = 0.9, C_d = 0.5$, respectively.

From Figures 7.7, 7.8 and 7.9, it can be seen that the PF-JPDAF algorithm using Layout 2 can track two crossing targets very well when the measurement origin uncertainty is low; and when the measurement origin uncertainty becomes large, the performance of the PF-JPDAF algorithm may be deteriorated. However, comparing Figures 7.7~7.9 with Figures 7.4~7.6, it can be seen that the PF-JPDAF algorithm using Layout 2 outperforms the PF-JPDAF algorithm using Layout 1. At the setting of $P_d = 0.9, C_d = 0.5$ (i.e. measurement origin uncertainty is large), for the PF-JPDAF algorithm adopting Layout 2, the RMSE values in 12 runs out of 50 runs exceed 50 m for target 2 and the RMSE values in all 50 runs are below 10 m for target 1. In contrast, for the PF-JPDAF algorithm adopting Layout 1, the RMSE values in 17 runs out of 50 runs exceed 50 m for target 2 and the RMSE values in 13 runs out of 50 runs exceed 50 m for target 1. The better performance of Layout 2 over Layout 1 inspired us to develop a distributive multiple target tracking scheme as described in the next section.

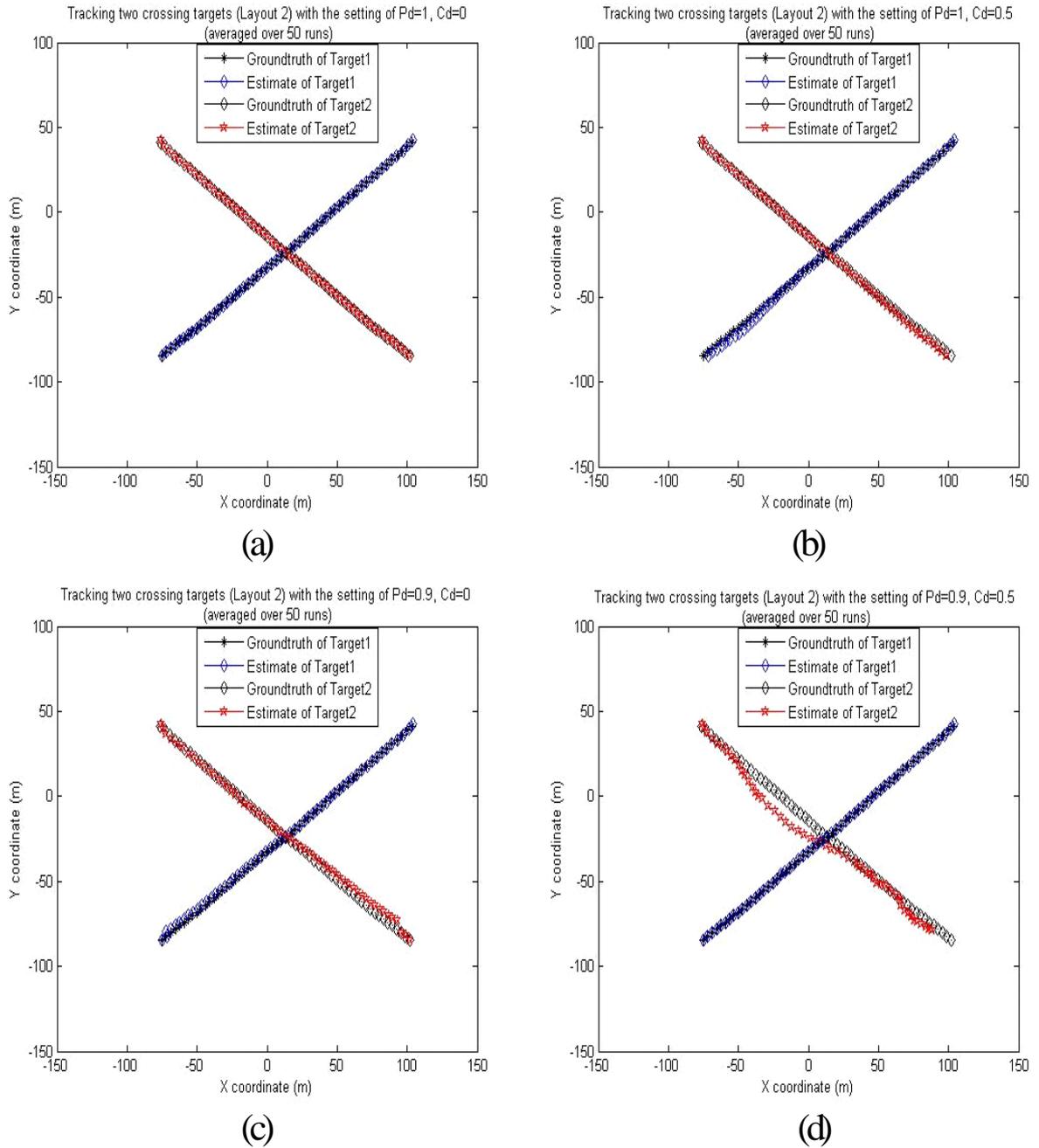


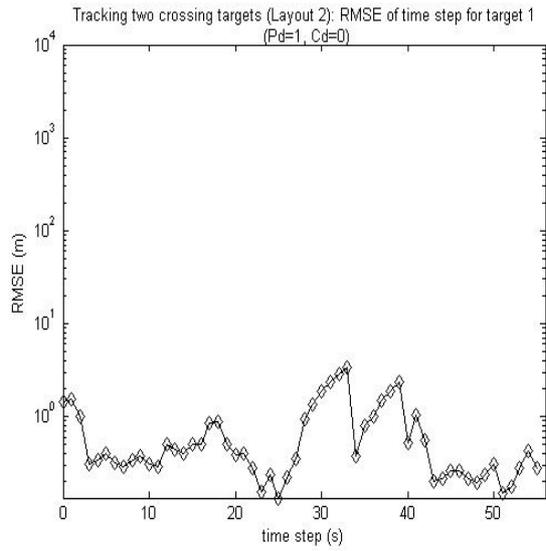
Figure 7.7 Estimated trajectories of two crossing targets under different settings of detection and clutter rates (**Layout 2**)

