# Joint Multitarget Tracking and Classification Using Aspect-Dependent Measurements

# JOINT MULTITARGET TRACKING AND CLASSIFICATION USING ASPECT-DEPENDENT MEASUREMENTS

By

SUTHARSAN SIVAGNANAM, B.Sc.Eng., M.A.Sc.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree of

Doctor of Philosophy

McMaster University

DOCTOR OF PHILOSOPHY (2009) MCMASTER UNIVERSITY

(Electrical and Computer Engineering) Hamilton, Ontario, Canada

TITLE: **Joint Multitarget Tracking and Classification Using Aspect-Dependent Measurements**

AUTHOR: Sutharsan Sivagnanam

B.Sc.Eng.

University of Peradeniya

Sri Lanka, 2002

M.A.Sc.

McMaster University

Canada, 2005

SUPERVISOR: Dr. T. Kirubarajan

NUMBER OF PAGES: xvii, 153

# Abstract

In this thesis new joint target tracking and classification techniques for aspect-dependent measurements are developed. Joint target tracking and classification methods can result in better tracking and classification performance than those treating these as two separate problems. Significant improvement in state estimation and classification performance can be achieved by exchanging useful information between the tracker and the classifier. Target classification in many target tracking algorithms is not typically done by taking into consideration the target-to-sensor orientation. However, the feature information extracted from the signal that originated from the target is generally a strong function of the target-to-sensor orientation. Since sensor returns are sensitive to this orientation, classification from a single sensor may not give exact target classes. Better classification results can be obtained by fusing feature measurements from multiple views of a target. In multitarget scenarios, handling the classification becomes more challenging due to the identifying the feature information corresponding to a target. That is, it is difficult to identify the origin of measurements. In this case, feature measurement origin ambiguities can be eliminated by integrating the classifier into multiframe data association. This technique reduces the ambiguity in feature measurements while improving track purity.

A closed form expression for multiaspect target classification is not feasible. Then,

training based statistical modeling can be used to model the unknown feature measurements of a target. In this thesis, the Observable Operator Model (OOM), a better alternative to the Hidden Markov Model (HMM), is used to capture unknown feature distribution of each target and thus can be used as a classifier. The proposed OOM based classification technique incorporates target-to-sensor orientation with a sequence of feature information from multiple sensors. Further, the multi-aspect classifier can be modeled using the OOM to handle unknown target orientation. The target orientation estimation using OOM can also be used to find improved estimates of the states of highly maneuverable targets with noisy kinematic measurements. One limiting factor in obtaining accurate estimates of highly maneuvering target states is the high level of uncertainty in velocity and acceleration components. The target orientation information is helpful in alleviating this problem to accurately determine the velocity and acceleration components.

Various simulation studies based on two-dimensional scenarios are presented in this thesis to demonstrate the merits of the proposed joint target tracking and classification algorithms that use aspect-dependent feature measurements.

*To my mother, brother and sister, who sacrificed so much for my well-being, and my wife Thushi, my son Sankeith.*

# Acknowledgements

I would like to thank all my friends, relatives and others who encouraged and supported me during the research leading to this thesis.

First, I would like to express my sincere gratitude to my advisor, Dr. T. Kirubarajan, for giving me an opportunity to work on a very interesting topic and for providing continuous guidance, advice and support throughout the course of this research work. I also thank Dr. A. Sinha and Dr. R. Tharmarasa, for their valuable comments and suggestions. I would also like to thank my committee members, Dr. I. Bruce, Dr. D. Sorina and Dr. X. Wu, for their valuable feedback on the ideas covered in this thesis.

I thank my fellow graduate students in the ECE department, in particular all my friends in the ETFLab, for their help and support. Further, my sincere thanks are due to the ECE department staff, especially Cheryl Gies and Helen Jachna.

Finally, I would like to thank my family for supporting and encouraging me to pursue this degree.

# Contents

# List of Tables

# List of Figures

# Nomenclature

## Acronyms

| | |
|---|---|
| ASW | Anti-Submarine Warfare |
| EKF | Extended Kalman Filter |
| EM | Expectation-Maximization |
| ES | Efficiency Sharpening |
| FIM | Fisher Information Matrix |
| HMM | Hidden Markov Model |
| HRR | High Resolution Range |
| IMM | Interacting Multiple Model |
| IRM | Information Reduction Matrix |
| JPDA | Joint Probabilistic Data Association |
| JMS | Jump Markov System |
| MAP | Maximum A Posteriori |
| MHT | Multiple Hypothesis Tracking |
| MIMO | Multiple Input Multiple Output |
| NN | Nearest Neighbor |
| OOM | Observable Operator Model |

| | |
|---|---|
| PCC | Probability of Correct Classification |
| PCL | Passive Coherent Localization |
| PCRLB | Posterior Cramér-Rao Lower Bound |
| PDA | Probabilistic Data Association |
| pdf | probability density function |
| RCS | Radar Cross Section |
| RMSE | Root Mean Square Error |
| SAR | Synthetic Aperture Radar |
| SIR | Sampling Importance Resampling |
| SMC | Sequential Monte Carlo |
| SN | Strongest Neighbor |
| SNR | Signal-to-Noise Ratio |

# Mathematical notations

| | |
|---|---|
| $e_k$ | association vector |
| $\mathbb{E}(\cdot)$ | expectation operator |
| $f_k$ | nonlinear state transition model at time $k$ |
| $F_k$ | state transition matrix at time $k$ |
| $\hat{F}_k$ | Jacobian of $f_k$ |
| $g$ | gate size |
| $h_k$ | nonlinear measurement model at time $k$ |
| $H_k$ | linear measurement matrix at time $k$ |
| $J$ | Fisher information matrix |
| $J_X$ | prior information matrix |
| $J_z$ | measurement information matrix |
| $\mathcal{N}$ | Gaussian distribution |
| $P$ | covariance of estimate |
| $P_D$ | probability of detection |
| $T$ | number of targets |
| $T_s$ | sampling time interval |
| $\tau$ | linear operator |
| $\Theta$ | target orientation |
| $\xi$ | target class |
| $\psi$ | feature measurement |
| $\mathcal{U}[\cdot]$ | uniform distribution |
| $V_s^f$ | surveillance region's hyper-volume |
| $\hat{x}$ | estimated target state |
| $x_k$ | target state at time $k$ |

| | |
|---|---|
| $x_k^t$ | target state of $t$-th target at time $k$ |
| $X_k$ | stacked target states of multiple targets at time $k$ |
| $y_k$ | state information vector at time $k$ |
| $Y_k$ | information matrix at time $k$ |
| $z_k$ | measurement vector at time $k$ |
| $Z_k$ | sequence of measurements up to time $k$ |
| $\delta(\cdot)$ | Dirac delta function |
| $\Gamma_k$ | covariance matrix of process noise at time $k$ |
| $\lambda$ | spatial density of the false alarms |
| $\lambda_b$ | spatial density of the new targets |
| $\mu_{FA}$ | probability mass function of number of false alarms |
| $\nu_k$ | process noise at time $k$ |
| $\omega_k$ | measurement noise at time $k$ |
| $\sigma$ | standard deviation of measurement error |
| $\Sigma_k$ | covariance matrix of measurement error at time $k$ |
| $\upsilon$ | false alarm distribution |
| $[\cdot]'$ | transpose |
| $[\cdot]_t$ | $t$-th block diagonal matrix |
| $\Delta_\alpha^\beta$ | second-order partial derivative operator |
| $\partial$ | partial derivative operator |

# Chapter 1

# Introduction

## 1.1 Motivation of the Thesis

Conventional target tracking algorithms use only target kinematics in measurement-to-track association. The target classification information can also be used together with kinematic measurements to improve data association [8] resulting in not only better target tracking performance but also improved classification. Furthermore, when tracking a target whose maneuvering capabilities are unknown, the use of a general motion model can lead to poor estimates. This uncertainty is often due to lack of knowledge about the type of the target object. Classification-aided tracking has been studied to some extent in the literature. It has been shown that joint target tracking and classification methods can result in better tracking and classification performances compared with algorithms that treat these two problems separately [16][17][49]. A significant improvement in performance requires two-way exchange of useful information between the tracker and the classifier. The kinematic output of the tracker (e.g., position, velocity and acceleration) should improve the performance

of the classifier, just as the target identity supplied by the classifier should improve track accuracy.

In [16], a Bayesian target classification method based only on the estimate of kinematic is introduced. Their major contribution was to point out the inter-dependence of the target state and the target class, and the integration of this dependence into joint tracking and classification framework. Within this framework, the estimators are provided by a grid-based algorithm, which is known to be very difficult to implement, especially in high dimensional space. In [36], the multitarget tracking and classification problem is addressed (for a fixed number of targets). Their joint tracking and classification implementation is based on nearest-neighbor data association. Moreover, the class estimate always settles on a fixed value. This is problematic when more than one classes appear to be similar during a certain period of time, since the class estimation might lock on the wrong class. Solution to this problem is proposed in [27] (for a single target) with the use of separate particle filter for each possible class. A method for comparing different filter is also given. Joint tracking and classification can be seen as simultaneously dealing with both a fixed model parameter (class) and state variables (position and velocity). Several methods have been proposed to deal with static parameters within a Sequential Monte Carlo (SMC) framework [67][55]. However, the parameters considered were continuous, which is fundamentally different from classification problems.

It has also been shown that the use of target class information in data association improves track purity and track continuity [8]. Classification information obtained from each feature measurement is incorporated into two-frame or two-dimensional data association (one list of tracks and one list of measurements at the latest time) as well as multi-frame (one list of tracks and multiple lists of measurements at the

latest time as well as at some time instants in the past) data association. Target recognition combined with multiple model state estimation is used in [50] for tracking and identification related problems. Amplitude-aided target tracking is presented in [42] for tracking low observable targets. In all these works, feature information from a single sensor is used for classification.

In general, target classification is based on a set of features or attributes that distinguish targets according to their shape, kinematic behavior and electro-magnetic emissions. The distinction between specific target classes and broad classes is the primary difference in recent classification approaches. In systems that define target classes broadly, classification is often based on class-dependent kinematic models [16][14][36]. For example, in [16], the target was assumed to belong to one of two possible classes. The first class was defined as highly agile airplanes such as fighter aircrafts. The second class was defined as lower speed, less maneuverable targets such as commercial aircrafts. However, there may be situations where a highly maneuverable target such as a fighter aircraft could maneuver slowly and travel straight in a line. In which case it will be classified as a commercial aircraft. Another approach uses at least one feature within its set of measurements with the potential discrimination between specific target type. In this work, the feature measurement is the key attribute to the target classification by enabling a trained model to classify target type. Different feature measurements are considered in the literature for target classification. In [59], it was suggested to use high-resolution range profiles. In [47][36], Radar Cross Section (RCS) was chosen. Signal amplitude [43] or signal-to-noise ratio (SNR) also can be considered as feature information for target classification.

## 1.2 OOM based joint target tracking and classification

Once the target classes have been defined, class-dependent statistical models that can be employed in the classifier need to be generated. Training based models such as Hidden Markov Model (HMM) or recently introduced Observable Operator Model (OOM) can be used to model the classifiers. Multiaspect classification is motivated by the fact that it is often difficult to discriminate the targets based on the measurements from a single target-to-sensor orientation. Furthermore, the target orientation as well as the target class are unknown a priori and there are often different orientations at which different targets look alike. For example, target scatter waves from two different targets can be nearly identical when observed from distinct target-to-sensor orientations. The absolute target-to-sensor orientation is typically unknown since the target is distant or concealed. The classification ambiguity may be resolved if the targets are observed at multiple orientations, since two targets are less likely to generate the same set of observations from multiple orientations.

Target classification in many target tracking algorithms is not typically done taking into consideration the target-to-sensor orientation. That is, the feature information from a target is typically modeled without considering the target-to-sensor orientation. However, target-to-sensor aspect angle is an important factor in determining the target class. Since the sensor returns are sensitive to the orientation, classification from a single sensor may not give exact target classes [11][15][57][65]. High Resolution Range (HRR) radar [53][78] and acoustic signals [39] are proven to be sensitive to target aspect angle. The feature information extracted from the signal that originated from the target is generally a strong function of the target-to-sensor

orientation [64]. This will allow to have a discrete set of states of target-to-sensor orientation. Significant amount of work has been done on multiaspect target classification based on HMM using matching pursuits [11][39][52][64][65]. These applications consider only the classification of a single target, specifically a stationary target, which may change its orientation. In multitarget scenarios, handling the classification becomes relatively difficult in terms of obtaining the feature information corresponding to each target. That is, it is difficult to decide which feature measurement corresponds to which target. This is similar to the measurement origin uncertainty in kinematic measurements, which is resolved using data association techniques like multi-assignment techniques or Probabilistic Data Association (PDA) algorithms. In this work, multiaspect classification is integrated into multiframe data association in order to overcome feature measurement origin ambiguities.

HMM based techniques have been used in many applications like speech recognition and computer vision. In addition, they have been successfully used in many classification problems, in particular those with stationary targets [39][52][64][65]. Recently, the OOM has been proposed as a better alternative to the HMM because of the former's solid mathematical basis and better algorithmic properties [37]. Also, the OOM has been proved to be computationally efficient compared to HMM [37][38]. The Efficiency Sharpening (ES) based learning algorithm [38], which is analogous to the Expectation-Maximization (EM) algorithm in HMM, has been proved to be faster than EM. It has also been shown in [38] that the model obtained via ES learning algorithm is more accurate than the corresponding model obtained via HMM. In addition to this, for the same level of model accuracy, OOM requires less dimension and thus lower computation, than HMM.

### 1.2.1 Contribution

In this work, multi-aspect feature data are fused together using training based OOM in order to obtain superior classification results. Also, accurate classification results are used to improve data association process and vice versa, resulting in better overall tracking and classification performance. The OOM classifier is integrated into multi-frame data association to give a complete classification/tracking solution. The joint classification and tracking is accounted for in the $(S+1)$ dimensional (i.e., $S+1$ lists of data matched together) or multiframe assignment framework by redefining the cost function. Also, the OOM based multi-aspect target classification algorithm is developed. Further more, the performance of the OOM based joint target tracking and classification is compared with the HMM based algorithm.

## 1.3 Improved Target Tracking using Orientation Estimates

Conventional tracking algorithms estimate current target position and motion parameters using a sequence of noisy measurements [6][8]. These algorithms typically use kinematic measurements and do not consider the target orientation measurements. Even though highly accurate measurements of range and azimuth are available, it is not always possible to estimate target states accurately. Most targets undergo maneuvers and accelerations causing the target's course to change often. Further, a target's course or orientation is not directly measured by sensors. Target orientation measurement is an additional piece of information that can be used in target tracking in order to obtain better estimates. Multiple Input Multiple Output (MIMO) radar uses widely dispersed antennas to view the target from several angles simultaneously [1].

This new evolving MIMO systems concept facilitates the development of tracking and classification algorithms based on the measurements from multiple views. Multitarget tracking using multistatic active sonar network [25] also utilizes sensor measurements from multiple bistatic pairs. Passive Coherent Localization (PCL) systems [47][36] that utilize commercial television and FM radio signals also use measurements from multiple receivers. These tracking applications provide motivation to fuse multiple sensor measurements to estimate target orientation information. Further, the estimated target orientation can be used in tracking to improve state estimation results.

To enhance the performance of a tracking system, additional information such as target orientation or target class information can be used in addition to kinematic data [44]. However, orientation information may not be as readily available as kinematic measurements and thus, there is a need to preprocess sensor data in order to get additional orientation information. Synthetic Aperture Radar (SAR) image [51] data can be processed to estimate target orientation that is used as additional information to estimate the target maneuvers [69]. Target orientation estimation from image data is presented in [70]. In [77], target orientation observations acquired from an image seeker are used for short-range air-to-air missile guidance problems. It has been shown that the missile guidance accuracy is significantly improved by incorporating the target orientation information. The achieved improvement is due to the correlation between the target's orientation and its maneuvers. The estimated target maneuver using a sequence of frames of the target orientation measurements from a single sensor is presented in [69]. In [41], an aircraft's normal (orthogonal to target course) acceleration is derived to represent the unknown target acceleration using orientation measurements. In that paper, aspect angle measurements are processed using an orientation filter where statistically weighted aspect angles are used together

with the current best estimates of target states to find the unknown acceleration. In [58], acceleration is estimated using a target's past position estimate and used as an additional information for the state estimation. The estimated acceleration term reduces tracking errors, especially during the target maneuvers. Common techniques for target motion modeling are based on kinematic parameters such as velocity and acceleration of target. However, only position measurements (range and angle) and occasionally range rate (Doppler) are directly available for estimating motion parameters. As a result, there can be a significant delay in adapting to unexpected target maneuvers. By incorporating target orientation measurements into tracking, it is possible to reduce the delay. A commonly used approach to determine the target orientation is by matching the feature information to a database of possible target at various orientations. Model based techniques have also been used to estimate the target orientation.

Target orientation information is very helpful in track initialization [43] as well as track maintenance [7] phases. A new track's position can easily be calculated using kinematic measurements. However, target velocity is often unknown and it can be initialized randomly or by using a two-point difference [6]. For example, in a two dimensional tracking scenario, target's course angle is selected uniformly in $[0, 2\pi)$ with some prior knowledge of its target velocities. In this case, most of the initialized tracks with zero velocities fail especially for maneuvering targets. If a target's course is assumed to be along its major axis, then orientation information is useful in determining the course. Hence, the velocity uncertainty can be reduced and therefore, track initialization can be significantly improved . Further, Doppler data cannot easily be incorporated into track initialization without knowing the course. However, if target orientation information is available, then Doppler data can be used

systematically in track initialization. Similarly, track maintenance can be improved by target orientation information. This work mainly focuses on incorporating target orientation information with target tracking and initialization problems.

### 1.3.1 Contribution

In this thesis, the coupling between target orientation and its course is taken into account with the assumption that the velocity of the target is nearly along the major axis of the target. This is however, only a loose assumption and is demonstrated in this thesis using a simulation. The new algorithm uses an Observable Operator Model (OOM) based orientation estimator to estimate the target orientation, which is then used as an additional measurement by the state estimator. In this work, the proposed OOM based orientation estimator uses feature/amplitude information from multiple sensors together with predicted target position from the tracker at each time step to estimate target orientation. The joint processing of the OOM orientation estimator and the target tracker improves both target tracking as well as the orientation measurement estimation. In this work, joint orientation and state estimation of highly maneuvering targets are considered.

## 1.4 Joint Target Tracking and Classification for Multistatic Active Sonar Network

It is increasingly difficult for sonar systems to detect conventional submarines, which have become quieter and with low target strength. The multi-static sonar concept presents a network-centric based approach through the use of active sonar transmitters with multiple receivers to improve target detection performance. Multistatic

sonar is an operational concept for jointly deploying and processing multiple sonar sources and receivers [48]. Several studies have already been done in the area of multistatic sonar techniques [45][46][29][30]. Multistatic sonar network has the potential to improve anti-submarine Warfare (ASW) detection and tracking performance against small, quiet targets in harsh reverberation-limited littoral operating areas. This improved performance comes from increased area coverage, expanded geometric diversity (greater coverage), increased target hold, robustness to sensor losses and jamming, increased probability of detection, and improved target classification via data fusion [19]. Moreover, multistatic systems are flexible. It is possible to use different waveforms at different sources. Therefore, the ping times can be chosen with greater freedom. However, even with its many advantages, tracking a target in a multistatic context remains a challenging proposition. Glint echo detections are few and thus any single receiver will probably not detect the target for longer than a small number of consecutive ping transmissions.

A group of researchers, called Multistatic Tracking Work Group (MSTWG), are currently working on the simulated data sets of several realistic scenarios to obtain suitable algorithms for multistatic active sensor network. This group developed several multitarget tracking algorithms based on MHT,PMHT, ML-PDA algorithm, PHD filter and IMM tracker with multiframe data association. However, the target classification for multistatic active sonar system has not been adequately addressed. This work mainly focuses on the target classification for multistatic system and aims to evaluate the tracking performance with target classification. The bistatic Radar Cross Section (RCS) is selected to perform targets classifications as multistatic consists of multiple bistatic pairs. The statistical models such as OOM and HMM cannot

be employed easily as in the case of monostatic configuration. The bistatic radar configuration requires huge amount amount of training data. For monstatic case, RCS training data from all views ($[0, 2\pi)$ for two dimensional targets)of a target is needed to train these statistical model. On the other hand, in bistatic case, RCS training data are to be obtained for bistatic angles in the range of $[0, 2\pi)$. For each bistatic angle, the target RCS has to be computed for the aspect angle range from $[0, 2\pi)$ [26]. This involves huge computation in model training as well as in likelihood calculation. Therefore, in this work, the fluctuating RCS for bistatic radar configuration is modeled using a Rayleigh distribution, where the aspect dependent average SNR is modeled by a Butterworth approximation [74].

By definition, Radar Cross Section (RCS) [63] is the ratio of power per unit area reflected toward a receiver divided by the power per unit area that was incident on a target. A radar receiver does not directly measure RCS, but it can be calculated using the ratio of the signal power received divided by the noise power (SNR). The radar range equation defines the relationship between SNR and RCS. Fluctuation in the target's RCS certainly cause fluctuation in measured SNR. In this work, it is assumed that the RCS fluctuations are the principal cause of SNR fluctuations [24]. The RCS of a typical target is very sensitive to aspect angles, target orientation and to frequencies. The exact nature of these fluctuations is difficult to model and even measurements are typically not fully repeatable. SNR fluctuation can also result from fluctuation in the environment, transmit power and scan loss. However, it is often impractical to measure RCS over all aspect angles in azimuth and elevation. Target RCS is also related to the target's physical size, but under certain circumstances it may be much larger. For example, a corner reflector has an extremely large target cross section in relation to its size, whereas a stealthy target has a very small cross section. In

many studies, targets are assumed to be a point. Therefore, the Radar Cross Section (RCS) or Sonar Cross Section is independent of the angle of illumination. In the case of bistatic systems, it is independent of the relative angle between the source and the receiver. Many practical targets however, do not show this behavior. A specular return from a target whose broadside is normal to the bistatic angle is much larger than a target that presents some other views. To incorporate aspect dependence, a target strength model needs to be used that represents the aspect angle in the signal return.

### 1.4.1 Contribution

A key feature that bridges the gap between tracking and classification is Radar Cross Section (RCS). By modeling the true deterministic relationship that exists between RCS and target aspect, it is possible to gain both valuable class information and an estimate of target orientation. However, the lack of a close-form relationship between RCS and target aspect prevents the use of Kalman filter or any of its variants. Therefore, Sequential Monte Carlo (SMC) method is selected to handle the nonlinear nature of bistatic RCS in multistatic sonar network. Rayleigh distribution together with approximated aspect dependent average SNR is selected to model bistatic RCS. The joint multitarget tracking and classification algorithm is developed within the SMC framework.

## 1.5 Organization of the Thesis

This thesis is organized as follows: Chapter 2 introduces the general target tracking and classification problem. Chapter 3 presents an overview of the Observable Operator Model. In addition, it reviews three different versions of the OOM learning algorithms. Then a new joint target tracking and classification algorithm based on Observable Operator Model is presented in Chapter 4, followed by an improved target tracking technique using kinematic measurements and orientation estimates in Chapter 5. Next another joint target tracking and classification algorithm for multistatic active sonar network using aspect dependent RCS is explained in Chapter 6. Finally, conclusions are drawn in Chapter 7 with some thoughts on future directions.

## 1.6 Related Publications

### 1.6.1 Journal articles

1. S. Sutharsan, T. Kirubarajan and Michel Pelletier, "Observable Operator Model-Based Joint Target Tracking and Classification", Submitted to IEEE Transactions on Aerospace and Electronic Systems, November, 2008.

2. S. Sutharsan, R. Tharmarasa and T. Kirubarajan, "Improved Target Tracking Using Kinematic Measurements and Target Orientation Estimates", Submitted to IEEE Transactions on Aerospace and Electronic Systems, February, 2009.

3. S. Sutharsan, R. Tharmarasa, T. Lang and T. Kirubarajan, "Joint Target Tracking and Classification Using Aspect Dependent Radar Cross Section For Multistatic Sonar Network", To be submitted to IEEE Transactions on Aerospace and Electronic Systems, July 2009.

### 1.6.2 Conference publications

1. S. Sutharsan and T. Kirubarajan, "Improved Target Tracking Using Kinematic Measurements and Target Orientation Estimates", Proc. SPIE Conference on Signal Processing, Sensor Fusion, and Target Recognition XVIII, Vol. 7336, Orlando, FL, April 2009.

2. S. Sutharsan, R. Tharmarasa, T. Lang and T. Kirubarajan, "Tracking and classification using aspect dependent RCS and kinematic data", Proc. SPIE Conference on Signal and Data Processing of Small Targets, Vol. 6969, Orlando, FL, March 2008.

3. S. Sutharsan and T. Kirubarajan, "Improved Observable Operator Model for Joint Target Tracking and Classification", Proc. SPIE Conf. Signal and Data Processing of Small Targets, Volume 6699, San Diego, CA, August 2007.

4. S. Sutharsan, T. Kirubarajan, A. Sinha and M. Farooq, "Observable Operator Model-Based Joint Target Tracking and Classification", Proc. Signal Processing, Sensor Fusion and Target Recognition XV, Orlando, FL, April 2006.

# Chapter 2

# Multitarget Tracking and Classification

## 2.1 Target Tracking

Determining the likely value of a quantity of interest from incomplete, inaccurate and uncertain observations is called estimation. Continuous estimation of the state of a moving object (also called target) with time is called tracking. The estimation of the current state of a dynamic system from noisy measurements is called filtering and estimating the future state using current measurements is called prediction. The tracking system should produce the measure of accuracy of the state estimates in addition to the state estimates [6][12]. The following sections describe how target tracking is performed.

### 2.1.1 Models

In order to track a target, at least two models are required [6]:

1. System model: describes the evolution of the state with time

2. Measurement model: relates the noisy measurements to the state

Possible system models are:

- nonlinear system model

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

- linear system model

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

The possible measurement models are:

- 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 $f_k$ and $h_k$ are nonlinear functions, $F_k$ and $H_k$ are known matrices, $x_k$ is the state of the target, $z_k$ is the measurement vector, $\nu_k$ is the process noise and $\omega_k$ is the measurement noise at measurement time $k$. In this thesis, it is assumed that $\nu_k$ is Gaussian with zero mean and covariance $\Gamma_k$, and $\omega_k$ is Gaussian with zero mean and covariance $\Sigma_k$.

### 2.1.2 Filtering algorithms

In the Bayesian approach to dynamic state estimation, it is important to construct the posterior probability density function (pdf) of the state given all the received measurements so far. Since this pdf contains all available statistical information, it may be the complete solution to the estimation problem. In principle, an optimal estimate of the state may be obtained from the pdf. In recursive filtering the received measurements are processed sequentially rather than as a batch so that it is not necessary to store the complete measurement set or to reprocess existing measurement if a new measurement becomes available. Such a filter consists of two stages: prediction and update.

The prediction stage uses the system model to predict the state pdf forward from one measurement time to the next. Suppose that the required pdf $p(x_k|Z_k)$ at measurement time $k$ is available, where $Z_k = [z_1, z_2, \ldots, z_k]$. The prediction stage involves using the system model (2.1) to obtain the prior pdf of the state at measurement 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 stage uses the latest measurement to modify the prediction pdf. At next measurement 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, cannot be determined analytically. Analytical solution exists only in a restrictive set of cases.

#### 2.1.2.1 Kalman Filter

The Kalman filter assumes the state and measurement models to be linear and the initial state error and all the noises entering into the system to be Gaussian. 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, and can be parameterized by a mean and covariance [6].

If the state and measurement equations are given by (2.2) and (2.4) respectively, then the Kalman filter algorithm can be viewed as the following recursive relationship [6]:

$$p(x_k|Z_k) = \mathcal{N}(x_k; \hat{x}_{k|k}, P_{k|k}) \tag{2.7}$$

$$p(x_{k+1}|Z_k) = \mathcal{N}(x_{k+1}; \hat{x}_{k+1|k}, P_{k+1|k}) \tag{2.8}$$

$$p(x_{k+1}|Z_{k+1}) = \mathcal{N}(x_{k+1}; \hat{x}_{k+1|k+1}, P_{k+1|k+1}) \tag{2.9}$$

where

$$\hat{x}_{k+1|k} = F_{k+1}\hat{x}_{k|k} \tag{2.10}$$

$$P_{k+1|k} = \Gamma_{k+1} + F_{k+1}P_{k|k}F'_{k+1} \tag{2.11}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}(z_{k+1} - H_{k+1}\hat{x}_{k+1|k}) \tag{2.12}$$

$$P_{k+1|k+1} = P_{k+1|k} - K_{k+1}H_{k+1}P_{k+1|k} \tag{2.13}$$

with

$$S_{k+1} = H_{k+1} P_{k+1|k} H'_{k+1} + \Sigma_{k+1} \tag{2.14}$$

$$K_{k+1} = P_{k+1|k} H'_{k+1} S^{-1}_{k+1} \tag{2.15}$$

In the above, $\mathcal{N}(x; \hat{x}, P)$ is a Gaussian density with argument $x$, mean $\hat{x}$ and covariance $P$, and $[\cdot]'$ denotes the transpose.

This is the optimal solution to the tracking problem if the above assumptions hold [6]. The implication is that no algorithm can do better than a Kalman filter in this linear Gaussian environment.

In many situations the assumptions made above do not hold. Hence the Kalman filter cannot be used as described above, and approximations are necessary.

