Distributed Tracking with Probability Hypothesis Density Filters Using Efficient Measurement Encoding

DISTRIBUTED TRACKING WITH PROBABILITY HYPOTHESIS DENSITY FILTERS USING EFFICIENT MEASUREMENT ENCODING

By

ARAVINTHAN AMPIKATHASAN, B.Sc.Eng., M.Sc.Eng.
University of Moratuwa, Sri Lanka.

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree
Master of Applied Science

McMaster University

MASTER OF APPLIED SCIENCE (2009) MCMASTER UNIVERSITY

(Electrical and Computer Engineering) Hamilton, Ontario, Canada

TITLE: **Distributed Tracking with Probability Hypothesis Density Filters Using Efficient Measurement Encoding**

AUTHOR: Aravinthan Ampikathasan

B.Sc.Eng.

University of Moratuwa

Sri Lanka, 2002

M.Sc.Eng.

University of Moratuwa

Sri Lanka, 2005

SUPERVISOR: Dr. T. Kirubarajan

NUMBER OF PAGES: x, 60

Abstract

Probability Hypothesis Density (PHD) filter is a unified framework for multitarget tracking that provides estimates for a number of targets as well as individual target states. Sequential Monte Carlo (SMC) implementation of a PHD filter can be used for nonlinear non-Gaussian problems. However, the application of PHD based state estimators for a distributed sensor network, where each tracking node runs its own PHD based state estimator, is more challenging compared with single sensor tracking due to communication limitations. A distributed state estimator should use the available communication resources efficiently in order to avoid the degradation of filter performance. In this thesis, a method that communicates encoded measurements between nodes efficiently while maintaining the filter accuracy is proposed. This coding is complicated in the presence of high clutter and instantaneous target births. This problem is mitigated using adaptive quantization and encoding techniques. The performance of the algorithm is quantified using a Posterior Cramér-Rao Lower Bound (PCRLB), which incorporates quantization errors. Simulation studies are performed to demonstrate the effectiveness of the proposed algorithm.

To my mother, sister and brother who sacrificed so much for my well-being and Esha, who has been a great source of motivation

Acknowledgements

Praise be to God for helping me finish this thesis and in all aspects of my life. I wish to express my deepest gratitude to my supervisor Dr. T. Kirubarajan for offering me an opportunity to be a graduate student. His guidance and inspiration throughout the entire research for this thesis were invaluable. His patience with students and willingness to always provide extra help are greatly appreciated. I would like to thank Dr. Tharmarasa and Dr. Nandakumaran for their guidance and kind suggestions throughout my work.

My appreciation also goes to all the instructors in the ECE Department who taught me various courses. What they have taught me in the class has greatly helped me understand the research papers. Department staff members, especially Cheryl Gies and Helen Jachna, are also gratefully acknowledged for their constant administrative help and support. I thank Terry Greenlay and Cosmin Coriou assisting in resolving computer related issues.

I am thankful to my fellow students in the ETF Laboratory for many thought-provoking and stimulating discussions. All above, I would like to acknowledge the immeasurable contribution of my parents in teaching me the value of education. Finally, I wish to thank Kowsi and her family, my roommates and my friends. Their continuous encouragement has been extraordinary and wonderful.

Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 Motivation and Contribution of the Thesis

The use of a large number of networked sensors, which can be deployed all over the surveillance region, in tracking applications has become feasible because of the availability of cheap sensors. Hence the importance of the research on distributed tracking has attracted many researchers in tracking community. Quantization and encoding play important role in distributed tracking as information needs to be shared.

Sensor networks provide large area of coverage and utilize a network of sensing devices to gather useful information. Further, the network provides the synergistic use of more information from multiple sources as sensing devices are densely deployed in the desired environment. A sensor node, which acts like a hub, connects a number of sensing devices and perform computations and communications. Sensor network is widely used in many application areas including target tracking. Centralized and distributed tracking architectures are commonly used in sensor networks for target tracking. The sensor data need to be fused in order to fully utilize the information obtained in the network. Distributed processing over the sensor network can be used to alleviate the problems inherent to the centralized architectures especially the communication bandwidth issue. Distributed approach, since the processing tasks are performed over multiple fusion nodes, requires lighter computational power and communication bandwidth.

Distributed algorithms based on particle filters to track a single object have gained much attention. One of the first developments for nonlinear/non-Gaussian systems was in [12] and it proposes two methods to use the distributed sensor network. One of them is based on likelihood factorization of particles of the filter and the other one based on adaptive data-encoding scheme. An improvement to this approach using a better encoding scheme and measurement vectorization has been presented in [13]. More particle based implementations are given in [14, 15]. The adaptive data-encoding scheme uses the histogram of expected measurements to encode the target generated measurements effectively. However, the effectiveness of the encoding scheme might degrade dramatically if no method is in place to identify and remove false measurements before transmitting over the network, since the false measurements might end up transmitting larger number of bits than transmitting raw measurements.

In this thesis, a decentralized version of the Probability Hypothesis Density (PHD) filter, a data-association free method, in contrast to the typical decentralized versions of data association techniques, is considered . The PHD filter has been shown to be an effective way of tracking time-varying multiple number of targets that avoids model-data association problems. The proposed algorithm uses a sensor network to track multiple targets based on the Sequential Monte Carlo(SMC) implementation of the PHD filter. There are a number of options available to perform distributed tracking with SMC-PHD filter in a sensor network. The first option is to send all the particles, which represent the posterior density of targets.

The second is to send Gaussian mixture representation of posterior density. The first and second options require high bandwidth communications, which can not be handled by practical wireless sensor networks. The third option is to send most appropriate measurements, after eliminating the false alarms, to update the global estimates of the targets. In this thesis, measurements are communicated among nodes to update the filters. Sharing the measurement directly is preferable over sharing of estimates, because when measurements are shared fusion can be done optimally as possible. In the later case, data transmission requires higher bandwidth channels unless the quantization of those data are done intelligently [16, 17]. Non-uniform quantization scheme could be made to match the distribution of the quantity to be discretized. Companding is a widely used method of implementing non-uniform quantizers[18]. It is observed in non-uniform quantization the communication is considerably cut with the right selection of the compander [16]. Quantized measurement need to be encoded before transmitted. It is assumed that an optimal noiseless source code will be employed to minimize transmission needs between nodes. In this thesis, Huffman coding is used to encode the quantized measurements. Handling multiple target originated measurements at the quantization stage and producing identical symbols for encoding and decoding at each node are challenging. This thesis proposes "cascaded companders" to non-linearly quantize multiple target measurements. Predicted probability density is used in generating identical set of symbols and to place the companders at right positions.

Among the various methods to quantify the performance, checking the closeness of its mean square error matrix to the lower bound is a commonly known method in target tracking applications. The Posterior Cramer-Rao Lower Bound (PCRLB) is defined to be inverse of the Fisher Information Matrix (FIM) for random vector and provides lower bound on the performance of unbiased estimators of the unknown target state [19]. PCRLB for state estimation with quantized measurements is complicated due to non-linearity of the quantizer. Previously, in [20] the PCRLB for dynamic target tracking with measurement origin uncertainty and in [21] the PCRLB for state estimation with quantized measurement were developed. In this thesis, the PCRLB calculation with quantized measurement is extended to incorporate measurement origin uncertainty for bearing only tracking.

1.2 Organization of the Thesis

This thesis is structured as follows. Chapter 2 describes the overview of target tracking and Chapter 3 explains PHD filter in detail and its implementation using particle filter. Theoretical background of quantization and encoding methods and its implementation details are reviewed in Chapter 4. Chapter 5 provides the derivation of the PCRLB with quantized measurements and measurement origin uncertainty. Simulations are presented that demonstrate the effectiveness of the proposed quantization strategy and modified PCRLB in Chapter 6.

1.3 Related Publications

1.3.1 Journal article

"Distributed Tracking with Probability Hypothesis Density Filters Using Efficient Measurement Encoding", *To be submitted to IEEE transactions on Aerospace and Electronic Systems.*

1.3.2 Conference publication

A. Aravinthan, R. Tharmarasa, Tom Lang, Mike McDonald and T. Kirubarajan, "Distributed Tracking with Probability Hypothesis Density Filters Using Efficient Measurement

Encoding", *Proceedings of the SPIE Conference on Signal and Data Processing of Small Targets*, San Diego, CA, Aug. 2009.

Chapter 2

TARGET TRACKING

2.1 Introduction

Tracking is described as the task of estimating the state at the current time and at any point in the future of a target from incomplete, inaccurate and uncertain observations. Also it produces the measure of the accuracy of the state estimates in addition to the state estimates [1]. Application of target tracking can be found in wide variety of areas ranging from tracking people on ground to tracking missiles on air. A typical tracking system, which is shown in figure 2.1 , has sensors, signal processing modules and information processing module. Depending on the complexity of the tracking system and number of sensors, the system can have multiple modules of signal processor and information processor either collocated or separated. When the sensors and/or information processor are separated, measurements and/or estimates need to be communicated efficiently to achieve better performance.

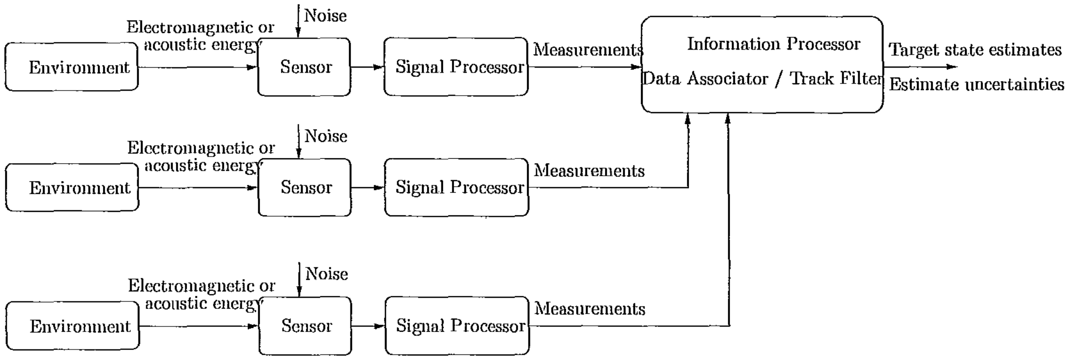

Figure 2.1: A typical tracking system

2.2 A target tracking system

Though a target tracking system consists of many different modules, which function together to achieve better results, the focus of this chapter is mainly on acquiring measurements and performing filtering. Most of the filtering algorithms are model based and depends on two models: a model describes the behaviors of the target called target model and the other for sensing the target behaviors called measurement model. The accuracy of tracking output significantly depends on these models [2].

2.2.1 Linear and nonlinear models

A Target in tracking is treated as a point object where the target model describes the evolution of the state with time. The most commonly used target models are in the following forms:

- nonlinear model

$$x_{k+1} = f_k(x_k) + \nu_k \tag{2.1}$$

- linear model

$$x_{k+1} = F_k \; x_k + \nu_k \tag{2.2}$$

Where x_k denotes the state at time k, f_k is a nonlinear function of state, F_k is a known matrix and ν_k is the process noise at time k. Target motions are broadly classified into two types: maneuver and nonmaneuver [2]. Measurement models describe how the target states are coupled to noisy measurements. The measurement model can be linear or nonlinear as given below.

- nonlinear measurement model

$$z_k = h_k(x_k) + \omega_k \tag{2.3}$$

- linear measurement model

$$z_k = H_k \; x_k + \omega_k \tag{2.4}$$

Where h_k is a nonlinear function and H_k is a known matrix. Measurement vector is denoted by z_k and measurement noise is by ω_k at time k. For simplicity it is assumed that v_k and ω_k are Gaussian with zero means and covariances Γ_k and Σ_k respectively.

2.2.2 The Bayesian approach

The Bayesian approach to dynamic state estimation constructs the posterior probability density function of the state based on all the past measurements and the measurements obtained in the current time. It can be considered the complete solution to the estimation problem as this pdf contains all available statistical information. An optimal estimate of the state may be obtained from the pdf in principle. Recursive filtering approach eliminates the necessity to store the complete data set nor to reprocess existing data for a new measurement update. In other wards, when a new measurement becomes available, the received data can be processed sequentially rather than as a batch. Such a filter consists of two stages: prediction and update. System model is used in prediction stage to predict the state pdf forward from one revisit to the next. Due unknown disturbances in state, prediction generally translates, deforms and spreads the state pdf. Suppose that the required pdf $p(x_k|Z_k)$ at time k is available, where $Z_k = [z_1, z_2, \ldots, z_k]$. The prediction stage involves using the system model 2.3 to obtain the prior pdf of the state at time $k+1$ and given by

$$p(x_{k+1}|Z_k) = \int p(x_{k+1}|x_k)p(x_k|Z_k)dx_k \tag{2.5}$$

The update operation uses the latest measurement to modify the prediction pdf. At the next time $k+1$, a measurement z_{k+1} becomes available and will be used to update the prior via Bayes' rule:

$$p(x_{k+1}|Z_{k+1}) = \frac{p(z_{k+1}|x_{k+1})p(x_{k+1}|Z_k)}{p(z_{k+1}|Z_k)} \tag{2.6}$$

In the above the likelihood function $p(z_{k+1}|x_{k+1})$ is defined by the measurement model (2.3). The above recursive propagation of the posterior density is only a conceptual solution, and in general it cannot be determined analytically. Analytical solution exists only in a restrictive set of cases. This single target Bayesian filtering can be extended to multitarget state and multitarget measurement.