(a) $P_d = 1$, $C_d = 0$

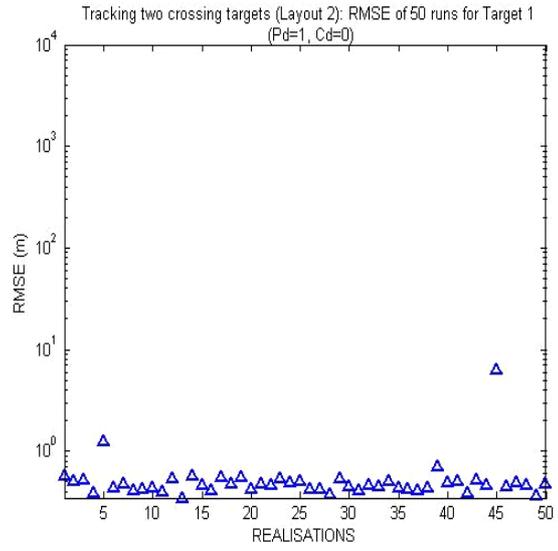
(b) $P_d = 1$, $C_d = 0.5$

(c) $P_d = 0.9$, $C_d = 0$

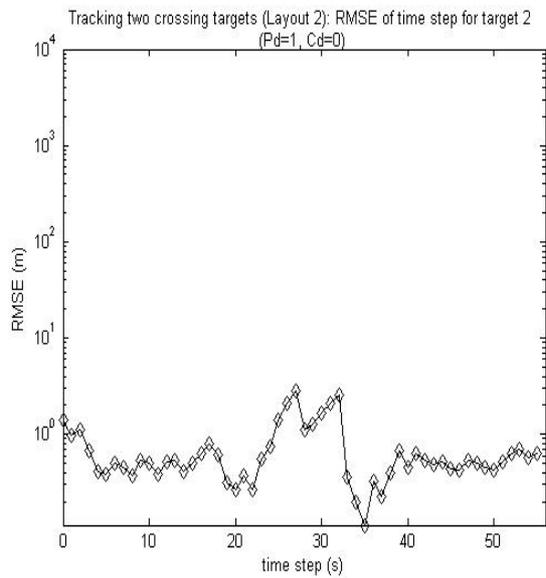
(d) $P_d = 0.9$, $C_d = 0.5$



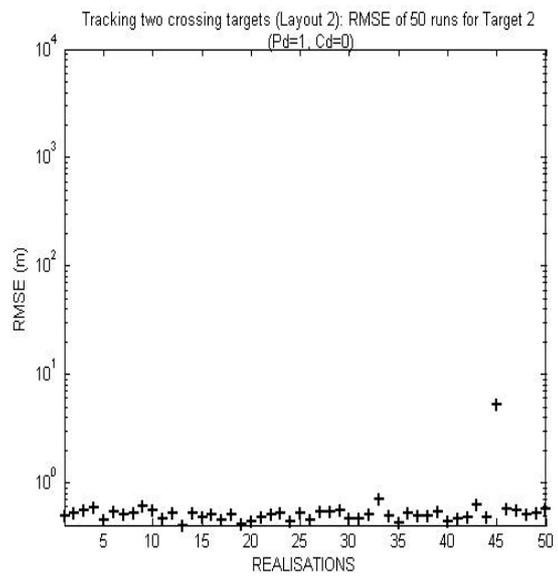
(a)



(b)



(c)



(d)

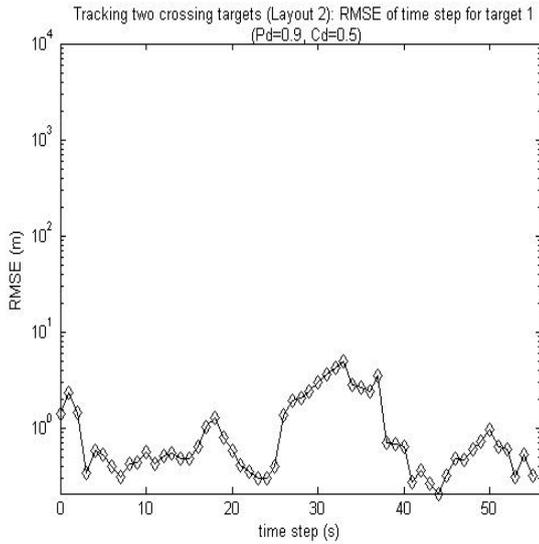
Figure 7.8 RMSE values of PF-JPDAF algorithm for tracking two crossing targets with the setting of $P_d = 1, C_d = 0$ (**Layout 2**)

(a) $RMSE_k$ of Target 1

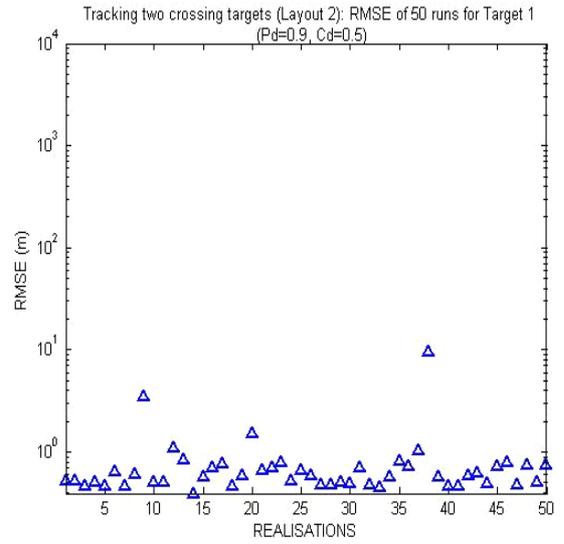
(b) $RMSE^n$ of Target 1

(c) $RMSE_k$ of Target 2

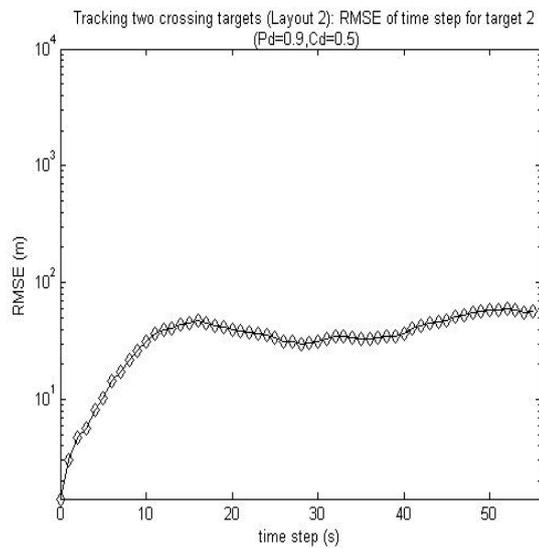
(d) $RMSE^n$ of Target 2



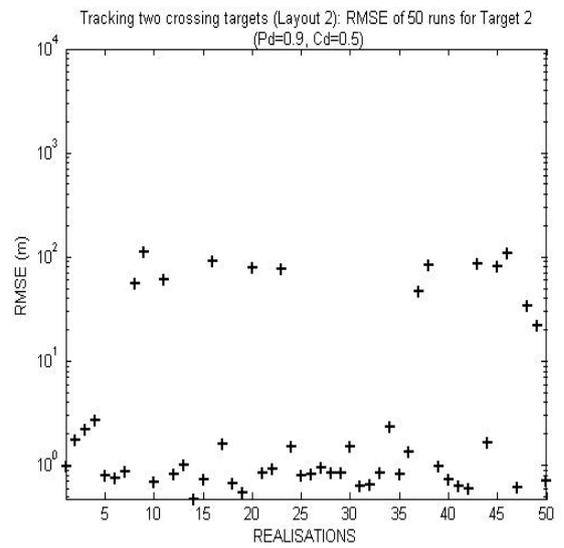
(a)



(b)



(c)



(d)

Figure 7.9 RMSE values of PF-JPDAF algorithm for tracking two crossing targets with the setting of $P_d = 0.9$, $C_d = 0.5$ (**Layout 2**)

(a) $RMSE_k$ of Target 1

(b) $RMSE^n$ of Target 1

(c) $RMSE_k$ of Target 2

(d) $RMSE^n$ of Target 2

7.5.2 Simulation Results of Distributively Tracking Two Crossing Targets

Figure 7.10 depicts the scenario of distributively tracking two crossing targets. The sensor field, the trajectories of two targets, the sensing nodes and the simulation set-up are all remained the same as in Layout 2 (Figure 7.3). However, instead of treating all the sensing nodes in the sensor field to form one sensor cluster, the whole sensor field in Figure 7.10 is divided into *five adjoining regions* and in each region several sensing nodes and one leader

node (the leader nodes are not drawn in Figure 7.10) form a sensor node cluster. Hence, there are total five sensor clusters. The above five adjoining regions are named as Region 1a, Region 2a, Region 1c, Region 2c and the Joint Region. There are 10 sensing nodes in each of the Region 1a, 2a, 1c and 2c and 16 sensing nodes in the Joint Region. Two targets start to move in Region 1a and Region 2a, respectively. In Region 1a and Region 2a, the single target tracking is performed separately for each target by using PF-PDAF algorithm developed in Chapter 5. When the two targets enter the Joint Region, the multiple target tracking is performed by using PF-JPDAF algorithm. After the two targets leaving the Joint Region and enter Region 1c and Region 2c, the single target tracking is performed again for each of them by using PF-PDAF algorithm. The GMM model developed in Chapter 6 is used for the propagation of the estimation results amongst sensor clusters.