#### 2.1.2.2 Information Filter

Information filter is an alternative form of Kalman filter [6][31]. Information state vector $y_{k|k-i}$ and information matrix $Y_{k|k-i}$ are defined as

$$y_{k|k-i} = P^{-1}_{k|k-i} \hat{x}_{k|k-i} \tag{2.16}$$

$$Y_{k|k-i} = P^{-1}_{k|k-i} \tag{2.17}$$

The measurement information vector and corresponding matrix are defined as

$$i_k = H'_k \Sigma^{-1}_k z_k \tag{2.18}$$

$$I_k = H'_k \Sigma^{-1}_k H_k \tag{2.19}$$

Then the estimate is given as

$$y_{k|k} = y_{k|k-1} + i_k \tag{2.20}$$

$$Y_{k|k} = Y_{k|k-1} + I_k \tag{2.21}$$

The advantage of information filter is that measurements from multiple sensors can be filtered easily by summing their information matrices and vectors

$$y_{k|k} = y_{k|k-1} + \sum_{j=1}^{n} i_{k,j} \tag{2.22}$$

$$Y_{k|k} = Y_{k|k-1} + \sum_{j=1}^{n} I_{k,j} \tag{2.23}$$

where $n$ is the number of sensors.

#### 2.1.2.3 Extended Kalman Filter (EKF)

If the state and measurement equations are given by (2.1) and (2.3) respectively, then a local linearization of the equations may be a sufficient description of the nonlinearity. Local linearizations of the above functions are

$$\hat{F}_k = \left.\frac{df_k(x)}{dx}\right|_{x=\hat{x}_{k-1|k-1}} \tag{2.24}$$

$$\hat{H}_k = \left.\frac{dh_k(x)}{dx}\right|_{x=\hat{x}_{k|k-1}} \tag{2.25}$$

The EKF is based on that $p(x_k|Z_k)$ is approximated by a Gaussian. Then all the equations of the Kalman filter can be used with this approximation and the linearized functions [6].

If the true density is substantially non-Gaussian, then a Gaussian cannot describe

it well. In such cases, particle filters will yield an improvement in performance in comparison to the EKF.

#### 2.1.2.4 Particle Filtering

The Particle Filter [28] (see also [4] and [23]) provides a mechanism for representing the density $p(x_k|Z_k)$ of the state vector $x_k$ at sampling time $k$ as a set of random samples (particles) $\{x_k^{(i)} : i = 1, 2, \ldots, m\}$, with associated weights $\{w_k^{(i)} : i = 1, 2, \ldots, m\}$, where $m$ is the number of particles. It then uses the principle of Importance Sampling to propagate and update these particles and their associated weights when new measurements become available [22].

Importance Density is taken to be the prior $p(x_k|x_{k-1})$ and uses the method of Sampling Importance Resampling (SIR) to produce a sample of equally weighted particles that approximate $p(x_k|Z_k)$, i.e.,

$$p(x_k|Z_k) \approx \frac{1}{m} \sum_{i=1}^{m} \delta(x_k - x_k{}^{(i)}) \tag{2.26}$$

where $\delta(.)$ is the Dirac delta function. The SIR method works as follows:

- **Prediction**: For each particle $x_{k-1}^{(i)}$, generate $\nu_{k-1}^{(i)}$ according to the known distribution of the transition noise and then a sample $x_{k|k-1}^{(i)}$ from the prior distribution $p(x_k|x_{k-1})$ can be obtained using the state propagation equation (2.1).

- **Weighting:** The information given by the observation can be utilized to find the importance weights. Each particle is given an importance weight $w_k^{(i)}$ using the formula

$$w_k^{(i)} = p(z_k|x_{k|k-1}^{(i)}) \tag{2.27}$$

- **Resampling:** The weighted samples will be resampled to eliminate those with low weights, multiply those with high weights and regenerate those with equal weights. The new $m$ particles are sampled with replacement from $\{x^{(1)}_{k|k-1}, x^{(2)}_{k|k-1}, \ldots, x^{(m)}_{k|k-1}\}$ so that the probability of sampling particle $i$ is proportional to $w^i_k$. Then new samples $\{x^{(1)}_k, x^{(2)}_k, \ldots, x^{(m)}_k\}$ will have equal weights $(1/m)$.

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

$$\hat{x}_k = \mathbb{E}[x_k|Z_k] \tag{2.28}$$

$$\approx \frac{1}{m}\sum_{i=1}^{m} x_k{}^{(i)} \tag{2.29}$$

The strength of the technique is that it is not restricted by the assumptions of linearity and Gaussian noise that are necessary in implementing Kalman filter based techniques.

### 2.1.3 Multiple Model Estimator

As mentioned earlier, since the Kalman Filter (KF) assumes a fixed model for the state evolution, its performance is not satisfactory in terms of estimation error when it is used to estimate the state of the system whose model varies with time. In such scenario, Multiple Model (MM) estimators perform better than a KF. In the MM approach, it is assumed that the system obeys one of a finite number of models and a Bayesian framework is used. By assuming that prior probabilities of each model being correct, (i.e., system is in particular mode), the corresponding posterior probabilities are obtained.

A MM estimator can be static or dynamic. In the case of static MM estimator,

the system model is fixed, that is, no switching from one mode to another occurs during the estimation process. Even though the system model is fixed, since each model in the mode set has its own dynamics, the overall estimator is dynamic. The static MM estimator is useful if there is an ambiguity in the system model,but it does not switch. The use of static MM estimator has become obsolete [6].

Dynamic MM estimator considers mode switching, i.e., the model that system obeys can change during the estimation process. As this is the case in the proposed joint tracking and classification system and orientation aided tracking systems, only the dynamic MM estimators are described in this section. First the optimal MM estimator is described. Since this is not computationally feasible, sub-optimal dynamic MM estimators have been developed and the most commonly used dynamic MM method Interacting Multiple Model (IMM) is explained in this chapter.

#### 2.1.3.1 The Optimal Dynamic Multiple Model Estimator

A linear system that can undergo model switching in time can be described by the following equations:

$$x(k) = F[M(k)]x(k-1) + \nu[k-1, M(k)] \tag{2.30}$$

$$z(k) = H[M(k)]x(k-1) + w[k, M(k)] \tag{2.31}$$

where $M(k)$ denotes the mode or model at time $k$. The mode at time $k$ is assumed to be among the possible $N$ modes,

$$M(k) \in \{M_j\}_{j=1}^{N} \tag{2.32}$$

The structure of the system and the statistics of noise can be different from model to model. For example

$$F[M_j(k)] = F_j \tag{2.33}$$

$$\nu[k-1, M_j(k)] \sim \mathcal{N}(\mu_j, Q_j) \tag{2.34}$$

The $l^{\text{th}}$ mode history or sequence of models through time $k$ is denoted as

$$M^{k,l} = \{M_{i_{1,l}}, \ldots, M_{i_{k,l}}\} \tag{2.35}$$

where $i_{\kappa,l}$ is the model index at time $\kappa$ from history $l$ and

$$1 \leq i_{\kappa,l} \leq N \qquad \kappa = 1, \ldots, k \tag{2.36}$$

It is assumed that the mode switching is a first order Markov process or Markov chain with known transition probabilities

$$p_{ij} \triangleq P\{M(k) = M_j | M(k-1) = M_i\} \tag{2.37}$$

These mode transition probabilities are assumed to be time-invariant and independent of the base state.

The event that the model $j$ is in effect at time $k$ is denoted as

$$M_j(k) = \{M(k) = M_j\} \tag{2.38}$$

The conditional probability of the $l^{\text{th}}$ mode history is denoted as

$$\mu^{k,l} \triangleq P\{M^{k,l}|Z^k\} \tag{2.39}$$

The $l^{\text{th}}$ sequence of models through time $k$ can be written as

$$M^{k,l} = \{M^{k-1,s}, M_j(k)\} \tag{2.40}$$

where sequence $s$ through $k-1$ is its parent sequence and $M_j$ is its last element. Then, in view of the Markov property,

$$P\{M_j(k)|M^{k-1,s}\} = P\{M_j(k)|M_i(k-1)\} = p_{ij} \tag{2.41}$$

where $i = s_{k-1}$, the index of the last model in the parent sequence $s$ through $k-1$.

The conditional pdf of the state at $k$ is obtained using the total probability theorem with respect to mutually exclusive and exhaustive set of events (2.40), as a Gaussian mixture with an exponentially increasing number of terms

$$p[x(k)|Z^k] = \sum_{l=1}^{N^k} p[x(k)|M^{k,l}, Z^k]p[M^{k,l}|Z^k] \tag{2.42}$$

Each mode sequence has to match a filter. Therefore, it can be seen that an exponentially increasing number of filters are needed to estimate the (base) state that makes the optimal approach infeasible.

The probability of the mode history is obtained using Bayes' formula as

$$\begin{aligned}
\mu^{k,l} &= P\{M^{k,l}|Z^k\} && (2.43)\\
&= P\{M^{k,l}|z(k), Z^{k-1}\} && (2.44)\\
&= \frac{1}{c}p[z(k)|M^{k,l}, Z^{k-1}]P\{M^{k,l}|Z^{k-1}\} && (2.45)\\
&= \frac{1}{c}p[z(k)|M^{k,l}, Z^{k-1}]P\{M_j(k), M^{k-1,s}|Z^{k-1}\} && (2.46)\\
&= \frac{1}{c}p[z(k)|M^{k,l}, Z^{k-1}]P\{M_j(k)|M^{k-1,s}, Z^{k-1}\}\mu^{k-1,s} && (2.47)\\
&= \frac{1}{c}p[z(k)|M^{k,l}, Z^{k-1}]P\{M_j(k)|M^{k-1,s}\}\mu^{k-1,s} && (2.48)
\end{aligned}$$

where $c$ is the normalization constant.

If the current mode depends only on the previous mode only (i.e., it is a Markov chain), then

$$\mu^{k,l} = \frac{1}{c}p[z(k)|M^{k,l}, Z^{k-1}]P\{M_j(k)|M_i(k-1)\}\mu^{k-1,s} \quad (2.49)$$

or

$$\mu^{k,l} = \frac{1}{c}p[z(k)|M^{k,l}, Z^{k-1}]p_{ij}\mu^{k-1,s} \quad (2.50)$$

where $i = s_{k-1}$ is the last model of the parent sequence $s$.

The above equation shows that the conditioning on the entire past history is needed even if the random parameters are Markov.

#### 2.1.3.2 Practical Implementation of Multiple Model Estimator

As the number of filters required in the optimal dynamic multiple model estimator increase exponentially, it is not computationally possible to realize it. Therefore suboptimal algorithms have been developed. However, most commonly used efficient

implementation of suboptimal MM estimator is the Interacting Multiple Model (IMM) estimator and explained below.

#### 2.1.3.3 IMM Estimator Algorithm

- Mixing probability calculation: Each cycle of the IMM estimator starts with the estimators of the individual filters from the previous cycle. These $N$ estimates are mixed probabilistically giving $N$ estimates, which are the input of the $N$ filters for this cycle. The weights given in the mixing are the mixing probabilities. Mixing probability is the probability that the model $M_i$ was in effect at time $k$ (previous cycle) given that the model $M_j$ is in effect at time $k+1$ (this cycle) and the measurements up to time $k$, $Z^k$.

$$\begin{aligned} \mu_{i|j}(k|k) &= P\{M_i(k)|M_j(k+1), Z^k\} && (2.51) \\ &= \frac{1}{\bar{c}_j} P\{M_j(k+1)|M_i(k), Z^k\} P\{M_i(k)|Z^k\} && (2.52) \\ &= \frac{1}{\bar{c}_j} p_{ij}\mu_i(k) \qquad i,j = 1,2,\cdots,N && (2.53) \end{aligned}$$

where $[p_{ij}]$ is the Markov chain transition probability matrix. These mode transition probabilities are assumed to be time invariant and independent of the base state. $\mu_i(k)$ is the probability that the state corresponds to model $M_i$ at time $k$. $\bar{c}_j$ is the normalizing constant and given by

$$\bar{c}_j = \sum_{i=1}^{N} p_{ij}\mu_i(k) \qquad (2.54)$$

- Mixing ($j = 1,2,\cdots,N$): With the mixing probabilities as weights, estimates of all the filters at time $k$ are mixed to produce the initial estimates for each

filter.

$$x^{0j}(k|k) = \sum_{i=1}^{N} \hat{x}^i(k|k)\mu_{i|j}(k|k) \tag{2.55}$$

where $\hat{x}^i(k|k)$ is the estimate of the filter matched model $M_i$ at time $k$ and the associated covariance is given by

$$P^{0j}(k|k) = \sum_{i=1}^{N} \mu_{i|j}(k|k)\{P^i(k|k) + [\hat{x}^i(k|k) - \hat{x}^{oj}(k|k)][\hat{x}^i(k|k) - \hat{x}^{oj}(k|k)]'\} \tag{2.56}$$

where $P^i(k|k)$ is the covariance associated with the estimate of the filter matched to model $M_i$ at time $k$.

- Model probability update ($j = 1, 2, \cdots, N$): Initial conditions obtained above and the measurement at time $k+1$ are input to each filter. In addition to the state estimate and associated covariance, filters also output the likelihood functions. The mode probabilities are updated using the likelihood function as

$$\mu_j(k+1) = \frac{1}{c}\Lambda_j(k+1)\bar{c}_j \tag{2.57}$$

where $\bar{c}_j$ is defined in (2.54) and the normalization constant $c$ is given by

$$c = \sum_{j=1}^{N} \Lambda_j(k+1)\bar{c}_j \tag{2.58}$$

- Overall estimate: State estimate of the IMM is calculated using the mixture equation

$$\hat{x}(k+1|k+1) = \sum_{j=1}^{N} \mu_j(k+1)\hat{x}^j(k+1|k+1) \tag{2.59}$$

and the covariance associated with this estimate is given by

$$P(k+1|k+1) = \sum_{j=1}^{N} \mu_j(k+1)\{P^j(k+1|k+1)[\hat{x}^j(k+1|k+1) - \hat{x}(k+1|k+1)][\hat{x}^j(k+1|k+1) - \hat{x}(k+1|k+1)]'\}. \quad (2.60)$$

It is important that even though the IMM estimator does not make a hard decision; it does not give unity probability for the map that is active, but the mode probability corresponding to the active map will be the highest. That is, based on the mode probabilities one can make the decision of which model is active at a particular time.

## 2.2 Multitarget Tracking

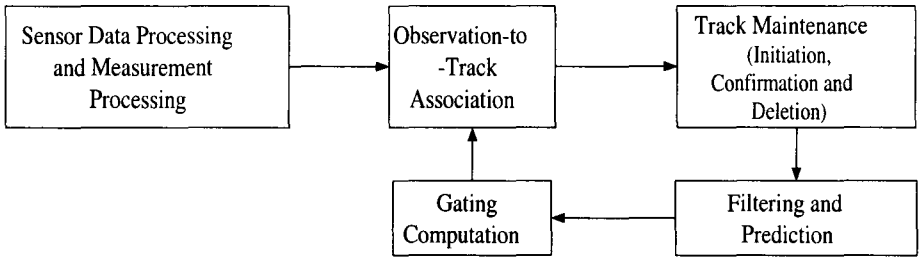



Figure 2.1: Basic elements of a conventional multitarget tracking system.

Figure 2.1 illustrates the block diagram of a conventional multitarget tracking system. A signal processing unit converts the signals from the sensor to measurements, which is the input of the multiple target tracking system. In the tracking system, track is a symbolic representation of a target moving through the surveillance region and is represented by a filter state. The incoming measurements are used for the track maintenance, which refers to the functions of track initiation, confirmation and

deletion [6][12]. Measurements, which are not assigned to any of the existing tracks, can initiate new tentative tracks, and becomes confirmed when the measurements included in that track satisfy the confirmation criteria. Similarly, a track that is not updated within a reasonable interval becomes degraded and will be deleted if not updated.

For existing tracks, a validation procedure is used to limit the region in the measurement space where the tracking system looks to find the measurement from the targets. Measurements outside the validation region can be ignored since they are unlikely to have originated from the target of interest. It is possible to have more than one measurement in the validation region, and a more detailed association technique is used to determine the final pairings. After updating the tracks with associated measurements, tracks are predicted ahead to the arrival time for the next set of measurements. Gates are placed around these predicted positions and the processing cycle repeats.

If the true measurement conditioned on the past is normally (Gaussian) distributed, then its probability density function given by

$$p(z_{k+1}|Z_k) = \mathcal{N}[z_{k+1}; \hat{z}_{k+1|k}, S_{k+1}] \tag{2.61}$$

where $\hat{z}_{k+1|k}$ is the predicted (mean) measurement at time $k+1$ and $S_k + 1$ is the measurement prediction covariance given by (2.14), then the true measurement will be in the following region

$$\mathcal{V}(k+1, \gamma) = \{z : [z - \hat{z}_{k+1|k}]' S_{k+1}^{-1} [z - \hat{z}_{k+1|k}] < \gamma\} \tag{2.62}$$

with the probability determined by the gate threshold $\gamma$. The region defined by (2.62)

is called gate or validation region.

### 2.2.1 Data association

#### 2.2.1.1 Well-separated targets

The problem of tracking well-separated 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:

- true measurement (if detected and falls within the gate)
- false alarms

The common mathematical model for such false measurements are with following assumptions:

- their spatial distribution is uniform within the surveillance region
- they are independent across time

Then the problem, data association, is that of associating the measurements in each validation region with the corresponding track (target). The possible approaches are [7]:

- Nearest Neighbor (NN): is the simplest possible approach, and uses the measurement nearest to the predicted measurement as if it were the correct one.
- Strongest Neighbor (SN): selects the strongest measurement among the validated ones.
- Probabilistic Data Association (PDA): is a Bayesian approach, and associates probabilistically all the validated measurements to the target of interest.

PDA is the standard technique used for data association in conjunction with the Kalman filter or the extended Kalman filter. In most particle filtering algorithms, NN is used since it requires less computation.

#### 2.2.1.2 Closely-spaced targets

When the targets are closely spaced, one measurement could originate from any one of the target or clutter. Then the following assumptions are made:

- one measurement is originated from at most one target
- one target can generate at most one measurement

Under the above assumptions, following approaches are possible for data association [7]:

- Joint Probabilistic Data Association (JPDA): is a target oriented approach, and it is an extension of PDA.
- Multiple Hypothesis Tracking (MHT): is a measurement oriented approach, in which probability that a measurement sequence is originated from an established target or a new target is calculated.

JPDA can only be applied to already established tracks, while MHT can be used for track initiation as well. Even though MHT is the optimal approach, it is not feasible when a large number of measurement steps are considered. Thus, suboptimal version of MHT, called S-D assignment, is the widely used technique for data association [20][60].

## 2.3 Target Classification

It is essential to clarify the terminology used to describe the various levels of specificity with which a sensor might distinguish between different targets.

- **Identification:** is the indication by any act or means of one's own friendly character or individuality. The determination by any act or means of the friendly or hostile nature of a detected person, object or phenomenon.

- **Recognition:** is the determination of the nature of a detected person, object or phenomenon, and possibly its class or type. This may include the determination of an individual within a particular class or type. There are consequently various degree of recognition.

  - **General recognition:** recognize an object by class. E.g., recognize a vehicle as tank, infantry fighting vehicle, truck or recognize an aircraft as either a bomber or a fighter. A lower level of general recognition might be to recognize a vehicle as tracked, wheeled or recognize an aircraft as swept winged or straight winged.

  - **Detailed recognition:** recognize an object by type. E.g., recognize a vehicle as either a T-80 tank or an M-1 Abrams tank, or recognize an aircraft as an Su-27 or a Tornado. It may entail the recognition of an individual person or object (e.g, finger printing).

The more general level can be given as below:

- **Classify:** associate with or assign to one of a number of sets (classes), which are distinguished by one or more criteria, irrespective of whether there is any prior knowledge of the class membership or class boundaries.

- **Recognize:** establish membership of one of the number of disjoint known sets (classes).
- **Identify:** establish the absolute sameness with one of a number of possible individual members of a class of known elements.

### 2.3.1 Bayesian Joint Target Tracking and Classification

Consider the following model of a discrete-time jump Markov system, describing the target dynamic and sensor measurements

$$x_k = F(m_{k-1})x_{k-1} + G(m_{k-1})u_{k-1} + w_{k-1} \tag{2.63}$$

$$z_k = h(m_{k-1}, x_k) + v_{k-1},\ k = 1, 2\ldots \tag{2.64}$$

where $x_k \in \mathbb{R}^{n_x}$ is the base (continuous) state vector with transition matrix $F$, $z_k \in \mathbb{R}^{n_z}$ is the measurement vector with measurement function $h$, and $u_k \in \mathbb{U}$ is a known control input. The noise processes $w_k$ and $v_k$ are independent identically distributed (i.i.d.) Gaussian having characteristics $w_k \sim \mathcal{N}(0, Q)$ and $v_k \sim \mathcal{N}(0, R)$, respectively. $w_k$ is the random input vector and $v_k$ is the random measurement error vector. All vectors and matrices are assumed of appropriate dimensions. The model (discrete) state $m_k \in \mathbb{S} \triangleq \{1, 2, \ldots, s\}$ is a homogeneous-time first-order Markov chain with transition probabilities

$$\pi_{ij} = P\{m_k = j | m_{k-1} = i\}, (i, j \in \mathbb{S}) \tag{2.65}$$

and initial probability distribution $P_1(i) \triangleq P\{m_1 = i\}$ for $i \in \mathbb{S}$, such that $P_1(i) \geq 0$, and $\sum_{i=1}^{s} P_1(i) = 1$. It is assumed that target belongs to one of the $M$ classes $c \in C$

where $C = \{c_1, c_2, \ldots, c_M\}$ represents the set of the target classes. Generally, the number of the discrete state $s = s(c)$, the initial probability distribution $P_1^c(i)$ and the transition probability matrix $\pi = [\pi_{ij}]^c$, $i, j \in \mathbb{S}$ are different for each target class.

The joint state and class is time varying with respect to the state and time invariant with respect to the class [27]. Let

$$\{Z^k, Y^k\} = \{z_i, y_i\} : \ i = 1, 2, \ldots, k \tag{2.66}$$

be the cumulative set of kinematic ($Z^k$) and class (feature) measurements ($Y^k$) up to time $k$.

The goal of the joint tracking and classification task is to estimate the state $x_k$ and the posterior classification probabilities $P(c|\{Z^k, Y^k\}), c \in C$ based on all available measurement information $\{Z^k, Y^k\}$.

If the posterior joint state-class probability density function (pdf) is $p(x_k, c|\{Z^k, Y^k\})$, then the posterior classification probabilities can be obtained by marginalization over $x_k$:

$$p(c|\{Z^k, Y^k\}) = \int_{x_k} p(x_k, c|\{Z^k, Y^k\}) dx_k \tag{2.67}$$

Suppose that the posterior joint state-class pdf is $p(x_{k-1}, c|\{Z^{k-1}, Y^{k-1}\})$ at time instant $k-1$. According to the Bayesian framework, $p(x_k, c|\{Z^k, Y^k\})$ can be computed recursively from $p(x_{k-1}, c|\{Z^{k-1}, Y^{k-1}\})$ in two steps - prediction and measurement update [27][56].

The predicted state-class pdf $p(x_k, c|\{Z^{k-1}, Y^{k-1}\})$ at time $k$ is given by

$$p(x_k, c|\{Z^{k-1}, Y^{k-1}\}) = \int_{x_{k-1}} p(x_k|x_{k-1}, c) p(x_{k-1}, c|\{Z^{k-1}, Y^{k-1}\}) dx_{k-1} \tag{2.68}$$

where the class and state conditioned pdf $p(x_k|x_{k-1}, c, \{Z^{k-1}, Y^{k-1}\})$ is obtained from the state transition equation (2.63)

$$\begin{aligned} p(x_k|x_{k-1}, c, \{Z^{k-1}, Y^{k-1}\}) &= \sum_{j=1}^{s(c)} p(x_k|x_{k-1}, m_k = j, \{Z^{k-1}, Y^{k-1}\}) \\ &\times P(m_k = j|x_{k-1}, c, \{Z^{k-1}, Y^{k-1}\}) \qquad (2.69) \\ &= \sum_{j=1}^{s(c)} p(x_k|x_{k-1}, m_k = j, \{Z^{k-1}, Y^{k-1}\}) \\ &\times \sum_{l=1}^{s(c)} \pi_{lj} P(m_{k-1} = l|c, \{Z^{k-1}, Y^{k-1}\}) \qquad (2.70) \end{aligned}$$

The form of the conditional pdf of the measurements

$$p(\{z_k, y_k\}|x_k, c) = \lambda_{\{x_k,c\}}(z_k, y_k) \qquad (2.71)$$

is usually known. This is the likelihood of the joint state and feature and has a key role in the classification algorithm.

When the measurements $\{z_k, y_k\}$ arrive, the update step can be completed

$$p(x_k, c|\{Z^k, Y^k\}) = \frac{\lambda_{\{x_k,c\}}(\{z_k, y_k\})p(x_k, c|\{Z^{k-1}, Y^{k-1}\})}{p(\{z_k, , y_k\}|\{Z^{k-1}, Y^{k-1}\})} \qquad (2.72)$$

where

$$p(\{z_k, y_k\}|\{Z^{k-1}, Y^{k-1}\}) = \sum_{c\in C} \int_{x_k} p(\{z_k, y_k\}|x_k, c)p(x_k, c|\{Z^{k-1}, Y^{k-1}\})dx_k \qquad (2.73)$$

The recursion (2.68)-(2.72) begins with the prior density $P\{x_1, c\}, x_1 \in \mathbb{R}^{n_x}, c \in C$, which is assumed known.

Using Bayes' theorem, the posterior probability of the discrete state $m_k$ for class

$c$ is expressed by

$$P(m_k = j|c, \{Z^k, Y^k\}) = \frac{1}{l_k} p(\{z_k, y_k\}|m_k = j, c, \{Z^{k-1}, Y^{k-1}\}) \times \sum_{l=1}^{s(c)} \pi_{lj} P(m_{k-1} = l|c, \{Z^{k-1}, Y^{k-1}\}), \quad (2.74)$$

where $l_k$ is a normalizing constant. The equation (2.74) is substituted in (2.70) in order to predict the state pdf at time $k+1$.

Then the target classification probability is calculated by the equation

$$P(c|\{Z^k, Y^k\}) = \frac{p(\{z_k, y_k\}|c, \{Z^{k-1}.Y^{k-1}\}) P(c|\{Z^{k-1}, Y^{k-1}\})}{\sum_{c \in C} p(\{z_k, y_k\}|c, \{Z^{k-1}, Y^{k-1}\}) P(c|\{Z^{k-1}, Y^{k-1}\})} \quad (2.75)$$

with an initial prior target classification probability $P_1(c)$, $\sum_{c \in C} P_1(c) = 1$.

The state estimate $\hat{x}_k^c$ for each class $c$

$$\hat{x}_k^c = \int_{x_k} x_k p(x_k, c|\{Z^k, Y^k\}) dx_k, \ c \in C \quad (2.76)$$

takes part in the calculation of the combined state estimate

$$\hat{x}_k = \sum_{c \in C} \hat{x}_k^c P(c|\{Z^k, Y^k\}). \quad (2.77)$$

It is obvious from (2.68) - (2.77) that the estimates, needed for each class, can be calculated independently from the other classes. Therefore, the joint target tracking and classification task can be accomplished by the simultaneous work of $M$ independent filters [22].

# Chapter 3

# Observable Operator Model (OOM)

A widely used class of models for stochastic systems is the HMM. Systems that can be modeled by HMMs are a proper subclass of linearly dependent processes, which is a class of stochastic systems known from mathematical investigations carried out over the past four decades. A recently introduced method, which is a simple characterization of linearly dependent processes, called OOM [37] is proved to outperform the HMM [37] [38]. The mathematical properties of OOMs have lead to a constructive learning algorithm for the identification of linearly dependent processes. Efficiency Sharpening (ES) algorithm for learning OOM is proved to be an efficient learning tool over the Expectation Maximization (EM) algorithm for learning HMM. The algorithm has a time complexity of $\mathcal{O}(N + nm3)$, where $N$ is the size of training data, $n$ is the number of distinguishable outcomes of observations, and $m$ is the model state-space dimension. In this work, OOM can be utilized to learn the probability distribution of the unknown feature measurement sequence from the training data.

## 3.1 Definition

A $m$-dimensional OOM is a triplet $\mathcal{A} = (\mathbb{R}^m, (\tau_a)_{a\in\Phi}, w_0)$, where $w_0 \in \mathbb{R}^m$ and $\tau_a$: $\mathbb{R}^m \longmapsto \mathbb{R}^m$ are linear operators, satisfying the following four properties.

1. $\mathbf{1}w_0 = 1$, where $\mathbf{1} = [1,\ 1, \ldots,\ 1] \in \mathbb{R}^m$
2. $\mu = \sum_{a\in\Phi} \tau_a$ has column sum of 1, where $\Phi$ is the quantized feature measurement set
3. $\mu w_0 = w_0$
4. for all sequences $a_1, a_2 \ldots a_k$ it holds that $\mathbf{1}(\tau_{a_k} \ldots \tau_{a_1} w_0) \in [0, 1]$.

An OOM describes the future knowledge about a stochastic system by means of linear operators chosen from a finite set of operators on the basis of current observations. Therefore, the stochastic trajectories in OOM can be considered as sequences of operators. Figure 3.1 illustrates a simple observed piece of trajectory $(a_1, a_2, a_3)$ that corresponds to a concatenation of operators $(\tau_{a_3}(\tau_{a_2}(\tau_{a_1})))$. Here, the initial state

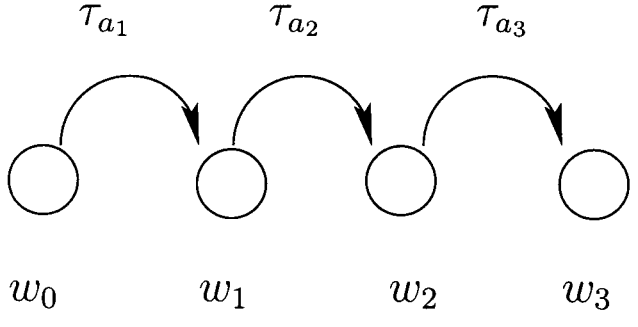



Figure 3.1: Stochastic trajectory as a series of operators.

vector is $w_0$ and operated with operators corresponding to the observed sequences one by one. Then

$$w_3 = \tau_{a_3}\tau_{a_2}\tau_{a_1} w_0 \tag{3.1}$$

This characterizes the linear property of the OOM. The one-to-one relationship between the selection of operator and the observation symbol has led to the naming of operators as observable operators. Here, $\mathbf{1}w_3 \in [0,1]$ describes a probability distribution as given in property 4.