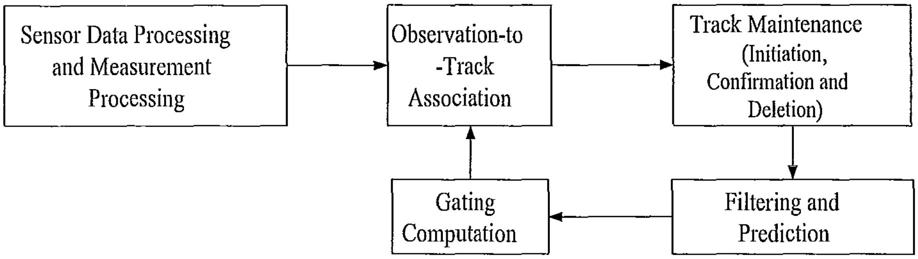

Figure 2.2: Basic elements of a conventional multitarget tracking system

2.3 Multitarget tracking

In the tracking system, a track is a state trajectory estimated from each new set of measurements, which is associated with the same target. It is a symbolic representation of a target moving through an area of interest. Basic elements of a conventional Multiple Target Tracking (MTT) are arranged as in Figure 2.2. A signal processing unit converts the signals from the sensor to measurements, which then become the input data to the MTT system. Track is maintained using incoming measurements. A track is maintained in one of the following states: initialized track, confirmed track or dead track [3]. An observation not associated to any existing tracks can initiate a new tentative track. A tentative track becomes confirmed when it meets confirmation criteria defined in the form of quality and number of measurements. Similarly, a track get degraded when not updated and the track not updated within some reasonable interval must be deleted. Gating tests evaluate which possible measurements-to-track pairings are reasonable and a more detailed association technique is used to determine final pairings. Tracks are predicted ahead to the arrival time for the next set of observations, after inclusion of measurements available in current time. Gates are placed around these predicted positions for track to measurement association and processing cycle repeats. If the true measurement conditioned on the past is normally (Gaussian) distributed with its Probability Density Function (PDF) give by

$$p(z_{k+1}|Z_k) \sim \mathcal{N}[z_{k+1}; \hat{z}_{k+1|k}, S_{k+1}] \tag{2.7}$$

Where z_{k+1} is the measurement at time $k+1$, $Z_k = [z_1, z_2, \dots, z_k]$, $\hat{z}_{k+1|k}$ is the predicted (mean) measurement at time $k+1$ and $S(k+1)$ is the measurement prediction covariance, then the true measurement will be in the following region

$$\mathcal{V}(k+1, \gamma) = \{z : [z - \hat{z}_{k+1|k}]' S_{k+1}^{-1} [z - \hat{z}_{k+1|k}] < \gamma\} \tag{2.8}$$

with the probability determined by the gate threshold γ. The region defined by 2.8 is called gate or validation region ($\mathcal{V}$) or association region. The validation procedure limits the region in the measurement space where the information processor looks to find the measurement from the target of interest. Measurements outside the validation region can be ignored, since they are too far from the predicted location and very unlikely to have originated from the target of interest. It can so happen that more than one measurement is found in the validation region.

2.3.1 Data association

The data association problem is that of associating the many measurements of a sensor with the underlying states or tracks that are being observed. Often the problem of tracking multiple targets in clutter considers the situation where there are possibly several measurements in the validation region of each target. The set of validated measurements consists of:

- the correct measurement
- the undesired measurements: false alarms

The simplest possible approach is to use the measurement nearest to the predicted measurement as if it were the correct one, which is called Nearest Neighbor(NN). An alternative approach, called Strongest Neighbor (SN), is to select the strongest measurement among the validated ones [3]. Since any of the validated measurements could have originated from the target, this suggests that all the measurements from the validation region should be used in some fashion. A Bayesian approach, called Probabilistic Data Association (PDA), associates probabilistically all the "neighbors" to the target of interest [3]. PDA is the standard technique used for data association in conjunction with the Kalman filter or the extended Kalman filter. The Kalman filter can be applied only if the models are linear and measurement and process noises are independent and white Gaussian.

2.4 Filtering Algorithms

Filters are used to estimate the target states at each time interval with appropriate available measurements. The recursive Bayesian type filters are easy to implement and well established. Here some well known recursive filters are reviewed. The Kalman filter is given in Section 2.4.1. Extended Kalman filter (EKF), a variant of the Kalman filter, is given in Section 2.4.2. Finally particle filter, which has proven to be effective for nonlinear non-Gaussian problems, is given in Section 2.4.3.

2.4.1 Kalman Filter

The Kalman filter assumes that the state and measurement models are linear and the initial state error and all the noises entering into the system are Gaussian and, hence, parameterized by a mean and covariance [1]. Under the above assumptions, if $p(x_k|Z_k)$ is Gaussian, it can be proved that $p(x_{k+1}|Z_{k+1})$ is also Gaussian. Then, the state and measurement equations are given by

$$x_{k+1} = F_k x_k + \nu_k \quad (2.9)$$

$$z_k = H_k x_k + \omega_k \quad (2.10)$$

If F_k and H_k are known matrices, $\nu_k \sim \mathcal{N}(0, \Gamma_k)$ and $\omega_k \sim \mathcal{N}(0, \Sigma_k)$, the Kalman filter algorithm can then be viewed as following recursive relationship [1]

$$p(x_k|Z_k) = \mathcal{N}(x_k; m_{k|k}, P_{k|k}) \quad (2.11)$$

$$p(x_{k+1}|Z_k) = \mathcal{N}(x_{k+1}; m_{k+1|k}, P_{k+1|k}) \quad (2.12)$$

$$p(x_{k+1}|Z_{k+1}) = \mathcal{N}(x_{k+1}; m_{k+1|k+1}, P_{k+1|k+1}) \quad (2.13)$$

where

$$m_{k+1|k} = F_{k+1} m_{k|k} \quad (2.14)$$

$$P_{k+1|k} = \Gamma_k + F_{k+1} P_{k|k} F_{k+1}^T \quad (2.15)$$

$$m_{k+1|k+1} = m_{k+1|k} + K_{k+1}(z_{k+1} - H_{k+1} m_{k+1|k}) \quad (2.16)$$

$$P_{k+1|k+1} = P_{k+1|k} - K_{k+1} H_{k+1} P_{k+1|k} \quad (2.17)$$

with

$$S_{k+1} = H_{k+1} P_{k+1|k} H_{k+1}^T + \Sigma_{k+1} \quad (2.18)$$

$$K_{k+1} = P_{k+1|k} H_{k+1}^T S_{k+1}^{-1} \quad (2.19)$$

In the above, $\mathcal{N}(x; m, P)$ is a Gaussian density with argument x, mean m and covariance P. This is the optimal solution to the tracking problem if the above assumptions hold. The implication is that no algorithm can perform better than a Kalman filter in this linear Gaussian environment.

2.4.2 Extended Kalman Filter

In many situations of interest, neither of these sets of assumptions hold. It is then necessary to make approximation. If $p(x_0)$, $p(x_{k+1}|x_k)$ and $p(Z_{k+1}|x_{k+1})$ are approximated as Gaussian and $h_k(x_k)$ and $f_k(x_{k+1})$ are approximated as linear, then the recursion above becomes the Extended Kalman Filter, EKF.

$$\begin{aligned}
p(x_k|Z_{1:k}) &\approx \mathcal{N}(m_{k|k}, C_{k|k}) \\
p(x_{k+1}|Z_{1:k}) &\approx \mathcal{N}(m_{k+1|k}, C_{k+1|k}) \\
&\approx \mathcal{N}(a_k(m_{k|k}), \Sigma_k + A_k C_{k|k} A_k^T) && (2.20) \\
p(x_{k+1}|Z_{1:k+1}) &\approx \mathcal{N}(m_{k+1|k+1}, C_{k+1|k+1}) \\
&\approx \mathcal{N}(m_{k+1|k} + W_k \tilde{Z}_{k+1|k}, C_{k+1|k} - W_k H_{k+1} C_{k+1|k}) && (2.21)
\end{aligned}$$

Such a local approximation of the equations may be a sufficient description of the nonlinearity. However, it is common that it is not. A better approximation can be made by considering the above approximation as using the first term in Taylor expansions of the nonlinear functions, $h_k(x_k)$ and $f_k(x_k)$. A higher order EKF that retains further terms in the Taylor expansions exists and results in a closer approximation to the true posterior. However, the additional complexity has prohibited its widespread use.

2.4.3 Particle Filter

Another approach, which also yields such an improvement over the EKF, is particle filtering. Rather than approximating the models in order to be able to fit a distribution of a given type to the posterior, a particle filter explicitly approximates the distribution so that it can handle high nonlinear non-Gaussian models. The approach has also been known as sequential Monte Carlo filtering [9].

In particle filtering, the required posterior density function is represented by a set of random samples ("particles") with associated weights [7,10]. Let $x_k^{(p)} : p = 1, \dots m$, with associated weights $w_k^p : p = 1, \dots m$ be the random samples representing the posterior density $p(x_k|Z_{1:k})$ of the state vector x_k at time epoch k, where $Z_{1:k}$ is the set of all measurements available at time k. The weights are normalized such that $\sum_{p=1}^{m} w_k^p = 1$. We use Sampling Importance Resampling (SIR) [11] to propagate and update the particles in which the $p(x_k|Z_{1:k})$ is represented by equally weighted particles. Then

$$p(x_k|Z_{1:k}) \approx \frac{1}{m} \sum_{p=1}^{m} \delta(x_k - x_k^p) \tag{2.22}$$

where $\delta(.)$ is the Dirac Delta function. The Importance Density is chosen to be the prior density, $p(x_k|Z_{1:k-1})$. The propagation and update of the particles in SIR method are given as follows.

Prediction Take each existing sample, $x_k^{(p)}$ and augment it with a sample $x_{k=1}^{*(p)} \sim p(x_{k+1}|x_k^{(p)})$, using the system model. The set $x_{k+1}^{*(p)} : p = 1, \dots, m$ gives an approximation of the prior, $p(x_{k+1}|Z_{1:k})$, at time $k+1$, i.e.

$$p(x_{k+1}|Z_{1:k}) \approx \frac{1}{m} \sum_{p=1}^{m} \delta(x_{k+1} - x_{k+1}^{*(p)}) \tag{2.23}$$

Update

Importance Weights: At each measurement epoch, to account for the fact that the samples, $x_{k+1}^{*(p)}$ are not drawn from $p(x_{k+1}|Z_{1:k+1})$, the weights are modified using the principle of Importance Sampling. When using the prior as the Importance Density, it can be shown that the weights are given by

$$w_{k+1}^{p} = p\left(Z_{k+1}|x_{k+1} = x_{k+1}^{*(p)}, Z_{1:k}\right) \tag{2.24}$$

Reselection: Resample (with replacement) from $x_{k+1}^{*(p)} : p = 1, \dots, m$, using the weights, $w_{k+1}^{*(p)} : p = 1, \dots, m$, to generate a new sample, $x_{k+1}^{(p)} : p = 1, \dots, m$, then set $w_{k+1}^{p} = 1/m$ for $p = 1, \dots, m$. Then:

$$p(x_{k+1}|Z_{1:k+1}) \approx \tfrac{1}{m} \textstyle\sum_{p=1}^{m} \delta(x_{k+1} - x_{k+1}^{(p)}) \tag{2.25}$$

At each stage the mean of the posterior distribution is used to estimate, $\hat{x}_k$ of the target state, x_k, i.e.

$$\hat{x}_{k+1} = \mathbb{E}[x_{k+1}|Z_{1:k+1}] \tag{2.26}$$

$$= \int_{x_{k+1}} x_{k+1}\ p(x_{k+1}|Z_{1:k+1}) dx_{k+1} \tag{2.27}$$

$$\approx \frac{1}{m} \sum_{p=1}^{m} x_{k+1}^{(p)} \tag{2.28}$$

2.5 Architectures

Centralized, distributed and decentralized are three major types of architectures used in multisensor-multitarget tracking. The details of these architectures are given in the following sections.