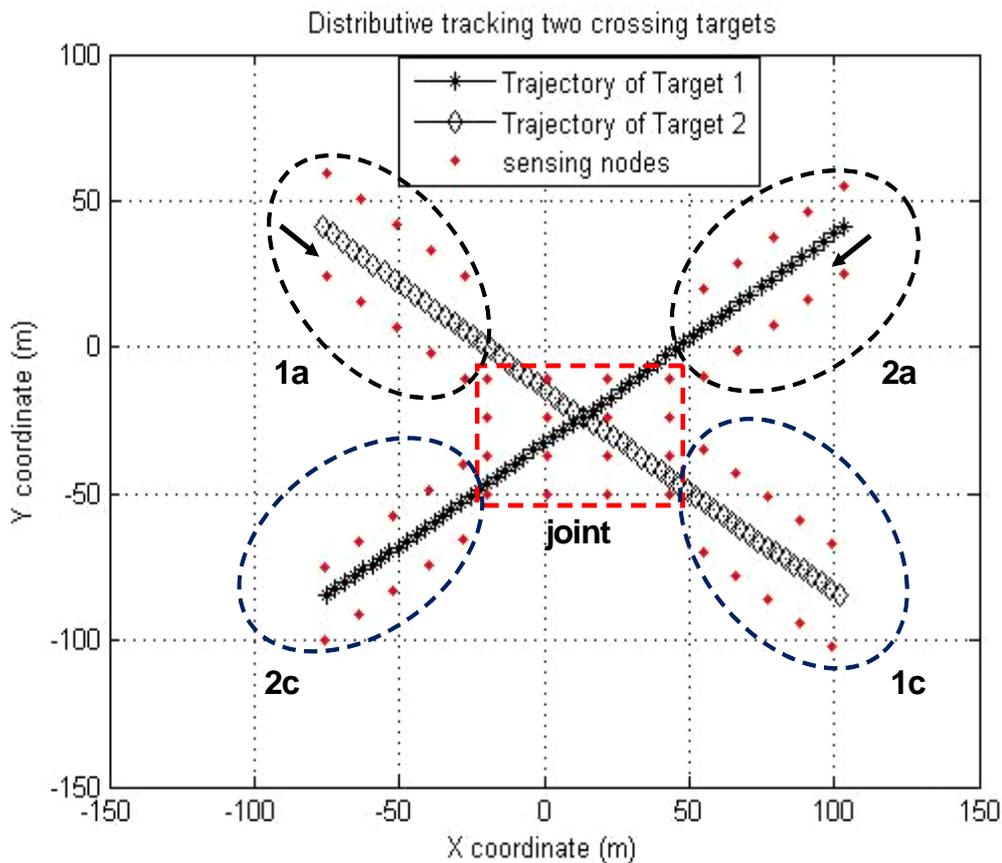


Figure 7.10 Distributively tracking two crossing targets in adjoining regions

Figure 7.11 depicts the estimated trajectories of two targets obtained by the above distributive two crossing targets tracking scheme under four different settings of clutter and detection rates: $P_d = 1, C_d = 0$; $P_d = 1, C_d = 0.5$; $P_d = 0.9, C_d = 0$; and $P_d = 0.9, C_d = 0.5$. In Figure 7.11, the circle represents the handover time steps that one cluster leader

transmits its estimation results to the next cluster leader (the details of the handover of estimation results can be found in Chapter 6). Figures 7.12 and 7.13 show the RMSE values of two targets obtained by the distributive two crossing targets tracking scheme under the settings of $P_d = 1, C_d = 0$ and $P_d = 0.9, C_d = 0.5$, respectively.

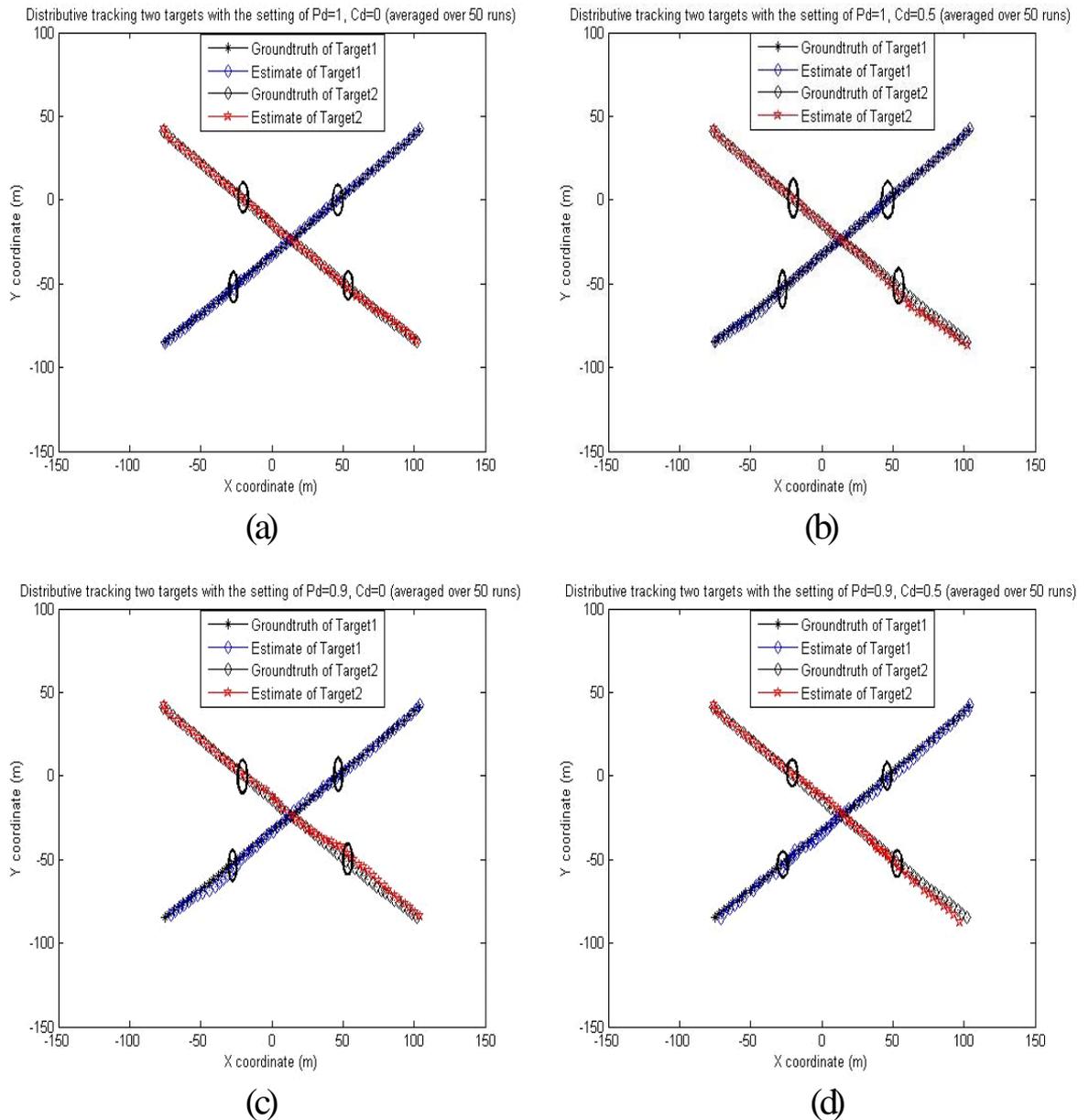
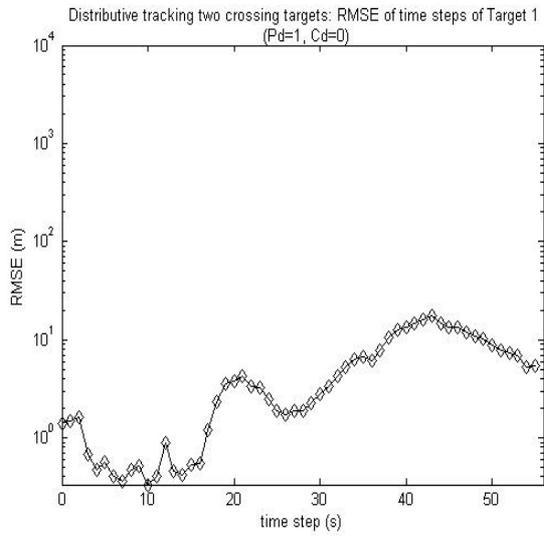
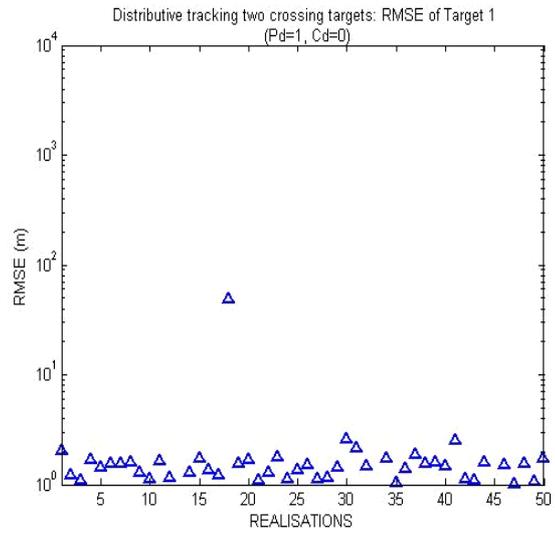