## 3.2 HMMs and OOMs

HMMs have a structure of hidden states and emission distributions whereas the theory of OOMs concentrates on the observations themselves. In OOM theory, the model trajectory is seen as a sequence of linear operators and not as sequences of states. This idea leads to the linear algebra structure of OOMs, which gives efficient methods in estimation and learning. Using OOMs instead of HMMs has advantages and disadvantages in the way the OOM and HMM are trained and learned. HMMs are widely known and have many applications including speech recognition. However, the class of OOMs is richer and combines the theory of linear algebra and stochastic processes. Thus, the training of OOMs can be done more efficiently than that of HMMs. Further, the training of OOMs is asymptotically correct whereas the corresponding EM algorithms for HMMs are not. The purpose here then, is to model target classes using feature information from a target by OOMs. Let $(Y_t)_t \in N$ be a discrete-time, discrete-valued stochastic observation process, where the random variables $Y_t$ have outcomes in an alphabet $\Sigma = \{a^1, a^2, \ldots, a^\alpha\}$. Assume a state process $(X_t)_t \in N$ having the Markov chain with finite number of hidden states $\{s_1, s_2, \ldots, s_m\}$. The state transition probability from state $s_j$ to $s_i$ is denoted by the $(i,j)^{\text{th}}$ element of the transition matrix $M$ of size $m \times m$. To characterize the HMM completely, one needs to specify three components: initial distribution $w_0$, transition matrix $M$ and

the observation matrix $O_{a_i}$, $\forall_{a_i} \in \Sigma$. The observation matrix can be constructed by placing the emission probability $P(Y_t = a_i | X_t = s_j)$ at the $j^{\text{th}}$ diagonal of observation matrix $O_{a_i}$.

The process described by the HMM is stationary if $w_0$ is an invariant distribution of the Markov chain, as shown by

$$M' w_0 = w_0 \tag{3.2}$$

Let the operator indexed over the symbol be taken from the alphabet $\Sigma$ be $\tau_{a_i} = M' O_{a_i}$. Then the probability of observing the sequance $a_0 \ldots, a_\tau$ can be obtained by

$$P(a_0, \ldots, a_\tau) = \mathbf{1} \tau_{a_\tau}, \ldots, \tau_{a_0} w_0 \tag{3.3}$$

$M$ can be recovered from the operator $\tau_a$ by observing that

$$\begin{aligned} \sum_{a \in \Sigma} \tau_a &= M' \sum_{a \in \Sigma} O_a && (3.4) \\ &= M' I && (3.5) \\ &= M' && (3.6) \end{aligned}$$

Consider an HMM with two hidden states $s_1$ and $s_2$ and the two outcomes to be $\Sigma = \{a, b\}$. The parameters needed to describe the HMM completely are shown in Figure 3.2. Here the state transition probabilities are indicated on the arrows between $s_1$ and $s_2$. As illustrated in this section, the HMM with modified structure,

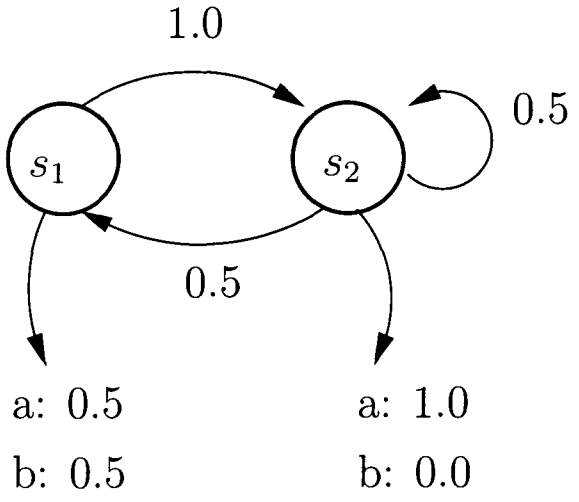



Figure 3.2: Hidden Markov Model example

i.e., $(M, O_a, O_b, w_0)$, becomes

$$M = \begin{pmatrix} 0.0 & 1.0 \\ 0.5 & 0.5 \end{pmatrix}, O_a = \begin{pmatrix} 0.5 & \\ & 1.0 \end{pmatrix}, O_b = \begin{pmatrix} 0.5 & \\ & 0.0 \end{pmatrix}$$

Since $O_a + O_b = I$, the HMM transition matrix can be written as

$$\begin{aligned} M' &= M'.I && (3.7) \\ &= M'(O_a + O_b) && (3.8) \\ &= M'O_a + M'O_b && (3.9) \\ &= \tau_a + \tau_b && (3.10) \end{aligned}$$

Therefore, similar to OOM structure, any given HMM can also be written as $(\mathbb{R}^m, (\tau_a)_{a\in\Phi}, w_0)$.

One may raise the question how the OOM components such as $m$, $w_0$ and matrix $\mu'$ of an OOM are associated with the number of states, the state probability vector and transition matrix $M$ of a corresponding hidden Markov model, respectively. However,

the OOM can be interpreted as follows:

1. Relax the requirement that the transpose of Markov matrix $M'$ be a stochastic matrix, to a weaker condition that the column of $M'$ each sum to 1.

2. Relax the requirement such that $w_0$ merely need to satisfy the component sum equal to 1. In other words, $w_0$ is allowed to assume negative values.

The mathematical construction of an OOM can therefore be considered as a generalization of HMM. This is one of the virtues of OOM. Since the learning algorithm is rooted in the efficiency sharpening (ES) principle, OOM learning algorithm is known as the ES algorithm. The ES algorithm will be discussed in detail later in this chapter. The following are some virtues of modeling and learning with OOMs:

- **Speed of Convergence:** For a given stochastic process, OOM/ES learning algorithm yields a model estimate in a fraction of the computational time than of HMM/EM algorithm [37] [38]. Typically, OOM/ES learning algorithm requires not more than five ES iterations to converge to a reasonably good model, whereas HMM/EM algorithm converges to the target model in more than one hundred iterations in general. Also, it should be noted that the computational load of one ES iteration is comparable to one EM iteration.

- **Accuracy:** The model obtained via the ES learning algorithm is markedly more accurate than the corresponding model obtained via HMM/EM algorithm. This has been proved after testing over a number of standard data sets [37]. With large data sets, HMM/EM algorithm often gets trapped in one of the suboptimal maxima of the likelihood function. In order to get a reasonably good model, a good guess of initialization is crucial in HMM/EM algorithm.

- **Expressiveness:** For the same level of model accuracy, OOM assumes less dimension than HMM. Another aspect of the enhanced expressiveness of OOM is that the class of processes have finite dimensional HMMs. There are certain linear dependant processes that can be captured by OOMs whereas HMMs cannot be employed (e.g., the probability clock [37][38] where the outcome probability fluctuate in time).

- **Tractability:** Since OOM is expressed in terms of linear algebra which is one of the well established fields in applied mathematics, OOM can be interpreted transparently.

There are two limitations associated with an OOM. One is the negativity issue associated with some model predicted probabilities. The other one is the instability problem associated with larger model dimension. However, there exists heuristic counter measures for both of these issues.

## 3.3 Learning OOMs

### 3.3.1 Interpretable OOM

Within the equivalent class of OOM, there is a proper subset of minimal dimensional OOMs. The state of a such minimal dimensional OOMs can be interpreted as future distributions of trajectories. This class of OOMs are collectively known as interpretable OOMs. Interpretable OOM is pivotal for OOM learning algorithm. It has three properties:

- Its state vectors are probability vectors.

- The components of its state vector provide probabilities of a certain well-defined future events, also known as characteristic events.
- It can constructively be obtained through a learning algorithm.

### 3.3.2 Characteristic Events

For a discrete time stochastic process $Y_t$ with the alphabet $\Sigma$, $k$-event B is defined as a non-empty subset of $\Sigma^k$. In mathematical notation, it can be written as

$$B \subset \Sigma^k \tag{3.11}$$

$P((Y_{t+1}, Y_{t+2}, \ldots, Y_{t+k}) \in B_i)$ denotes the probability of observing the process trajectory passing through $B_i$ in the time window of $[t+1, t+k]$. For notational ease, $P(B)$ will be used instead of $P((Y_{t+1}, Y_{t+2}, \ldots, Y_{t+k}) \in B_i)$ in this thesis.

Now, let $(Y_n)_{n\geq 0}$ be an $m$ dimensional stationary process with observable from an alphabet $\Sigma$. For a sufficiently large $k$, $\Sigma^k = B_1 \cup \ldots \cup B_m$ is the partition of the set of strings of length $k$ into $m$ disjoint, non-empty sets $B_i$. Then, this partition is called a set of characteristic events $B_i, (i = 1, \ldots, m)$, if some sequences $\bar{a}_1, \ldots, \bar{a}_m$ exist such that the matrix $V = (P(B_i|\bar{a}_j))_{1\leq i,j\leq m}$ is nonsingular. Here $P(B_i|\bar{a}_j)$ means $\Sigma_{\bar{b}\in B_i} P(\bar{b}|\bar{a}_j)$. The invertibility of $V$ is an important phenomenon in OOM learning process.

**Definition:** Let $\mathcal{A} = (\mathbb{R}^m, (\tau_a)_{a\in\Sigma}, w_0)$ be a finite dimensional OOM and let $(B_i), (i = 1, \ldots, m)$ be the characteristic events of $\mathcal{A}$. Then $\mathcal{A}$ is called interpretable with respect to characteristic events $(B_i), (i = 1, \ldots, m)$ if

$$P(B_i|w_n) = (w_n)_i, \forall n \in N, i \in \{1, \ldots, m\} \tag{3.12}$$

where $P(B_i|w_n)$ denotes the probability of observing $(X_{n+1}, \ldots, X_{n+k}) \in B_i$ given that the OOM was in state $w_n$ at time $n$. Further, $(w_n)_i$ denotes the $i$th component of the state vector $w_n$. Since $B_1 \cup \ldots, \cup B_m$ is an exhaustive and disjoint partitioning of $\Sigma^k$, it follows that $\sum_i P(B_i|w_n) = 1$ and hence $\mathbf{1}w_n = 1$. That is, $w_n$ is a probability vector.

Therefore, an interpretable OOM (interpretable with respect to characteristic events $B_1, \ldots, B_m$) holds the following properties:

1. $w_0 = (P(B_1), \ldots, P(B_m))^T$
2. $\tau_{\bar{a}} w_0 = (P((\bar{a}B_1)), \ldots, P((\bar{a}B_m)))^T$

where $P((\bar{a}B)) = \sum_{\bar{b} \in B} P(\bar{a}\bar{b})$.

### 3.3.3 Basic OOM Learning Algorithm

The HMM/EM algorithm finds the local maximum of a likelihood function when it is trained on a finite data set of interest. But with large amounts of data, there can be many maxima of the likelihood function and the EM algorithm may fail to find the global optimal solution. The selection of a reasonably good initial guess for the starting point would give an optimal solution, however it is not guaranteed. Since OOM relaxes the limitation of having many local maxima, it can be employed as an alternative to HMM. Even though the OOM has similar structural properties as HMM, the learning algorithm is completely different from that of HMM. Given the OOM $\mathcal{A} = (\mathbb{R}^m, (\tau_a)_{a \in \Phi}, w_0)$, the OOM learning is simply computing the estimates of each operators $(\tau_a)_{a \in \Phi}$ using the training samples of finite length produced by an unknown stationary process.

The interpretable OOM is the key factor in the OOM learning process. In deriving the learning algorithm, it is assumed that the training sequence $s$ is produced by a stationary process, which can be modeled by some OOM, $\mathcal{A} = (\mathbb{R}^m, (\tau_a)_{a\in\Sigma}, w_0)$ of minimal dimension $m$. If the OOM $\mathcal{A}$ is assumed to be interpretable with respect to characteristic events $B_1, \ldots, B_m$, then the argument value pair for operator $(\tau_a)_{a\in\Sigma}$ can be obtained from

$$\begin{aligned} \tau_a((P(\bar{a}B_1), \ldots, P(\bar{a}B_m))') &= \tau_a(\tau_{\bar{a}} w_0) \\ &= \tau_{\bar{a}a} w_0 \\ &= (P(\bar{a}aB_1), \ldots, P(\bar{a}aB_m))' \end{aligned} \tag{3.13}$$

A linear operator on $\mathbb{R}^m$ is determined by $m$-argument value pairs provided that arguments are linearly independent. It should be noted that similar to characteristic events, indicative events can also be defined by the partitioning of sequence space. The only difference is that the indicative events are perceived as events that describe the past process trajectories whereas characteristic events describe the future. In order to differentiate indicative and characteristic events, indicative events are denoted by $A_i$ and characteristic events by $B_i$ in this thesis. Now, the procedure for the learning algorithm can be summarized in the following steps:

- **Step 1:** Choose characteristic events $B_1, \ldots, B_m$ and indicative sequences $\bar{a}_1, \ldots, \bar{a}_m$ such that the matrix $\hat{V} = (\hat{P}(\bar{a}_j B_i))_{i,j=1,\ldots,m}$ is invertible. This matrix contains $m$ linearly independent argument vectors for operators $\tau_a$ in its columns.

- **Step 2:** For each $a \in \Sigma$, collect the corresponding value vectors in a matrix $\hat{W}_a = (\hat{P}_s(\bar{a}_j a B_i))_{i,j=1,\ldots,m}$.

- **Step 3:** Obtain an estimates for $\tau_a$ by

$$\hat{\tau}_a = \hat{W}_a \hat{V}^{-1} \tag{3.14}$$

To further simplify the counting, $\hat{V}$ and $\hat{W}_a$ can be substituted with $V^{\text{raw}}$ and $W_a^{\text{raw}}$ as follows:

$$V^{\text{raw}} = (\text{count the number of events } A_j B_i \text{ in } s)_{i,j=1,\ldots,m} \tag{3.15}$$

$$\hat{W}_a^{\text{raw}} = (\text{count the number of events } A_j a B_i \text{ in } s)_{i,j=1,\ldots,m} \tag{3.16}$$

It can be proved that $W_a^{\text{raw}}(V^{\text{raw}})^{-1} \approx \hat{W}_a(\hat{V}^{-1})$.

#### 3.3.3.1 Toy Example

Consider a generator of binary symbols 'a' and 'b' and the training sequence $s$ of length 20 is given as 'abbbaaaabaabbbabbbbb'. Assume that the estimated model dimension is 2. The characteristic events $B_1 = \{a\}$ and $B_2 = \{b\}$ and indicative events $A_1 = \{a\}$ and $A_2 = \{b\}$. The initial state vector $w_0$ can be estimated by

$$\begin{aligned} \hat{w_0} &= \begin{pmatrix} \frac{\#a}{N} & \frac{\#b}{N} \end{pmatrix} \\ &= \begin{pmatrix} \frac{8}{20} & \frac{12}{20} \end{pmatrix} \end{aligned}$$

Using (3.15) and (3.16), $V^{\text{raw}}$ and $W_a^{\text{raw}}$ can be calculated by a single sweep of the inspection window of certain length over $s$ as follows:

$$V^{\text{raw}} = \begin{pmatrix} \#aa & \#ba \\ \#ab & \#bb \end{pmatrix} \tag{3.17}$$

$$= \begin{pmatrix} 4 & 3 \\ 4 & 8 \end{pmatrix} \tag{3.18}$$

$$W_a^{\text{raw}} = \begin{pmatrix} \#aaa & \#baa \\ \#aab & \#bab \end{pmatrix} \tag{3.19}$$

$$= \begin{pmatrix} 2 & 2 \\ 2 & 1 \end{pmatrix} \tag{3.20}$$

Hence

$$\hat{\tau}_a = W_a^{\text{raw}}(V^{\text{raw}})^{-1} \tag{3.21}$$

$$= \begin{pmatrix} 0.4 & 0.1 \\ 0.6 & -0.1 \end{pmatrix} \tag{3.22}$$

Similarly, $\hat{\tau}_b$ can be found as

$$\hat{\tau}_b = \begin{pmatrix} 0.00 & 0.25 \\ 0.20 & 0.55 \end{pmatrix} \tag{3.23}$$

It is interesting to note that $\hat{\tau}_a$ is no longer a stochastic matrix as it takes a negative entry.

### 3.3.4 Properties of the Basic Learning Algorithm

This section summarizes the properties of the basic OOM learning algorithm.

1. Conceptually transparent and computationally cheap.

2. Complexity: Run time complexity is $O(N + m^3)$ and the space complexity is $O(N \log(m) + |\Sigma| m^2)$, where $N$ is the length of the sample, $m$ is the model dimension and $|\Sigma|$ is the length of the alphabet.

3. Event selection: The statistical efficiency of the basic learning algorithm heavily depends on the selection of indicative and characteristic events. In other words, the characteristic and indicative events have to be chosen such that counting matrix $V$ is a matrix with condition number closer to 1.

4. Counting statistics: Small portion of the counting statistics, typically twice the length of the characteristic events are entered into the estimation algorithm. Much information contained in the training data is thus ignored.

As outlined earlier in this section, the major challenge in the basic version of the algorithm is to find good indicative and characteristic events so that the estimated OOM could capture the $m$ significant dimension of the underlying dynamics. In other words, the task is not to find the true dimension but to find the dimension that capture the data such that the data is neither over-fitted nor under-exploited. In order to achieve this objective, the counting matrix $V^{\text{raw}}$ has to be selected with the numerical rank closer to 1. That is, the smallest singular value of $V^{\text{raw}}$ should be significantly larger than 0 to become the full ranked matrix. The question now is how to find $V^{\text{raw}}$ with numerical rank closer to 1 with less computational overhead.

Applying a small perturbation in $V^{\text{raw}}$, it can be shown that

$$\text{num.rank}(V^{\text{raw}}) = m \iff \sigma_{min}(V^{\text{raw}}) > \epsilon. \parallel V^{\text{raw}} \parallel_{\infty} \tag{3.24}$$

where,

$\sigma_{min}$ is the smallest singular value of $V^{\text{raw}}$, $\epsilon$ is the average relative error of matrix entries ($\parallel V^{\text{raw}} - V^{\text{raw}}_{perturbated} \parallel$) and $\parallel . \parallel_{\infty}$ refers to the infinity norm.

### 3.3.5 Learning Algorithm Based on Efficiency Sharpening Principle

The Efficiency Sharpening (ES) method is iterative. In each iteration, the model estimated in the previous step is used to construct an estimator with a better statistical efficiency than the previous one, hence the term efficiency sharpening. In this framework, though the characteristic and indicative events are used in the initial model, the subsequent models obtained iteratively exploit only the states of the previous model as replacement to characteristic and indicative events. The influence on the choice of good characteristic and indicative events is thus eliminated in ES based learning algorithm. The computational load is however, comparable to HMM/EM iteration, typically 2-5 iterations are needed to get a reasonably good estimated model. Moreover, the accuracy of OOM models (training and testing likelihoods) is superior to HMM, except on data sets that have been generated by HMMs in the first place. This chapter presents a number of relevant theorems and propositions required to establish the learning algorithm. This chapter also discusses the two versions of OOM/ES learning algorithm – Poor Man's version and the suffix tree based version are also discussed.

#### 3.3.5.1 Poor Man's Version of ES Principle

This training algorithm is rooted in the estimates $\hat{V}$ and $\hat{W}_a$. This model variance across different training sequence (i.e., the statistical efficiency of the estimator) hinges among factors such as the condition of $\hat{V}$, and the variance of estimates $\hat{V}$ and $\hat{W}_a$ etc. The initial model is obtained using the basic learning algorithm. The model variance is minimized when the characterizer $C = UU^r$. It is proved [37] that $\hat{V}^{(n)}$ and $\hat{W}_a^{(n)}$ can be written in a recursive manner as follows:

$$\hat{V}^{(n)} = \hat{C}^{(n)}\hat{V}^{(n-1)} \tag{3.25}$$

$$\hat{W}_a^{(n)} = \hat{C}^{(n)}\hat{W}_a^{(n-1)} \tag{3.26}$$

Hence the estimates for the operator can be obtained using (3.15) and (3.16). The iterative procedure to find a new model estimate is terminated when the training log-likelihood of models appear to settle on a constant value. Although the poor man's strategy is simple and computationally inexpensive, the only limitation is that the indicative sequences $(\bar{a}_i)_{l<k} = O^k$ do not adapt to the training sequence. For example, some $\bar{a}_i$ may not occur in the training sequence and some may occur only a few times. This phenomena leads to poor estimates of the probability through relative frequencies.

#### 3.3.5.2 Suffix Tree-Based ES Principle

This second version is rooted in a data structure called Suffix Tree (ST). Using ST representation of the training sequence, one can exploit characteristic or indicative events of all possible lengths simultaneously. This subsequence counting statistics are

stored in the nodes of the ST. Instead of obtaining the operator $\hat{\tau}_a$ from $m$-argument value pairs continued in the counting matrices $V^{\text{raw}}$ and $W_a^{\text{raw}}$, all the counting values are exploited to obtain argument vectors. It should be noted that the number of used argument value pairs is in the order of the training data size. In short, ST is exploited to represent the following:

1. training sample
2. partitioning of the characteristic and indicative events
3. for counting statistics

The basics of the suffix tree is illustrated below.

**Suffix Tree:**

Suffix tree (ST) is a data structure that exposes the internal structure of a symbol sequence in a deep way. It turns out that all the possible substrings found inside a string. ST belongs to the member of a Trie family. The word trie comes from the word retrieval. A trie is a $k$-ary position tree. It is constructed from input strings. That is, the input is a set of $n$ strings called $s_1, s_2, \ldots, s_n$, where each $s_i$ consists of symbols belonging to a finite alphabet and has a unique terminal symbol (also known as sentinel symbol) which is denoted by \$.

**Common Applications of ST**

1. To perform basic operations in a large text (e.g., searching, insertion and deletion of the text found in a dictionary)
2. To store data in a more compressed form
3. To search the pattern in a picture file

Detailed description of OOM learning using suffix tree is illustrated in [37].

## 3.4 Summary

This chapter presents overview of Observable Operator Model and three different versions of the OOM learning algorithms namely, the basic version, ES/Poor man's version and ES/suffix tree version. The basic version of the learning algorithm is rooted in the counting statistics of a training sequence. Although it is simple and transparent, the statistical efficiency depends on the selection of the indicative and characteristic events. In order to overcome this limitation, ES principle is introduced. Finally, two different versions of ES principle-based learning algorithms are presented.

# Chapter 4

# Observable Operator Model-Based Joint Multitarget Tracking with Multiaspect Classification

## 4.1 Joint Tracking and Multiaspect Classification Problem

In many target classification problems, multiple views of a target can reasonably improve the robustness and reliability in decision making. The use of information from several target-to-sensor aspects is motivated by the difficulty in distinguishing between different target classes from single view of a target. The returns from two different targets at certain orientations can be so similar that they may easily confuse the classifier. As a result, all targets may be classified as same type. For example, consider the target orientation shown in Figure 4.1. Here, the same target at different orientations might give different sensor returns, especially different feature

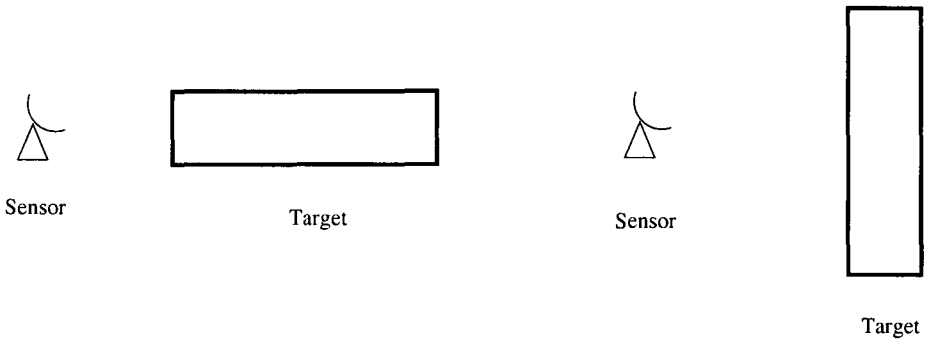



Figure 4.1: Same target generating different sensor feature returns.

measurements. In Figure 4.2, different targets tend to produce similar sensor feature measurements that may give false classification results. Thus, more reliable target classification algorithm needs to be obtained based on the received signals at multiple aspect angles. This would allows to accumulate more information about the size, shape and orientation of the target. Further, when the feature space undergoes changes due to target orientation, multi-aspect classification is necessary in order to maintain accurate target classification results.

The general formulation of the multi-aspect classification can determine the target type when multiple sensors observe an evolving scene. Based on the kinematic measurements and feature measurements such as amplitude and shape, the estimates of kinematic states of the targets and the target types can be obtained. For example, assume that there are $S$ number of sensors reporting observations of a target. At the $k$th observation time, the sensor $s$ produces the vector $\mathbf{y}_k^s$, which contains kinematic measurements $z_k^s$ as well as the feature information $\psi_k^s$. Let $\mathbf{x}_k$ be the state of a target, which consists of position coordinates, velocity and possibly acceleration of the target, and $\Theta_k$ be the orientation of the target. Then the collection of kinematic measurements is given by

$$\mathbf{z}_k = \{z_k^1, z_k^2, \ldots, z_k^S\} \tag{4.1}$$

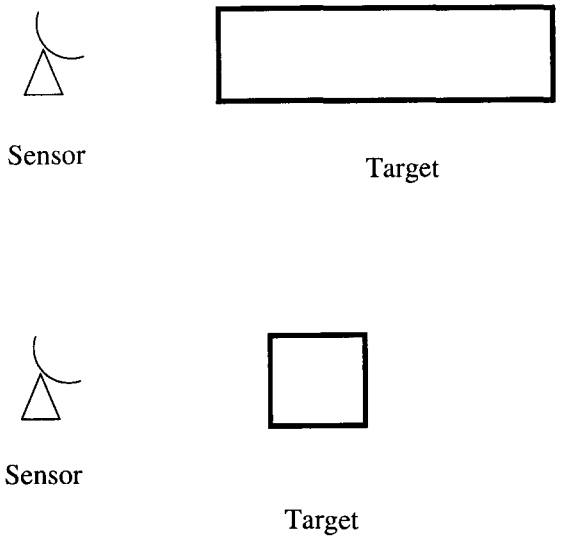



Figure 4.2: Different targets generating similar sensor feature returns.

and the collection of feature measurement from all sensors is given by

$$\chi_k = \{\psi_k^1, \psi_k^2, \ldots, \psi_k^S\} \tag{4.2}$$

The joint posterior density of target state, orientation and target class conditioned on the kinematic and feature measurements can be written in terms of prior density of the parameters and the likelihood $\Lambda$ for the sensor observations given the parameters using Bayes' rule as

$$p(\mathbf{x}_k, \Theta_k, \xi | \mathbf{z}_k, \chi_k) \propto p(\mathbf{x}_k, \Theta_k, \xi) \Lambda(\mathbf{z}_k, \chi_k | \mathbf{x}_k, \Theta_k, \xi) \tag{4.3}$$

where the target type or class is denoted by $\xi \in \mathcal{C}$. Here, $\mathcal{C}$ is an exhaustive set of template target classes. In this formulation, it is assumed that kinematic data or tracking data are insensitive to the target type and its orientation, and that the

feature measurements are invariant to target position. However the feature measurement depends on the target orientation. Even though the target position is invariant to the feature measurements, the target position is needed in deriving the OOM based classifier as it is required to find the relative sensor position with respect to target position. Also, in multiple target scenarios, the feature measurement association to each target is necessary to estimate the target types of each target. Further, the target classification information is advantageous in data association of kinematic measurements across sensors. These fundamental links between target tracking and multi-aspect classification motivated this study to propose and solve the joint target tracking and multi-aspect classification problem in a single framework by incorporating OOM based classifier. Now, assume that $\mathbf{z}_k$ and $\chi_k$ are conditionally independent given the state $\mathbf{x}_k$, the orientation $\Theta_k$ and the target type $\xi$. Thus, (4.3) can be decomposed into

$$p(\mathbf{x}_k, \Theta_k, \xi | \mathbf{z}_k, \chi_k) \propto p(\mathbf{x}_k, \Theta_k, \xi) \Lambda(\mathbf{z}_k | \mathbf{x}_k, \Theta_k, \xi) \Lambda(\chi_k | \mathbf{x}_k, \Theta_k, \xi) \tag{4.4}$$

Here $\Lambda(\mathbf{z}_k | \mathbf{x}_k, \Theta_k, \xi)$ and $\Lambda(\chi_k | \mathbf{x}_k, \Theta_k, \xi)$ are the likelihoods due to kinematic measurements and feature measurements, respectively. With the assumption that the kinematic measurements are insensitive to the target class and its orientation, the kinematic likelihood $\Lambda_{\mathcal{K}}$ can be written as

$$\Lambda_{\mathcal{K}}(\mathbf{z}_k | \mathbf{x}_k) = \Lambda(\mathbf{z}_k | \mathbf{x}_k, \Theta_k, \xi) \tag{4.5}$$

Also assume that the kinematic measurements from different sensors are independent

and thus the kinematic likelihood can be given as

$$\Lambda_{\mathcal{K}}(\mathbf{z}_k|\mathbf{x}_k) = \Lambda(\mathbf{z}_k|\mathbf{x}_k, \Theta_k, \xi) \tag{4.6}$$

$$= \prod_{l=1}^{N_s} \exp(L(z_k^l|\mathbf{x}_k)) \tag{4.7}$$

where $L(z_k^l|\mathbf{x}_k)$ is the log-likelihood of $l$th sensor kinematic measurement. This can be calculated since there exist deterministic target motion models and measurement models. However, the likelihood due to feature measurement in (4.4) cannot be explicitly calculated as in the case of kinematic likelihood. By modeling the deterministic relationship between the feature measurement and the target-to-sensor orientation, it is possible to obtain both target class or type information as well as an estimate of the target orientation. However, the lack of close-form relationship between target orientations and feature measurements prevent the calculation of the likelihood due to the feature measurements. The training based models such as HMM and OOM can be employed to model the unknown distribution with some training data. Due to the superior performances of OOM compared to HMM [37], the main focus is on modeling the feature likelihood or namely classifier using training based OOM, which overcomes the limitation to compute the likelihood $\Lambda_{\mathcal{C}} = \Lambda(\chi_k|\mathbf{x}_k, \Theta_k, \xi)$ of feature measurements generated from a target.