2.5.1 Centralized Tracking

The centralized architecture in general has more than one sensor monitoring the region of interest in detecting and tracking the targets. All the sensors generate the measurements at each revisit time and report those measurements to a central fusion center that fuses all the measurements and updates the tracks. This architecture gives optimal tracking performance. However in a very large surveillance region with many sensors, this architecture may not be feasible because of limited resources such as communication bandwidth and computation power. Centralized architectures are generally simpler to execute since the processing of data at one location can reduce the computational requirement of the algorithm.

2.5.2 Distributed Tracking

An alternative architecture called distributed architecture, also called as "hierarchical architecture", is used alleviate heavy communication and computation requirements of centralized architecture and it is shown in Figure 2.3. Distributed architecture, sensors are connected to Local Fusion Centers (LFCs). Each LFC updates its local tracks based on the measurements from the local sensors and sends its updates to other LFCs. Then, the LFCs performs the track-to-track fusion and may sends back the updated tracks to the other LFCs, if feedback path is available.

2.5.3 Decentralized Tracking

When there is no fusion center that can communicate with all the sensors or LFC in a large surveillance region, neither centralized nor distributed tracking is possible. In this case another architecture called decentralized architecture, in which one has multiple FCs and no CFC, is used. In this architecture, each FC gets the measurements from one or more sensors

Chapter 3

THE PROBABILITY HYPOTHESIS DENSITY FILTER

3.1 Background

In tracking multiple targets, if the number of targets is unknown and varying with time, it is not possible to compare states with different dimensions using ordinary Bayesian statistics of fixed dimensional spaces. However, the problem can be addressed by using Finite Set Statistics (FISST) [23] to incorporate comparisons of state spaces of different dimensions. FISST facilitates the construction of multitarget densities from multiple-target transition functions by computing set derivatives of belief-mass functions [23], which makes it possible to combine states of different dimensions. The main practical difficulty with this approach is that the dimension of the full state space becomes large when many targets are present, which increases the computational load exponentially in the number of targets. Since the PHD is defined over the state space of one target in contrast to the full posterior distribution, which is defined over the state space of all the targets, the computational cost of propagating the full posterior density over time is much lower than propagating the full posterior density. In general, a PHD-based multitarget tracker will experience more difficulty in resolving closely-spaced targets than the tracker based on the full target posterior. However, if the probability density functions of individual targets is highly concentrated around their means compared to the target separation, so that the individual target pdfs do not overlap significantly, it will become possible to resolve targets using the PHD filter as well. A theoretical explanation about the capability of the PHD filter to resolve closely-spaced targets in Gaussian context is given in [23]. By definition, the PHD $D_{k|k}(x_k|Z_{1:k})$, with single target state vector x_k, and given all the measurements up to and time step k, is the density whose integral on any region S of the state space is the expected number of targets $N_{k|k}$ contains in S. That is,

$$N_{k|k} = \int_X D_{k|k}(x_k|Z_{1:k})dx_k \tag{3.29}$$

Since this property uniquely characterizes the PHD and the first-order statistical moment of the full target posterior distribution possesses this property, the first-order statistical moment of the full target posterior, or the PHD, given all the measurement $Z_{1:k}$ up to time step k, is given by the set integral [22].

$$D_{k|k}(x_k|Z_{1:k}) = \int f_{k|k}(\{x_k\} \cup Y|Z_{1:k})\delta(Y) \tag{3.30}$$

More detailed mathematical explanations and derivation of the PHD filter can be found in [22]. The approximate expected target states are given by the local maxima of the PHD. The prediction and update steps of one cycle of PHD filter are given in the following section.

3.1.1 Prediction

In a general scenario of interest, there are target disappearances, target spawning and entry of new targets. We denote the probability that a target with state x_{k-1} at time step $(k-1)$ will survive at time step k by $e_{k|k-1}(x_{k-1})$, the PHD of spawned targets at time step k from a target with state x_{k-1} by $b_{k|k-1}(x_k|x_{k-1})$, and the PHD of new-born spontaneous targets at time step k by $\gamma_k(x_k)$. Then, the predicted PHD, $D_{k|k-1}(x_k|Z_{1:k-1})$, at time k given all measurements up to time $k-1$ is given by

$$
\begin{aligned}
&D_{k|k-1}(x_k|Z_{1:k-1}) \\
&\quad = \gamma_k(x_k) + \int [e_{k|k-1}(x_{k-1}) f_{k|k-1}(x_k|x_{k-1}) + b_{k|k-1}(x_k|x_{k-1})] \\
&\quad \times D_{k-1}|k-1(x_{k-1}|Z_{1:k-1}) dx_{k-1}
\end{aligned} \tag{3.31}
$$

where $f_{k|k-1}(x_k|x_{k-1})$ denotes the single-target Markov transition density. The prediction equation 3.31 is lossless since there are no approximations.

3.1.2 Update

The predicted PHD can be corrected with the availability of measurements Z_k at time step k to get the updated PHD. It is assumed that the number of false alarms is Poisson-distributed with the average rate of λ_k and that the probability density of the spatial distribution of false alarms is $c_k(z_k)$. Let the detection probability of a target with state x_k at time step k be $p_D(x_k)$. Then, the updated PHD at time step k is given by

$$
D_{k|k}(x_k|Z_{1:k}) \cong \left[\sum_{z_k \in Z_k} \frac{p_D(x_k) f_{k|k}(z_k|x_k)}{\lambda_k c_k(z_k) + \psi_k(z_k|Z_{1:k-1})} + (1 - p_D(x_k)) \right] D_{k|k-1}(x_k|Z_{1:k-1}) \tag{3.32}
$$

where the likelihood function $\psi(\cdot)$ is given by

$$\psi_k(z_k|Z_{1:k-1}) = \int p_D(x_k) f_{k|k}(z_k|x_k) D_{k|k-1}(x_k|Z_{1:k-1}) dx_k \tag{3.33}$$

and $f_{k|K}(z_k|x_k)$ denotes the single-sensor/single-target likelihood. The update equation 3.32 is not lossless since approximations are made on predicted multitarget posterior to obtain the closed-form solution 3.32. The reader is reffered to [22] for further explanation.

3.2 Sequential Monte Carlo PHD Filter

This section describes the SMC approach to the PHD filter. This approach provides a mechanism to represent the posterior probability hypothesis density by a set of random samples or particles, which consist of state information with associated weights, to approximate the PHD. The advantage of this method is that the number of particles can be adaptively allocated in a way that a constant ratio between the number of particles and the expected number of targets is maintained. This has a significant effect on the computational complexity of the algorithm that the complexity does not increase exponentially but linear with the increasing number of targets. The SMC implementation considered here is structurally similar to the Sampling Importance Resampling (SIR) type of particle filter [10]. Let the posterior PHD $D_{k-1|k-1}(x_{k-1}|Z_{1:k-1})$ be represented by a set of particles $\left\{w_{k-1}^{(p)}, x_{k-1}^{(p)}\right\}_{p=1}^{L_{k-1}}$. That is

$$D_{k-1|k-1}(\mathrm{x}_{k-1}|Z_{1:k-1}) = \sum_{p=1}^{L_{k-1}} w_{k-1}^{(p)} \delta(\mathrm{x}_{k-1} - \mathrm{x}_{k-1}^{(p)}) \tag{3.34}$$

where $\delta(\cdot)$ is the Dirac Delta function. In contrast to particle filters, the total weight $\sum_{p=1}^{L_{k-1}} w_{k-1}^{(p)}$ is not equal to one; instead, it gives the expected number of targets n_{k-1}^X at time step $(k-1)$, which follows from the property that the integral of the PHD over the state space gives the expected number of targets.

3.2.1 Prediction

We now apply importance sampling to generate state samples that approximate the predicted PHD $D_{k|k-1}(x_k|Z_{1:k-1})$. We generate $\{x_{k|k-1}^{(s)}\}_{p=1}^{L_{k-1}}$ state samples from the proposal density $q_k(\cdot|x_{k-1}, Z_k)$ and i.i.d. state samples $\{x_{k|k-1}^{(p)}\}_{p=L_{k-1}+1}^{L_{k-1}+J_k}$ corresponding to new spontaneously born targets from another proposal density $p_k(\cdot|Z_k)$. That is

$$x_{k|k-1}^{(s)} \sim \begin{cases} q_k(\cdot|x_{k-1}, Z_k) & p = 1, \ldots, L_{k-1} \\ p_k(\cdot|Z_k) & p = L_{k-1}+1, \ldots, L_{k-1}+J_k \end{cases} \tag{3.35}$$

Then, the weighted approximation of the predicted PHD is given by

$$D_{k|k-1}(x_k|Z_{1:k-1}) = \sum_{p=1}^{L_{k-1}+J_k} w_{k|k-1}^{(p)} \delta(x_k - x_{k|k-1}^{(p)}) \tag{3.36}$$

where

$$w_{k|k-1}^{(p)} = \begin{cases} \frac{e_{k|k-1}(\mathrm{x}_{k|k-1}^{(p)}) f_{k|k-1}(\mathrm{x}_{k|k-1}^{(p)}|\mathrm{x}_{k-1}^{(p)}) + b_{k|k-1}(\mathrm{x}_{k|k-1}^{(p)}|\mathrm{x}_{k-1}^{(p)})}{q_k(\mathrm{x}_{k|k-1}^{(p)}|\mathrm{x}_{k-1}^{(p)}, Z_k)} w_{k-1}^{(s)} & p = 1, \ldots, L_{k-1} \\ \frac{\gamma_k(\mathrm{x}_{k|k-1}^{(p)})}{p_k(\mathrm{x}_{k|k-1}^{(p)}|Z_k)} \frac{1}{J_k} & p = L_{k-1}+1, \ldots, L_{k-1}+J_k \end{cases} \tag{3.37}$$

The functions that characterize the Markov target transition density $f_{k|k-1}(.)$, target spawning $b_{k|k-1}$ and entry of new targets $\gamma_k(\cdot)$ in 3.37 are conditioned on the target motion model.

3.2.2 Update

With the available set of measurements Z_k at time step k, the updated particle weights can be calculated by

$$w_k^{*(p)} = \left[(1 - p_D(\mathrm{x}_{k|k-1}^{(p)})) + \sum_{i=1}^{N_k^Z} \frac{p_D(\mathrm{x}_{k|k-1}^{(p)}) f_{k|k}(\mathrm{z}_k^i | \mathrm{x}_{k|k-1}^{(p)})}{\lambda_k c_k(\mathrm{z}_k^i) + \Psi_k(\mathrm{z}_k^i)} \right] w_{k|k-1}^{(p)} \tag{3.38}$$

where

$$\Psi_k(z_k^i) = \sum_{p=1}^{L_{k-1}+J_k} p_D(\mathrm{x}_{k|k-1}^{(p)}) f_{k|k}(\mathrm{z}_k^i | \mathrm{x}_{k|k-1}^{(s)},) w_{k|k-1}^{(p)} \tag{3.39}$$

and $f_{k|k}(\cdot)$ is the single-target/single-sensor measurement likelihood function.

3.2.3 Resample

To perform resampling, since the weights are not normalized to unity in PHD filters, the expected number of targets is calculated by summing up the total weights, i.e.,

$$\hat{n}_k^X = \sum_{p=1}^{L_{k-1}+J_k} w_k^{*(p)} \tag{3.40}$$

Then the updated particle set $\left\{ w_k^{*(p)} / n_k^X, \mathrm{x}_{k|k-1}^{(p)} \right\}_{p=1}^{L_{k-1}+J_k}$ is resampled to get $\left\{ w_k^{(p)} / n_k^X, \mathrm{x}_k^{(p)} \right\}_{p=1}^{L_k}$ such that the total weight after resampling remains n_k^X. Now, the discrete approximation of the updated posterior PHD at time step k is given by

$$D_{k|k}(\mathrm{x}_k | Z_{1:k}) = \sum_{p=1}^{L_k} w_k^{(p)} \delta(\mathrm{x}_k - \mathrm{x}_k^{(p)}) \tag{3.41}$$

3.3 SMC-PHD Filter in Distributed Tracking

Distributed processing over the sensor network can be used to alleviate the problem inherent with the centralized architectures. Among several distributed algorithms implemented using different filters, recently, the particle filter implementation of distributed algorithm have gained much attention. One of the first developments for nonliner/non-Gaussian systems was in [12]. Efficient utilization of communication resources is essential to all distributed algorithms operated in sensor networks.

3.3.1 Distributed Tracking Framework

The underlying sensor network architecture consists of two different types of devices: sensors and nodes. Sensors collect measurements from the targets and report them to computational nodes. Nodes are responsible for running filters to track targets. Information gathered at nodes are shard among nodes. The nodes are interconnected using wireless communication.