Figure 7.11 Estimated trajectories of two crossing targets with different settings of detection and clutter rates using distributive tracking scheme
(a) $P_d = 1, C_d = 0$ (b) $P_d = 1, C_d = 0.5$
(c) $P_d = 0.9, C_d = 0$ (d) $P_d = 0.9, C_d = 0.5$

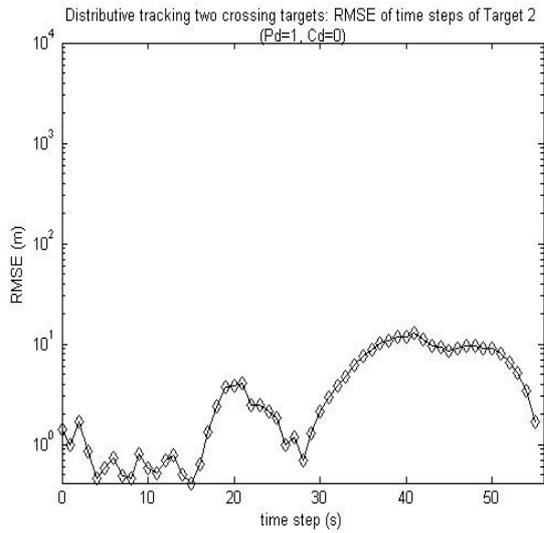
(The circles denote the handover time step that one cluster leader transmits its estimation results to the next cluster leader.)



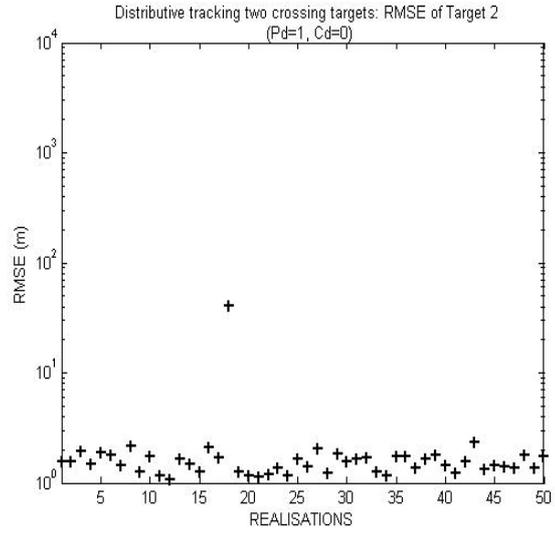
(a)



(b)



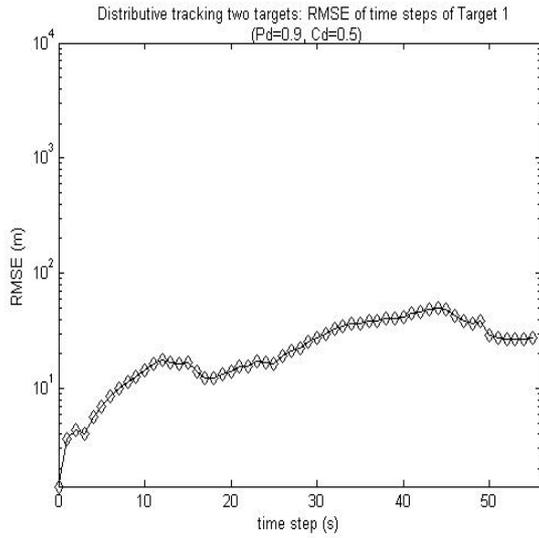
(c)



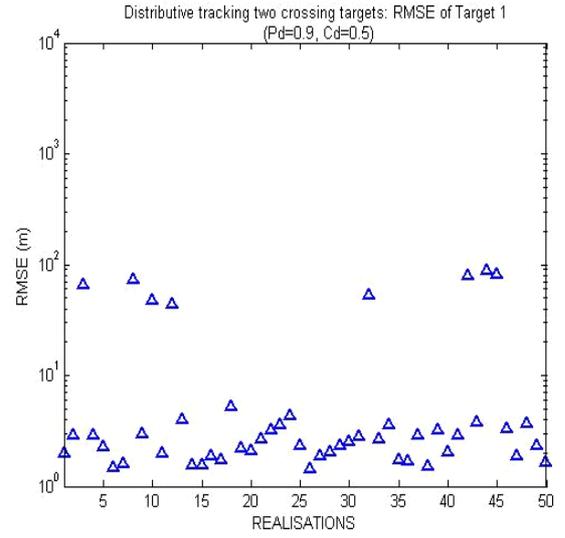
(d)

Figure 7.12 RMSE values for distributively tracking two crossing targets with the setting of $P_d = 1, C_d = 0$

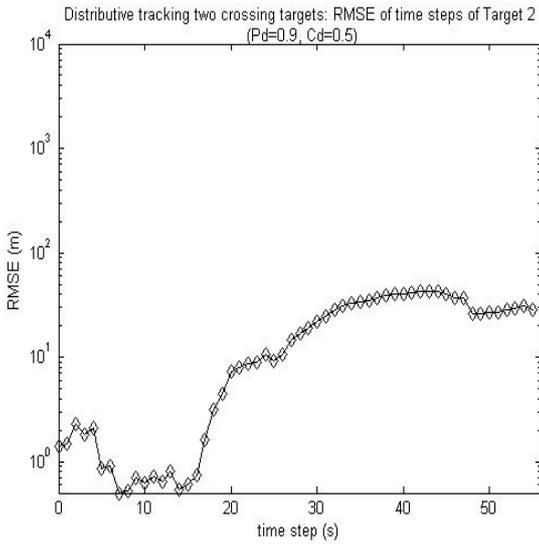
- (a) $RMSE_k$ of Target 1
- (b) $RMSE^n$ of Target 1
- (c) $RMSE_k$ of Target 2
- (d) $RMSE^n$ of Target 2



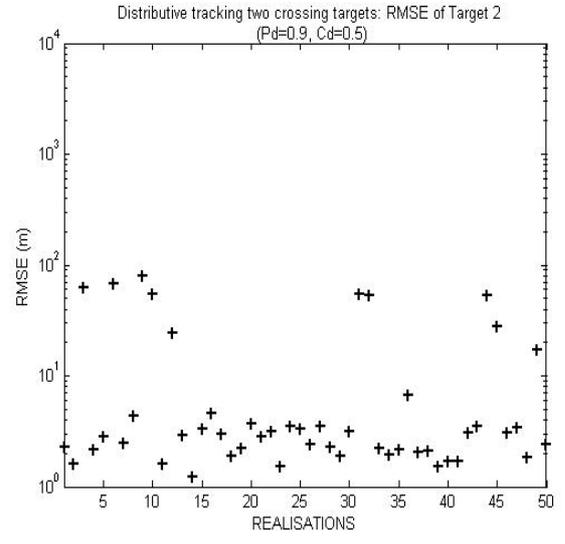
(a)