## 4.2 Hidden Markov Model Classifier

As discussed in Chapter 1, the underlying feature information from targets can vary with target-to-sensor orientation. Further, the degree of variability depends on the detailed target geometry. However, one can define angular sectors that represent

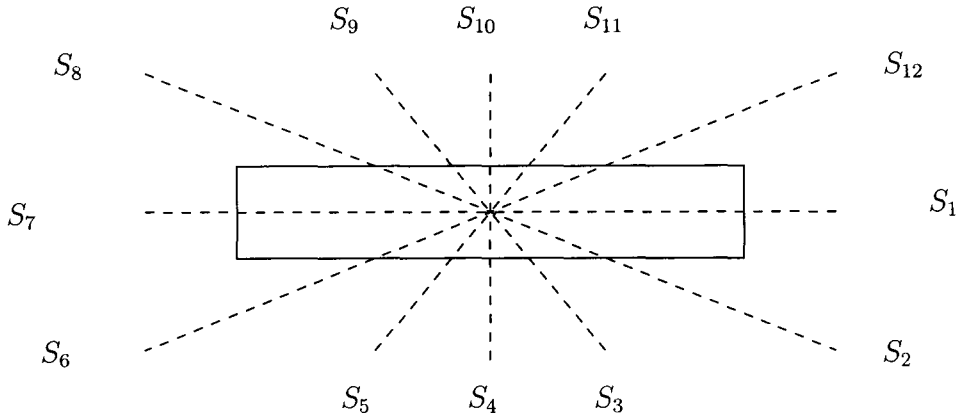



Figure 4.3: Discrete aspect states of a target.

different target-to-sensor orientations, over which the associated scattered signal is approximately stationary as a function of orientation. This assumption is valid as the scattered signals within a particular angular sector vary slowly. A HMM system is typically represented by the following quantities [62]:

1. A set of states (angular sectors that represent target-to-sensor orientations) $S_i$, $i \in [1, N]$ that are unobservable and thus are hidden.

2. A set of $M$ observations, where $M$ represents the number of all possible code book feature data vector. Since the feature measurement is in continuous space, there is a need for quantization so that any continuous-valued feature measurement $f_t$ can be converted into the set $\{\psi_1, \psi_2, \ldots, \psi_M\}$, where $\psi_m$ is the $m$th code book vector.

3. A set of state transition probabilities represented by a matrix $A = [a_{ij}]$ where $a_{ij} = P(q_t = S_j | q_{t-1} = S_i, \xi)$. Here $q_t$ is the system state at time $t$, $S_i$ is state $i$ and $\xi$ is the model defined by the object class and the corresponding training data.

4. A set of observation probabilities represented by matrix $B = [b_i(\psi_k)]$ where $b_i(\psi_k) = P(\psi_k|q_t = S_i, \xi)$ is the emission probability of the $k$th quantized observation, $\psi_k$, at time $t$ from state $S_i$. If the emission processes are assumed to be stationary for an HMM, the probability $b_i(\psi_k)$ is simply reduced to $P(\psi_k|S_i, \xi)$.

5. An initial state distribution $\pi = [\pi_1, \pi_2, \ldots, \pi_N]'$. Here the element $\pi_j$ is the probability of being in state $S_j$ at time $t = 1$, i.e., $P(q_1 = S_j|\xi) = \pi_j$.

Consider a sequence of $T$ states $Q = \{q_1, q_2, \ldots, q_T\}$ and a sequence of $T$ observations $V = \{\psi_1, \psi_2, \ldots, \psi_T\}_k$ at time $k$. The conditional probability of the observation sequence given the state sequence and model is given by

$$P(V|Q, \xi) = \Pi_{t=1}^{T} P(v_t|q_t = S_i, \xi) = \Pi_{t=1}^{T} b_i(v_t) \tag{4.8}$$

where $v_t$ is the feature measurement received at time $k$. It is assumed that the observations are statistically independent. The probability of a given state sequence $Q$ is given by the product of the state-transition probabilities, along with the initial state probabilities

$$P(Q|\xi) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \ldots a_{q_{T-1} q_T} \tag{4.9}$$

Here $\xi$ is the HMM model. Since the state sequence can be modeled as a Markov process, the joint probability of a vector sequence $V$, and the state sequence $Q$ is

$$P(V, Q|\xi) = P(V|Q, \xi) P(Q|\xi) \tag{4.10}$$

The probability of the observed sequence is computed by considering all possible state paths as follows:

$$P(V|\xi) = \sum_{\text{all}Q} \pi_{q_1} b_{q_1}(V_1) a_{q_1 q_2} b_{q_2}(V_2) a_{q_2 q_3} \dots a_{q_{T-1} q_T} b_{q_T}(V_T) \tag{4.11}$$

This is the likelihood of the feature measurement of being in class $\xi$. Section 3 describes how the feature measurement likelihood can be calculated using OOM that produces more accurate results than HMM.

## 4.3 OOM Based Classifier

It is assumed that the response of an unknown target is measured at $L$ aspects $\mathcal{A} = \theta_1, \dots, \theta_L$ with corresponding scattered signals. The signal is then converted into a set of feature vectors, $\mathbf{f} = f_1 \dots, f_L$. The feature extraction methods have been well studied in [52][65]. The statistics of the sequences of $L$ feature vector is modeled via an OOM. Vector quantization [21] is performed by matching the feature vector onto a set of discrete points, representative of continuous distribution of feature $\mathbf{f} \rightarrow \psi = \psi_1, \dots, \psi_L$. An OOM is designed to characterize each of the $r$ classes of targets as follows:

$$\xi(k) \in \{\xi_i\}_{i=1}^{r} \tag{4.12}$$

In order to incorporate target class switching in OOM based target classification, a Markovian process with a known class transition probabilities can be used. Target class switching is possible in real applications where target spawning occurs as in

ballistic missile target tracking. The target class transition probabilities are given by

$$p_{ij} = P(\xi(k) = \xi_j | \xi(k-1) = \xi_i) \tag{4.13}$$

The target class transition probability is also assumed to be time invariant. In other words, it is a homogeneous Markov chain. The target class estimation algorithm includes the following three steps:

- Step 1: Target class likelihood estimation
- Step 2: Target class probability update
- Step 3: Current target class estimation.

### 4.3.1 Class Likelihood Estimation

Assume that a set of training sequences of each target class is available. The following tasks are performed in order to estimate the likelihood:

- For each target class in $\{\xi_i\}_{i=1}^{r}$, an OOM, $(\mathbb{R}^m, (\tau_a)_{a\in\Phi}, w_0)$, is built and the model parameters that optimize the likelihood of the training set of the feature observations for the $i^{\text{th}}$ target class are estimated. The OOMs obtained at the end of this step are called class-specific pre-trained OOMs.
- During the estimation operation, the feature measurements from the sensors are recognized by employing these class-specific pre-trained OOMs.

The OOM classifier gives the likelihoods of being in each target class that a set of quantized feature vector belongs to. In [8], the classifier output is a target class and is modeled using a confusion matrix to handle uncertainties in the classification output.

In this work, the OOM classifier's output is a set of class likelihoods that a set of feature measurements belong to, i.e., the likelihood of being in each possible target class. Therefore, the class likelihoods are used directly to update the classification probabilities. The likelihood probability vector of the OOM classifier $\mathbf{\Omega}$ is

$$\mathbf{\Omega} = \{\Omega_1(k), \Omega_2(k), \ldots, \Omega_r(k)\} \tag{4.14}$$

where $\Omega_i(k)$ is the likelihood of the feature measurement received at time step $k$ for target class $i$ and is evaluated as

$$\Omega_i(k) = p(\{\psi_1, \ldots, \psi_S\}_k | \xi = \xi_i) \tag{4.15}$$

$$= \mathbf{1}[\tau_{\{\psi_1,\ldots,\psi_S\}_k} w_0]_i \tag{4.16}$$

Where $\{\psi_1, \ldots, \psi_S\}_k$ are the feature measurements received from $S$ sensors at time $k$.

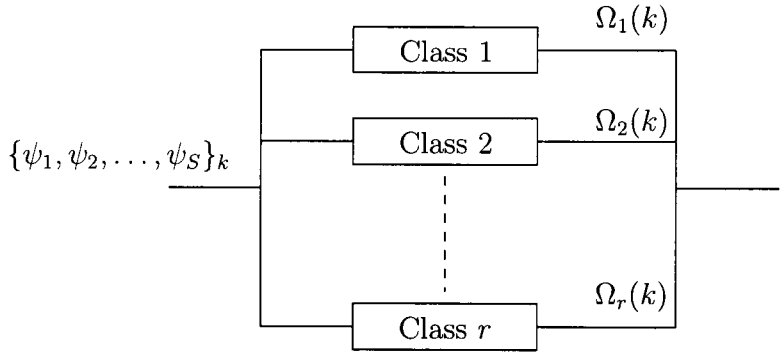



Figure 4.4: Pre-trained OOMs for $r$ classes of targets.

Figure 4.4 shows the schematic block diagram of the OOM classifier. For each class of known targets, an OOM has to be trained using the training data from the

particular class of targets. When a sequence of feature information becomes available from multiple sensors, it is fed into the OOM classifier algorithm where the likelihood of being in each target class will be produced at the output of the OOM classifier.

The maximum likelihood (ML) estimate, $\xi^{\mathrm{ML}}(k)$ is given as

$$\xi^{\mathrm{ML}}(k) = \arg \max_{\xi_1 \leq \xi_i \leq \xi_r} \Omega_i(k) \tag{4.17}$$

In the simulation, pre-trained HMMs are also employed for the likelihood estimation for the purpose of comparing the performance of OOM. The simplest way to determine the target class for a sequence of sensor data is to get the class type that has the maximum likelihood. However, the prior information from established tracks and class likelihoods from the OOM may be used to update the target class. When multiple targets are present in the surveillance region, extraction of the set of feature measurements corresponding each target is difficult. This problem is overcome by jointly treating the classification and multiframe data association, which is explained in Section 4.4.

### 4.3.2 Class Probability Update

The ML estimate is unreliable as it does not use any prior knowledge. Therefore Bayesian sequential estimation of target class can be performed by incorporating the target class switching process as the a priori knowledge of the target class. The prediction of target class is performed using the Markovian process with the known class transition probabilities. If the prior class probabilities $\pi$ of a target being in

possible $r$ classes is given by

$$\pi_{k-1|k-1} = \{\pi^1_{k-1|k-1}, \pi^2_{k-1|k-1}, \ldots, \pi^r_{k-1|k-1}\} \tag{4.18}$$

then the prediction (4.19) and update (4.21) equations of $i$th class of the classifier in target class estimation are given by

$$\pi^i_{k|k-1} = \sum_{j=1}^{r} \pi^i_{k-1|k-1} p_{ij} \tag{4.19}$$

and

$$\pi^i_{k|k} = \frac{\pi^i_{k|k-1}\Omega_i(k)}{\sum_{j=1}^{r} \pi^j_{k|k-1}\Omega_j(k)} \tag{4.20}$$

$$= \frac{\pi^i_{k|k-1}\mathbf{1}[\tau_{\{\psi_1,\ldots,\psi_S\}_k} w_0]_i}{\sum_{j=1}^{r} \pi^j_{k|k-1}\mathbf{1}[\tau_{\{\psi_1,\ldots,\psi_S\}_k} w_0]_j} \tag{4.21}$$

respectively. The target class likelihood $\Omega_i(k)$ is obtained from the first step of the OOM based classifier and the class transition probability $p_{ij}$ is assumed to be known. A reasonably good choice of the target class transition probabilities is helpful in determining the target classes and target class switches accurately. The prior class probability at initial time is not known and therefore uniform prior is used at time $k = 0$.

### 4.3.3 Current Class Estimation

Finally, the target current class at time $k$ is estimated in the maximum a posteriori (MAP) sense as follows

$$\xi^{\text{MAP}}(k) = \arg \max_{\xi_1 \leq \xi_j \leq \xi_r} \pi^i_{k|k} \tag{4.22}$$

The target class estimation using MAP method gives better results compared to ML method as it uses the prior information.

## 4.4 OOM Classifier Integrated Multiframe Data Association

When the targets are close to one another, separately extracting the measurements corresponding each target is difficult. That is, there is an uncertainty in the origin of the feature measurements, which is similar to the measurement uncertainty in kinematic measurements [7]. Therefore, the data association and the OOM based target classification are integrated together. In the measurement-to-measurement data association problem, at each scan, $S$ lists of measurements obtained from $S$ sensors are given. Sensor fusion techniques using $S$ dimensional (or $S$-D) assignment approaches have been extensively studied in [7][43]. After performing the $S$-D assignment during the $k^{\text{th}}$ scan, one has a set of $S$-tuples of measurements, now measurement-to-track association can be performed as a 2-D assignment problem in order to get position estimates. A slightly different technique called $(S+1)$-D association algorithm, which assigns the measurements from the sensors directly to the tracks from the previous scan, is discussed in [66]. In $(S+1)$-D data association, the first list is the one-step predicted states of the available tracks from previous scan. The other $S$ lists are the measurement lists from the $S$ sensors. This is illustrated in Figure 4.5. The cost of assigning an $S$ tuple measurements $(i_1, i_2, \ldots, i_S)$ to a particular track $\mathcal{T}$, i.e., the cost of an (S+1)-tuple $(\mathcal{T}, i_1, i_2, \ldots, i_S)$ is defined as

$$C_{\mathcal{T}i_1i_2\ldots i_S} = -\log \frac{p(Z_{i_1i_2\ldots i_S}, \psi_{i_1i_2\ldots i_S} | \hat{X}_{\mathcal{T}}^{k|k-1}, \hat{\xi}_{\mathcal{T}}^{k|k-1})}{p(Z_{i_1i_2\ldots i_S}, \psi_{i_1i_2\ldots i_S} | \mathcal{T} = \emptyset)} \tag{4.23}$$

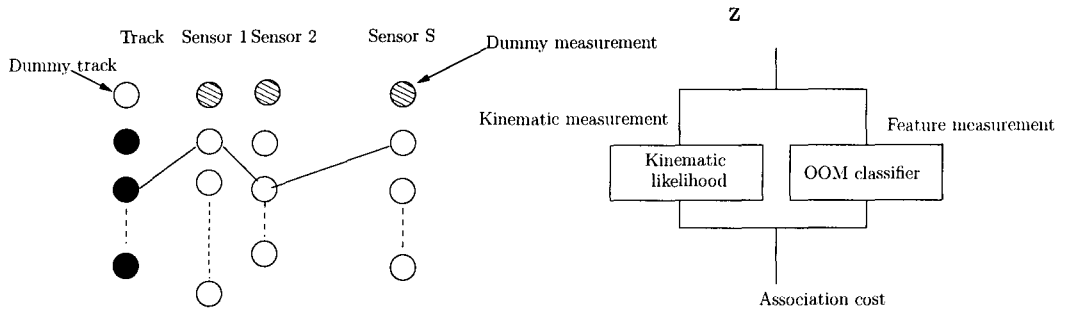



Figure 4.5: OOM integrated $(S+1)D$ data association.

where $p(Z_{i_1 i_2 \ldots i_S}, \psi_{i_1 i_2 \ldots i_S} | \hat{X}_{\mathcal{T}}^{k|k-1}, \hat{\xi}_{\mathcal{T}}^{k|k-1})$ denotes the likelihood that the $S$-tuple has originated from the target with predicted state $\hat{X}_{\mathcal{T}}^{k|k-1}$ and predicted target class $\hat{\xi}_{\mathcal{T}}^{k|k-1}$. The $p(Z_{i_1 i_2 \ldots i_S}, \psi_{i_1 i_2 \ldots i_S} | \mathcal{T} = \emptyset)$ is the likelihood that all measurements in the $S$-tuple are from an extraneous source. The cost defined in (4.23) is to assign the $S$-tuple of measurements to already established tracks, whereas in typical $SD$ assignment techniques the cost of an $S$-tuple is for the measurement-to-measurement association. In $SD$ assignment techniques there is a need for calculating the ML estimate of the unknown target state with the following assumptions:

- Feature measurements and kinematic measurements are independent
- Kinematic measurement is independent of given target class information
- Feature measurement is dependent on target state and class

the cost function in (4.23) can be re-written as

$$C_{\mathcal{T} i_1 i_2 \ldots i_S} = -\log \frac{p(Z_{i_1 i_2 \ldots i_S} | \hat{X}_{\mathcal{T}}^{k|k-1}) p(\psi_{i_1 i_2 \ldots i_S} | \hat{X}_{\mathcal{T}}^{k|k-1}, \hat{\xi}_{\mathcal{T}}^{k|k-1})}{p(Z_{i_1 i_2 \ldots i_S}, \psi_{i_1 i_2 \ldots i_S} | \mathcal{T} = \emptyset)} \tag{4.24}$$

Let us denote the numerator part in (4.24) as $\Lambda(Z_{i_1 i_2 \ldots i_S} | \hat{X}_{\mathcal{T}}^{k|k-1}, \hat{\xi}_{\mathcal{T}}^{k|k-1})$ given by

$$\Lambda(Z_{i_1 i_2 \ldots i_S} | \hat{X}_{\mathcal{T}}^{k|k-1}, \hat{\xi}_{\mathcal{T}}^{k|k-1}) = \Lambda_{\mathcal{K}}(z_{i_1 i_2 \ldots i_S} | \hat{X}_{\mathcal{T}}^{k|k-1}) \Lambda_{\mathcal{C}}(\psi_{i_1 i_2 \ldots i_S} | \hat{X}_{\mathcal{T}}^{k|k-1}, \hat{\xi}_{\mathcal{T}}^{k|k-1}) \tag{4.25}$$

The calculation of the kinematic part (i.e., $\Lambda_{\mathcal{K}}(.)$) is described in [7],[8] and the cost calculation for the feature part (i.e., $\Lambda_{\mathcal{C}}(.)$) can be obtained using the OOM classifier output. When calculating the cost for each branch of the assignment tree, the OOM classifier likelihood has to be obtained for the set of feature measurements corresponding to the branches in the assignment tree. For a set of feature measurements $\psi_{1...S}$, the class likelihoods of being each class is taken at the output of the OOM classifier. Therefore, the likelihood due to feature measurement $\Lambda_{\mathcal{C}}(\psi_{i_1 i_2 ... i_S} | \hat{X}_{\mathcal{T}}^{k|k-1}, \hat{\xi}_{\mathcal{T}}^{k|k-1})$ can be calculated as below

$$\Lambda_{\mathcal{C}}(\psi_{i_1 i_2 ... i_S} | \hat{X}_{\mathcal{T}}^{k|k-1}, \hat{\xi}_{\mathcal{T}}^{k|k-1}) = \arg \max_{\xi_1 \leq \xi_i \leq \xi_r} \pi_{k|k-1}^{i} \mathbf{1}[\tau_{\{\psi_1,...,\psi_S\}_k} w_0]_{\xi_i} \quad (4.26)$$

where $\pi_{k|k-1}^{i}$ is the predicted class of target $\mathcal{T}$ using Makovian process at time $k$ and $\mathbf{1}[\tau_{\{\psi_1,...,\psi_S\}_k} w_0]_{\xi_i}$ is the likelihood calculated using OOM classifier for the target class $\xi_i$.

Now, define a binary indicator function $u(i_S)$ as

$$u(i_S) = \begin{cases} 0 & \text{if } i_S = 0 \\ 1 & \text{otherwise} \end{cases} \quad (4.27)$$

Since the false alarms are assumed to be uniformly distributed, the likelihood that all the measurements in the $S$-tuple are from a false source is given by

$$p(Z_{i_1 i_2 ... i_S}, \psi_{i_1 i_2 ... i_S} | \mathcal{T} = \emptyset) = \prod_{s=1}^{S} \left[ \frac{1}{V_s^f} \right]^{u(i_s)} \quad (4.28)$$

where $V_s^f$ is the spatial false alarm of the field of view of sensor $s$.

Then, the objective of the $(S+1)D$ assignment is to find the best association and the problem is formulated as follows:

$$\min_{\rho_{i_1 i_2 \ldots i_S}} \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \cdots \sum_{i_S=0}^{n_S} C_{\mathcal{T} i_1 i_2 \ldots i_S} \rho_{\mathcal{T} i_1 i_2 \ldots i_S} \tag{4.29}$$

subject to

$$\begin{aligned}
\sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \cdots \sum_{i_S=0}^{n_S} \rho_{\mathcal{T} i_1 i_2 \ldots i_S} &= 1, \qquad i_1 = 1, 2, \ldots, n_1 \qquad (4.30)\\
\sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \cdots \sum_{i_S=0}^{n_S} \rho_{\mathcal{T} i_1 i_2 \ldots i_S} &= 1, \qquad i_2 = 1, 2, \ldots, n_2 \\
&\vdots \\
\sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \cdots \sum_{i_S=0}^{n_{S-1}} \rho_{\mathcal{T} i_1 i_2 \ldots i_S} &= 1, \qquad i_S = 1, 2, \ldots, n_S \qquad (4.31)
\end{aligned}$$

where $\rho_{\mathcal{T} i_1 i_2 \ldots i_S}$ is a binary variable such that

$$\rho_{\mathcal{T} i_1 i_2 \ldots i_S} = \begin{cases} 1 & \text{if } (S+1)\text{-tuple } \rho_{\mathcal{T} i_1 i_2 \ldots i_S} \text{ is included in the solution set} \\ 0 & \text{otherwise} \end{cases} \tag{4.32}$$

The $(S+1)D$ assignment problem is NP-hard, thus it is not possible to find the optimal solution in polynomial time. Therefore, it is necessary to find suboptimal solutions to solve the problem in real-time. The algorithm proposed in [20] is used to solve the assignment problem.

## 4.5 Results

In this section, the results of the simulation studies for the proposed joint tracking and classification algorithm that was developed in this work are presented.

### 4.5.1 Target Motion and Measurement Models

In the simulations, the tracking of targets in two dimensional space is considered and therefore the $F(\Delta_k)$ for constant velocity and coordinated-turn models are given as

$$F(\Delta_k) = \begin{bmatrix} 1 & \Delta_k & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.33}$$

and

$$F(\Delta_k) = \begin{bmatrix} 1 & \frac{\sin(\omega\Delta_k)}{\omega} & 0 & \frac{-(1-\cos(\omega\Delta_k))}{\omega} & 0 \\ 0 & \cos(\omega\Delta_k) & 0 & -\sin(\omega\Delta_k) & 0 \\ 0 & \frac{(1-\cos(\omega\Delta_k))}{\omega} & 1 & \frac{(\sin(\omega\Delta_k))}{\omega} & 0 \\ 0 & \sin(\omega\Delta_k) & 0 & \cos(\omega\Delta_k) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.34}$$

respectively. In the above, $\Gamma(\Delta_k)$ is defined as

$$\Gamma(\Delta_k) = \begin{bmatrix} \frac{\Delta_k^3}{3} & \frac{\Delta_k^2}{2} & 0 & 0 \\ \frac{\Delta_k^2}{2} & \Delta_k & 0 & 0 \\ 0 & 0 & \frac{\Delta_k^3}{3} & \frac{\Delta_k^2}{2} \\ 0 & 0 & \frac{\Delta_k^2}{2} & \Delta_k \end{bmatrix} l \tag{4.35}$$

where $l = 1\mathbf{x}10^{-4}\mathrm{m}^2\mathrm{s}^{-3}$ and the sampling time $\Delta_k$ is taken as $5s$.

The measurements from three sensors are available at discrete time sampling intervals of $T = 5$ seconds. The target-generated measurements corresponding to a target are given by

$$z(t_k) = \begin{bmatrix} \sqrt{(x(t_k) - x_s)^2 + (y(t_k) - y_s)^2} \\ \tan^{-1}((y(t_k) - y_s)/(x(t_k) - x_s)) \end{bmatrix} + v_k \tag{4.36}$$

where $v_k$ is the zero-mean Gaussian noise vector of dimension two with covariance matrix diag($1 \times 10^4$ m$^2$, $3 \times 10^{-4}$ rad$^2$). The pairs $(x(t_k), y(t_k))$ and $(x_s, y_s)$ denote the locations of a target and the sensor at time step $t_k$, respectively.

The Interacting Multiple Model (IMM) estimator is used as the tracker that consists of two models: a constant-velocity and coordinated-turn models [6]. The process noise variance for position is $0.01\mathrm{m}^2/\mathrm{s}^4$ and for turn rate it is $0.0001\mathrm{rad}^2/\mathrm{s}^2$. It is assumed that the probability of detection $P_d = 0.95$ and the probability of getting false alarm $P_{FA} = 0.03$. However, the algorithm can work with even lower detection probability and higher false alarm rates.

The Markov chain transition matrix for the IMM estimator is

$$[p_{ij}] = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix} \tag{4.37}$$

### 4.5.2 Simulation Scenario

Figure 4.6 and Figure 6.4 show the true target trajectories of the scenario that is used to evaluate the performance of the proposed OOM based joint target tracking and classification algorithm. The aspect-dependent amplitude information from the target is used as the feature information for this simulation. The training data was

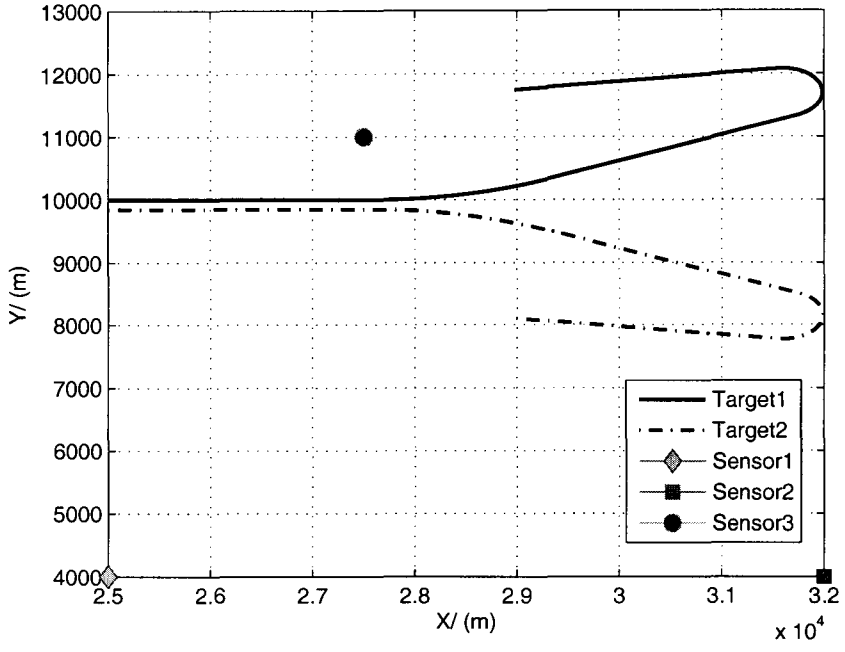



Figure 4.6: Scenario 2: True target trajectories and sensor locations.

also simulated to train the OOM by accounting for target orientation and sensor positions. The targets are assumed to be of hollow cylindrical shape with amplitude measurements being generated according to cylindrical spreading. The cylindrical target dimension can be given by $(L, R)$, where $L$ is the length and $R$ is the radius of the target. The algorithm is tested on a dual Pentium 2.4 GHz Xeon processor machine with 1 GB of memory. The results are based on 100 Monte Carlo runs.

### 4.5.3 Results – Scenario 1

The simple scenario shown in Figure 4.6 was chosen to show the robustness of the proposed algorithm when the target changes its target class. In this scenario, two targets are in a convoy with a constant velocity model for the first 125s and then a

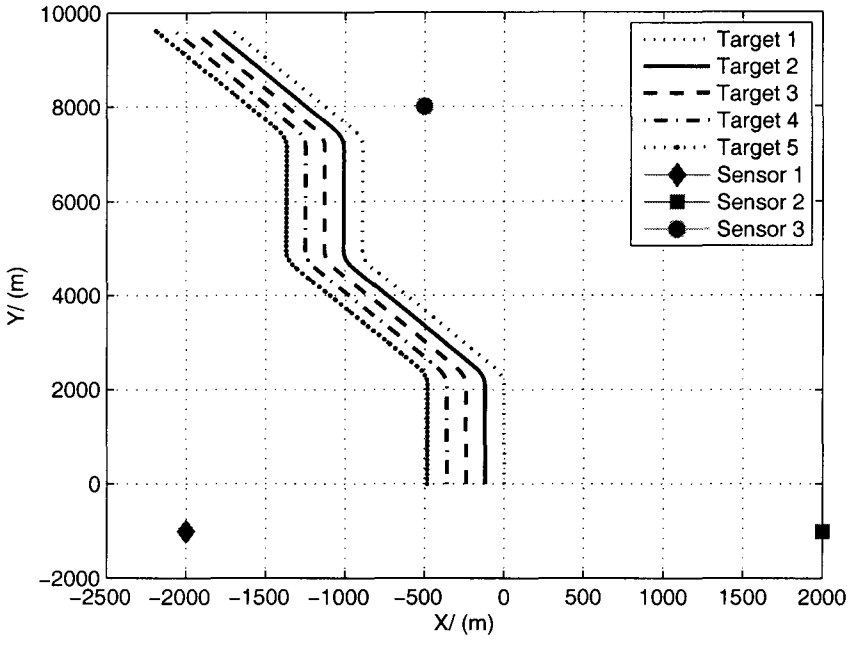



Figure 4.7: Scenario 1: True target trajectories and sensor locations.

coordinated turn model for a period of 85s. Again, targets follow constant velocity motion before making a coordinated turn at time 235s. Finally, targets follow constant velocity motion for the rest of the simulation period. The three sensors are located at (20000m, −10000m), (50000m, 15000m) and (80000m, −10000m). The range and azimuth standard deviation of these sensors are assumed to be 100m and $10^{-2}$ rad, respectively. Targets change their target classes at time 210s. In this scenario, three known types of targets were learned using the OOM with the target class switching probability matrix

$$[p_{ij}] = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{bmatrix} \tag{4.38}$$

The dimensions of three target classes, class 1 to class 3, are (60m, 20m), (40m, 20m) and (10m, 5m), respectively.