3.3.2 Proposed Distributed Algorithm

Our objective in this thesis is to create distributed algorithm based on SMC-PHD filter, which minimizes communication requirements relating to sensor data fusion when multiple time varying targets are present in the region of interest. We assume optimization of sensor resources to collect data and communication issues such as network protocols are already efficient enough. The proposed algorithm maintains SMC-PHD filters in all the computational nodes. This may seem to have high communication costs than centralized computational node due to the maintenance of additional PHD filters that are not needed in centralized architecture. However, a proposal can be given to maintain lower communication cost compared to centralized fusion approach. There are a number of different options available to perform distributed tracking with SMC-PHD filter in a sensor network. The first option is to send all the particles, which represent the posterior density of targets. The second is to send Gaussian mixture representation of posterior density. The first and second options require high bandwidth communications, which can not be handled by practical wireless sensor networks. The third option is to send most appropriate measurements, after eliminating the false alarms, to update the global estimates of the targets. In this thesis, we use the option of communicating measurements among nodes to update the filters. In sensor network, it is quite possible that each node may have enough active sensors to track an object by itself with reasonable tracking accuracy. Therefore, a PHD filter can be used to obtain estimates based on the measurements collected from sensors local to that node. Since these nodes maintain PHD filters based on local measurements, they can also be used in encoding strategy. The proposed framework will be performed in two layers. The first layer collects the measurement data that are local to each node and maintain a local PHD filter using its associated sensors. In the second layer, all the measurement are exchanged to all other nodes in the network and the global PHD filters are maintained.

3.3.2.1 Algorithm

In the proposed algorithm, identical copy of the SMC-PHD filter is maintained at each node. Initially, this is achieved by initializing all the filters using the same random seed. In order to encode the measurement data an intelligent quantization and encoding strategy is used. From the time step $k-1$ to k, particles are propagated while taking into account of measurement prediction covariance. The range of expected measurements is divided into bins depending on the accuracy level required. The contribution of each propagated particle's distribution is integrated over the bins to form the probability density. The measurements are quantized with a nonuniform quantizer where companders are used to perform nonlinear quantization. The probability density in the measurement space is then transformed to companded measurement space. Then, the quantized measurements are encoded using Huffman encoding algorithm with the transformed bin probabilities. The encoded measurements are transmitted to all the other nodes and each node decodes and decompands the data to obtain the quantized measurements. The details of quantization and encoding strategy used in this algorithm is presented in chapter 4.

Each node performs filtering using quantized measurements to obtain the target state estimates. All the nodes use the same set of measurement data to update the filter and thus identical copy of filter is maintained.

The distributed SMC-PHD filter is detailed below.

1. Initialization $k = 0$
   - Initialize SMC PHD filter on each node $n = 1, \ldots, N$ using the same random seed to generate identical particle distribution on all the nodes.
   - For each node $n = 1, \ldots, N$
     - Generate samples $\left\{\mathrm{x}_0^{(p)}\right\}_{p=0}^{L_0}$
2. Quantization and Encoding (For the detail implementation of this step the reader is referred to Section 4.3 of Chapter 4)
   - Local Estimation
     - Perform filtering using the SMC PHD filter acting only on the measurements local to the node.
   - Quantization
     - For each node $n = 1, \ldots, N$
       * For $s = 1, \ldots, L_{k-1}$, predict $\mathrm{x}_{k|k-1}^{(p)}$

    * Calculatc the bin probabilities, $p(z_k|b_j, \mathbf{z}^{(p)}_{1:k-1})$, in the measurement space using predicted measurements and construct the probability density.
    * Identify the regions where the companders need to be placed and the number of companders needed. We use one compander per each target and width of the companding region is limited to $3\sigma_p^c$, where the σ_p^c is the standard deviation of the c th cluster. The compander is placed on the mean value, μ_p^c, of the cluster. In other regions linear quantizer is used.
    * Quantize the measurements, $\tilde{\mathbf{z}}_k$

- Encoding
  - For each node $n = 1, \ldots, N$
    * Calculate the bin probabilities, $\tilde{p}(z_k|b_j, \mathbf{z}^{(p)}_{1:k-1})$, in the transformed measurement space.
    * Use the bin probabilities to form Huffman tree H_f^{k-1} and encode quantized measurements.

3. Reducing the false measurements transmitted over the network

   - Remove the measurements from the queue, if the number of bits in each encoded measurement exceeds a predefined threshold, l. This process is done using the local estimates of the target.

4. Global Estimate

   - For each node $n = 1, \ldots, N$, create the Huffman tree H_f^{k-1} and the quantizer, to reconstruct the quantized data, $\tilde{\mathbf{z}}'_k$.
   - Using set of measurements obtained, perform filtering to obtain the global state estimates.

Chapter 4

QUANTIZATION AND ENCODING

Measurements reported by sensors in a sensor network need to be transmitted in order to perform tracking at high computational nodes called fusion centers. Quantization and encoding play crucial role whereby measurements are quantized and encoded before transmitted. Intelligent quantization and encoding scheme is important to effectively use the communication resources. The goal of this chapter is to provide the necessary theoretical background on quantization and encoding. In addition, this chapter also explains how quantization and encoding can be effectively implemented to perform distributed target tracking with SMC-PHD filters.

4.1 Quantization

An one dimensional quantizer Q with L levels may be defined by a set of $L+1$ decision levels $a_0, a_1, \ldots, a_L$ and a set of L out put levels $y_1, y_2, \ldots, y_L$, as shown in Figure 4.1. When a sample x, the quantity to be quantized, lies in the i^{th} quantizer interval $S_i = a_{i-1} < x \leq a_i$ the quantizer produces the output value $Q(x) = y_i$ [24]. The value of y_i is usually chosen to lie within the interval S_i. The end levels a_0 and a_L are generally chosen to be the smallest

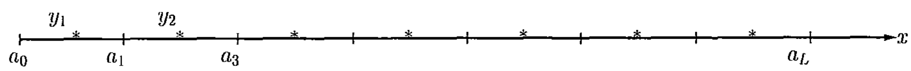

Figure 4.1: Quantization

and largest values the input samples may obtain. The L output levels generally have a finite value and if $L = 2^n$, a unique n-bit binary word can identify a particular output level. The input-output characteristic of a one-dimensional quantizer resembles a staircase. The quantizer intervals, or steps may vary in size.

4.1.1 Uniform Quantization

In the uniform quantization, the step sizes are identical, and the size is determined by the maximum error of the quantizer. The output points are located at the mid-point of these intervals. If the step size is denoted by $\triangle$, then the maximum absolute error is given by $\triangle/2$. In general, uniform quantization is not the most effective way to obtain good quantizer performance.

4.1.2 Nonuniform Quantization

The nonuniform quantization essentially has a nonuniform spacing of decision levels based upon the input probability density [16]. The general model used to represent the nonuniform quantizer is shown in Figure 4.2. The combined function of compression, quantization and expansion is termed companding [18]. It is simply an equivalent way of viewing the operation of a nonuniform quantizer. The quantized samples are transmitted over the network and at the receiver end of the network the quantized samples are decompanded to its original values plus the quantization noise. The variance of the quantization noise associated with the received samples are related to the shape of the companding function $G(.)$ and the number of the bits used for quantization,n. A typical companding function is shown in Figure 4.3.

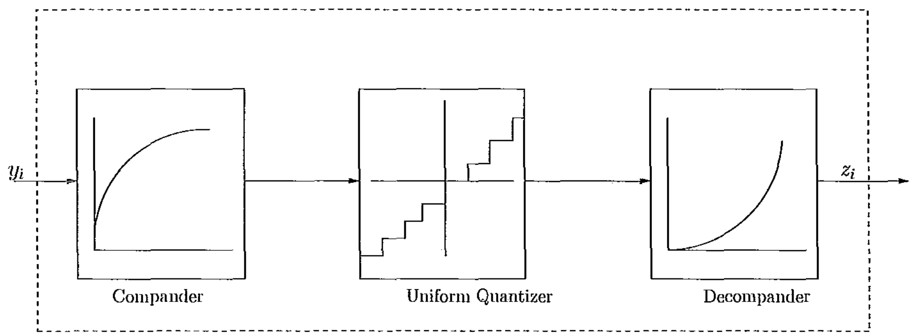

Figure 4.2: Nonuniform quantization

With reference to the Figure 4.3, it is noted that

$$G(y + \Delta_y) - G(y) = \delta \tag{4.42}$$

in which the right hand side is the resolution of the uniform quantizer. Using standard companding techniques, we get

$$\Delta_y \approx \frac{\delta}{\dot{G}(y)} \tag{4.43}$$

where $\dot{G}$ denotes differentiation.

4.2 Encoding

In information theory, an entropy coding is a lossless data compression scheme that is independent of the specific characteristics of the medium. A common method of entropy coding defines a codebook through assigning a code to each symbol. By assigning smaller codes to the more frequent symbols, the average size of each coded symbol can be minimized. This leads to compression over sufficiently large number of encoded symbols. This technique is

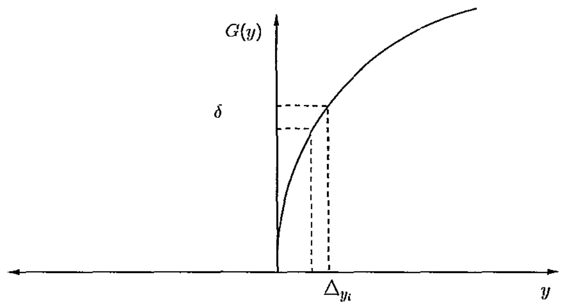

Figure 4.3: A typical compander

known as variable length coding. Generally, variable length coding shows a better performance than fixed-length codes where same size is assigned to all symbols [25]. Two widely used entropy coding techniques are Huffman coding and arithmetic coding. Huffman coding is very simple to implement and it is very efficient when the probabilities of symbols to be sent can be calculated in advance. Hence it is best suited for our application in this thesis.

4.2.1 Huffman coding

Huffman coding assigns a variable length codes to each input symbol where the code and its size are based on the probability of occurrence of the associated symbol. It is necessary to calculate probability of symbols beforehand to the assignment and construction of dictionary. By sorting and analyzing the probability of symbols, a conversion table is constructed so that the symbols with higher probability have the fewest number of bits and no symbol is a prefix to another symbol [25]. Greater compression can be achieved with accurate probability distribution.

4.2.2 Building Huffman codes

The construction of Huffman encoding table is a lengthy process. The probabilities must be sorted so that the lowest two probabilities can be found. These probabilities are added

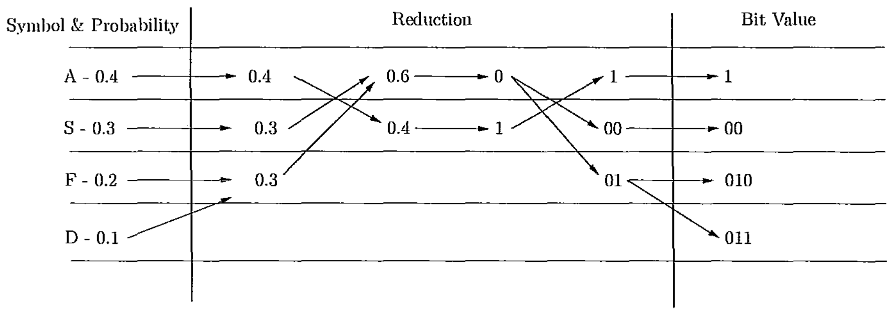

Figure 4.4: Construction of Huffman Encoding Table

together to create a new probability table. This table is sorted, and the process repeats until only two probabilities are left. These probabilities are assigned a value of zero and one. The process is now reversed. At each stage the two expanded probabilities are given a one or zero as they are expanded. The process continues until the table is expanded to its original state. For example assume the message "ASAFAFDAS" is being encoded. The first step is to find the probability for each symbol. A has a probability of 0.4, while S is 0.3, D is 0.1 and F is 0.2. These probabilities are sorted and added to create the table as in Figure 4.4. Once the table has been constructed, the data can be compressed. The compression process is accomplished by a direct conversion of symbols. The entire message is encoded as "10010101010011100", which requires 17 bits. The message normally require 18 bits.

4.3 Measurement quantization and encoding with cascaded companders

The proposed algorithm needs an efficient nonlinear quantization for measurements. Therefore, "cascaded companders", which can quantize measurements from multiple targets, is proposed. This section briefly explains the process of developing the compander. The first step is to construct a probability density of expected measurements to identify the regions where the target originated measurements would lie. The details of this process is given in Section 4.3.1. Measurements that fall in this region is quantized with minimum quantization error via Gaussian companders. Section 4.3.2 explains the cascaded companders. Details of encoding and decoding process using Huffman coding are given in Section 4.3.3. Sections 4.3.5 and 4.3.4 provide details on false alarm elimination process and incorporation of quantization errors into tracking, respectively.