(b)



(c)



(d)

Figure 7.13 RMSE values for distributively tracking two crossing targets with the setting of $P_d = 0.9$, $C_d = 0.5$

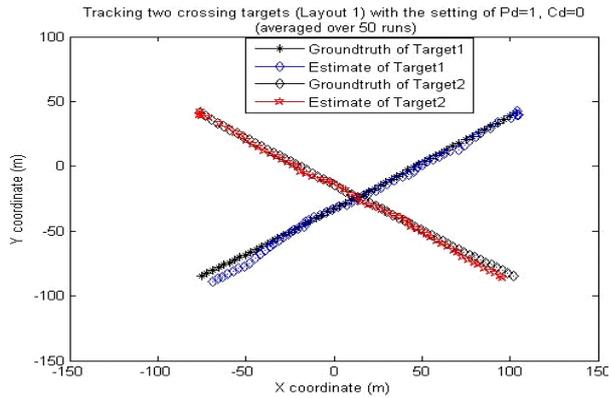
- (a) $RMSE_k$ of Target 1 (b) $RMSE^n$ of Target 1
(c) $RMSE_k$ of Target 2 (d) $RMSE^n$ of Target 2

From Figure 7.11, it can be seen that the handover of estimation results between the cluster leaders does not cause the significant distortion on the overall performance of the distributive tracking scheme. This is because in multiple target tracking under measurement origin uncertainty due to the presence of multiple targets and clutter, the overall performance of the tracking algorithm is greatly decided by the data association problem, i.e. to which extent the measurement generated by one target can be correctly picked up from the measurements generated by other targets or clutter. It can be seen from Figures

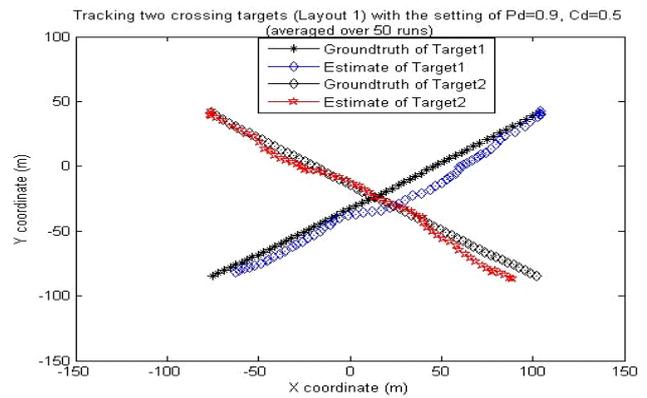
7.12 and 7.13 that the higher measurement origin uncertainty degrades the tracking accuracy of the distributive tracking scheme as it does in the normal PF-JPDAF algorithm adopting Layouts 1 and 2.

Figure 7.14 compares the performance of PF-JPDAF algorithm using Layout 1, PF-JPDAF algorithm using Layout 2, and the distributive tracking scheme for tracking two crossing targets. Figures 7.14 (a)-(c) show the result under the setting of $P_d = 1, C_d = 0$ and Figures 7.14 (d)-(f) show the result under the setting of $P_d = 0.9, C_d = 0.5$. It can be seen that the distributive tracking scheme outperforms other two schemes (Layout 1 and Layout 2) under both settings.

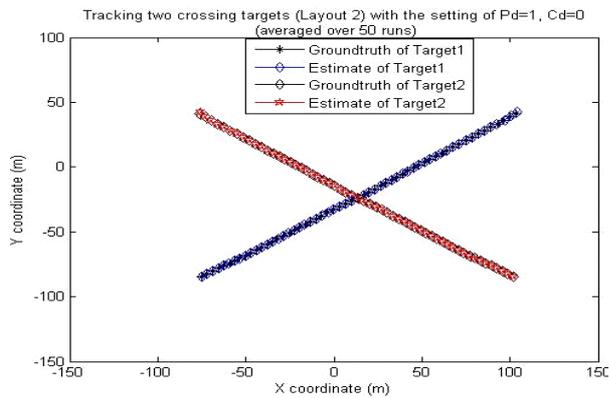
The superior performance of the distributive tracking scheme is due to the use of two separate PF-PDAF algorithms in the Regions 1a and 2a and in Regions 1c and 2c (Figure 7.10), instead of using a single PF-JPDAF to jointly estimate the states of the two targets throughout the whole sensor field. Although the GMM approximation is used in the distributive tracking scheme when one cluster leader propagates its estimation result to the next cluster leader and this may have some impacts on the tracking accuracy, however, the simulation results show that the performance of the distributive tracking scheme does not degrade much. Moreover, the distributive tracking scheme is more computational efficient than other two schemes. In the simulations, the distributive tracking scheme takes 23.66 seconds for 56 time steps estimation of two targets states while the PF-JPDAF algorithm adopting Layout 1 takes 40.51 seconds and PF-JPDAF algorithm adopting Layout 2 takes 85.82 seconds.



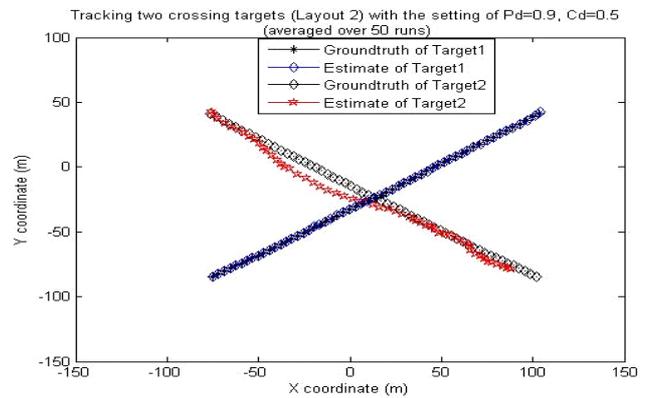
(a) Layout 1, $P_d=1$, $C_d=0$



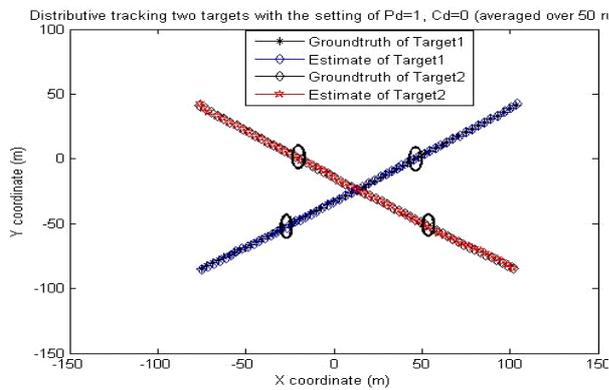
(d) Layout 1, $P_d=0.9$, $C_d=0.5$



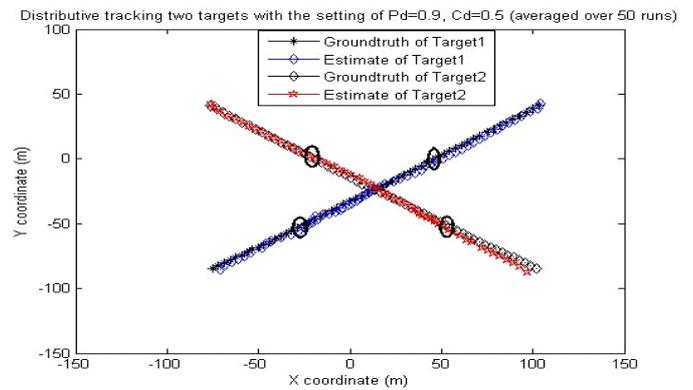
(b) Layout 2, $P_d=1$, $C_d=0$



(e) Layout 2, $P_d=0.9$, $C_d=0.5$



(c) Distributive tracking, $P_d=1$, $C_d=0$



(f) Distributive tracking, $P_d=0.9$, $C_d=0.5$

Figure 7.14 RMSE values of PF-JPDAF algorithm using Layout 1, PF-JPDAF algorithm using layout2, and the distributive tracking scheme