Target 1 in the scenario was initially in class 1 and switches to class 2 while target 2 changes its type from class 2 to class 3. Figure 4.8 shows the target class estimates by the proposed OOM based joint target tracking and classification algorithm. The algorithm correctly estimates the target types as well as the target class switches. The estimated target classes for target 1 are target type 1 initially and then switches to type 2 while target 2 is estimated as target class 2 and then class 3. Three time steps were needed to correctly switch the target class at time 210s. The target class probabilities for HMM based classifier are shown in Figure 4.9. The Probability of Correct Classification (PCC) of target 1 and target 2 are 0.74 and 0.76, respectively. When the HMM is employed in target classification, the average PCC of target 1 and target 2 are 0.62 and 0.64, respectively. The simulation with varying number of sensors is carried out for scenario 1 to explore the classification performance. The PCC for varying number of sensors is shown in Table 4.1 and Table 4.2, and it is clear that better classification is achieved with both OOM and HMM based classification algorithms as the number of sensors increases. However, the OOM based classifier outperforms the HMM classifier. When two sensors are used, the OOM classifier and the HMM classifier tend to classify the targets incorrectly. However, when three sensors or four sensors are used, OOM classifier classifies the target types correctly where as the HMM shows misclassifications. Further, for the four sensor case, both the OOM and HMM classifiers classify targets correctly and there were no misclassifications. It should be noted that poor classification results may worsen the target tracking performance[8]. Therefore, a certain minimum number of sensors that capture target information at different views are required for accurate tracking and

classification results. The target structure is also a factor in determining the number of sensors that are needed to provide better results. Complex targets may require more sensors to capture target type information. Using three sensors is adequate for the example illustrated here, where a simple cylindrical target structure is considered.

Table 4.1: PCC for Different Number of Sensors

| Target 1 | | |
|---|---|---|
| Number of sensors | OOM | HMM |
| 2 | 0.62 | 0.55 |
| 3 | 0.74 | 0.62 |
| 4 | 0.92 | 0.87 |
| 5 | 0.94 | 0.88 |

Table 4.2: PCC for Different Number of Sensors

| Target 2 | | |
|---|---|---|
| Number of sensors | OOM | HMM |
| 2 | 0.66 | 0.56 |
| 3 | 0.76 | 0.64 |
| 4 | 0.93 | 0.87 |
| 5 | 0.95 | 0.88 |

### 4.5.4 Results – Scenario 2

The scenario in Figure 6.4 is similar to the one used in [8]. In this scenario, five targets of different classes are following a convoy movement for 500s in the upward directions (positive $\mathbf{Y}$ direction). In this scenario, target class switching is not considered. All five targets maintain around 120m separation among each other. Targets follow a constant velocity motion with three different coordinated turns during (110s – 130s), (240s – 260s) and (370s – 390s), respectively. The sensors are located at $(-2000\text{m}, -1000\text{m})$, $(2000\text{m}, -1000\text{m})$ and $(-500\text{m}, 8000\text{m})$. The range and azimuth

standard deviation of these sensors are 40m and $10^{-3}$ rad, respectively. Synchronized measurements from all three sensors are obtained at an interval of 5s. Here, four known template target classes are trained for OOM and HMM classifiers. The dimensions of four target classes, class 1 to class 4, are (10m, 5m), (20m, 10m), (30m, 5m) and (50m, 20m), respectively. The following target class switching probability matrix is used for scenario 2.

$$[p_{ij}] = \begin{bmatrix} 0.8 & 0.1 & 0.05 & 0.05 \\ 0.05 & 0.8 & 0.1 & 0.5 \\ 0.05 & 0.05 & 0.8 & 0.1 \\ 0.1 & 0.05 & 0.05 & 0.8 \end{bmatrix} \tag{4.39}$$

The average time taken for training an OOM model using 20000 training data is 234s whereas for the HMM model it is 607s. Also the average times taken for a single Monte Carlo run for OOM model and HMM model are 582s and 1124s, respectively. Figures 4.10 – 4.14 compare the target classification performances for OOM and HMM based algorithms. The OOM based method outperformed HMM based algorithm. On average, about 10 to 15 percentage of classification performance improvement is achieved using the OOM method. The average position and velocity Root Mean Squared Errors (RMSE) for each target for scenario 2 are shown in Table 4.3 and Table 4.4, respectively. The RMSE results show that incorporating classification information into data association improves target tracking accuracies. However, the OOM based algorithm outperforms the HMM based algorithm. Figure 4.15 shows the estimated target trajectories for a single run obtained using the OOM based classifier whereas Figure 4.16 and Figure 4.17 show the estimated target trajectories obtained using HMM based classifier and without using any classification

information, respectively. This shows that the tracking performance is improved by incorporating the target class information into data association framework. It can be clearly seen that the performance of the joint target tracking and classification algorithm is significantly better.

Table 4.3: Average Position RMSE Comparison for Scenario 2 / $m$

| Target Label | OOM Classifier | HMM Classifier | Without classification |
|---|---|---|---|
| Target 1 | 5.34 | 6.84 | 10.71 |
| Target 2 | 5.71 | 7.02 | 11.54 |
| Target 3 | 5.12 | 6.34 | 10.48 |
| Target 4 | 5.94 | 7.37 | 12.16 |
| Target 5 | 5.07 | 6.17 | 10.46 |

Table 4.4: Average Velocity RMSE Comparison for Scenario 2 / $ms^{-1}$

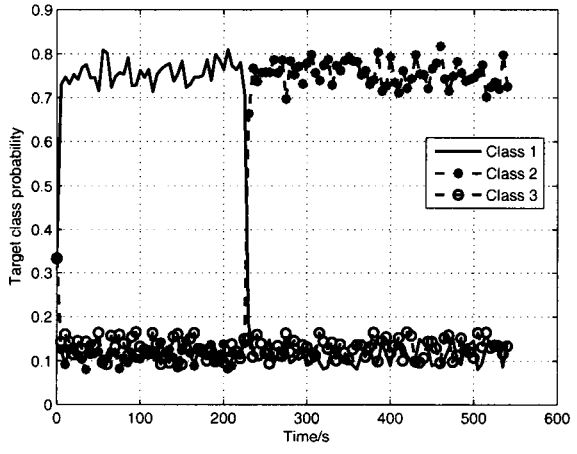
| Target Label | OOM Classifier | HMM Classifier | Without classification |
|---|---|---|---|
| Target 1 | 9.65 | 12.58 | 14.45 |
| Target 2 | 9.16 | 12.09 | 14.24 |
| Target 3 | 10.12 | 12.93 | 15.09 |
| Target 4 | 9.87 | 12.67 | 14.95 |
| Target 5 | 10.73 | 13.04 | 15.26 |

## 4.6 Summary

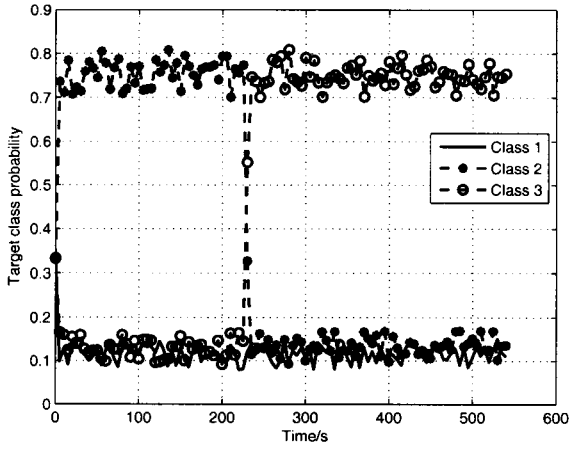
In this chapter, a new joint target tracking and classification algorithm based on Observable Operator Model was presented. The OOM is used to model the classifier as it affords both mathematical simplicity and better algorithmic efficiency. Target classification and multisensor fusion are integrated together using multiframe data association framework, which ensures better classification and tracking results. A significant improvement in performance was achieved by using two-way exchange of

useful information between the tracker and the classifier. The simulation studies explicitly show the superior performance of the proposed algorithm. Therefore, the fundamental relationship between the target classification and tracking can be used to improve data association resulting not only in better target tracking performance but also improved classification.

The results on the relative performance of the proposed OOM classification and HMM based classification indicate that more accurate target class estimation and better track purity are feasible with the former. Higher probability of correct classification (close to 0.95) is achievable with more sensors and therefore, the number of sensors is also a factor in determining the correct target class and to obtain better tracking accuracies.

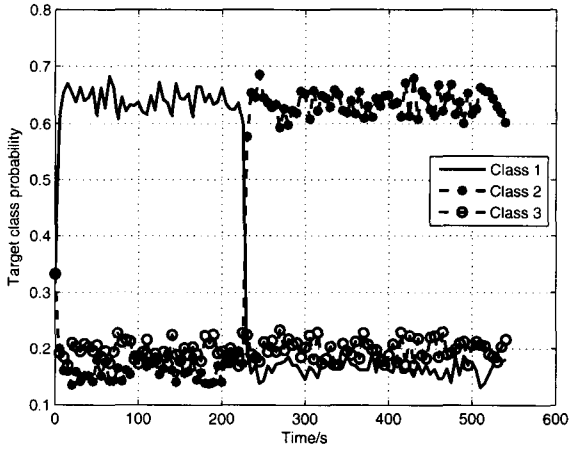



(a) Target 1

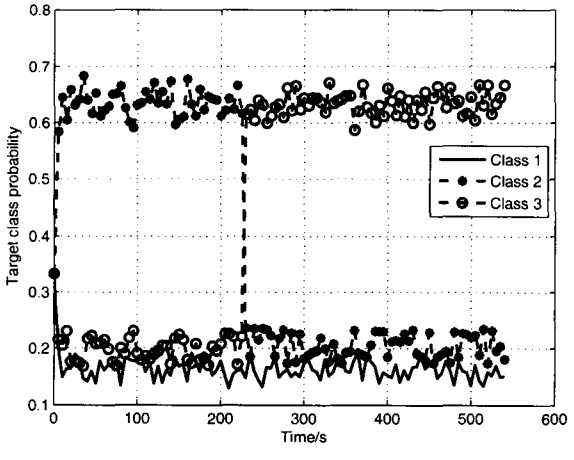



(b) Target 2

Figure 4.8: Scenario 1: Target class estimate using OOM.

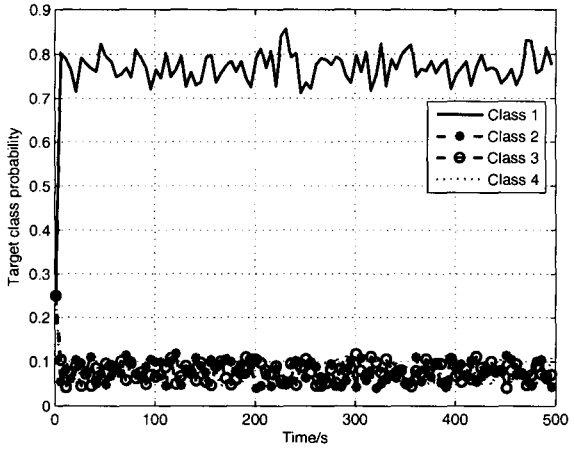



(a) Target 1

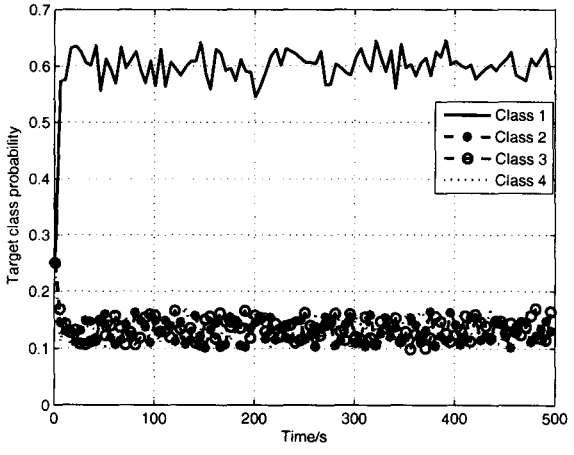



(b) Target 2

Figure 4.9: Scenario 1: Target class estimate using HMM.

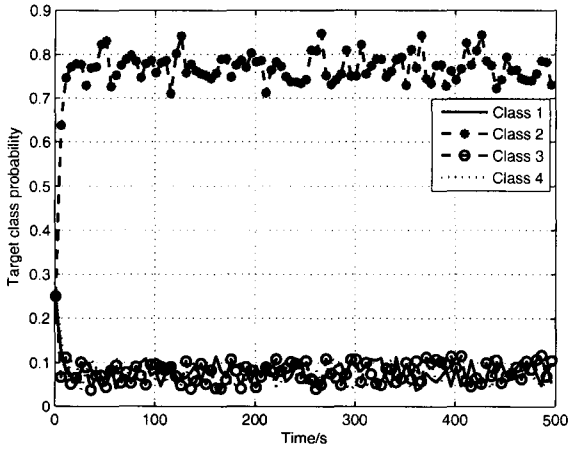



(a) OOM

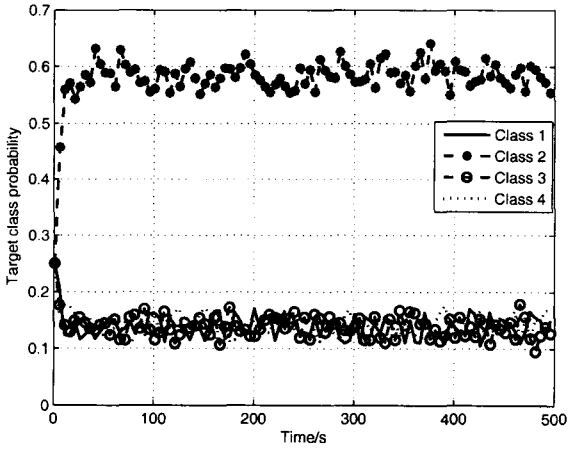



(b) HMM

Figure 4.10: Scenario 2: Target class estimate for target 1.

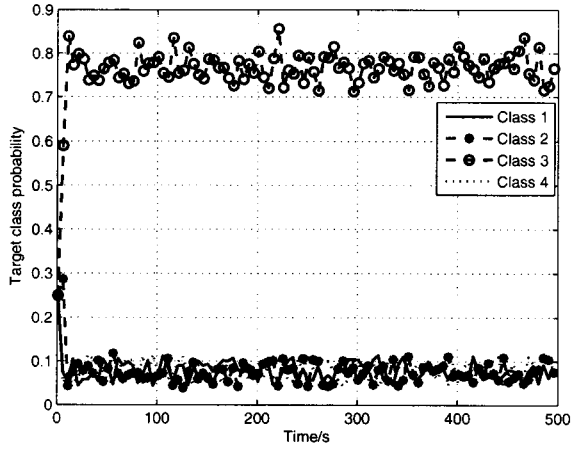



(a) OOM

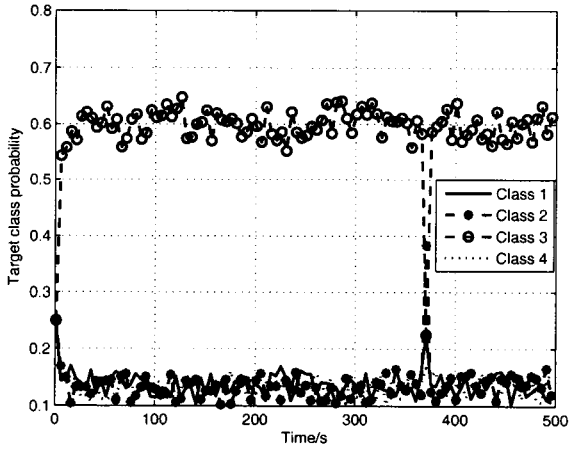



(b) HMM

Figure 4.11: Scenario 2: Target class estimate for target 2.

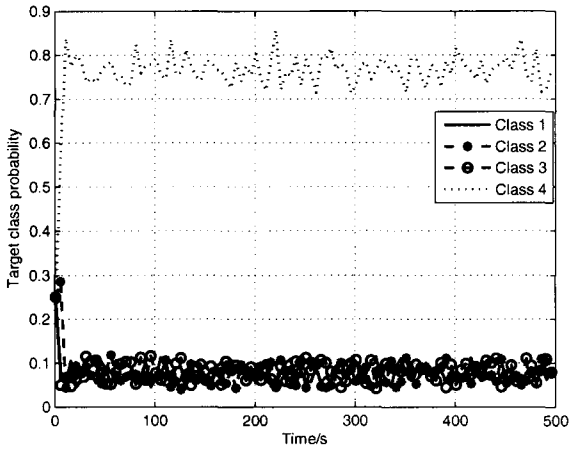



(a) HMM

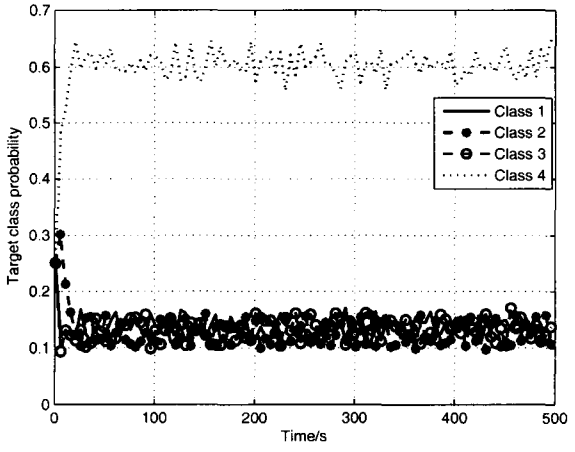



(b) HMM

Figure 4.12: Scenario 2: Target class estimate for target 3.

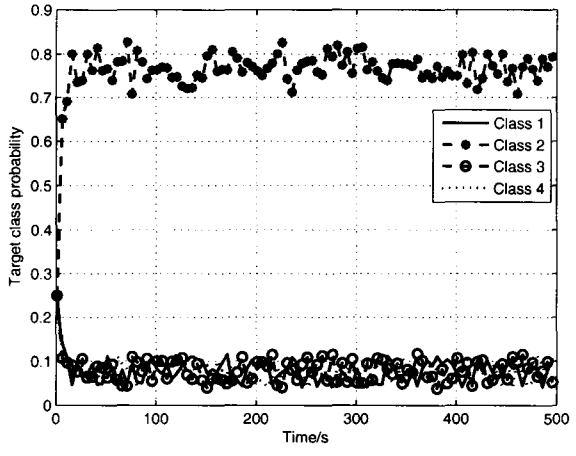



(a) OOM

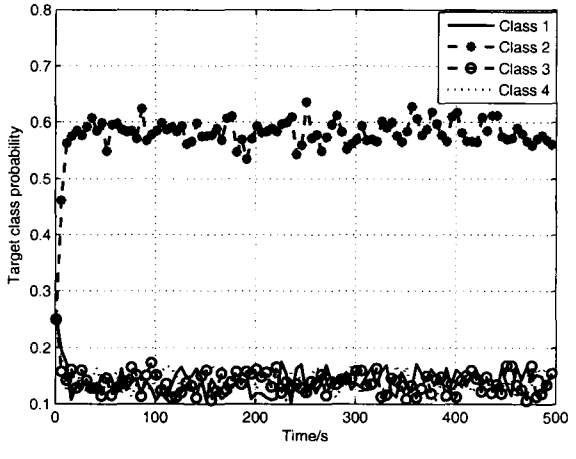



(b) HMM

Figure 4.13: Scenario 2: Target class estimate for target 4.




(a) OOM




(b) HMM

Figure 4.14: Scenario 2: Target class estimate for target 5.

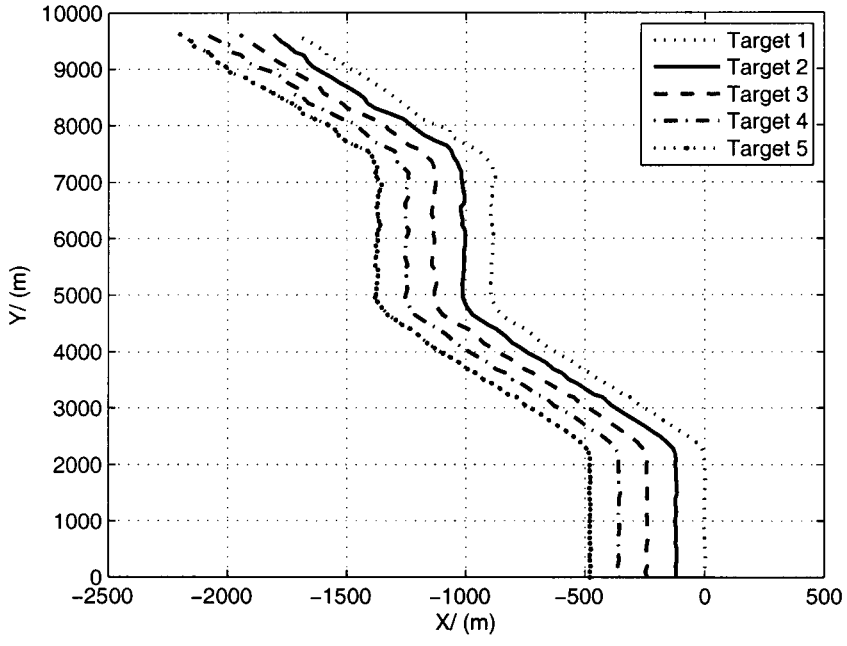



Figure 4.15: Scenario 2: Estimated target trajectories using OOM.

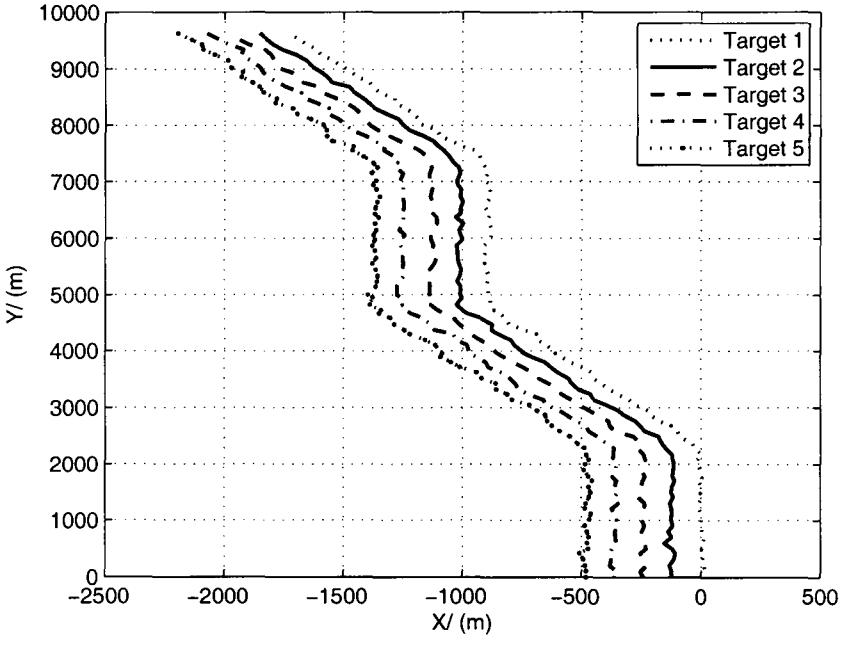



Figure 4.16: Scenario 2: Estimated target trajectories using HMM.

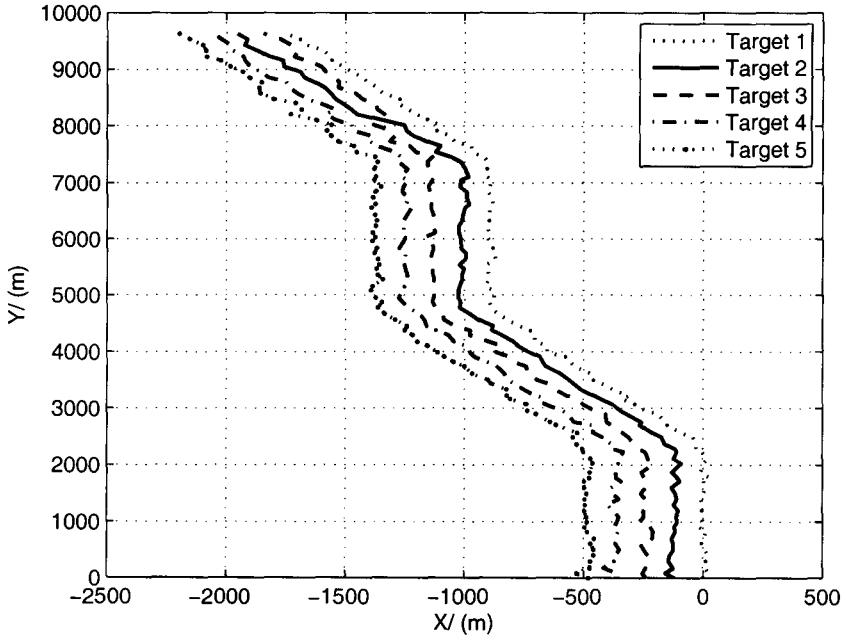



Figure 4.17: Scenario 2: Estimated target trajectories without classification.

# Chapter 5

# Improved Target Tracking Using Kinematic Measurements and Orientation Estimates

## 5.1 Problem Formulation

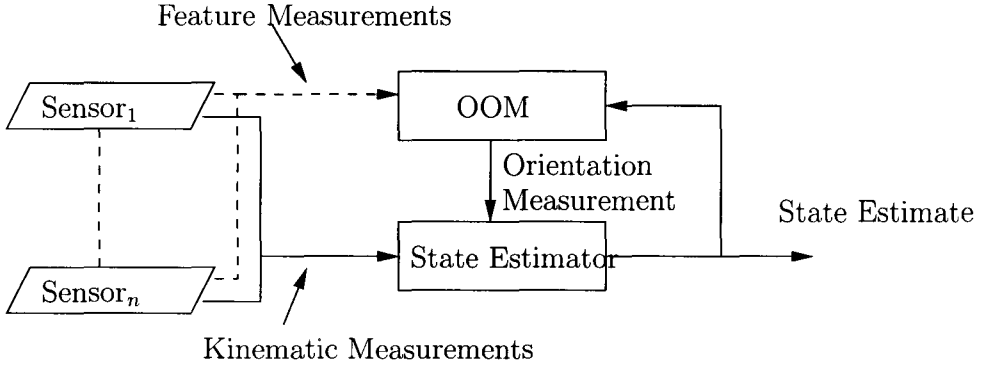



Figure 5.1: Block diagram of proposed algorithm.

In target tracking problems, the information from multiple views of a target can

reasonably improve the tracking accuracies. The feature measurements from multiple views together with Observable Operator Model (OOM) is used to estimate the target orientation as well as in target tracking process to obtain better accuracies. The block diagram of the proposed algorithm is shown in Figure 5.1. Each sensor produces kinematic as well as the feature measurements. In this work, the sensors are assumed to be reporting bearing (kinematic) measurement and amplitude (feature) information at every time step. The feature measurements from all sensors are given to the OOM based orientation measurement generator. This module also uses the estimated state of previous time step from the tracker to predict the current target position. The output of the OOM module and the kinematic measurements directly from the sensors are used to estimate the current target state. The use of orientation information in the state estimation improves the filter performance, which is discussed in Section 6.6.

The correlation between the target orientation information and the current direction of target movement is taken into account with the assumption that the course of the target lie nearly along the major axis of the target [44]. The significant coupling between the target orientation and the velocity direction may improve target tracking, especially when the target maneuvers. Suppose the target is moving in the $(X, Y)$ plane as indicated in Figure 5.2. Multiple sensors located in different location in the $(X, Y)$ plane are used to observe the target motion as well as the target orientation changes.

Assume that there are $n$ number of sensors that report bearing and feature information at time $k$. The state of the target at time $k$ is defined by the four dimensional vector

$$\mathbf{x}_k = [x_k \quad \dot{x}_k \quad y_k \quad \dot{y}_k]' \tag{5.1}$$

where $x_k$ and $y_k$ are the positions of the target, and $\dot{x}_k$ and $\dot{y}_k$ are the velocities of

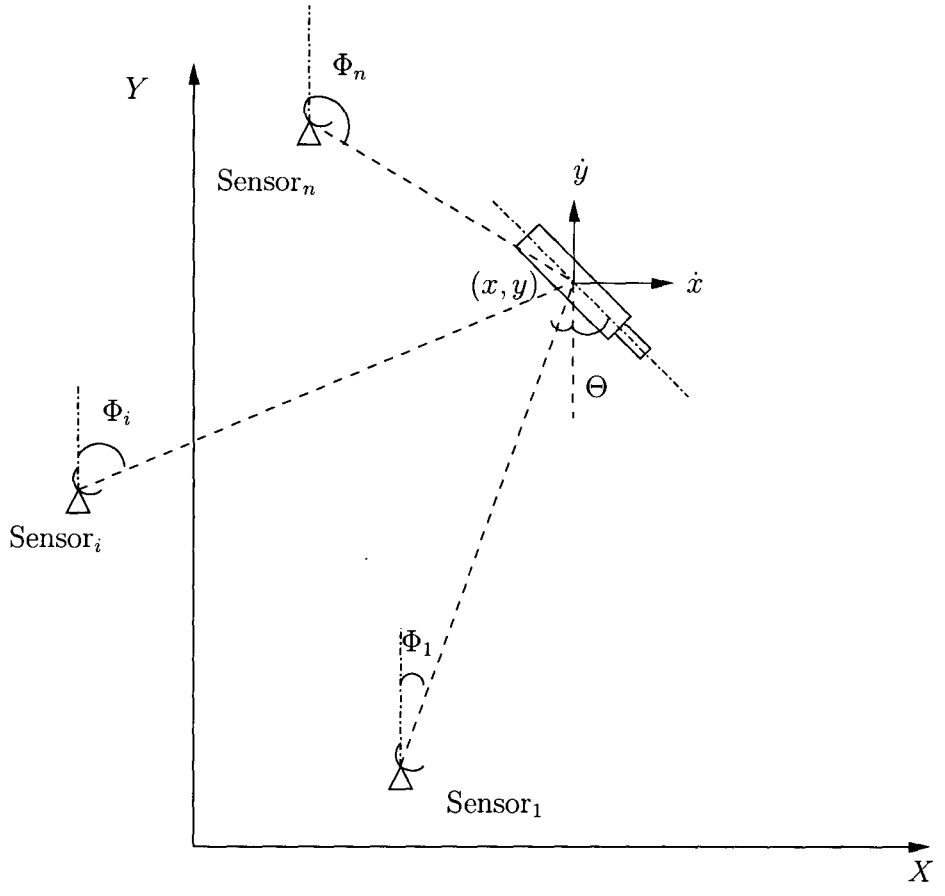



Figure 5.2: Tracking diagram

the target in the **X** and **Y** coordinates, respectively.