4.3.1 Construction of a Probability Density

The necessity to have identical and accurate probability densities of targets at each nodes, where global SMC-PHD filter is running, is eminent from the fact that the measurements are quantized, encoded and communicated between these nodes based on the probability density. The construction of probability density commences by propagating the densities of particles from the time step $k-1$ to k, taking into account of measurement prediction covariance. The range of expected measurements is divided into bins depending on the accuracy level required. The contribution of each propagated particle's distribution is intergraded over the bins to form the probability density. The Figure 4.5 shows distribution of few sample particles and the quantizer decision boundaries a_{i-1} and a_i.

The probability density of predicted particles $p(z_k^s)$ in the measurement space are given by

$$p(z_k^p) = \mathcal{N}\left(z_k^s; h_k(x_{k|k-1}^p), S_k\right) \tag{4.44}$$

where $h_k(.)$ is a nonlinear function and S_k is the measurement prediction covariance.

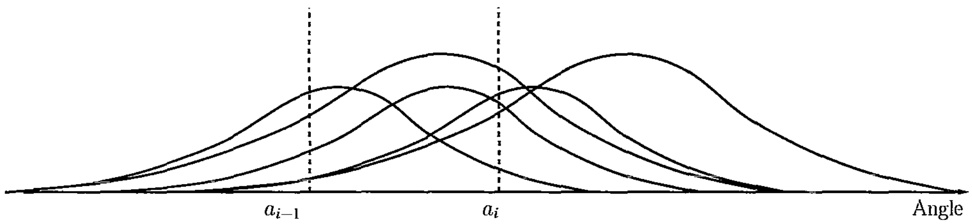

Figure 4.5: Calculation of bin probablities

Then the bin probability is given by

$$p(z_k|b_j, \mathbf{z}_{1:k-1}^{(p)}) = \sum_{s=1}^{L_{k-1}} \int_{a_{i-1}}^{a_i} p(z_k^s)dz \tag{4.45}$$

4.3.2 Measurement Quantizer

Two quantizer strategies are investigated in this thesis: uniform and nonuniform. Uniform quantizer is simple where the measurement space is divided into equal bins based on the number of bits used to encode. The nonuniform quantizer is complex and proposed based on probability density of the targets. Figures 4.6 and 4.7 show quantizers at two different time steps, when one and two targets were present in the environment respectively. The companders are placed in measurement space such that the target originated measurements have less quantization errors than other measurements. In this thesis a Gaussian compander law is used, which is centered on the expected target position and whose curvature is dictated by the standard deviation of the expected position [16]. The compander and expander functions are as follows:

- Compander : $\mathrm{erf}(\frac{\xi}{\sigma\sqrt{6}})$
- Expander : $\sigma\sqrt{6}\ \mathrm{erf}(\xi)$

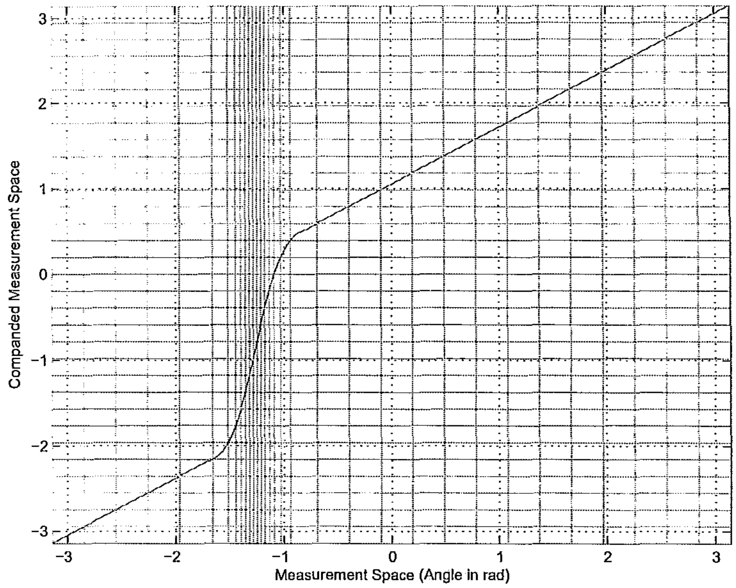

Figure 4.6: A 32 bins compander with one target

where $\mathrm{erf}(\xi) = 2/\sqrt{pi}\int_0^{\xi} exp(-t^2)dt$. We use one compander per each target and width of the companding region is limited to $3\sigma_p^c$, where the σ_p^c is the standard deviation of the c th cluster. The compander is placed on the mean value, μ_p^c, of the cluster. A maximum quantization error is set in other regions of the measurement space, where the compander is not placed, by a liner quantizer. The companders are cascaded when multiple targets measurements are to be quantized.

4.3.3 Measurement encoding and decoding

The original probability density constructed based on expected measurements is transformed to companded measurement space in order to create global Huffman dictionary for encoding. The term global refers to the process or information that is related to global SMC-PHD filter running on every node. Companded measurements are encoded and transmitted over the network. In the receiver, the measurements are decoded before expanding. Same steps are

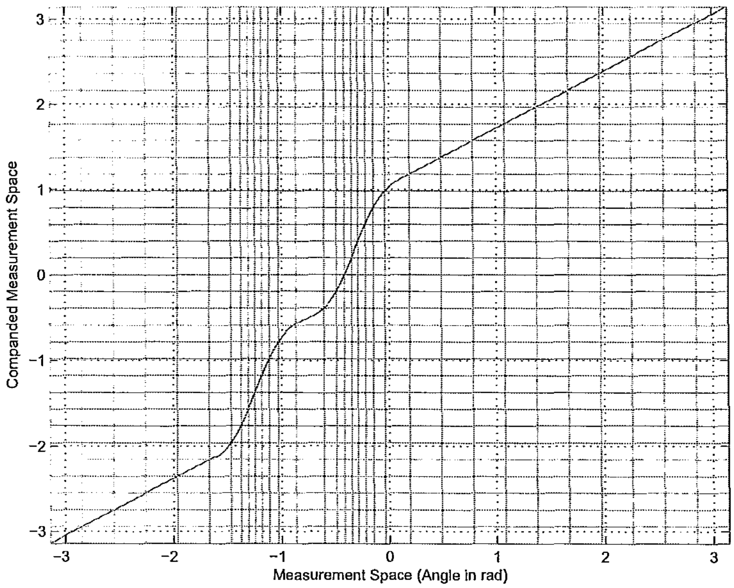

Figure 4.7: A 32 bins compander with two targets

followed to construct a decoding dictionary.

4.3.4 Incorporating Quantization Errors

The insertion of quantized measurement to the SMC-PHD filter is done by updating the current particles by the quantized measurements while taking into account the extra error introduced by the quantization. The error arising from quantization has a uniform distribution. The variances of errors introduced due to quantization is given by,

- Uniform quantization

$$Var(z_k^i|x_k) = \sigma_w^2 + \frac{\delta^2}{12} \quad (4.46)$$

- Non-uniform quantization

$$Var(z_k^i|x_k) = \sigma_w^2 + \frac{\delta^2}{12\dot{G}(y_k^i)^2} \tag{4.47}$$

4.3.5 False Alarms Elimination

Reducing the number of false measurements communicated over the network is important as it will consume much communication resources otherwise. The number of bits in each encoded measurements, based on local Huffman dictionary, can effectively be used to reduce the number of false measurements transmitted over the sensor network. In this approach, we assume since the local PHD filters have most updated information including a birth of a new target and the target generated measurements are most likely to be in a region in which the value of the probability is high. Thus the target generated measurements are most likely to have lesser number of bits in their encoded form compared to false measurements when encoded with local Huffman dictionary. We may assume the measurements that have higher number of bits are not target generated and, by having a threshold value on the number of bits, they can be removed from the set of measurements that are transmitted over the network. Once the measurements are selected to be transmitted, those measurements are encoded with global Huffman dictionary in order to transmit over sensor network. However, when measurements corresponding to new targets are encoded with global Huffman dictionary may produce higher number of bits. It could be noted that the new targets can be identified by the global PHD filter quickly. An indicator function, $\mathcal{I}_h^{(k,i)}$ is used to identify whether the measurement has been communicated or not.

$$\mathcal{I}_h^{(k,i)} = \begin{cases} 1 & H_f^{k-1}(\tilde{\mathbf{z}}_k^i) \leq l \\ 0 & H_f^{k-1}(\tilde{\mathbf{z}}_k^i) > l \end{cases} \tag{4.48}$$

$H_f^{k-1}(\tilde{\mathrm{z}}_k^i)$ is a function that generates Huffman codes for each measurement. l is the cutoff number of bits per measurement. If a measurement in their encoded form is less than the cutoff number of bits, then the measurement is communicated else not.

Chapter 5

POSTERIOR CRAMER-RAO LOWER BOUND

In this Chapter, the posterior covariance of the target state is derived to assess the performance bounds for the proposed algorithm. The recursive Riccati-like formula for the Posterior Cramer-Rao Lower Bound (PCRLB) for state estimation with quantized measurements [21] is used to derive the posterior covariance with measurement origin uncertainty. The fundamental theory behind the CRLB for dynamic trarget tracking can be found in [20]. The Section 5.1 provides a brief review on PCRLB. Incorporating the measurement origin uncertainty in PCRLB is discussed in Section 5.2. In Section 5.3 the PCRLB with quantized measurements is derived.

5.1 Background

Consider the estimation of the state of a dynamical system given by 2.2 and 2.3. The quantized measurements at time k are denoted by $\tilde{\mathbf{z}}_k$. Let $\hat{x}_{k|k}$ denote the updated state estimate at time instant k, using measurement $\tilde{\mathbf{z}}_{1:k}$. The estimation error covariance matrix,

$P_{k|k}$, for unbiased estimator is bounded as follows,

$$P_{k|k} = E[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})^T] \geq J_k^{-1} \tag{5.49}$$

where J_k is the Fisher information matrix. This bound is called PCRLB. The information matrix, J_k, can be computed using Riccati-like recursion:

$$J_{k+1} = D_k^{22} - D_k^{21}(J_k + D_k^{11})^{-1} D_k^{12} \tag{5.50}$$

where

$$D_k^{11} = E[-\Delta_{x_k}^{x_k} \log p(x_{k+1}|x_k)] \tag{5.51}$$

$$D_k^{12} = E[-\Delta_{x_k}^{x_{k+1}} \log p(x_{k+1}|x_k)] \tag{5.52}$$

$$D_k^{21} = (D_k^{12})^T = E[-\Delta_{x_{k+1}}^{x_k} \log p(x_{k+1}|x_k)] \tag{5.53}$$

$$D_k^{22} = E[-\Delta_{x_{k+1}}^{x_{k+1}} \log p(x_{k+1}|x_k)] + \underbrace{E[-\Delta_{x_{k+1}}^{x_{k+1}} \log p(\tilde{\mathbf{z}}_k|x_{k+1})]}_{J_k(\tilde{\mathbf{z}})} \tag{5.54}$$

For linear Gaussian systems, it can be shown that [19]

$$J_{k+1}^{-1} = \left((Q_k + F_k J_k^{-1} F_k^T)^{-1} + E\left[-\Delta_{k+1}^{k+1} \log p(\tilde{\mathbf{z}}_{k+1}|x_{k+1})\right]\right)^{-1} \tag{5.55}$$

with $J_0^{-1} = P_0$.

5.2 PCRLB with Measurement Origin Uncertainty

Let us consider $n_s (\geq 1)$ sensors, and let $\tilde{\mathbf{z}}_k^s$ be the quantized measurement vector from sensor s. It is assumed that measurement noises of sensors are independent. Also due to false alarms, the total number of measurements can vary among sensors at each time step k. Let m_k^s be the total number of measurements from sensor s at time k. Let the observation set at time k from sensor s be

$$\tilde{\mathbf{z}}_k^s = \{\tilde{\mathbf{z}}(i)_k^s\}_{i=1}^{m_k^s} \tag{5.56}$$

where m_k^s in general is random quantity. The are three different possibilities that could make m_k^s number of measurements. First, these m_k^s observations can all be false alarms. Second, there is one true detection and $(m_k^s - 1)$ false alarms and third possibility is that there might be no observations $(m_k^s = 0)$.