(a) Layout 1 $P_d = 1$, $C_d = 0$

(d) Layout 1 $P_d = 0.9$, $C_d = 0.5$

(b) Layout 2 $P_d = 1$, $C_d = 0$

(e) Layout 2 $P_d = 0.9$, $C_d = 0.5$

(c) Distributive tracking $P_d = 1$, $C_d = 0$

(f) Distributive tracking $P_d = 0.9$, $C_d = 0.5$

7.5.3 Simulation Results of Tracking Two Close-spaced Paralleling Targets

Figure 7.15 depicts the simulation setup for tracking two close-spaced paralleling targets as depicted in Figure 7.1 (b). It is assumed that the tracking task takes place in one sensor cluster which consists of one cluster leader (the cluster leader is not drawn in the figure) and 18 sensing nodes deployed along the road on which the targets traverse. At each time step, a set of six sensing nodes are selected from these 18 sensing nodes to make measurements and transmit their measurements to the cluster leader. The above sensing nodes selection is based on their positions and is empirically decided for the PF-JPDAF algorithm to attain desirable tracking accuracy at the reasonable computation cost.

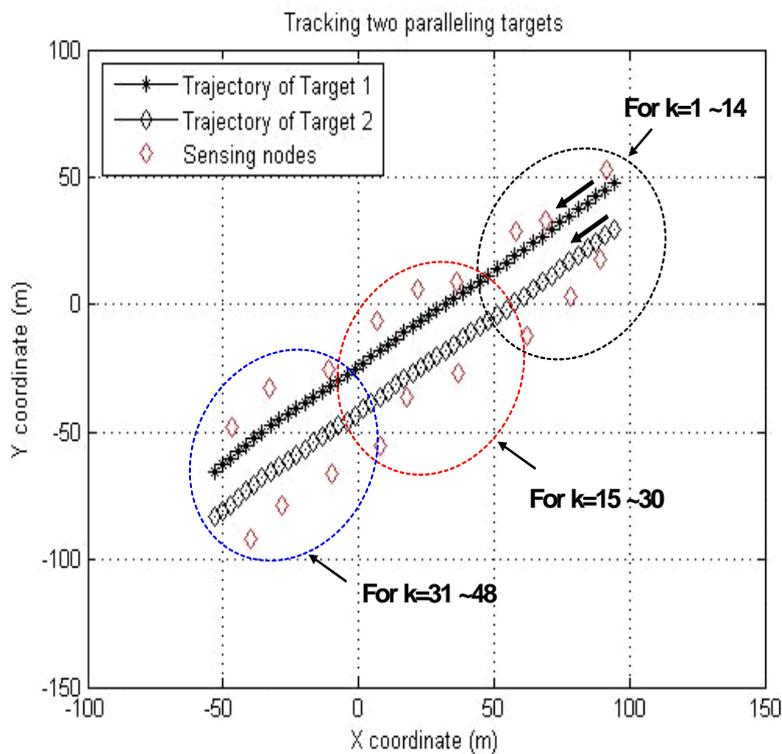


Figure 7.15 The scenario of tracking two close-spaced paralleling targets (During time step $k=1 \sim 14$, the sensing nodes within the black ellipse are selected; During time step $k=15 \sim 30$, the sensing nodes within the red ellipse are selected; During time step $k=31 \sim 48$, the sensing nodes within the blue ellipse are selected)

Figure 7.16 depicts the estimated trajectories of two close-spaced paralleling targets obtained by the PF-JPDAF algorithm under four different sets of clutter and detection rates: $P_d = 1, C_d = 0$; $P_d = 1, C_d = 0.5$; $P_d = 0.9, C_d = 0$; and $P_d = 0.9, C_d = 0.5$. Figures 7.17 and 7.18 show the RMSE values of the two targets under the setting of $P_d = 1, C_d = 0$ and $P_d = 0.9, C_d = 0.5$, respectively.