The target motion model in the Cartesian coordinate system is

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}) + \Gamma_k \, \nu_{k-1} \tag{5.2}$$

where $\nu_{k-1}$ is the white Gaussian process noise with covariance matrix $\Gamma$. Here $f_k(.)$ is linear/nonlinear system model.

The bearing $\Phi_i$ of the target from sensor $r$ located at $(x_k^r, y_k^r)$ at time $k$ is given

by

$$\Phi_k^r = \text{atan}\left(\frac{x_k - x_k^r}{y_k - y_k^r}\right) + v_k^r \tag{5.3}$$

$$= h_r + v_k^r \tag{5.4}$$

where $(x_k^r, y_k^r)$ is the position of sensor $r$. The range of possible bearing measurements is $[0, 2\pi)$. If the bearing measurement of sensor $r$ is range azimuth $\Phi_k^r$, then the augmented kinematic measurements and the feature measurement $\psi_i$ of sensor $r$ is given by

$$\mathbf{z}_k = \begin{bmatrix} \Phi_k^r \\ \psi_k^r \end{bmatrix} \tag{5.5}$$

.

The target orientation measurement $\Theta_k$ is defined as the angle between the major axis of the target in moving direction and the negative $Y$ axis, and given by

$$\Theta_k = \text{atan}\left(-\frac{\dot{x}_k}{\dot{y}_k}\right) + v_k^\Theta \tag{5.6}$$

$$= h_\Theta + v_k^\Theta \tag{5.7}$$

where $v_k^\Theta$ is the white Gaussian process noise. However, there is no sensor available that efficiently measures the target orientation. Therefore, the target orientation measurement $\Theta$ is estimated using the training-based OOM model by incorporating Gaussian mixture model for target feature returns. The detailed description of this algorithm is illustrated in Section 5.2. The OOM based target orientation estimator first classifies the target class (when more than one possible target class exists) using the feature measurement [68] and then process the feature measurements to model

the target orientation density. The estimated target orientation is used as measurement for target tracking. The effective measurement model can be obtained by augmenting the bearing measurements from all receivers. The estimated orientation measurements and given by

$$\mathbf{Z}_k = \begin{bmatrix} \Phi_k^1 \\ \Phi_k^2 \\ \vdots \\ \Phi_k^n \\ \Theta_k \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \\ h_\Theta \end{bmatrix} + \begin{bmatrix} v_k^1 \\ v_k^2 \\ \vdots \\ v_k^n \\ v_k^\Theta \end{bmatrix}. \tag{5.8}$$

A highly maneuvering target can be tracked using (5.2) and (5.8) by employing an Interacting Multiple Model (IMM) [6] based tracking algorithms. However, the better performance of the target tracking algorithm totally depends on the target orientation measurement estimator.

### 5.1.1 Track Initialization

In this sub-section, a simple track initialization technique that uses target orientation measurement is presented. Track initialization [18][33][10] is one of the crucial task in target tracking, especially for high maneuvering targets with high velocities [40]. Practical implementation of track initialization is usually done using the kinematic measurements and prior knowledge of target velocities [6]. The approximate target position can be evaluated using the kinematic measurements and the velocities in $X$ and $Y$ directions are initialized randomly with known Gaussian prior information. Typical prior information are the possible minimum velocity $v_{\min}$ and maximum velocity $v_{\max}$ of the target. In such case one can use $v_{\min}$ as initial velocity estimate and

$\frac{1}{2}(v_{\max} - v_{\min})$ as the standard deviation [6]. This method randomly sets the target heading directions in $[0, 2\pi)$. The orientation measurement adds more information to prior knowledge and therefore reduces the uncertainty in target heading direction. Let initial target velocities in $X$ and $Y$ directions be $\dot{x}_0$ and $\dot{y}_0$, respectively and the initial target orientation measurement be $\Theta_0$. Therefore, the velocity in the target heading can be written as

$$\dot{x}_0 \sin\Theta_0 - \dot{y}_0 \cos\Theta_0 \sim \mathcal{N}\left(v(0); v_{\min}, \frac{1}{4}(v_{\max} - v_{\min})^2\right). \tag{5.9}$$

The velocity orthogonal to target heading is zero and thus

$$\dot{x}_0 \cos\Theta_0 + \dot{y}_0 \sin\Theta_0 = 0. \tag{5.10}$$

The equations (5.9) and (5.10) can be solved for initial velocities $\dot{x}_0$ and $\dot{y}_0$.

Another advantage of using target orientation information in track initialization is the accountability of doppler measurement. Suppose sensors return range $R$ and doppler $\dot{R}$ measurements. The approximate initial target position can be found using range measurements. The doppler measurements cannot be utilized efficiently in track initialization without knowing the target heading. However, the doppler measurement can be efficiently utilized by accounting target orientation information. The doppler can be written in terms of velocities in $X$ and $Y$ direction and azimuth information $\Phi$ as

$$\dot{R}_0 = -(\dot{x}_0 \sin\Phi_0 + \dot{y}_0 \cos\Phi_0). \tag{5.11}$$

Here the azimuth information from a sensor can be calculated once the approximate

target position is estimated. If the target orientation information $\Theta_0$ is available, then the doppler measurement can be given in terms of $\Theta_0$ and $\Phi$ as

$$\dot{R}_0 \cos(\Theta_0 + \Phi_0) = \dot{x}_0 \sin\Theta_0 - \dot{y}_0 \cos\Theta_0. \tag{5.12}$$

Since doppler measurement $\dot{R}_0$, azimuth $\Phi_0$ and target orientation $\Theta_0$ are available, equations (5.11) and (5.12) can be solved for initial velocities $\dot{x}_0$ and $\dot{y}_0$ in $X$ and $Y$ directions, respectively.

## 5.2 Observable Operator Model Based Orientation Measurement Estimation

### 5.2.1 OOM Classification

There may be more than one possible target classes available in the scenario. Thus, the target class has to be determined before the target orientation measurement is generated. Assume that there are $\alpha$ possible template target classes available. An OOM is designed to characterize each of the $\alpha$ class of targets as follows:

$$\xi(k) \in \{\xi_i\}_{i=1}^{\alpha} \tag{5.13}$$

Assume that a set of training sequences of each target class and the following tasks are performed in order to estimate the likelihood:

- For each target class in $\{\xi_i\}_{i=1}^{r}$, an OOM is built $(\mathbb{R}^m, (\tau_a)_{a\in\Phi}, w_0)$ and the model parameters that optimize the likelihood of the training set of the feature observations for the $i^{\text{th}}$ target class are estimated. The OOMs obtained at the

end of this step are called class-specific pre-trained OOMs.

- During the estimation operation, the feature measurements from the sensors are recognized by employing these class-specific pre-trained OOMs.

The likelihood $\Omega_i(k)$ of the feature measurement received at time step $k$ for target class $i$ can be evaluated as

$$\Omega_i(k) = p(\{\psi_1, \ldots, \psi_S\}_k | \xi = \xi_i) \tag{5.14}$$

$$= \mathbf{1}[\tau_{\{\psi_1,\ldots,\psi_S\}_k} w_0]_i \tag{5.15}$$

Now, the target class can be estimated using the maximum likelihood (ML) estimator and given by

$$\xi^{\mathrm{ML}}(k) = \arg \max_{\xi_1 \leq \xi_i \leq \xi_r} \Omega_i(k) \tag{5.16}$$

Once the target class is determined, the target orientation measurement has to be estimated.

### 5.2.2 OOM for Orientation Measurement

Once the target class is evaluated from the sensor feature measurement set, the target orientation measurement has to be generated. The observed feature measurements depend on target-to-sensor aspect angle and therefore the discrete aspect angle need to be incorporated in order to find the target orientation. Moreover, the aspect angle needs to be discretized as the OOM theory based on discrete variables. Further, the OOM model explained above does not handle target-to-sensor aspect angle explicitly. Thus, another OOM Model that is capable of handling target-to-sensor discrete state needs to be developed. The new OOM can be derived from the already learned OOM

as the OOM model is based on linear algebra theory [37][38]. Assume that the target-to-sensor aspect state can be divided into finite set $\mathcal{S} = \{\theta_1, \theta_2, \ldots, \theta_n\}$ of hidden outcomes. A hidden $m$-dimensional linearly dependent process can be specified by an OOM $\mathcal{H} = (\mathbb{R}^m, (\tau_{\theta_i})_{\theta_i \in \mathcal{S}}, w_0)$ [37]. Assume that a receiver obtains measurement when the target is at state $\theta_i$. Then the observable outcome $\psi_j$ from a finite set $\mathcal{O} = \{\psi_1, \psi_2, \ldots, \psi_n\}$ is emitted with a probability of $g_{\psi_j|\theta_i} = P[Y_k = \psi_j | \theta = \theta_i]$. Choose emission probability arbitrarily, except for the constraint that the emission probability matrix be nonsingular.

Observable operators $\tau_{\theta_i}, i = 1, \ldots, n$, of the hidden linear dependent process satisfy the following system of linear equations:

$$\begin{aligned} \tau_{\psi_1} &= g_{\psi_1|\theta_1}\tau_{\theta_1} + \cdots + g_{\psi_1|\theta_n}\tau_{\theta_n} && (5.17) \\ \vdots &= \qquad\qquad \vdots \\ \tau_{\psi_n} &= g_{\psi_n|\theta_1}\tau_{\theta_1} + \cdots + g_{\psi_n|\theta_n}\tau_{\theta_n} && (5.18) \end{aligned}$$

These equations, (5.17) – (5.18), can be written in matrix form as follows:

$$\tau_\psi = \begin{bmatrix} g_{\psi_1|\theta_1} & \cdots & g_{\psi_1|\theta_n} \\ \vdots & \vdots & \vdots \\ g_{\psi_n|\theta_1} & \cdots & g_{\psi_n|\theta_n} \end{bmatrix} \tau_\theta \qquad (5.19)$$

where $\tau_\psi = [\tau_{\psi_1} \quad \tau_{\psi_2} \quad \ldots \quad \tau_{\psi_n}]'$ and $\tau_\theta = [\tau_{\theta_1} \quad \tau_{\theta_2} \quad \ldots \quad \tau_{\theta_n}]'$. Here $\tau_{\psi_i}$ and $\tau_{\theta_i}$ are square matrices and $i^{\text{th}}$ block of $\tau_\psi$ and $\tau_\theta$, respectively. This system can be solved for $\tau_\psi$ since the emission transition matrix is nonsingular. The feature information of an unknown target is measured at $L$ aspects $\Theta = \theta_1, \theta_2, \ldots, \theta_L$ with

corresponding time-dependent feature information $\mathcal{F}(t, \theta_l), l = 1, \ldots, L$. The sequence of continuous-valued feature information $Y = y_1, y_2, \ldots, y_L$ is converted into a set of discrete-valued feature information using a vector quantization technique [21]. The vector quantization is performed by mapping the feature vectors onto a set of discrete points, representation of continuous-valued distribution of the features $Y \rightarrow \psi = \{\psi_1, \psi_2, \ldots, \psi_L\}$ with $\psi_l \in \mathcal{C}$, where $\mathcal{C}$ is code book for possible discrete feature measurement symbols for a specific target..

### 5.2.3 Emission Probability Distribution

Now, the emission probability matrix has to be derived in order to use with new OOM. The emission probability depends on target type and shape. Since the typical targets have complex shapes and sizes. There is no closed form expression to calculate the emission probabilities. However, the emission probabilities can be approximated through a Gaussian mixture density. Let $g_{y|\theta_m}$ be the probability density of observing continuous valued feature vector in state $\theta_m$. The continuous valued feature density for state $\theta_m$ can be modeled using a continuous Gaussian mixture model as given below:

$$g_{y|\theta_m}(y|\theta_m) = \sum_{k=1}^{K} w_{km} \mathcal{N}(\mu_{km}, \Sigma_{km}) \tag{5.20}$$

where $\mathcal{N}(\mu_{km}, \Sigma_{km})$ is a multivariate Gaussian density with mean $\mu_{km}$ and covariance $\Sigma_{km}$ and $w_{km}$ is the mixture coefficient for $k^{\text{th}}$ component of state $\theta_m$. The variable dimension is equal to the dimension of the feature vector dimension. The coefficient $w_{km}$ should be a non-negative value and satisfies

$$\sum_{k=1}^{K} w_{km} = 1, \ \text{m} = 1, \ldots, \text{n} \tag{5.21}$$

The mixture density parameters can be estimated using a clustering technique of the training data for each target state. A K-mean algorithm is employed to cluster the data from state $\theta_m$ into $K$ distinct regions using the mean $\mu_{km}$ as the centroid of the $k^{\text{th}}$ region. Once the model for probability density of observing continuous valued feature vector in each state have been defined, target emission probabilities of equivalent OOM model can be found.

### 5.2.4 Orientation Measurement Estimation

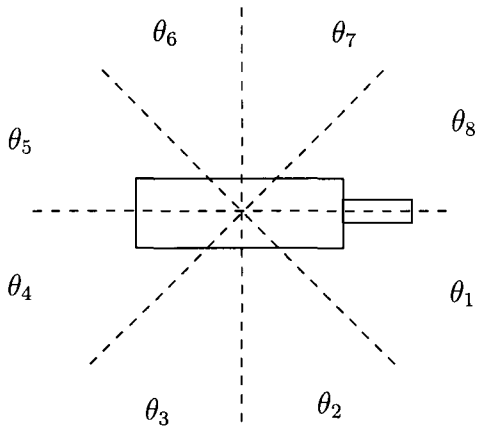



Figure 5.3: Discretized target-to-sensor aspect angle.

Since the OOM-based orientation estimator is used together with target tracking algorithm, the approximate current target position is known and thus, the sensors can be arranged such that feature measurement received are arranged in clockwise or anti-clockwise pattern. The same convention has to be followed while learning the OOM as well. This eliminates the unwanted possible target-to-sensor aspect sequences. Since the predicted target position from the tracker is used to determine the target-to-sensor aspect sequence, there is uncertainty in the estimated target position. This uncertainty leads to an error in target-to-sensor aspect state sequence

calculation, especially when the sensors are very close to the target. If the sensor locations are assumed to be far away from the target location, then the error in aspect state sequence calculation can be eliminated. This assumption is reasonable as the targets and sensors are closely located at less frequent occurrences. As the target evolves, there may be occasions where the target and sensors are close to each other. However, those sensors that are located very close to the targets can be omitted when calculating the target orientation measurement. Let $\Theta$ be the target

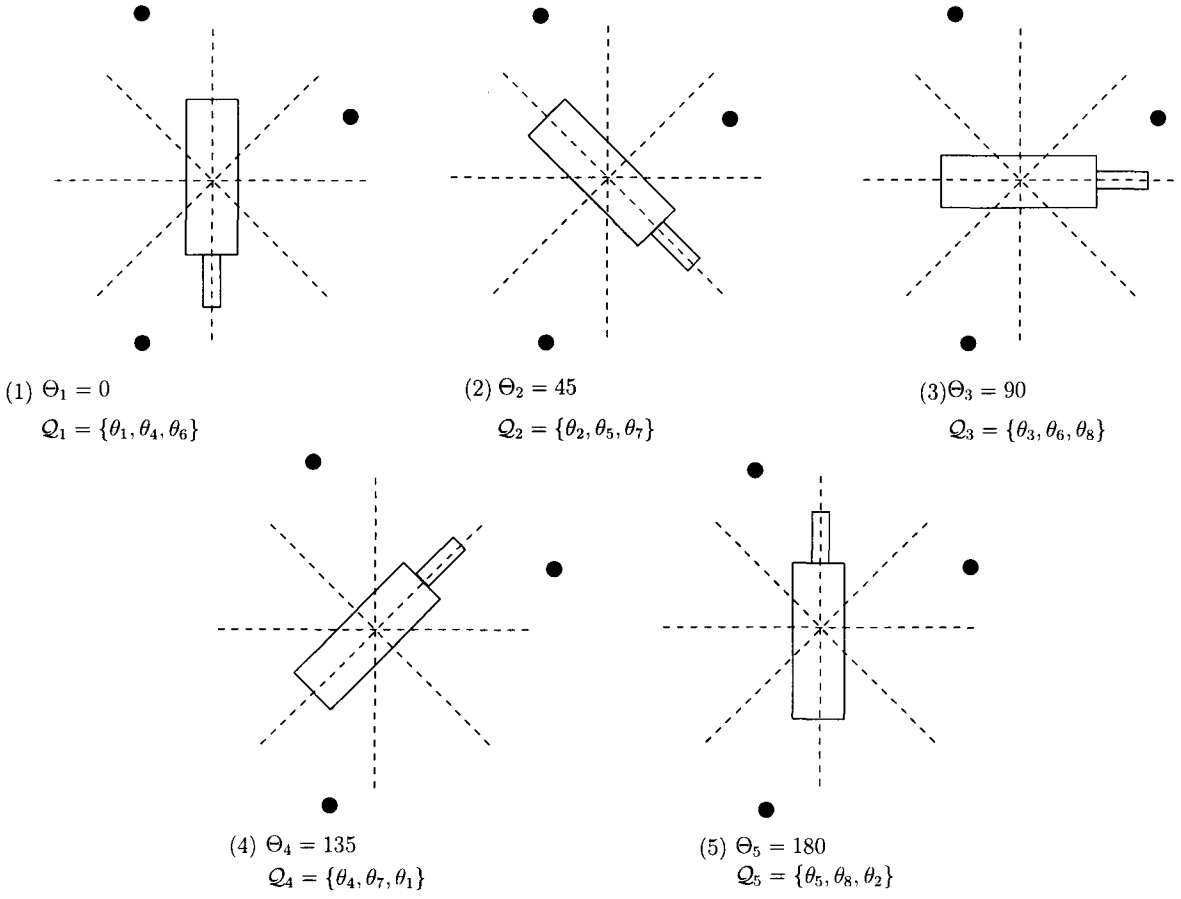



Figure 5.4: Different target orientation and corresponding aspect sequences.

orientation, which has to be discretized in order to estimate use with OOM. In this

work a number of discrete aspect states are used as target orientation division as well. Two dimensional target is considered and therefore, these angle can be discretized in $[0, 2\pi)$. A sample target-to-sensor aspect state decomposition is shown in Figure 5.3. This can be extended for three dimensional targets by having angular divisions in $4\pi$ sphere. However, this needs a huge amount of training data for each target to learn the OOM and it is computationally involved. There is a direct connection between the target orientation $\Theta$ and the target-to-sensor aspect sequence $\mathcal{Q}$ as shown in Figure 5.4. At a particular time step, sensor locations are fixed. Therefore, each possible target orientation $\Theta_j$ can be directly mapped into a specific target-to-sensor aspect sequence $\mathcal{Q}_j$. For an OOM model of a given target $\xi_T$, the probability of an observed feature measurement sequence for any underlying target-to-sensor aspect state sequences $\mathcal{Q}_j = \{q_1, \cdots, q_N\}$ is given by

$$P(\psi|\mathcal{Q}_j, \xi_T) = g^T_{y_1|q_1}\, g^T_{y_2|q_2} \cdots g^T_{y_n|q_n} \tag{5.22}$$

Using the Baye's rule $P(\mathcal{Q}_j|\psi, \xi_T)$ can be written as

$$P(\mathcal{Q}_j|\psi, \xi_T) = \frac{P(\psi|\mathcal{Q}_j, \xi_T)P(\mathcal{Q}_j|\xi_T)}{P(\psi|\xi_T)} \tag{5.23}$$

Here $P(\psi|\xi_T)$ is the normalizing constant. Therefore, $P(\mathcal{Q}_j|\psi, \xi_T)$ can be further simplified by

$$P(\mathcal{Q}_j|\psi, \xi_T) = \frac{1}{c}P(\psi|\mathcal{Q}_j, \xi_T)P(\mathcal{Q}_j|\xi_T) \tag{5.24}$$

Further, the probability of having a sequence of target-to-sensor angles $\mathcal{Q}_j$ can be obtained by using the aspect angle incorporated OOM and is given by

$$P(\mathcal{Q}_j|\xi_T) = \tau_{q_1} \cdots \tau_{q_N} w_0^{\Theta} \tag{5.25}$$

here, $\tau_{q_1} \cdots \tau_{q_N}$ are the operator multiplication of the aspect angle incorporated OOM. Therefore

$$P(\mathcal{Q}_j|\psi, \xi_T) = g^T_{y_1|q_1}\ g^T_{y_2|q_2} \cdots g^T_{y_n|q_n}\ \tau_{q_1} \cdots \tau_{q_N} w_0^{\Theta} \tag{5.26}$$

Each target-to-sensor aspect state corresponds to a particular target orientation. If the target-to-sensor aspect state sequence $\mathcal{Q}_j$ represents target orientation $\Theta_j$, then the first and second moment of the orientation measurement can be written as

$$\Theta = \sum_{j=1}^{N} \Theta_j P(\mathcal{Q}_j|\psi) \tag{5.27}$$

$$\sigma_\Theta = \sum_{j=1}^{N} P(\mathcal{Q}_j|\psi)(\Theta_j - \Theta)(\Theta_j - \Theta)' \tag{5.28}$$

respectively. The mean $\Theta$ and covariance $\sigma_\Theta$ can be as an additional information together with bearing measurements. In the absence of proper model for sensor measurement, it is assumed to be a Gaussian conditioned on aspect angular state sequences.

## 5.3 Posterior Cramer-Rao Lower Bound with Orientation Measurements

Discrete-time nonlinear filtering or the associated problem of adaptive system identification arise in various application such as adaptive control, analysis and prediction of nonstationary time series. However, the optimal estimator for these problems cannot be built in general, and it is necessary to turn to one of the large number of existing suboptimal filtering techniques [2]. Assessing the achievable performance may be difficult and therefore, simulation based approaches have to be obtained to compare the

lower bounds corresponding to optimum performance. Lower bounds give an indication of performance limitations, and consequently, they can also be used to determine whether imposed performance requirements are realistic or not. In time invariant statistical models, a commonly used lower bound is the Cramer-Rao Bound (CRB), given by the inverse of the Fisher information matrix. In the time-varying systems, the estimated parameter vector has to be considered as random since it corresponds to an underlying nonlinear, randomly driven model. A lower bound that is analogous to the CRB for random parameters was derived in [75]. This bound is usually referred to as the Van Trees version of the CRB or Posterior CRB (PCRB).

The PCRLB [34] provides a mean square error bound on the performance of any unbiased estimator of an unknown and stochastic parameter vector. The PCRLB provides a powerful tool, enabling one to determine a bound on the optimal achievable accuracy of target state estimation. This provides a mechanism for establishing the maximum degree to which sub-optimal filtering algorithms could potentially be improved [9]. In the context of target tracking PCRLB provides an indication as to where future algorithmic development should be focused. The PCRLB also provides a means of establishing limitations in current sensor technology, enabling one to easily determine impossible scenarios (i.e., scenarios in which operational requirements cannot possibly be met) given current sensor capabilities.

In this section the PCRLB for target state estimation error is derived using recursive computation method for the Fisher information matrix for single maneuvering target model. Suppose $\mathbf{x}_k$ is the true target state at time $k$, which is to be estimated by the unbiased estimator. Let $C_k$ be the error covariance. The PCRLB is defined to be the inverse of the Fisher information matrix (FIM), $J_k$ [73], gives a lower bound

of the error covariance matrix $C_k$, i.e.,

$$C_k \triangleq \mathbb{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T\right] \geq J_k^{-1} \tag{5.29}$$

where $E$ denotes expectation and $[.]^T$ denotes the transpose and $\hat{\mathbf{x}}_k$ is the estimated state.

The sequence of information matrices $\{J_k\}$ for the target state vector $\{\mathbf{x_k}\}$ obeys the following recursion [73]:

$$J_k = D_{k-1}^{22} - D_{k-1}^{21}(J_{k-1} + D_{k-1}^{11})^{-1}D_{k-1}^{12} \tag{5.30}$$

where $J_k$ and $J_{k-1}$ are the Fisher information matrices of the target at time $k$ and $k-1$, respectively, and

$$\begin{aligned}
D_{k-1}^{11} &= \mathbb{E}\{-\Delta_{\mathbf{x}_{k-1}}^{\mathbf{x}_{k-1}} \log\ p(\mathbf{x}_k|\mathbf{x}_{k-1})\} \\
&= F^T\Gamma_k^{-1}F \qquad (5.31) \\
D_{k-1}^{12} &= \mathbb{E}\{-\Delta_{\mathbf{x}_{k-1}}^{\mathbf{x}_k} \log\ p(\mathbf{x}_k|\mathbf{x}_{k-1})\} \\
&= -\Gamma_k^{-1}F \qquad (5.32) \\
D_{k-1}^{21} &= \mathbb{E}\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_{k-1}} \log\ p(\mathbf{x}_k|\mathbf{x}_{k-1})\} \\
&= -F^T\Gamma_k^{-1} \qquad (5.33) \\
D_{k-1}^{22} &= \mathbb{E}\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log\ p(\mathbf{x}_k|\mathbf{x}_{k-1})\} + E\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log\ p(\mathbf{Z}_k|\mathbf{x}_k)\} \\
&= \Gamma_k^{-1} + J_{\mathbf{Z}_k} \qquad (5.34)
\end{aligned}$$

where $\Delta$ is the Hessian operator, $F$ is the Jacobian of the linear/nonlinear process

function $f$, $Q$ is the covariance matrix of the additive process noise and

$$J_{\mathbf{Z}_k} = \mathbb{E}\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log\ p(\mathbf{Z}_k|\mathbf{x}_k)\}. \tag{5.35}$$

The Hessian $\Delta_a^b$ is a second-order partial derivative operator whose $(i,j)^{\text{th}}$ term is given by

$$\Delta_a^b(i,j) = \frac{\partial^2}{\partial a(i)\partial b(j)} \tag{5.36}$$

and $a(i)$ and $b(j)$ are the $i^{\text{th}}$ and $j^{\text{th}}$ components of vector $a$ and $b$, respectively. The equation (5.30) can be further simplified using the matrix inverse lemma [6], and given by

$$J_k = \left[\Gamma_k + F(J_{k-1})^{-1}F^T\right]^{-1} + J_{\mathbf{Z}_k} \tag{5.37}$$

where $J_{\mathbf{Z}_k}$ is the information contribution of augmented measurement vector $\mathbf{Z}_k$. Further, by assuming each sensor has independent measurement process, $J_{\mathbf{Z}_k}$ can be written as [71]

$$\begin{aligned} J_{\mathbf{Z}_k} &= \mathbb{E}\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log\ p(\mathbf{Z}_k|\mathbf{x}_k)\} && (5.38)\\ &= \sum_{i=1}^{n} \mathbb{E}\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log\ p(\Phi_k^i|\mathbf{x}_k)\} + \mathbb{E}\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log\ p(\Theta_k|\mathbf{x}_k)\} && (5.39)\\ &= \sum_{i=1}^{n} J_{\Phi_k^i} + J_{\Theta_k} && (5.40) \end{aligned}$$

where

$$\begin{aligned} J_{\Phi_k^i} &= \mathbb{E}\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log\ p(\Phi_k^i|\mathbf{x}_k)\} && (5.41)\\ J_{\Theta_k} &= \mathbb{E}\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log\ p(\Theta_k|\mathbf{x}_k)\}. && (5.42) \end{aligned}$$

The information contribution due to bearing measurement of sensor $i$ can also be written as

$$J_{\Phi_k^i} = \mathbb{E}\{[\Delta_{\mathbf{x}_k} \log\ p(\Phi_k^i|\mathbf{x}_k)][\Delta_{\mathbf{x}_k} \log\ p(\Phi_k^i|\mathbf{x}_k)]^T\}. \tag{5.43}$$

The bearing measurement uncertainty is assumed to be Gaussian noise and thus

$$p(\Phi_k^i|\mathbf{x}_k) = \mathcal{N}(\hat{\Phi}_k^i; \Phi_k^i, \sigma^{r2}) \tag{5.44}$$

where $\hat{\Phi}_k^i$ is the bearing estimate of the target and $\sigma^{r2}$ is the bearing measurement variance. Therefore, the gradient $\Delta_{\mathbf{x}_k} \log\ p(\Phi_k^i|\mathbf{x}_k)$ can be given as

$$\Delta_{\mathbf{x}_k} \log\ p(\Phi_k^i|\mathbf{x}_k) = \frac{(\hat{\Phi}_k^i - \Phi_k^i)}{\sigma_k^{r2}} \Delta_{\mathbf{x}_k} \hat{\Phi}_k^i \tag{5.45}$$

Therefore

$$J_{\Phi_k^i} = \mathbb{E}\left\{ \left[ \frac{(\hat{\Phi}_k^i - \Phi_k^i)}{\sigma^{r2}} \Delta_{\mathbf{x}_k} \hat{\Phi}_k^i \right] \left[ \frac{(\hat{\Phi}_k^i - \Phi_k^i)}{\sigma^{r2}} \Delta_{\mathbf{x}_k} \hat{\Phi}_k^i \right]^T \right\}. \tag{5.46}$$

Now the information contribution by target orientation measurement $J_{\Theta_k}$ needs to be derived. The target orientation measurement is obtained via OOM based orientation measurement estimator. Since the distribution of the target orientation measurement uncertainty is unknown, it is assumed to be a Gaussian noise and given by

$$p(\Theta_k|\mathbf{x}_k) = \mathcal{N}(\hat{\Theta}_k; \Phi_k, \sigma_k^{\Theta^2}) \tag{5.47}$$

where $\hat{\Theta}_k$ is the target orientation estimate of the target and $\sigma_k^{\Theta^2}$ is the OOM based orientation measurement variance. Similar to (5.43), the information contribution by

orientation measurement can be written using gradient operator as

$$J_{\Theta_k} = \mathbb{E}\{[\Delta_{\mathbf{x}_k} \log\ p(\Theta_k|\mathbf{x}_k)][\Delta_{\mathbf{x}_k} \log\ p(\Theta_k|\mathbf{x}_k)]^T\} \tag{5.48}$$

and further simplified to

$$J_{\Theta_k} = \mathbb{E}\left\{\left[\frac{(\hat{\Theta}_k - \Theta_k)}{\sigma_k^{\Theta^2}}\Delta_{\mathbf{x}_k}\hat{\Theta}_k\right]\left[\frac{(\hat{\Theta}_k - \Theta_k)}{\sigma_k^{\Theta^2}}\Delta_{\mathbf{x}_k}\hat{\Theta}_k\right]^T\right\} \tag{5.49}$$

as in the case of bearing information contribution (5.46).