Using measurement independent assumption, the measurement information, $J_k(\tilde{\mathbf{z}})$, is given by

$$\begin{aligned} J_k(\tilde{\mathbf{z}}) &= E\left[-\Delta_{x_k}^{x_k} \log[\prod_{s=1}^{n_s} p(\tilde{\mathbf{z}}_k^s|x_k)]\right] \\ &= \sum_{s=1}^{n_s} \left[E[-\Delta_{x_k}^{x_k}(\log p(\tilde{\mathbf{z}}_k^s|x_k))]\right] \\ &= \sum_{s=1}^{n_s} J_k^s(\tilde{\mathbf{z}}) \end{aligned} \tag{5.57}$$

Now, from [26] and [20] we get

$$J_k^s(\tilde{\mathbf{z}}) = \sum_{m_k^s=1}^{\infty} p(m_k^s) J_k^s(z, m_k^s) \tag{5.58}$$

where

$$J_k^s(z, m_k^s) = E[-\Delta_{x_k}^{x_k}(\log p(\tilde{\mathbf{z}}_k^s(i)_{i=1}^{m_k^s}|x_k))] \tag{5.59}$$

and

$$p(\tilde{\mathbf{z}}_k^s(i)_{i=1}^{m_k^s}|x_k) = \prod_{i=1}^{m_k^s} p_0(z_k^s(i)) \times \left[(1-\epsilon(m_k^s)) + \frac{\epsilon(m_k^s)}{m_k^s}\sum_{i=1}^{m_k^s}\frac{p_1(z_k^s(i))}{p_0(z_k^s(i))}\right] \tag{5.60}$$

We assume that false alarms have a uniformly distributed over the observation volume V. Then $p_0(z_k^s(i))$ is given by

$$p_0(z_k^s(i)) = \frac{1}{V} \tag{5.61}$$

Equation 5.60 then becomes

$$p(\tilde{\mathbf{z}}_k^s(i)_{i=1}^{m_k^s}|x_k) = \left[\frac{(1-\epsilon(m_k^s))}{V^{m_k^s}} + \frac{\epsilon(m_k^s)}{m_k^s V^{m_k^s-1}}\sum_{i=1}^{m_k^s} p_1(\tilde{\mathbf{z}}_k^s(i))\right] \tag{5.62}$$

where, $p_1(\tilde{\mathbf{z}}_k^s(i))$ is the pdf of the true observation, which is depend on x_k. The details of obtaining $p_1(\tilde{\mathbf{z}}_k^s(i))$ with the quantized measurement are given in following section.

5.3 PCRLB with Quantized Measurement

Due to the essence of quantization, we know that $\tilde{\mathbf{z}}_k^s(i)$ has a discrete distribution and the only thing we can infer from $\tilde{\mathbf{z}}_k^s(i) = Q(z_k^s(i))$ about $\tilde{\mathbf{z}}_k^s(i)$ is that $a_{(i,k)}^s \le \tilde{\mathbf{z}}_k^s(i) < a_{(i+1,k)}^s$ [21]. The encoding algorithms gives priority to send quantized measurements that can be encoded with less number of bits. Lets assume the cutoff length for encoded measurement is l bits. The equation 4.48 from Chapter 4 can be used to define $p_1(\tilde{\mathbf{z}}_k^s(i))$,

$$\begin{aligned} p(\tilde{\mathbf{z}}_k^s(i)|x_k) &= \mathcal{I}_h^{(k,i)} \times P\{\tilde{\mathbf{z}}_k^s(i)) = Q(y_k^s(i))|x_k\} \\ &= \mathcal{I}_h^{(k,i)} \times P\{a_{(i,k)}^s \le h_k(x_k) + v_k < a_{(i+1,k)}^s|x_k\} \end{aligned} \tag{5.63}$$

For bearing only tracking,

$$z_k^s(i) = \underbrace{tan^{-1}\left(\frac{(y_k - y_k^s)}{(x_k - x_k^s)}\right)}_{h(x_k, x_k(s))} + v_k^s \tag{5.64}$$

where (x_k, y_k) is the position of the target at time k and (x_k^s, y_k^s) is the location of the sth sensor at time k. The measurement error v_k^s is assumed to be a zero mean Gaussian random variable with standard deviation σ_w^s. Then

$$\begin{aligned} p_1(\tilde{\mathbf{z}}_k^s(i)|x_k) &= \mathcal{I}_h^{(k,i)} \times P\{a_{(i,k)}^s - h(x_k, x_k(s)) \leq v_k < a_{(i+1,k)}^s - h(x_k, x_k(s))\,|x_k\} \\ &= \mathcal{I}_h^{(k,i)} \times \int_{a_{(i,k)}^s - h(x_k, x_k(s))}^{a_{(i+1,k)}^s - h(x_k, x_k(s))} \frac{1}{\sigma_w^s \sqrt{2\pi}} \exp\left\{-\frac{t^2}{2(\sigma_w^s)^2}\right\} dt \\ &= \mathcal{I}_h^{(k,i)} \times f_k^{i,s}(x_k) \end{aligned} \tag{5.65}$$

where

$$f_k^{i,s}(x_k) = \left[\Phi\left(\frac{a_{(i+1,k)}^s - h(x_k, x_k(s))}{\sigma_w^s}\right) - \Phi\left(\frac{a_{(i,k)}^s - h(x_k, x_k(s))}{\sigma_w^s}\right)\right] \tag{5.66}$$

It has been proven that the $J_k^s(z)$ is related to $\tilde{J}_k^s(z)$ by following proposition. Where $\tilde{J}_k^s(z)$ is the counter part of $J_k^s(z)$ when there is no measurement origin uncertainty [26].

$$J_k^s(\tilde{\mathbf{z}}) = q_2(P_d, \lambda, \Sigma_k, V)\tilde{J}_k^s(\tilde{\mathbf{z}}) \tag{5.67}$$

where $q_2(P_d, \lambda, \Sigma_k, V)$ is a constant scalar dependant on the probability of detection, the false alarm density, the covariance of observation noise and on the volume of the observation region. Now we calculate the $\tilde{J}_k^s(\tilde{\mathbf{z}})$ with quantized measurements. From equation 5.65 we can define $E\left[-\Delta_{x_k}^{x_k} logp_1(\tilde{\mathrm{z}}_k^s(i)|x_k)\right]$ as follows

$$\begin{aligned}\tilde{J}_k^s(\tilde{\mathrm{z}}) &= E\left[-\Delta_{x_k}^{x_k}\log p_1(\tilde{\mathrm{z}}_k^s(i)|x_k)\right] \\ &= E\left[-\Delta_{x_k}^{x_k}\log \mathcal{I}_h^{(k,i)} \times f_k^{i,s}(x_k)\right]\end{aligned} \tag{5.68}$$

The equation 5.68 can be reduced to following form [21].

$$\tilde{J}_k^s(\tilde{\mathrm{z}}) = -E\left[\mathcal{I}_h^{(k,i)} \times g(x_k, \tilde{\mathrm{z}}_k^s(i))\right] \tag{5.69}$$

$$g(x_k, \tilde{\mathrm{z}}_k^s(i)) = \begin{bmatrix} \frac{\partial^2 \log(f_k^{i,s}(x_k))}{\partial (x_k^1)^2} & \cdots & \frac{\partial^2 \log(f_k^{i,s}(x_k))}{\partial x_k^1 \partial x_k^p} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 \log(f_k^{i,s}(x_k))}{\partial x_k^1 \partial x_k^p} & \cdots & \frac{\partial^2 \log(f_k^{i,s}(x_k))}{\partial (x_k^p)^2} \end{bmatrix} \tag{5.70}$$

where

$$\frac{\partial^2 \log(f_k^{i,s}(x_k))}{\partial x_k^p \partial x_k^q} = \frac{f_k^{i,s}(x_k)\frac{\partial^2 f_k^{i,s}(x_k)}{\partial x_k^p \partial x_k^q} - \frac{\partial f_k^{i,s}(x_k)}{\partial x_k^p}\frac{\partial f_k^{i,s}(x_k)}{\partial x_k^q}}{(f_k^{i,s}(x_k))^2}$$

$$\frac{\partial f_k^{i,s}(x_k)}{\partial x_k^p} = -\frac{H_k^p}{\sqrt{2\pi\sigma_s^2}} \times \left(\exp^{-\frac{(a_{(i+1,k)}^s - h(xk))^2}{2\sigma_x^2}} - \exp^{-\frac{(a_{(i,k)}^s - h(xk))^2}{2\sigma_x^2}}\right)$$

$$\begin{aligned}\frac{\partial^2 f_k^{i,s}(x_k)}{\partial x_k^p \partial x_k^q} &= -\frac{H_k^p H_k^q}{\sqrt{2\pi\sigma_s^3}} \times \\ &\left(\left[a_{(i+1,k)}^s - h(xk)\right]\exp^{-\frac{[a_{(i+1,k)}^s - h(xk)]^2}{2\sigma_x^2}} - \left[a_{(i,k)}^s - h(xk)\right]\exp^{-\frac{[a_{(i,k)}^s - h(xk)]^2}{2\sigma_x^2}}\right)\end{aligned}$$

It is not easy to obtain $E\left[-\Delta_{x_k}^{x_k} logp_1(\tilde{\mathrm{z}}_k^s(i)|x_k)\right]$ analytically. So a numerical method is used with M-run Monte-Carlo simulation.

$$\tilde{J}_k^s(\tilde{\mathbf{z}}) \approx -\frac{1}{M}\sum_{j=1}^{M}\sum_{i=1}^{m_k^{(s,j)}} \mathcal{I}_h^{(k,i)} \times g(x_k, \tilde{\mathbf{z}}_k^s(i)) \mid_{x_k=x_k^j,\ \tilde{\mathbf{z}}_k^s(i)=\tilde{\mathbf{z}}_k^{(s,j)}(i)} \tag{5.71}$$

where x_k^j and $\tilde{\mathbf{z}}_k^{(s,j)}(i)$ are the realizations of state and measurement during the jth Monte-Carlo simulation run. Using equations 5.71 and 5.57 we can write,

$$\tilde{J}_k(\tilde{\mathbf{z}}) \approx -\frac{1}{M}\sum_{s=1}^{n_s}\sum_{j=1}^{M}\sum_{i=1}^{m_k^{(s,j)}} \mathcal{I}_h^{(k,i)} \times g(x_k, \tilde{\mathbf{z}}_k^s(i))) \mid_{x_k=x_k^j,\ \tilde{\mathbf{z}}_k^s(i)=\tilde{\mathbf{z}}_k^{(s,j)}(i)} \tag{5.72}$$

where the $\tilde{J}_k(\tilde{\mathbf{z}})$ is information contribution of measurements from all the sensors. From the equations 5.72 and 5.67 the following equation for calculating $J_k(z)$ with measurement origin uncertainty and quantized measurement is derived.

$$J_k(z) = q_2(P_d, \lambda, \Sigma_k, V)\tilde{J}_k(\tilde{\mathbf{z}}) \tag{5.73}$$

Chapter 6

SIMULATION STUDIES

In this chapter, results of the simulation studies for the proposed distributed algorithm and the quantization and encoding strategy are presented.

6.1 Simulation setup

In the simulations studies we consider a two dimensional typical tracking example to show the effectiveness of the proposed algorithms. As shown in Figure 6.1, it consists two computational nodes placed at $(-15 \times 10^3, 15 \times 10^3)$ and $(15 \times 10^3, 15 \times 10^3)$. Each node has three sensors reporting braring-only observations at a time interval of $T = 30s$. The target motion model, which is nearly constant velocity, has the following linear-Gaussian target dynamics,

$$\mathrm{x_{k+1}} = \mathrm{Fx_k} + \mathrm{v_k} \tag{6.74}$$

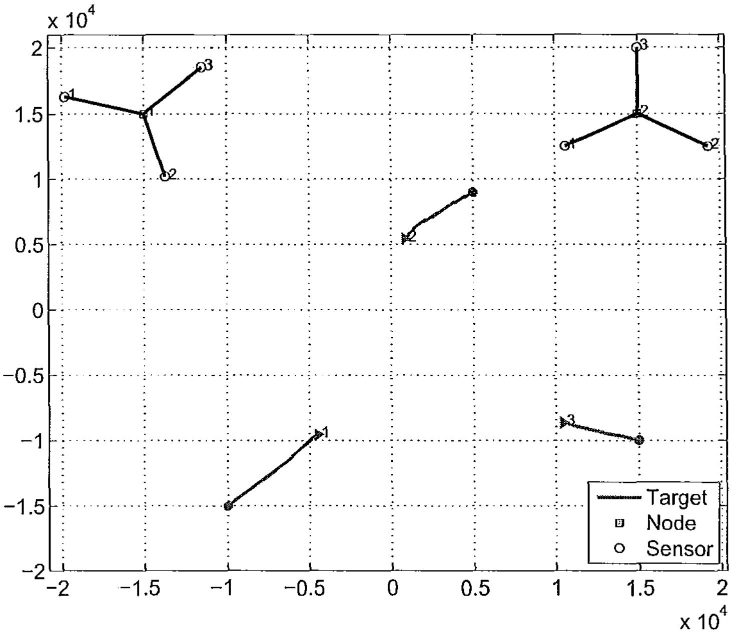

Figure 6.1: The simulation environment

where the target transition matrix F is given by

$$\mathbf{F} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.75}$$

and $\mathbf{v}_k$ is zero-mean white Gaussian noise with covariance $\mathbf{Q}$ given by

$$\mathbf{Q} = \begin{bmatrix} \frac{1}{3}T^3 & \frac{1}{2}T^2 & 0 & 0 \\ \frac{1}{2}T^2 & T & 0 & 0 \\ 0 & 0 & \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ 0 & 0 & \frac{1}{2}T^2 & T \end{bmatrix} q \tag{6.76}$$

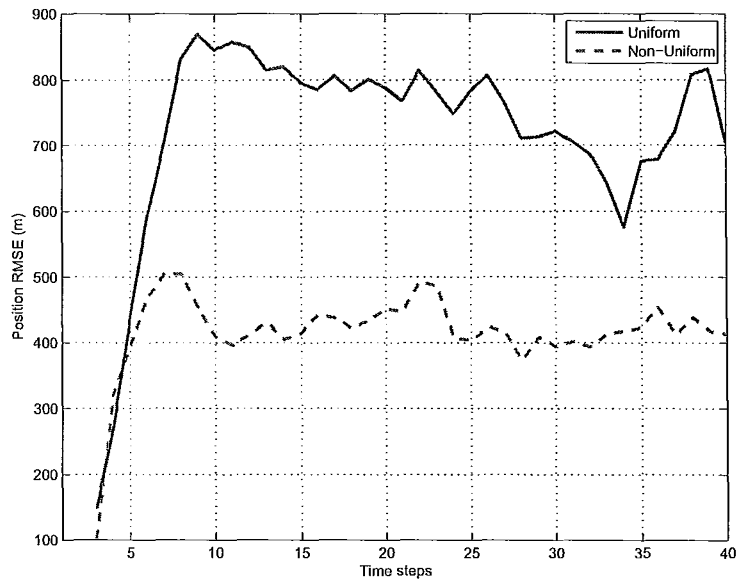

Figure 6.2: Position RMSE with 64 bins quantization for target 1

where $q = 0.001$ is the level of process noise in target motion.