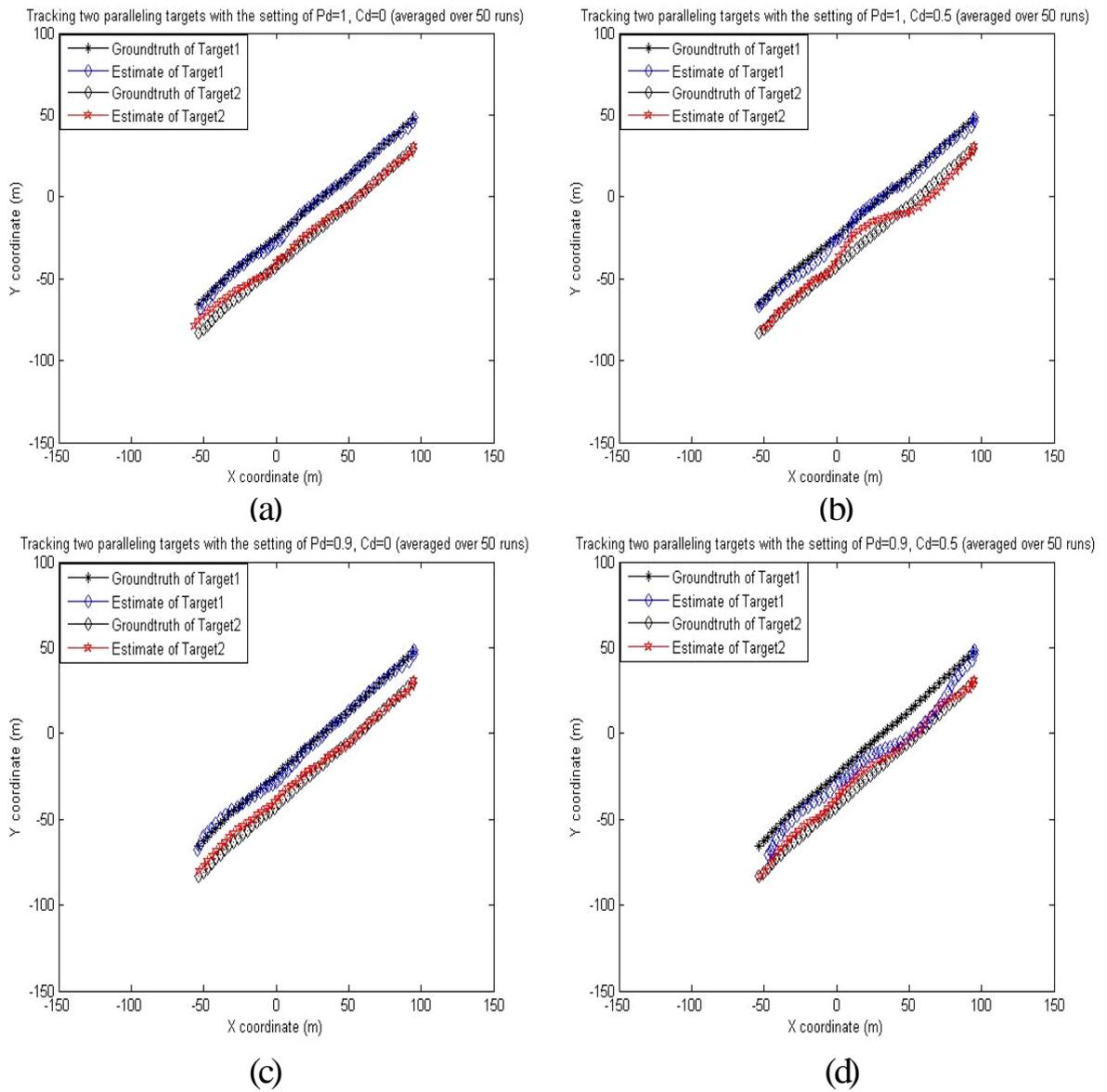


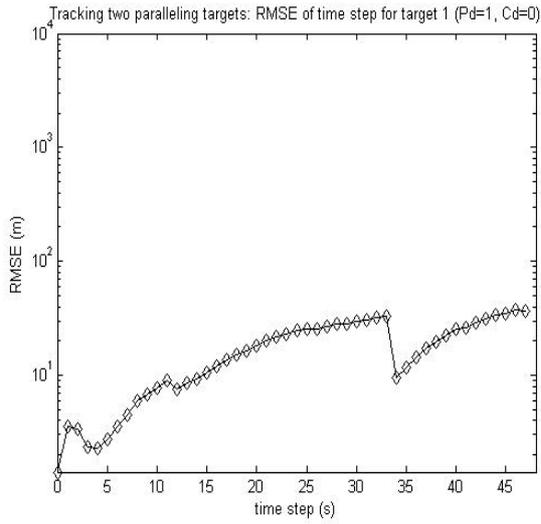
Figure 7.16 Estimated trajectories of two paralleling targets under different settings of detections and clutter rates

(a) $P_d = 1$, $C_d = 0$

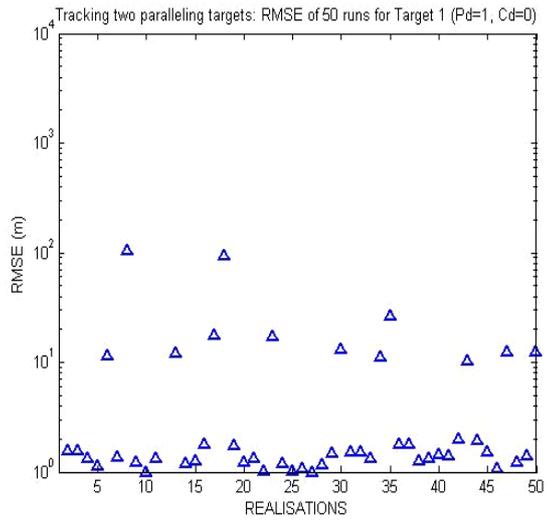
(b) $P_d = 1$, $C_d = 0.5$

(c) $P_d = 0.9$, $C_d = 0$

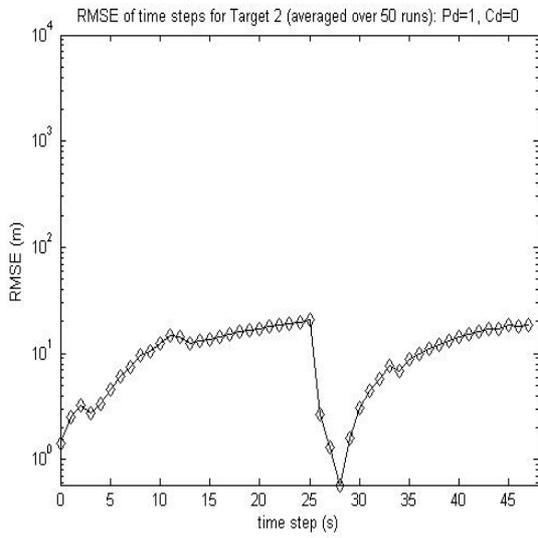
(d) $P_d = 0.9$, $C_d = 0.5$



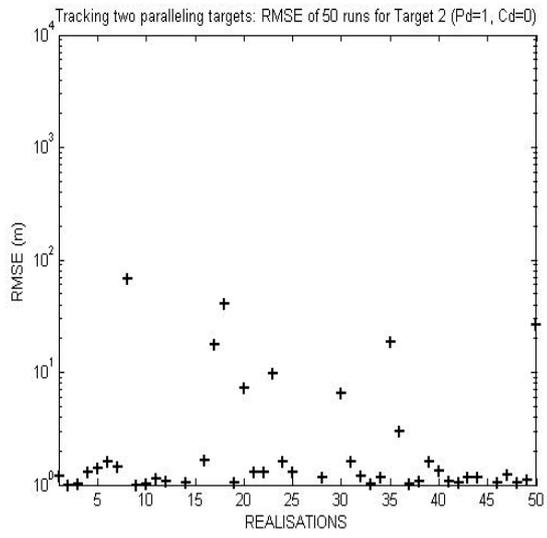
(a)



(b)



(c)



(d)

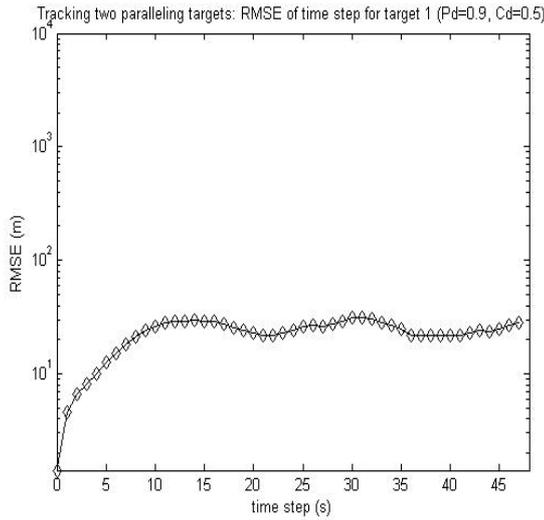
Figure 7.17 RMSE values of PF-JPDAF algorithm for tracking two paralleling targets with the setting of $P_d = 1$, $C_d = 0$

(a) $RMSE_k$ of Target 1

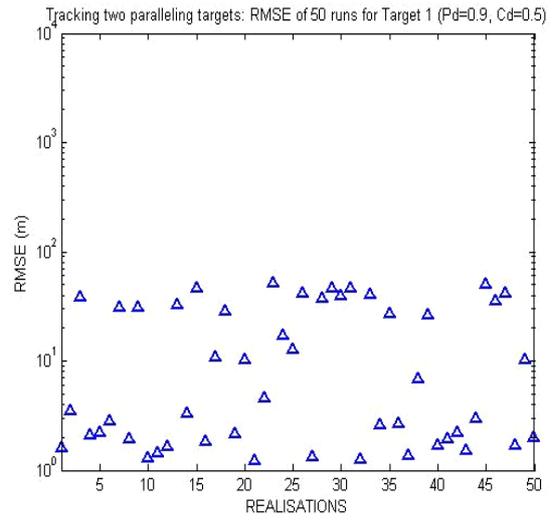
(b) $RMSE^n$ of Target 1

(c) $RMSE_k$ of Target 2

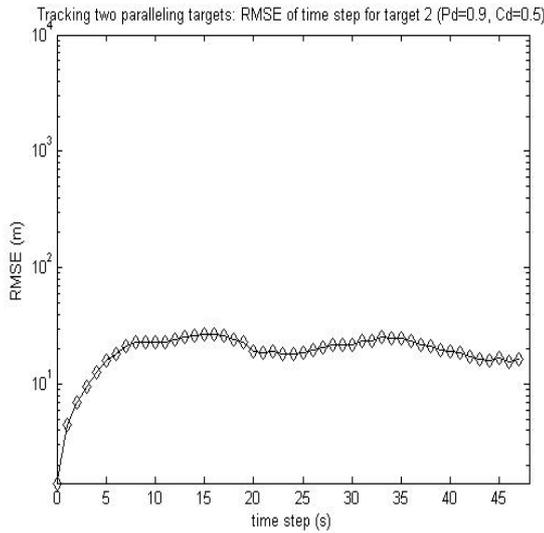
(d) $RMSE^n$ of Target 2



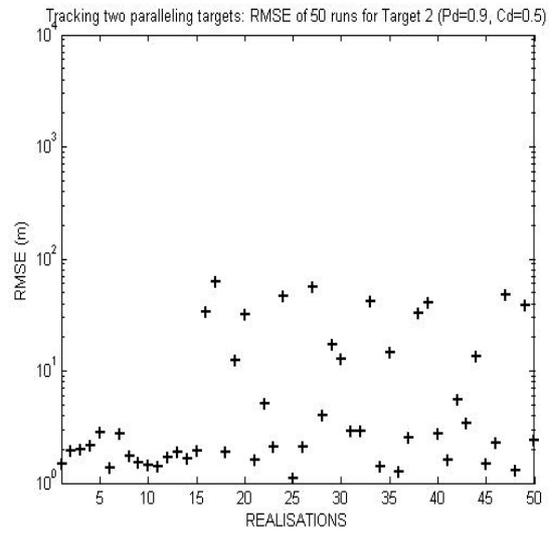
(a)