## 5.4 Simulation Results

In this section, the results of the simulation studies for the proposed target tracking algorithm, which utilizes the OOM-based estimated target orientation measurements that was developed in this work, are presented.

### 5.4.1 Target Motion and Measurement Models

The true state of the target at time $k$ is

$$\mathbf{x}_k) = [x_k \quad \dot{x}_k \quad y_k \quad \dot{y}_k]' \tag{5.50}$$

where $x_k$ and $y_k$ are the positions of the target, and $\dot{x}_k$ and $\dot{y}_k$ are the velocities of the target in the $\mathbf{X}$ and $\mathbf{Y}$ coordinates, respectively.

The target motion model in the Cartesian coordinate system is

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}) + \Gamma_k\ \nu_{k-1} \tag{5.51}$$

with white Gaussian process noise $\nu(t_{k-1})$ and has a covariance matrix $\Gamma_k$. The Interacting Multiple Model (IMM) estimator is used as the tracker that consists of two models, namely, a constant-velocity and coordinated-turn models [6]. The process noise variance for position is $0.01\text{m}^2/\text{s}^4$, and for turn rate it is $0.0001\text{rad}^2/\text{s}^2$. The Markov chain transition matrix for the IMM estimator is

$$[p_{ij}] = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix} \tag{5.52}$$

The tracking of targets in two dimensional space is considered and therefore the $f_k(.)$ for constant velocity and coordinated-turn models are given as

$$f_k = \begin{bmatrix} 1 & \Delta_k & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.53}$$

and

$$f_k = \begin{bmatrix} 1 & \frac{\sin(\omega\Delta_k)}{\omega} & 0 & \frac{-(1-\cos(\omega\Delta_k))}{\omega} & 0 \\ 0 & \cos(\omega\Delta_k) & 0 & -\sin(\omega\Delta_k) & 0 \\ 0 & \frac{(1-\cos(\omega\Delta_k))}{\omega} & 1 & \frac{(\sin(\omega\Delta_k))}{\omega} & 0 \\ 0 & \sin(\omega\Delta_k) & 0 & \cos(\omega\Delta_k) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.54}$$

respectively. Here $\Delta_k$ is the sampling time and $\omega$ is the turn rate. In the above, $\Gamma_k$

is defined as

$$\Gamma_k = \begin{bmatrix} \frac{\Delta_k^3}{3} & \frac{\Delta_k^2}{2} & 0 & 0 \\ \frac{\Delta_k^2}{2} & \Delta_k & 0 & 0 \\ 0 & 0 & \frac{\Delta_k^3}{3} & \frac{\Delta_k^2}{2} \\ 0 & 0 & \frac{\Delta_k^2}{2} & \Delta_k \end{bmatrix} l \tag{5.55}$$

where $l = 1\mathbf{x}10^{-4}\mathrm{m}^2\mathrm{s}^{-3}$ and the sampling time $k$ is taken as 10 seconds.

The measurements from three sensors are available at discrete time sampling intervals of $T = 10$ seconds. The target-generated bearing measurements corresponding to a target are given by (5.4) and the augmented measurement model is given by (5.8).

### 5.4.2 Results

The scenario shown in Figure 5.5 was chosen to show the robustness of the proposed algorithm with OOM based target orientation measurement. In this scenario, a single target is considered, where it follows a constant velocity motion model for the first 50 seconds and then a coordinated turn motion model for a period of 125 seconds. Again, it follows constant velocity motion before making a coordinated turn from 225 seconds to 425 seconds. Finally, it follows constant velocity motion for the period of 15 seconds followed by coordinated turn motion for the rest of the simulation period. The three sensors are located at $(20000\mathrm{m}, -50000\mathrm{m})$, $(90000\mathrm{m}, -10000\mathrm{m})$ and $(60000\mathrm{m}, 20000\mathrm{m})$. The bearing standard deviation of these sensors are assumed to be 0.05 rad. The turn rate for the coordinated turn is 0.0175rad/s. Initial target velocity is $120\mathrm{ms}^{-1}$ in positive $X$ direction. The target aspect state is decomposed into 10 degree intervals and the OOM is learned for a single target using simulated training sequences.

Target in the scenario is highly maneuvering with higher velocity. The simulation

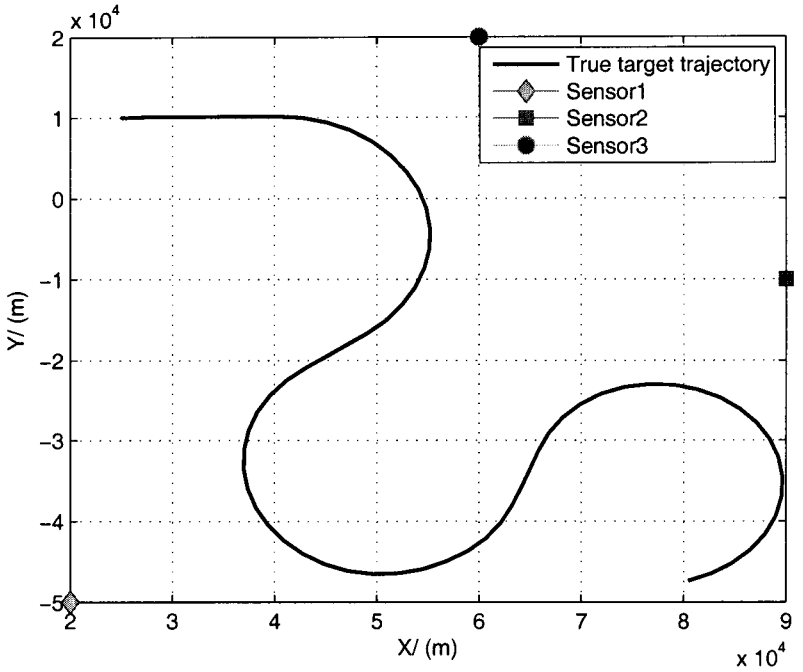



Figure 5.5: True target trajectory and sensor locations.

results were obtained with and without orientation measurements. All the results presented in this work are based on 100 Monte Carlo (MC) experiments. The target orientation measurement is obtained by processing the amplitude measurements from all the sensors together with the predicted target positions using the OOM. The orientation measurement and its variance are only available during the tracking process. Thus, the PCRLB calculation can be performed together with tracking as the orientation measurement is available only at tracking process.

Track initialization is done using the method proposed in Section 5.1.1 when the orientation measurement is incorporated into tracking. Eight percent of the initialized tracks failed when the orientation information was not used whereas the orientation information incorporated track initialization had no failures in track initialization. The initial target state RMSE in Figure 5.6 without orientation information is much

higher than that of orientation measurement incorporated tracking. This improvement is achieved by reducing the velocity uncertainties by incorporating target orientation information. Further, on average the PCRLB with orientation measurement is lower than that without the orientation information. It is noted that the PCRLB with the target orientation information is far below the PCRLB without orientation information when the target maneuvers. This is a clear evidence that use of target orientation information is beneficial in target tracking.

Figure 5.7 shows the estimated target trajectories for 100 runs obtained when using the OOM based orientation information and without orientation information. The orientation information incorporated trajectory following very closely with the true target trajectories, whereas the latter slightly deviate from the true target trajectory, especially when the target maneuvers. The true target model probabilities are given in Figure 5.8 and the estimated target models with and without orientation information is given in Figures 5.9 and 5.10, respectively. The target model switching is faster with orientation information when compared to the one without the orientation information. Further, the probability of being in each target model (constant velocity and coordinated turn) is consistent with the orientation information aided tracking as seen in Figure 5.9. The mode probability of being in coordinated turn model with orientation information is close to 1 whereas the estimation without orientation information fluctuates between 0.7 to 1. Figure 5.11 compares the estimated target orientation. The estimated target orientation when the OOM based orientation information is used shows superior performance of the proposed algorithm.

In the above simulations, three sensors are used and the average orientation measurement standard deviation from the OOM is 17.19 degree. However, it improved the target tracking performance. The algorithm is tested by using five sensors and the

orientation measurement standard deviation comes down to 14 degree and the target tracking performance is further improved. This can be further reduced by increasing the number of target-to-sensor aspect states at additional computational cost. However, when two sensors were used, a performance degradation in target tracking was observed. This could be due to the violation of conditional Gaussian assumption measurement model for OOM target orientation when fewer number of sensors are used. Therefore, further study is needed to address the model mismatch between the OOM and the state estimator. This problem may be overcome by incorporating a suitable bias model, which is capable of handling varying bias, into the measurement model.

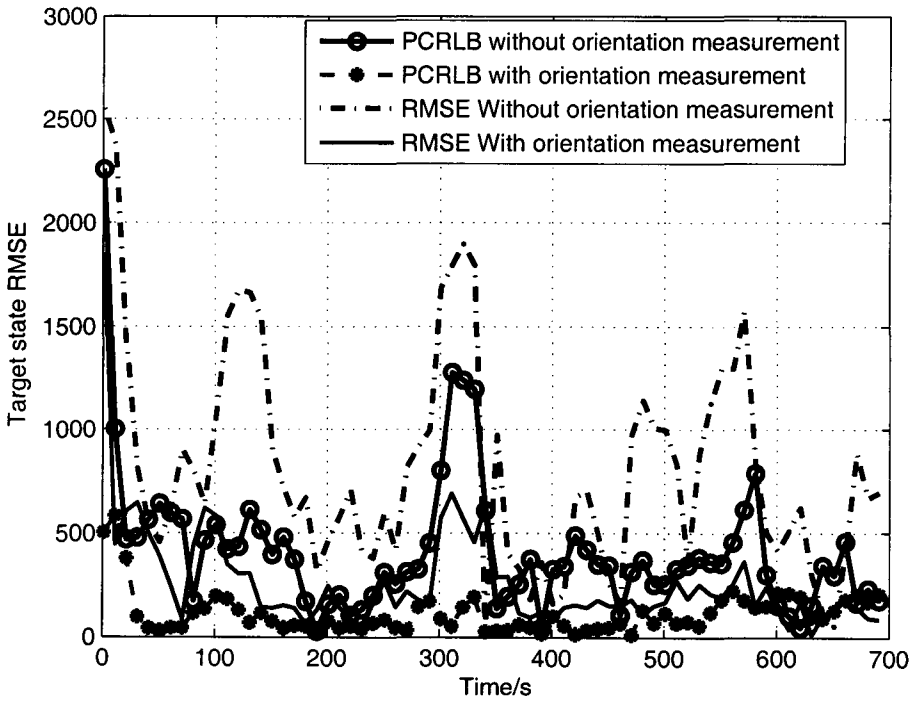



Figure 5.6: Target state RMSE and PCRLB

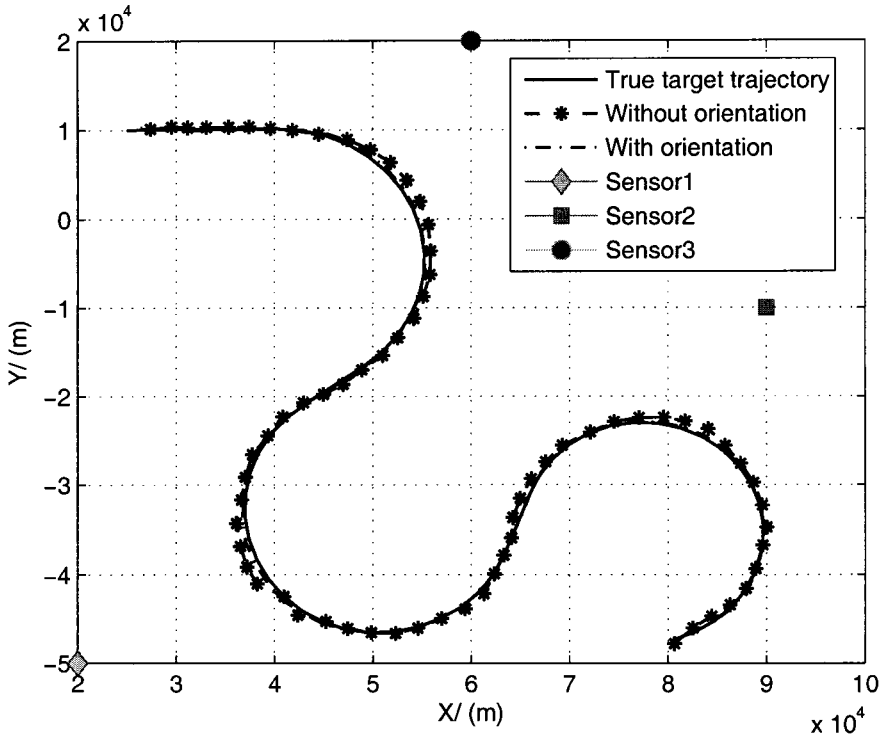



Figure 5.7: True target trajectory and the estimated target trajectories

## 5.5 Summary

This chapter discusses a target tracking system that provides improved estimates of target states using target orientation information in addition to standard kinematic measurements. The improved state estimation of highly maneuverable targets with noisy kinematic measurements, using a set of feature information from multiple sensors, is achieved. The Observable Operator Model (OOM) is used together with multiple sensor feature information to estimate the target orientation. It is shown that proposed OOM based target orientation information is helpful in accurately determining the target state. Further, the proposed track initialization technique

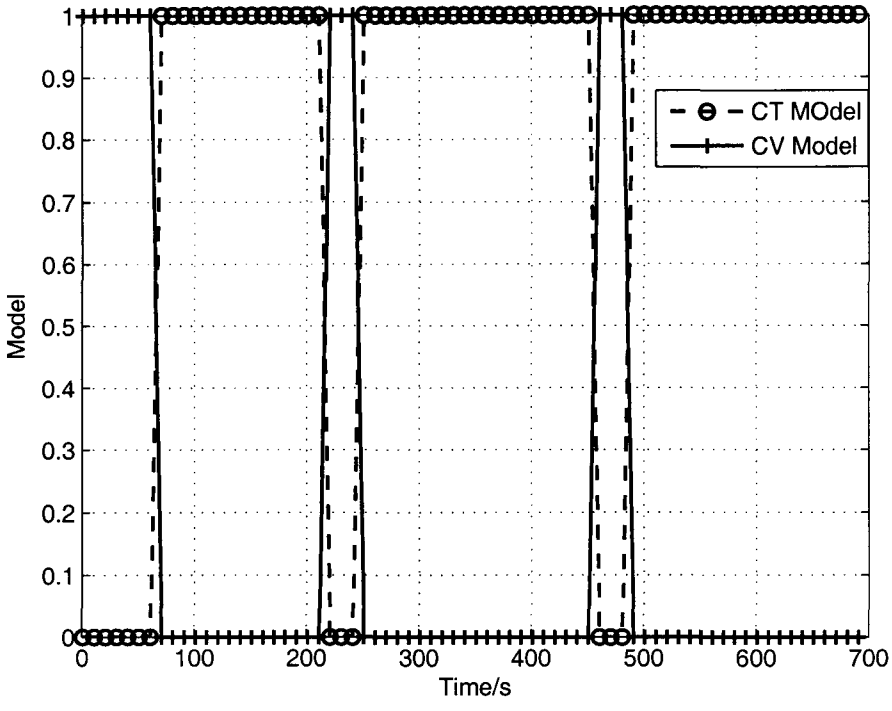



Figure 5.8: True target model switching

that incorporates the target orientation information also improves the tracking performance. The experiments described in the chapter confirm that the OOM based target orientation estimation can be used successfully in determining the orientation of targets to a reasonable degree of accuracy. Also it has been shown that incorporating estimated target orientation into tracking problem improves target tracking performance while reducing the computational complexity of OOM module by using the predicted target position from the tracker at every time step.

Simulation results show that the incorporation of target orientation can enhance the tracking performance in the presence of fast moving or maneuvering targets. In addition, the Posterior Cramer-Rao lower bound (PCRLB) that quantifies the

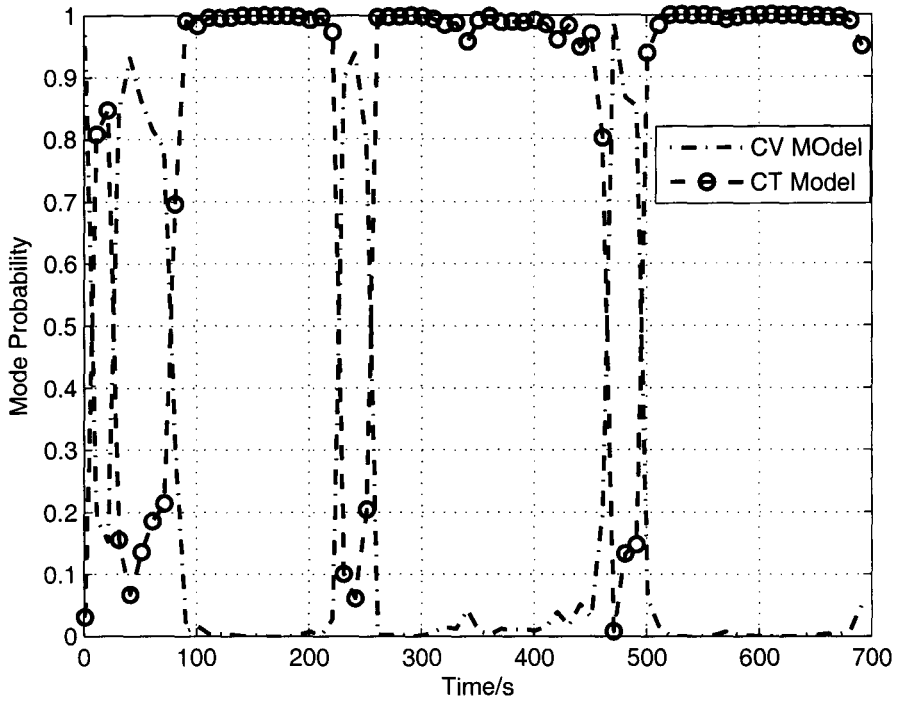



Figure 5.9: Target mode probability with orientation measurement

achievable performance is derived. It is shown that the proposed estimator meets the PCRLB. In this work, somewhat restrictive, but realistic scenario is considered. However, the proposed algorithm is best suited for tracking missiles and super sonic fighter aircrafts that make frequent maneuvers.

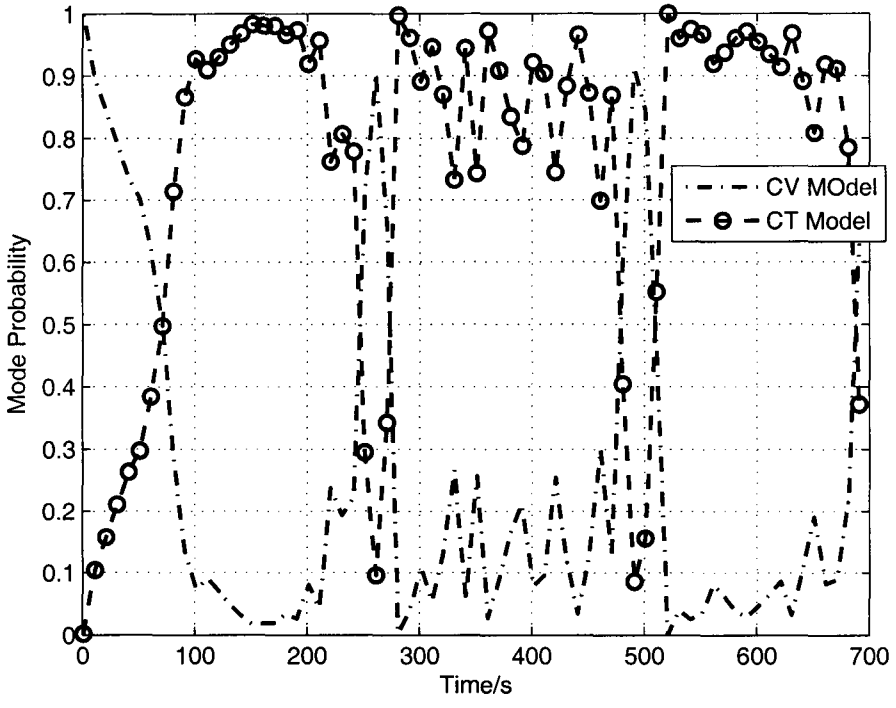



Figure 5.10: Target mode probability without orientation measurement

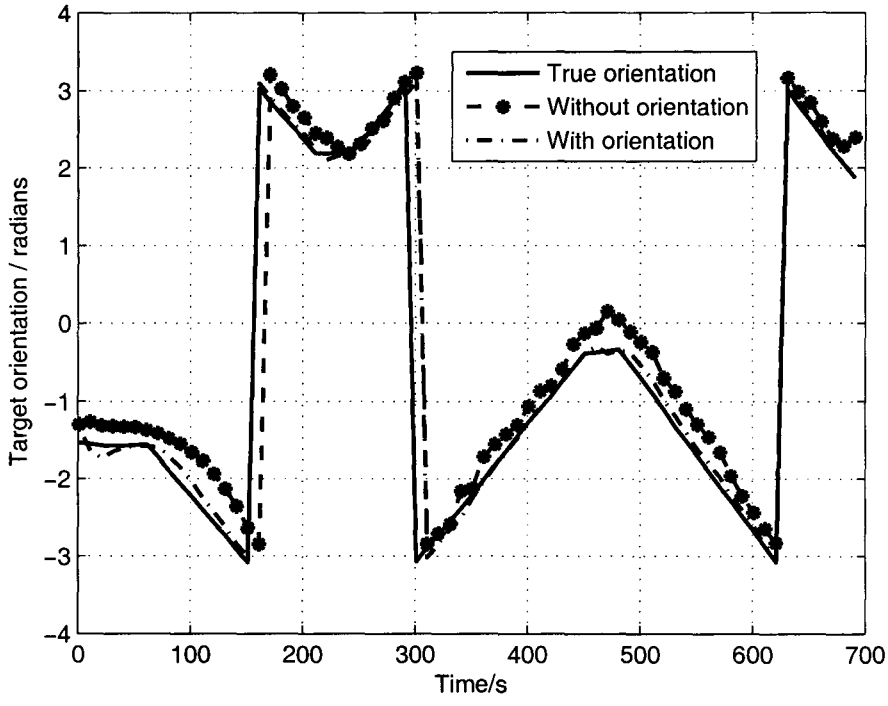



Figure 5.11: Target orientation estimate

# Chapter 6

# Joint Target Tracking and Classification for Multistatic Active Sonar Network

## 6.1 Bistatic Radar Configurations

In this section the fundamental building block of bistatic radar, emphasizing similarities and differences with the more common monostatic (Figure 6.1) counter part is illustrated. Figure 6.2 shows a typical bistatic geometry with clear separation of the transmitter and the receiver. Bistatic radar has two geometrical characteristics, which differentiate it from conventional monostatic systems. These include

- the transmitter receiver separation
- transmitter-target-receiver triangulation

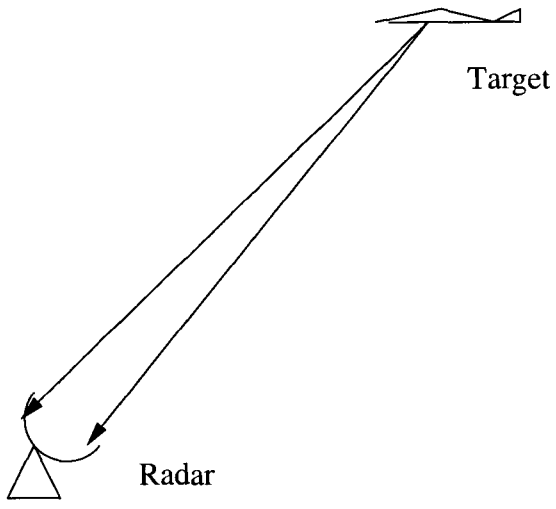



Figure 6.1: Monostatic radar configuration

Distance $d$ represents the bistatic baseline. $r_o$ is the transmitter to target separation and $r$ is the target to receiver separation. $\beta$ is referred to as the bistatic angle. The baseline distance is usually fixed and slowly varying. The system performance depends on target position relative to the bistatic baseline and hence triangulation. Three target position areas can be described with substantially different characteristics. There is the broadside area with a bistatic angle less than $180^0$, which provides the most common form of bistatic radar. There is also the baseline area with a bistaic angle equal to $180^0$. This situation corresponds to the forward scatter geometry and poses several implications. The final area is the extended baseline behind both the transmitter and the receiver.

The bistatic radar range equation is given by

$$\frac{P_r}{P_n} = \frac{P_t G_t G_r \lambda^2 L_{pt} L_{pr} \sigma_b}{(4\pi)^3 r_0^2 r^2 k T_0 B F} \tag{6.1}$$

where

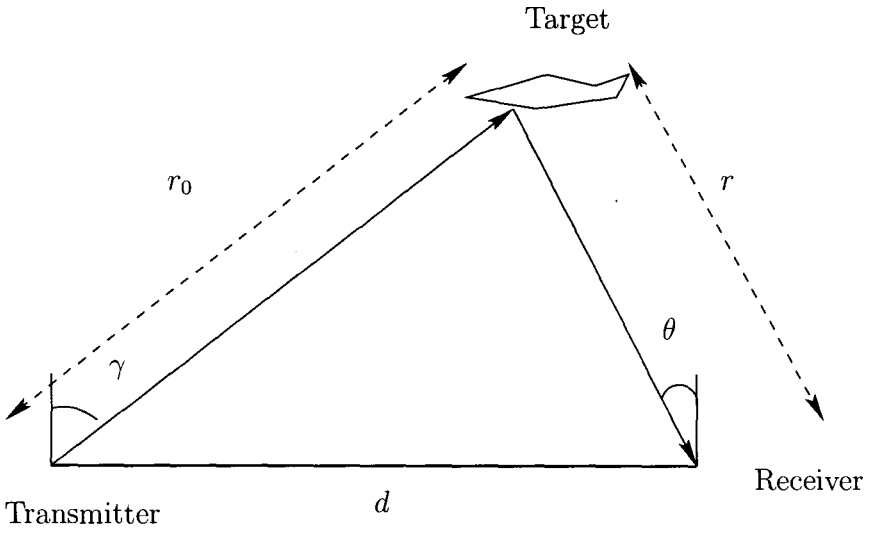



Figure 6.2: Bistatic radar configuration

- $P_t$ - transmitter power in Watts
- $G_t$ - the gain of the transmitter antenna
- $G_r$ - gain of the receiving antenna
- $\lambda$ - radar wavelength in meters
- $L_{pt}$ - loss from transmitter to target ($\leq 1$)
- $L_{pr}$ - loss from target to receiver ($\leq 1$)
- $r_0$ - distance between the transmitter and target in meter
- $r$ - distance between the receiver and target in meter
- $k$ - Boltzmann's constant
- $T_0$ - noise reference temperature in Kelvin
- $B$ - receiver bandwidth in Hertz

- $F$ - receiver noise figure
- $\sigma_b$ - bistatic Radar Cross Section in $m^2$

The main difference between bistatic and monostatic radar is the separation of transmitter and receiver ranger. These determine the bistatic geometry. From the bistatic geometry, it can be observed that targets of constant bistatic range are described by ellipses with the transmitter and receiver as two foci. In the monostatic radar case range are described by circles. In bistatic radar, these ellipses are same as the contours of zero Doppler. The contours of maximum Doppler shift form hyperbolae as they must also cross these ellipses orthogonally. In monostatic radar the orthogonal condition still holds but leads to a series of lines emanating radially out from the co-located transmitter receiver pair. In bistatic radar a moving target will not present zero Doppler to two receiving sites simultaneously. This can usefully be exploited in multistatic radar systems.

The bistatic radar cross section of the target is not necessarily the same as the monostatic one. For small bistatic angles of less than approximately $5^0$, the bistatic RCS of a complex target is equal to the monostatic RCS measured on the bisector of the bistatic angle at a frequency lower by a factor of $\cos(\beta/2)$. When operating in the broadside area, bistatic radar may be well suited to detecting stealthy targets. This is because a target is very unlikely to present a low bistatic cross section to two receiving sites simultaneously. This potential makes the detection of stealthy targets easier as the reflections from it in other directions will be detected by bistatic receivers.

When a target crosses the baseline of the bistatic radar, the RCS can be greatly enhanced. This is due to the forward scatter phenomenon. Here the RCS of a target

at the bistatic baseline is calculated from

$$\sigma_b = \frac{4\pi A^2}{\lambda^2} \tag{6.2}$$

where

$A$ is the geometric area of the target in $m^2$ and $\lambda$ is the radar wavelength in meters.