Targets have different stating time and starting positions within the surveillance region. Target 1 and target 2 are present at $k = 0$, and their initial target positions are $(-10 \times 10^3, -15 \times 10^3)$ and $(-5 \times 10^3, 9 \times 10^3)$ m. Target 3 enters late at time $k = 10$ from the position $(15 \times 10^3, -10 \times 10^3)$ m. The targets' initial velocities are $(5, 5)$, $(-4, 3)$, $(-5, 2)$ ms^{-1}. The target trajectories and sensor network arrangement are shown in Figure 6.1. The target generated measurements corresponding to target j on sensor i

$$z_k^{i,j} = \tan^{-1}\left(\frac{y_k^j - y_S^i}{x_k^j - x_S^i}\right) + v_k^i \tag{6.77}$$

where v_k^i is i.i.d sequence of zero-mean Gaussian variables with standard deviation 0.01 rad. The target jth location is dented by (x_k^j, y_k^j) and of ith sensor are denoted by (x_S^i, y_S^i). Additional parameters used in the simulations: The probability of target survival $= 0.99$;

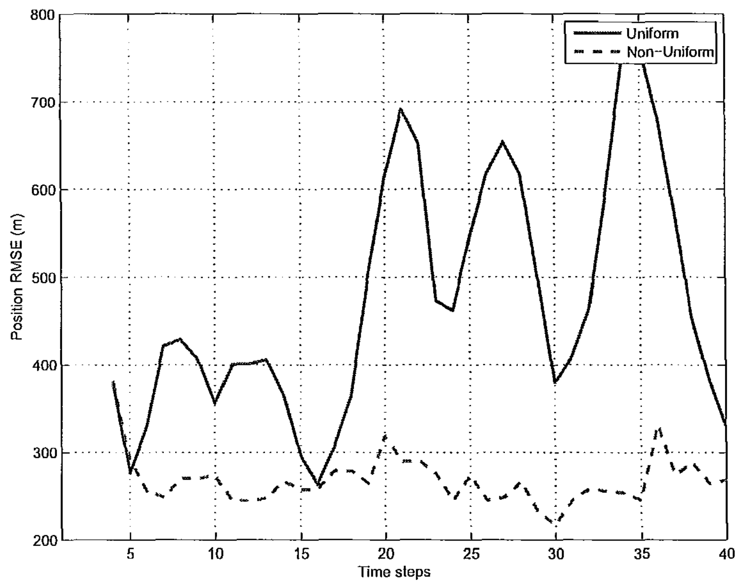

Figure 6.3: Position RMSE with 64 bins quantization for target 2

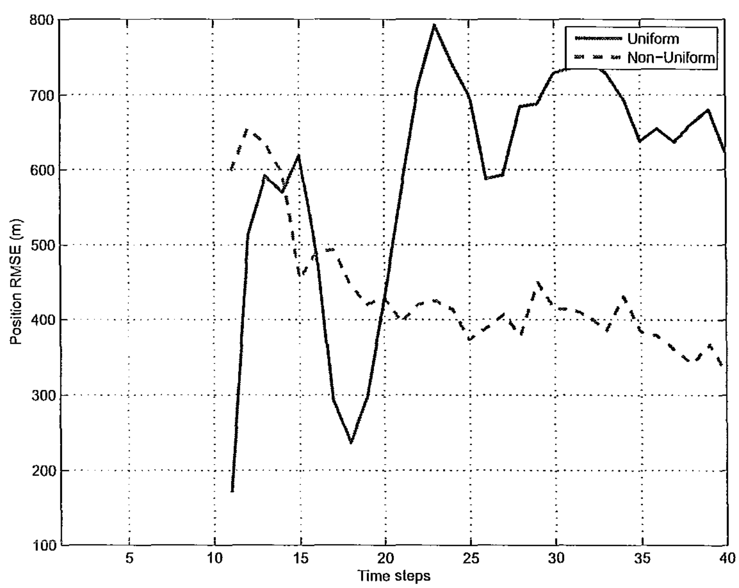

Figure 6.4: Position RMSE with 64 bins quantization for target 3

The probability of target birth = 0.05; The probability of target spawning = 0; Number of particles representing one target = 1000; The average false alarm rate $\lambda = 4 \times 10^{-3} rad^{-1}$. The simulation results are based on 100 Monte Carlo runs.

6.2 Simulation results

Figures 6.2, 6.3 and 6.4 show position Root Mean Square Errors (RMSEs) when the measurements quantized with uniform and nonuniform quantizations are used in tracking target 1, 2 and 3. It is evident from the figures that the proposed non-linear quantization performs better than linear quantization when same number of bins are used to quantize the measurements.

To achieve the same tracking performance as uniform quantization, the proposed nonuniform quantization requires less number of bits for quantization than uniform quantization. This intern reduces the communications significantly. The Figures 6.5, 6.6 and 6.7 show that RMSEs of 64 bins nonuniform quantization and 128 bins uniform quantization are close.

The Figure 6.8 shows the number of bits transmitted when false alarms are eliminated compared with no false alarm elimination and number of bits transmitted without any encoding. The figure indicates that the average number of bits transmitted is substantially reduced. When new targets are introduced the number of bits transmitted increases as the global estimates of the targets are not available. Once the target is initialized the Huffman dictionary takes into account of new target so the encoded measurements have less number of bits. Figures 6.9, 6.10 and 6.11 show that the performance does not change even when false alarm elimination process is in place. PCRLB bounds plotted with estimates of the targets confirms the effectiveness of the proposed distributed SMC-PHD filter algorithm. Figure 6.12 displays the results for Target-1.

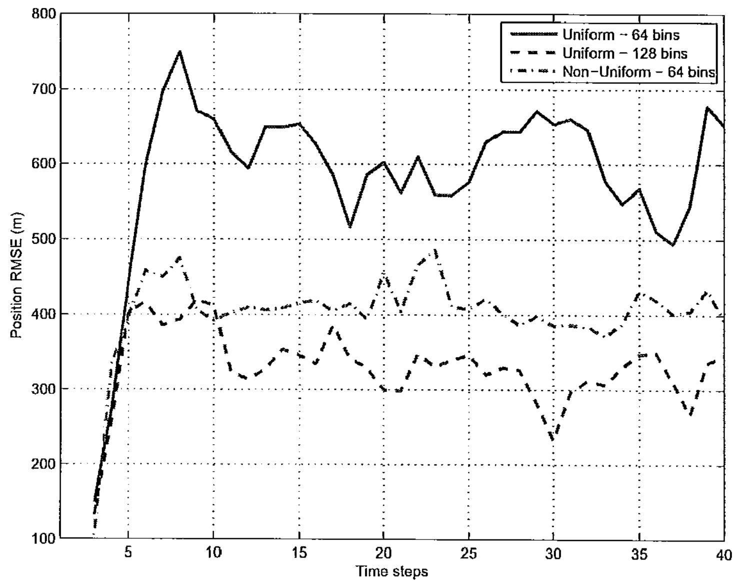

Figure 6.5: Position RMSE, performance comparison for target 1

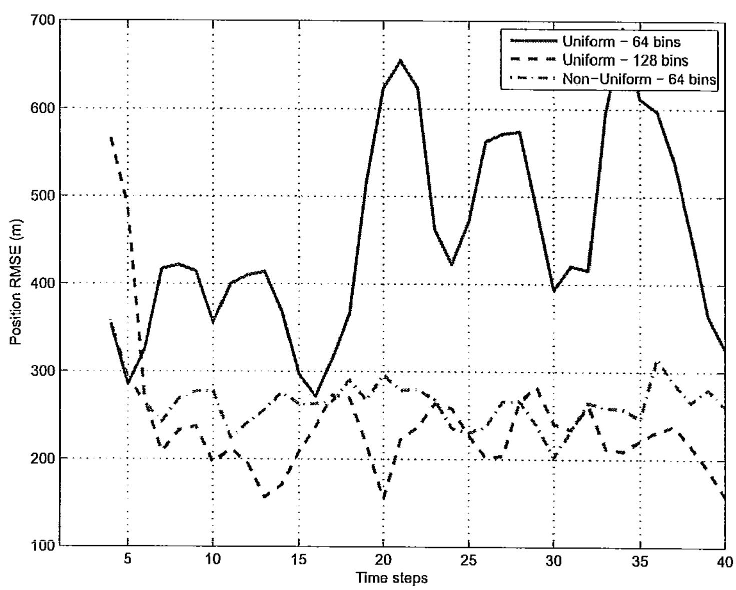

Figure 6.6: Position RMSE, performance comparison for target 2

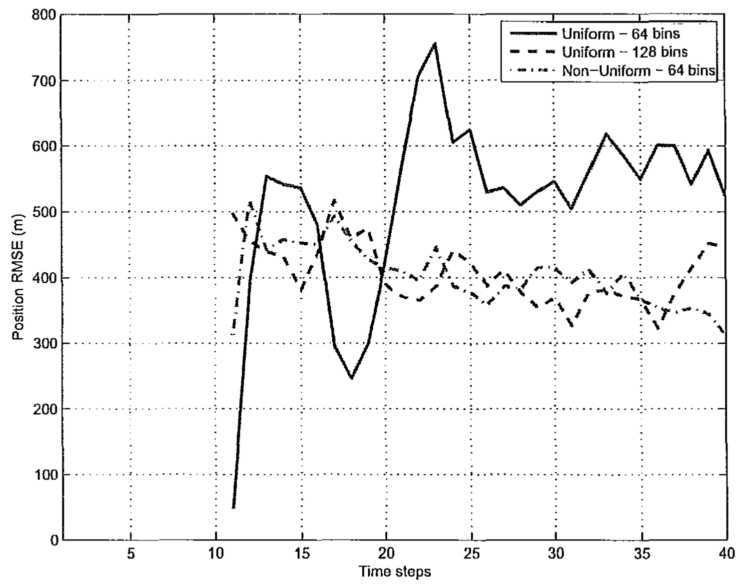

Figure 6.7: Position RMSE, performance comparison for Target 3

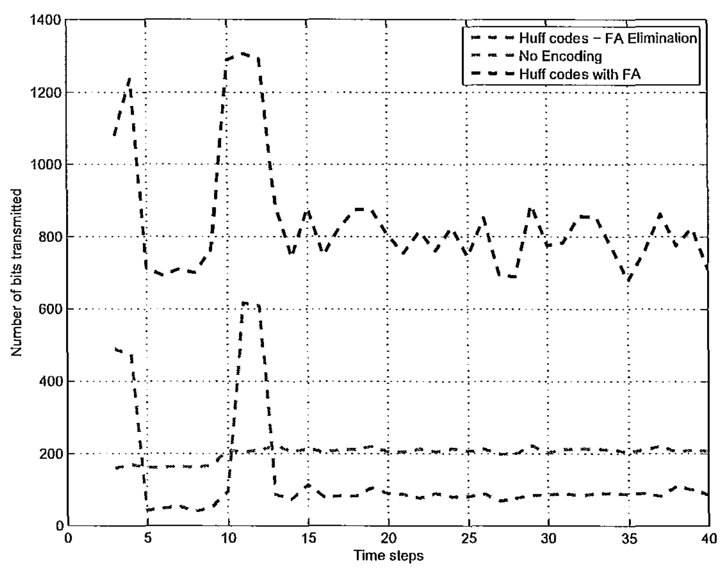

Figure 6.8: Number of bits transmitted with and without false alarms elimination when $\lambda = 3.183 \times 10^{-1} rad^{-1}$