(b)



(c)



(d)

Figure 7.18 RMSE values of PF-JPDAF algorithm for tracking two paralleling targets with the setting of $P_d = 0.9$, $C_d = 0.5$

(a) $RMSE_k$ of Target 1

(b) $RMSE^n$ of Target 1

(c) $RMSE_k$ of Target 2

(d) $RMSE^n$ of Target 2

From these figures, it can be seen that when there is no clutter, the PF-JPDAF algorithm can track two close-spaced paralleling targets well (Figures 7.16 (a) and 7.16 (c)). However, when clutter exists, the performance of PF-JPDAF algorithm is much deteriorated (Figures 7.16 (b) and 7.16 (d)). This is because under measurement origin uncertainty, two close-spaced paralleling targets means a greater challenge is posed for the data association problem. This may be even more challenging than the scenario of tracking two crossing targets, since in close-spaced paralleling targets tracking, two targets generate almost the

same measurements at the sensing nodes (recalled that it is assumed that the acoustic intensity takes the same value for both targets).

Although the above simulations are conducted for tracking two targets, the PF-JPDAF algorithm developed in this chapter can be used to track more than two targets without any change in the algorithmic design.

7.6 Summary

This chapter addresses the problem of tracking multiple targets under measurement origin uncertainty in the wireless sensor networks. The PF-JPDAF multiple target tracking algorithm has been developed. By making use of the particle's representation of the probability density functions of the target state and effectively solving the challenging data association problem, PF-JPDAF algorithm can be applied to track multiple targets with nonlinear measurement model and under measurement origin uncertainty in wireless sensor networks.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

Target tracking is a representative application of wireless sensor networks. Nevertheless, it remains to be a challenging and non-trivial task to design target tracking algorithms for 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. To address these challenges, this thesis developed the collaborative information processing techniques that jointly tackle both information processing and networking issues for the distributive estimation of target state in the highly dynamic and resources constrained wireless sensor networks.

Taking into account the interplay between information processing and sensor network architecture, this thesis proposed a collaborative information processing framework for target tracking applications in wireless sensor networks. By jointly addresses the information processing issue which is responsible for the representation, fusion and processing of data and information and the networking issues which caters for the network formation, data and information delivery and wireless channel management, this framework could provide a unified approach for the distributive target state estimation in wireless sensor networks.

Within the proposed collaborative information processing framework, this thesis developed a suite of target tracking algorithms on the basis of the recursive Bayesian estimation method. For tracking a single target in wireless sensor networks, four tracking algorithms were developed, namely 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). The PF and EKPF tracking algorithms use discrete samples (particles) to approximate the probability density function of the target state and thus they can be applied to more general non-Gaussian and non-linear target

tracking problems in wireless sensor networks. Especially, the novel EKPF algorithm integrates the EKF into PF to propagate particles to higher measurement likelihood area in the state-space. The simulation results showed that the EKPF outperformed 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. Moreover, to help evaluate the performance of the developed tracking algorithms, the posterior Cramer-Rao lower bound (PCRLB) which is the theoretical lower bound on the mean square error (RME) of the target state estimation was computed and compared for the above four tracking algorithms.

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*. To tackle the data association problem, this thesis developed a hybrid algorithm which integrate the Particle filter (PF) with the probability density association filter (PDAF), named as PF-PDAF tracking algorithm for single target tracking under the dual assumptions of clutter and missed detections. The PF-PDAF tracking algorithm combines the advantages of PDAF algorithm in effectively solving the data association problem with the merits of PF that can accommodate the general non-Gaussian, nonlinear state-space model. In order to provide a theoretical bound on the performance of PF-PDAF algorithm, the PCRLB under measurement origin uncertainty was also derived and computed. It was shown that under measurement origin uncertainty, the measurement contribution to the PCRLB is a product of PCRLB when there is no measurement origin uncertainty and a scalar information reduction factor (IRF). Similar to PF-PDAF, this thesis also developed a hybrid algorithm which integrate the Particle filter (PF) with the joint probability density association filter (JPDAF), named as PF-JPDAF tracking algorithm for multiple target tracking under the measurement origin uncertainty in wireless sensor networks. By adopting the particles' representation of the probability density function of the target state, the PF-JPDAF extends the JPDAF to solve the general non-linear non-Gaussian multiple targets tracking problems in wireless sensor networks.

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 and energy consumption. 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, distributive EKPF, distributive PF-PDAF tracking algorithms for single target tracking and the distributive PF-JPDAF tracking algorithm for multiple target tracking in wireless sensor networks. In these distributive tracking algorithms, a number of sensor clusters are dynamically formed within the sensor field and each sensor cluster occupies a smaller region. At each time step, in the active sensor cluster, a portion of sensing nodes are selected based on their information utility and energy consumption measures; these sensing nodes are then activated to sense and provide their measurements to the cluster leader, which then runs the tracking algorithm to update the probability density function of the target state. When the target moves out of the current sensor cluster, the current cluster leader forwards its estimation results in the GMM format to the new cluster leader. This process will continue until the target moves out of the sensor field. To facilitate the sensing nodes selection in the distributive tracking algorithms, a composite objective function incorporating both the information utility and the energy consumption measures was developed in this thesis. This composite objective function enables the distributive tracking algorithms to achieve the desirable tracking accuracy while still maintaining the lowered energy consumption.

8.2 Future Work

The work reported in this thesis can be extended in several ways. Some potential topics of further research are provided as follows.

Extension of PF-JPDAF Algorithm to Track Unknown and Varying Number Targets

In this thesis, the PF-JPDAF algorithm is successfully applied to track multiple targets under measurement origin uncertainty in wireless sensor networks. However, in the practical tracking problems the number of targets may vary significantly over the tracking period since the targets may enter or leave the area under observation. As a consequence, both the number and identity of the targets needed to be estimated. Therefore, one extension of PF-JPDAF is to account for an unknown and variable number of targets: correctly detect the entering and leaving targets, maintain a unique identification for each

target, and update the target state estimate throughout the whole tracking period in the wireless sensor networks.

Computation of PCRLB for Multiple Target Tracking

In previous chapters, the PCRLB has been calculated for single target tracking. We also want to calculate the PCRLB to provide a theoretical lower bound on the performance of the multiple target tracking algorithms developed for wireless sensor networks. However, in multiple target tracking, the existence of multiple targets and clutter poses a major challenge in the PCRLB computation.

Sensing Nodes Selection for Multiple Target Tracking

As in single target tracking, we want to develop a composite objective function for sensing node selection in multiple target tracking. However, in the presence of multiple targets and clutter, the calculation of the composite objective function will be complicated, and with the increasing number of targets, the enumeration of all possible combinations of sensing nodes become intractable and it is necessary to develop the approximation solutions.

Networking Algorithms for Target Tracking in Wireless Sensor Networks

In chapter 3, the networking algorithms including the hierarchical routing algorithm and the hybrid MAC algorithm have been proposed. In the future research, we aim to implement these algorithms and integrate them into distributive target state estimation algorithms. However, this is a challenging task and need to consider many aspects of system requirements and tradeoffs in wireless sensor networks.