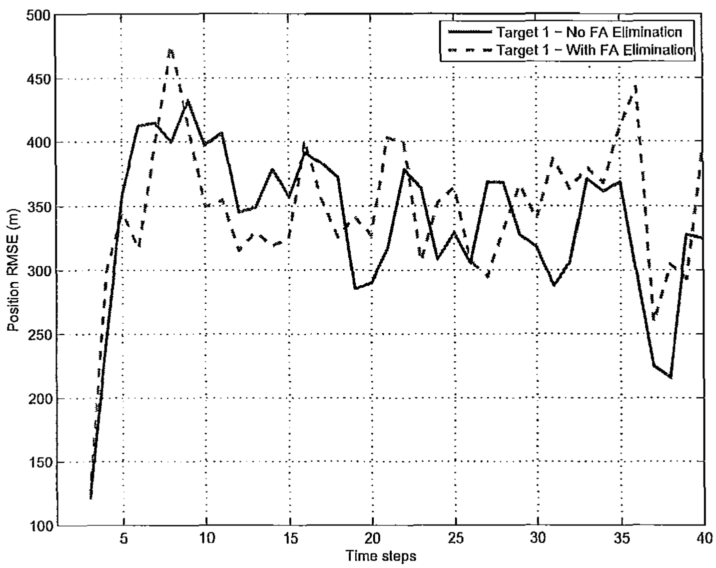

Figure 6.9: Position RMSE, with and without false alarms elimination for target 1

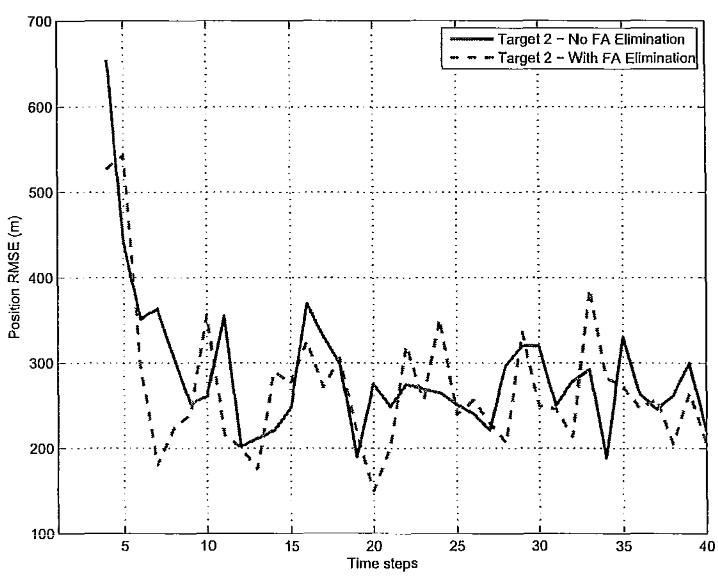

Figure 6.10: Position RMSE, with and without false alarms elimination for target 2

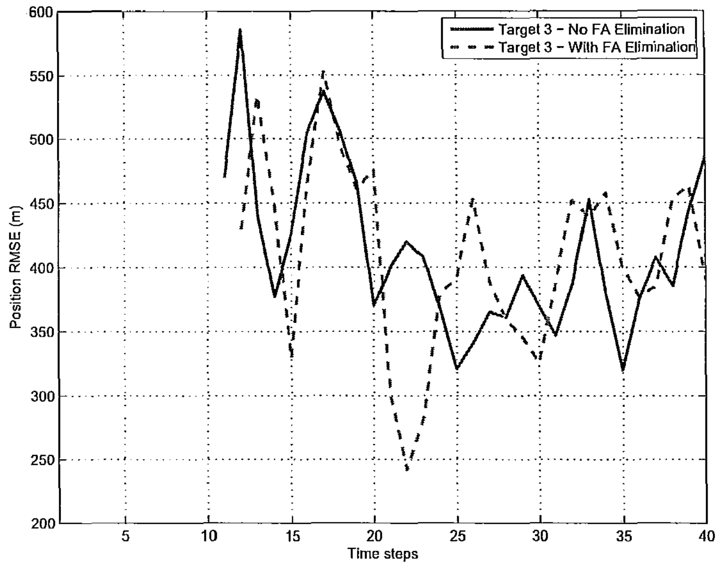

Figure 6.11: Position RMSE, with and without false alarms elimination for target 3

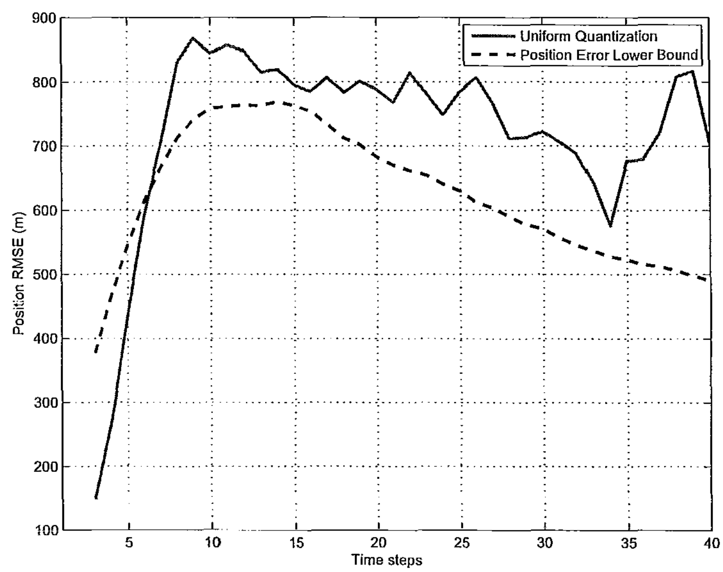

Figure 6.12: Position RMSE and PCRLB with 64 bins uniform quantization for target 1

Chapter 7

SUMMARY

7.1 Conclusions

In this thesis we have considered a distributed implementation of SMC-PHD filter and an efficient quantization and encoding for communicating measurements. Communication resources need to be handled efficiently in sensor networks while maximizing the tracking performance. False alarms take significant communication resources unless it is handled properly. A nonuniform quantization via companding was implemented taking the advantages of the filter properties. It ensures that the target originated measurements are quantized with less errors than others. An effective way of eliminating the false alarms was also implemented. Posterior covariance was derived to access the algorithm using recursive formula for the Fisher Information Matrix. Simulation studies confirms, that the proposed quantization, encoding and false alarm elimination are shown to be more efficient in terms of communication resource utilization and tracking performance. The proposed distributed algorithm for SMC-PHD filter is also shown effective when the results were compared to its performance bound.

Bibliography

[1] Bar-Shalom, Y., Li, X., and Kirubarajan, T., *Estimation with Applications to Tracking and Navigation.* New York: Wiley, 2001.

[2] Li, X. R. and Jilkov, V. P, "Survey of Maneuvering Target Tracking. Part I: Dynamic Models," *IEEE Trans. Aerospace and Electronic Systems,* vol. 39, no. 4, pp. 1333–1364, Oct. 2003.

[3] Bar-Shalom, Y. and Li, X., *Multitarget-Multisensor Tracking: Principles and Techniques.* YBS Publishing, 1995.

[4] Xiong, N. and Sevensson, P.,"Multi-Sensor Management for Information Fusion: Issues and Approaches," *Information Fusion* vol. 3, no. 1, pp. 163–186, Jun. 2002.

[5] Julier, S. "A Skewed Approach to Filtering, "*Proceedings of SPIE on Signal and Data processing of Small Targets*, vol. 3373, pp. 271–282, Orlando, FL, Apr. 1998.

[6] Arulampalam, M. S and Ristic, B., "Comparison of the Particle Filter with Range Parameterized and Modified Polar EKFs for Angle-Only Tracking," *Proceedings of SPIE on Singal and Data Processing of Small Targets*, vol. 4048, pp. 288–299, Orlando, FL, Apr. 2000.

[7] Gordon, N. J, Salmond, D. J, and Smith, A. F. M, "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation," *IEE proceedings-F*, vol. 140, no. 2, pp. 107–113, Apr. 1993.

[8] MacCormic, J. and Blake, A., "Proababilty Exclusion Principle for Tracking Multiple Objects," *Proceedings of the IEEE International Conferences on Computer Vision*, vol. 1, pp. 572–578, Kerkyra, Greece, Sep. 1999.

[9] Doucet, A., De Freitas, J. F. G and Gordon, N. J, *An Introduction to Sequential Monte Carlo Methods*, Springer-Verlag, New York, 2001.

[10] Arulampalam, M. S., Maskell, S., Gordon, N. and Clapp, T., "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking, *IEEE Transactions on Singal Processing*, vol. 50, no. 2, pp. 173–188, Feb. 2002.

[11] Rubin, D. B., "Using the SIR Algorithm to Simulate Posterior Distributions", *Bayesian Statistics*, 3, Oxford Univeristy Press, 1992.

[12] Coates, M. J., "Distributed Particle Filters for Sensor Networks," *Third International Symposium on Information Processing in Sensor Networks*, pp. 99–107, Apr. 2004.

[13] Ing, G. and Coates, M. J., "Parallel Particle Filters for Tracking in Wireless Sensor Networks," *IEEE 6th Workshop on Signal Processing Advances in Wireless Communications*, pp. 935–939, Jun. 2005.

[14] Rosencrant, M., Gordon, G., and Thrun, S., "Decentralized Sensor Fusion with Distributed Particle Filters", *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, Acapulco, Mexico, Aug. 2003.

[15] Sheng, X., Hu, Y. H., and Ramanathan, P., "Distributed Particle Filter with GMM Approximation for Multiple Targets Localization and Tracking in Wireless Sensor Network," *Fourth International Symposium on Information Processing in Sensor Networks*, pp. 181–188, Apr. 2005.

[16] Palmieri, F., Marano, S., and Willett, P., "Measurement Fusion for Target Tracking Under Bandwidth Constraints," *Proceedings of the 2001 IEEE Aerospace Conference on Information Fusion*, Big Sky, MT, Mar. 2001.

[17] Ruan, Y., Willett, P., Marrs, A., Palmieri, F., and Marano, S., "Practical Fusion of Quantized Measurements via Particle Filtering," *IEEE Transaction of Aerospace and Electronic Systems*, vol. 44, no. 1, Jan. 2008.

[18] Peebles, P., *Digital Communication Systems.* Upper Saddle River, NJ: Prentice-Hall, 1987.

[19] Hernandez, M. L., Kirubarajan, T., and Bar-Shalom, Y., "Multisensor Resource Deployment Using Posterior Cramer-Rao Bounds," *IEEE Transaction on Aerospace and Electronic Systems*, vol. 40, no. 2, Apr. 2004.

[20] Zhang, X., Willett, P., and Bar-Shalom, Y., "The Cramer-Rao Bound for Dynamic Target Tracking with Measurement Origin Uncertainty," *Proceedings of the 41st IEEE Conference on Decision and Control*, Les Vegas, Nevada USA, Dec. 2002

[21] Duan, Z., Jilkov, V. P., and Li, X. R., "Posterior Cramer-Rao Bounds for State Estimation with Quantized Measurement", *40th Southestern Symposium on System Theory*, IEEE 2008.

[22] Mahler, R. P. S., "Multitarget Bayes Filtering via First-Order Multitarget Moments," *IEEE Transactions on Aerospace and Electronics Systems*, vol. 39, no. 4, pp. 1152–1178, Oct. 2003.

[23] Mahler, R. P. S., "Multitarget moments and their application to muli-target tracking," *Proceedings of the Workshop on Estimation, Tracking and Fusion: A Tribute to Yaakov Bar-Shalom*, pp. 134-166, Monterey, CA, 2001.

[24] Gray, R. M., "Quantization," *IEEE Transactions on Information Theroy*, vol. 44, no. 6, Oct. 1998.

[25] Sayood, K., *Introduction to Data Compression.* Morgan Kaufman Publishers, 2nd edition, 2000.

[26] Zhang, X. and Willett, P., "Cramer-Rao Bounds for Discrete-Time Linear Filtering with Measurement Origin Uncertainties," *Proceedings of the Workshop on Estimation, Tracking and Fusion: A Tribute to Yaakov Bar-Shalom*, pp. 546–560, Monterey, CA, 2001.