For a sphere of radius $a$ meters, the monostatic RCS is equal to the projected area of a sphere given by $\pi a^2$. Considering a sphere with monostatic RCS equal to $0.25m^2$ and at a wavelength equal to $0.1m$, the forward scatter RCS is

$$\sigma_b = \frac{4\pi(0.25)^2}{(0.1)^2} = 78.5m^2 \tag{6.3}$$

This corresponds to an enhancement of $25dB$. The forward scatter RCS will decrease as the bistatic angle decreases and ultimately reaches the monostatic RCS in the limit where the angle is equal to zero. Nevertheless, significant RCS enhancement is generally achieved at bistatic angle of $165^0$. An important factor is that the forward scatter RCS does not depend on material composition. As a result, bistatic radars operating in the forward scatter region may be able to detect stealthy targets and will give appreciable forward scatter RCS despite their designed low monostatic RCS. In addition, the angular width of scattering is a function of the wavelength and hence favors low frequencies. For real targets, the geometric area $A$ of the target is not constant and it depends on target-to-sensor aspect and the target orientation.

Another important difference between bistatic and monostatic radar is that a directional receive antenna must scan at a non-uniform rate to follow the position of the transmitted signal through space, a process known as pulse chasing. This can be challenging for designs based on mechanical scanning, hence an alternative is to

use one or more electronically agile beams as in phased array radar. Such phased array antennas can be expensive and in some applications will prohibit the use of the bistatic technique.

## 6.2 Aspect Dependent Bistatic Radar Cross Section

As illustrated in Chapter 1, the RCS of a typical target is very sensitive to aspect angles and the exact nature of RCS fluctuations is difficult to model using a close form expression. For these reasons, computer simulations of radar systems adopt various statistical models for RCS fluctuations. The most common models are the so-called Swerling models [13]. Both Swerling type 0 and type 5 refer to a model with no RCS fluctuation. Swerling types 1 and 2 model RCS fluctuations using an exponential distribution that assumes the target consists of many equal-amplitude scattering centers. Swerling types 3 and 4 also use an exponential distribution by assuming that the target have one dominant scattering center along with many smaller equal-amplitude scattering centers. In types 1 and 3, the RCS fluctuations are relatively slow so that all pulses of the same frequency in a single dwell experience the same RCS. In types 2 and 4, the RCS fluctuates more rapid so that each pulse in a single dwell experiences a different RCS. In this work, target with rapid RCS fluctuation is considered together with target orientation dependence SNR.

Denote $\mathrm{SNR}_n$ as the average Signal to Noise Ratio (SNR) at the $n^{\mathrm{th}}$ receiver corresponding to the return wave from the target. Assuming a Rayleigh fading signal model [24][32], the received signal at the correlated output is exponentially distributed with variance proportional to $(1+\mathrm{SNR}_n)$. The probability density function of output

$\rho$ is given by

$$p(\rho) = \frac{\rho}{\zeta} \exp\left(-\frac{\rho^2}{2(1+\zeta)}\right) \tag{6.4}$$

Here $\zeta$ is Rayleigh parameter proportional to $\text{SNR}_n$.

Now the dependence of $\text{SNR}_n$ on propagation distance and propagation angle need to be specified. The latter is also the Direction of Arrival (DOA) from the sensor field point of view. It is assumed that the source and the sensors are approximately at the same depth as the target. Ignoring small depth differences, the travel distance of the signal from the source to the $r^{\text{th}}$ receiver via reflection on the target is

$$r_r = \sqrt{(x(k) - x_t(k))^2 + (y(k) - y_t(k))^2} + \sqrt{(x(k) - x_r(k))^2 + (y(k) - y_r(k))^2}. \tag{6.5}$$

Denote the propagation angle as $\alpha$ and the angle from receiver to the target as $\varphi_r$. Therefore

$$\varphi_r = \text{atan}\left(-\frac{x_r(k) - x(k)}{y_r(k) - y(k)}\right) \tag{6.6}$$

There is no ambiguity in determining $\varphi_r$. The $\text{SNR}_n$ can be written as

$$\text{SNR}_n = c_0 f_1(r_r) f_2(\varphi_r, \alpha) \tag{6.7}$$

where $c_0$ is a constant, $f_1(.)$ describes the dependence on the propagation distance and $f_2(.)$ specifies the dependence on the propagation angle. In this work, it is assumed that the average signal strength is inversely proportional to the propagation distance and thus

$$f_1(r_r) = r_r^{-1} \tag{6.8}$$

The angle dependency can be modeled by a Butterworth approximation [74] as

$$f_2(\varphi_r, \alpha) = \frac{1}{1 + \left(\frac{\varphi_r - \alpha}{W}\right)^{2K}} \tag{6.9}$$

where $2W$ is the 3-dB bandwidth and $K$ is the filter order.

Now the propagation angle should be related to the source-target geometry and the target orientation. In this work, a specular reflection model [74], where target surface acts as a reflecting mirror, is used. If the target orientation is defined as $\Theta$, then

$$\alpha = 2\Theta - \varphi_r \tag{6.10}$$

This way, all the propagation angles are decided by the target orientation together with the geometry of the transmitter and target.

## 6.3 Multistatic Sonar Network

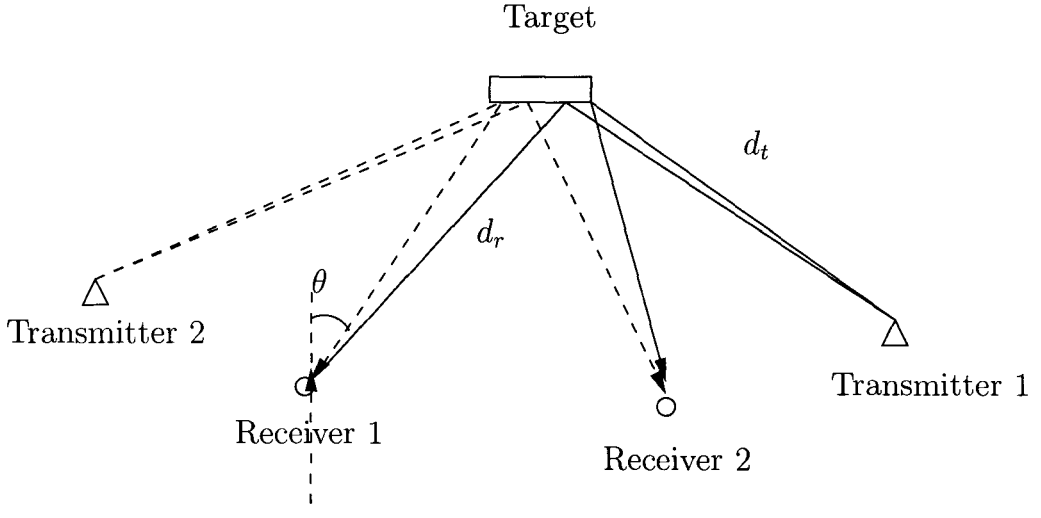



Figure 6.3: Multistatic sensor network

In a multistatic sonar network, there will be multiple transmitters and multiple receivers. In other words, multistatic sonar network consists of multiple bistatic pairs. Therefore, the theoretical derivations for multistatic configuration is same as bistatic radar configuration. However, the multistatic systems should be able to handle multiple bistatic together in a single framework. According to the transmitter-receiver pair, each pair will act as a monostatic or bistatic configuration. The main advantage of the multistatic sensor network over the monostatic sensor is the better target detection and probability due to geometric diversity. A sample scenario with two transmitters and two receivers in shown in Figure 6.3. In multistatic data, measurements are bistatic range ($r_b$) and bearing $\Phi$, and the $j^{\text{th}}$ measurement at sensor $s$ at time $k$ is given by

$$z_s(j,k) = \begin{bmatrix} r_b(j,k) \\ \Phi(j,k) \end{bmatrix} \tag{6.11}$$

$$= \begin{bmatrix} d_r(k) + d_t(k) \\ \text{atan}\left(\frac{x(k)-x_r(k)}{y(k)-y_r(k)}\right) \end{bmatrix} + \omega_s(j,k) \tag{6.12}$$

where

$$d_r(k) = \sqrt{(x(k)-x_r(k))^2 + (y(k)-y_r(k))^2} \tag{6.13}$$

$$d_t(k) = \sqrt{(x(k)-x_t(k))^2 + (y(k)-y_t(k))^2} \tag{6.14}$$

and $[x, y]$, $[x_r, y_r]$ and $[x_t, y_t]$ are the $x$ and $y$ coordinates of the target, receiver and transmitter, respectively. atan(.) is the four quadrant arctangent and $\omega_s(j,k)$ is a zero mean Gaussian random variable with covariance $Q_s(j,k)$. If a measurement is a false alarm, then it is assumed to be uniformly distributed across the surveillance

region. Note that monostatic measurement can also be considered as a special case of bistatic with transmitter and receiver at the same location. Even if the time difference of arrival are given instead of bistatic range, they can be converted to the bistatic range by multiplying by speed of the signal in the given medium and then adding the distance between the receiver and transmitter.

## 6.4 Multiple Target Tracking and Classification

When performing joint target tracking and classification, a good estimate of the state of the target (position, velocity and target class) can be expected. A model-based target tracking method is considered, where the target motion and observations can be represented by state-space models. The state of the system is the concatenation of the states of the targets. A class-dependent motion models are considered where each target is considered to be generating aspect dependent signal. Those motion models are essential to any model-based tracking algorithm and, thus, need to be well fit to the tracked targets.

Let $p_k$ denotes the number of targets at time $k$ within the field of the considered multistatic sensor network. Let $T_k$ be the set of active targets at time $k$. Thus, $T_k = \{1\}$ if $p_k = 1$ and $T_k = \{1, 2\}$ if $p_k = 2$. The first target is always the tracked target and the second a generic interfering one. If $n_k = n_{k-1} = 2$, then it is assumed that the second target remains the same. Denote $X_k = \{x_{t,i}, \xi_i\}_{i \in T_k}$ be the state of the targets at time $k$, where $\{x_{t,i}, \xi_i\}$ stands for the state (position, velocity and class) of the $i^{\text{th}}$ target. The system at time $k$ is thus, characterized by the vector $(X_k, n_k)$ where $n_k$ is introduced to emphasize the dependence of $X_k$ on the number of targets. Conditioned on the number of targets at time $k$ and $k+1$, the system dynamic model

is described by $p(X_k|X_{k-1}, n_k, n_{k-1})$.

It is assumed that each target is moving independently from the other according to a Markov transition dynamic. This dynamic depends on the class $\xi_i$ of the $i$ target. The dynamic of the system (conditioned on the number of targets at time $k$ and $k+1$) can, thus, be decomposed as

$$x_{k,i} = F_{\xi_i,i}(x_{k-1,i}, u_{k,i}) \; \forall_i \in T_k \tag{6.15}$$

$$\xi_{k,i} = \xi_{k-1,i} \; \forall_i \in T_k \tag{6.16}$$

The noise term $u_{t,i}$ are assumed to be white and pairwise independent.

The evolution of the number of target $n_k$ is independent of the previous state of the targets $X_{k-1}$. The state transition dynamic is, thus, given by (6.15), (6.16), and

$$\pi_{n_k,n_{k-1}} = p(n_k|n_{k-1}). \tag{6.17}$$

The probability of birth is denoted by $p_b$ (e.g., switching from $T_k = \{1\}$ to $T_k = \{1, 2\}$) and the probability of death is denoted by $p_v$ (e.g., switching from $T_k = \{1, 2\}$ to $T_k = \{1\}$). With these assumptions, the problem of ordering ambiguity is eliminated [3][76].

Sensor nodes are usually prone to errors and the measurements available can either arise from the targets of interest when they are detected or be spurious clutter noise (e.g., returns from nearby objects or electromagnetic interferences). One of the major problems in such a system arises from the generally unknown association between the available measurements and the targets of interest. Traditionally, data association is handled by methods such as the nearest neighbor or the Joint Probabilistic Data Association (JPDA) [5] algorithm. However, when dealing with nonlinear models and

unknown number of targets, none of these methods can be applied directly. In this work, statistical data association scheme is used.

Let $m_k$ be the number of bistatic pair in a multistatic sensor network. One sensor may receive more than one measurement from a target due to multiple transmitters [72]. Therefore, here, one single bistatic pair is considered as a single sensor. Let $\mathbf{z}_k^1$ be the $j^{\text{th}}$ bistatic pair of measurements, which is given by

$$\mathbf{z}_k^j = \{z_k^j, \psi_k^j\} \tag{6.18}$$

where $z_k^j$ and $\psi_k^j$ are the kinematic and RCS measurements, respectively. Therefore, the total measurement received at time $k$ is

$$\mathbf{Z}_k = \left(\mathbf{z}_k^1, \ldots, \mathbf{z}_k^{m_k}\right) \tag{6.19}$$

At most, one measurement can arise from each target to a bistatic pair of sensor and several measurement can arise from the clutter. The data association vector is denoted by $e_k$, which is, a vector of length $m_k$ whose components take values in $T_k \bigcup \{0\}$. Note that $e_k(m) = i$ means that the $m^{\text{th}}$ measurement has been generated by the $i^{\text{th}}$ target, whereas $e_k(m) = 0$ means that it is a spurious one. Conditioned upon the data association and the state of the target, the measurements are assumed to be independent. The general model for the measurement is given by

$$\mathbf{z}_k^m = H_k(x_{k,e_k(m)}, v_{k,m}),\ m \in \{m' | e_k(m') \neq 0\} \tag{6.20}$$

$$\mathbf{z}_k^m \sim p_c(\mathbf{z}),\ m \in \{m' | e_k(m') = 0\} \tag{6.21}$$

i.e., a measurement is given by the measurement function $H_k$ if it arises from a target

and is specified by some probability distribution $p_c$ if it arises from the clutter. The noise terms $v_{k,m}$ are assumed to be white Gaussian and pairwise independent.

The false detections are spurious and measurement is assumed to be uniformly distributed in the measurement area whose volume is denoted by $V$. The number of false detection, $m_k^0$, is typically generated by a Poisson distribution with parameter $\lambda V$, where $\lambda$ is the number of clutter measurements per unit volume and per time step. Hence

$$P(m_k^o = l) = e^{-\lambda V} \frac{(\lambda V)^l}{l!} \tag{6.22}$$

In order to deal with an unknown or varying number of targets $n_k$, several alternatives are available. A classical approach is to estimate $n_k$ separately from the rest of the state-space by using a hypothesis test, and then to treat the estimated $n_k$ as the true number of targets for the estimation of the other variables [35]. Another possibility is to compare several tracking hypotheses with different number of targets. In [3] and [23], it is proposed to cast the multiple target tracking problem into a Jump Markov System (JMS) filtering, where the number of targets and, thus, the dimensionality of the state follows a Markov chain. In this work, JMS is used to model the varying number of targets.

## 6.5 Sequential Monte Carlo Implementation

The target state is $(X_k, n_k)$ with $X_k = \{x_{k,i}, \xi_i\}_{i \in T_k}$. The posterior distribution can be expressed by Bayes' rule as

$$p(X_{0:k}, n_{0:k} | Z_{1:k}) = \frac{p(Z_{1:k} | X_{0:k}, n_{0:k}) p(X_{0:k}, n_{0:k})}{p(Z_{1:k})} \tag{6.23}$$

Therefore, the following recursive formula can be obtained:

$$p(X_{0:k}, n_{0:k}|Z_{1:k}) = p(X_{0:k-1}, n_{0:k-1}|Z_{1:k-1}) \frac{p(Z_k|X_k, n_k)p(X_k, n_k|X_{k-1}, n_{k-1})}{p(Z_k|Z_{1:k-1})} \quad (6.24)$$

This motivates to adopt a recursive importance sampling strategy by choosing a proposal density, which can be factorized as

$$q(X_{0:k}, n_{0:k}|Z_{1:k}) = q(X_{0:k-1}, n_{0:k-1}|Z_{1:k-1})q(X_k, n_k|X_{0:k-1}, n_{0:k-1}, Z_{0:k}) \quad (6.25)$$

It is then possible to sequentially draw from $q(X_{0:k}, n_{0:k}|Z_{1:k})$ by keeping the past particle sets $\{X_{0:k-1}^{(j)}, n_{0:k-1}^{(j)} w_{k-1}^{(j)}\}$ and then drawing $X_k^{(j)}$ from $q(X_k, n_k|X_{0:k-1}^{(j)}, n_{0:k-1}^{(j)}, Z_{1:k})$. Therefore, the weight update also becomes recursive manner as

$$w_k^{(j)} \propto w_{k-1}^{(j)} \frac{p(Z_k|X_k^{(j)}, n_k^{(j)})p(X_k^{(j)}, n_k^{(j)}|X_{k-1}^{(j)}, n_{k-1}^{(j)})}{q(X_k^{(j)}, n_k^{(j)}|X_{0:k-1}, n_{0:k-1}Z_{1:k})} \quad (6.26)$$

The goal is to sequentially estimate $p(X_k, n_k|Z_{1:k})$. The optimal choice of importance sampling density [23] is

$$q(X_k, n_k|X_{0:k-1}^{(j)}, n_{0:k-1}^{(j)}, Z_{1:k}) = p(X_k, n_k|X_{k-1}^{(j)}, n_{k-1}^{(j)}, Z_k) \quad (6.27)$$

Clearly, it is impossible to sample directly from this distribution. Further, weight update also requires the evaluation of $p(Z_k|X_{k-1}^{(j)})$, which does not admit a closed form expression. Therefore, the reasonable choice is to use prior distribution as sampling density.

### 6.5.1 MMSE State Estimate and MAP Classification

As discussed earlier, the Bayesian recursion for particle filter is initialized by drawing $N$ independent samples from the initial prior $p(x_0)$. Since target label $\xi$ is one of the state component, $p(x_0)$ will incorporate the a priori probability of class $\xi$ entering the sensor's surveillance volume. In the case of a uniform marginal density over the target classes, the initial set of samples would contain roughly $N/L$ number of particles from each target class, where $L$ is the number of possible target classes. However, as the recursion proceeds, the resampling performed during measurement updates can change this initial distribution drastically. More specifically, targets whose RCS value lead to poor matches with the observed data can end up with negligible likelihoods and little chance of being selected during resampling. As a practical matter, a minimum number of particles from each target class has to be selected at each resampling stage. Neglecting to do so sometimes could eliminate all particles corresponding to the correct target class by a single noisy RCS measurement.

Immediately prior to prediction, the approximate Maximum A Posteriori (MAP) estimate of the target class is simply the class that has the most particles. This corresponds to integrating out the target class marginal and selecting the mode. However, this elimination is more accurately performed before resampling, since resampling introduces extra random variation within the particle set. This, after calculating the normalized likelihoods, the MAP estimate of target class is

$$\hat{\xi}_k = \arg \max_{\xi \in \{1,2,\ldots,L\}} \{\varsigma_k(\xi)\} \tag{6.28}$$

where the particle counts are defined as

$$\varsigma_k(\xi) = N \sum_{i=1}^{N} b_i \mathbf{1}(\mathbf{x}_{k|k-1}^{(i)} \in \xi) \tag{6.29}$$

and the indicator function $\mathbf{1}(.)$ is used to determine a particle that belongs to target class $\xi$. The MMSE estimate of the current position of the MAP target class can be found by

$$\hat{\mathbf{x}}_k^{\mathbf{c}} = \frac{N}{\varsigma_k(\hat{\xi})} \sum_{i=1}^{N} b_i \mathbf{x}_k^{\mathbf{c}(i)} \mathbf{1}(\mathbf{x}_{k|k-1}^{(i)} \in \xi) \tag{6.30}$$

where $\mathbf{x}_k^{\mathbf{c}(i)}$ is the position vector extracted from the state of the $i^{\text{th}}$ particle, $\mathbf{x}_{k|k-1}^{(i)}$. According to the law of large numbers, this is an approximation to the expected target position conditioned on the observation and the $\hat{\xi}_k$ is the correct class.

## 6.6 Simulation Results

In this section, the results of the simulation studies for the proposed joint tracking and classification algorithm for multistatic active sensor network using aspect dependent RCS measurements that was developed in this work are presented.

Figure 6.4, which is the same scenario used in Chapter 4, shows the true target trajectories of the five targets and three sensor locations. Each sensor has a transmitter and a receiver, where transmitters operates on distinguished frequencies. Therefore, there is no ambiguity in different bistatic measurement pairs. The aspect-dependent bistatic RCS from targets are used to classify the targets. SMC approach is used due to high nonlinearity in bistatic RCS measurements. The results are based on 50 Monte Carlo runs. It is assumed that the false alarm rate $Pfa$ is 0.01. Each target class is described by a constant $c_0$, filter order $K$ and bandwidth $W$. It is assumed

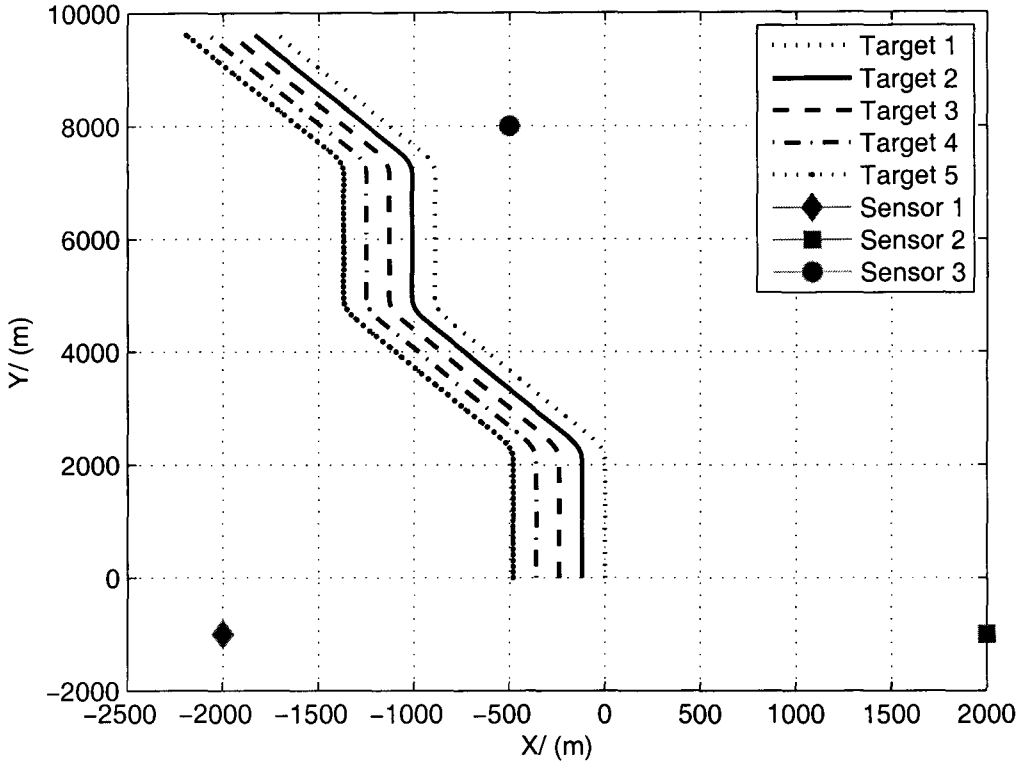



Figure 6.4: Scenario 1: True target trajectories and sensor locations.

that there are four possible target classes available. In the simulations, the parameters $K = 4$ and $W = 5\pi/180$ are fixed for all four target classes. However, different target class parameters can also be used. These target class parameters can also be estimated by formulating a maximum likelihood problem when the training measurements are available at different target orientations. Each target class is distinguished by the coefficient $c_0$. The coefficient $c_0$ for target class 1 to class 4 are 500, 900, 1100 and 1500, respectively. The coefficient for class 2 and class 3 are selected closer (900 and 1100) to observe the performance of the classification algorithm.

The average position and velocity Root Mean Squared Errors (RMSE) for each

target for scenario 2 are shown in Table 6.1 and Table 6.2, respectively. The RMSE results show that incorporating classification information into data association improves target tracking accuracies. Figures 6.5 – 6.9 compare the target classification performances. The joint tracking and classification algorithm correctly determined the target type except for target 2. The Figure 6.6 shows that target class as type 3 instead of type 2 in some time steps. Class 2 and class 3 has some similarities and this misclassification can be avoided by using more number of sensors. .

Table 6.1: Average Position RMSE Comparison/ $m$

| Target Label | With classification | Without classification |
|---|---|---|
| Target 1 | 4.76 | 9.43 |
| Target 2 | 5.06 | 10.73 |
| Target 3 | 4.99 | 9.84 |
| Target 4 | 5.27 | 11.35 |
| Target 5 | 4.97 | 9.57 |

Table 6.2: Average Velocity RMSE Comparison/ $ms^{-1}$

| Target Label | With classification | Without classification |
|---|---|---|
| Target 1 | 9.04 | 13.65 |
| Target 2 | 8.87 | 13.36 |
| Target 3 | 9.52 | 14.65 |
| Target 4 | 9.24 | 14.25 |
| Target 5 | 10.06 | 14.85 |

## 6.7 Summary

Active sonar tracking using measurements from multistatic sensors has benefit in terms of robustness and due to the increased probability of detection. The aspect-dependent bistatic Radar Cross Section (RCS) measurements and kinematic data are

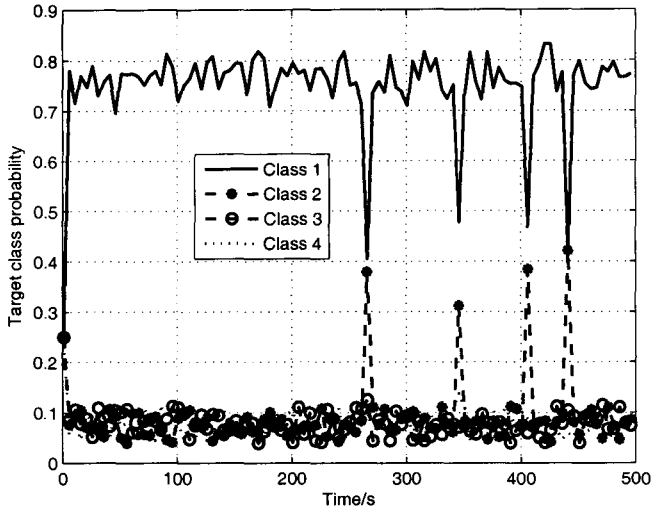



Figure 6.5: Target class estimate for target 1.

used. The scattered signals measured from different orientations of a target may vary due to aspect-dependant RCS. A complex target may contain several dozen significant scattering centers and dozens of other less significant scatterers. Because of this multiplicity of scatterers, the net RCS pattern exhibits high variation with aspect angle. Thus, radar cross sections from multiple aspects of a target, which are obtained via multiple bistatic pairs, will help in accurately determining the target class. By modeling the deterministic relationship that exits between RCS and target aspect, both the target class information and the target orientation can be estimated. Kinematic data are also very helpful in determining the target class as it describes the target motion pattern and its orientation. A Sequential Monte Carlo (SMC) approach was used to implement joint tracking and classification algorithm for multiaspect active sonar network system.

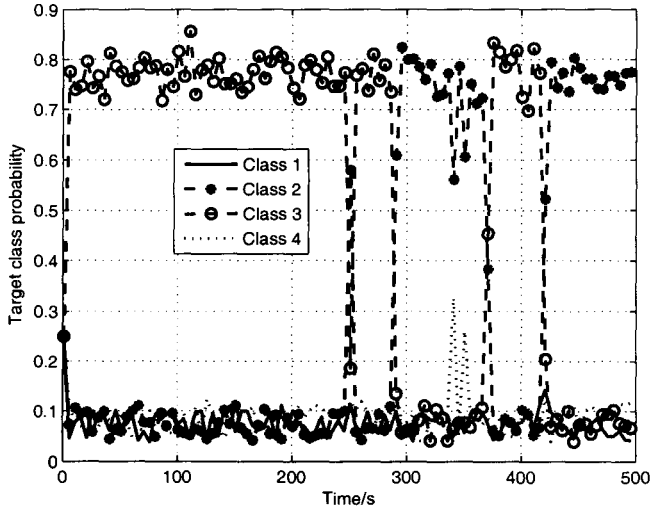



Figure 6.6: Target class estimate for target 2.

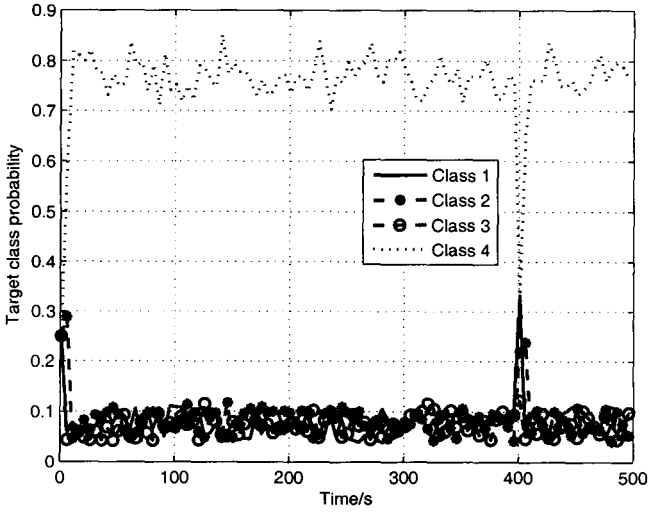



Figure 6.7: Target class estimate for target 3.

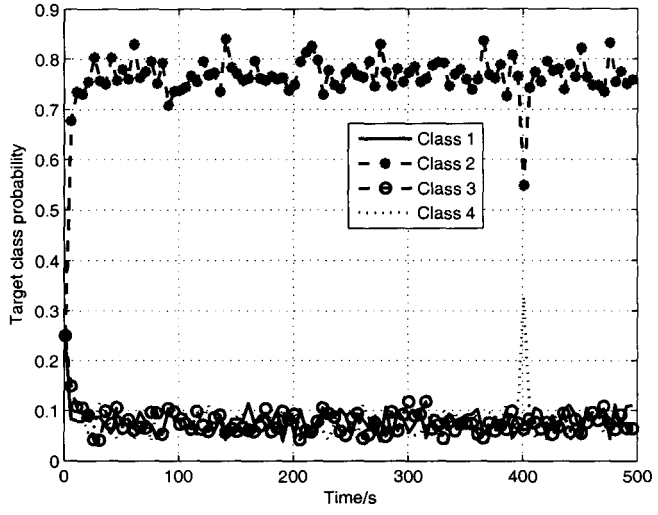



Figure 6.8: Target class estimate for target 4.

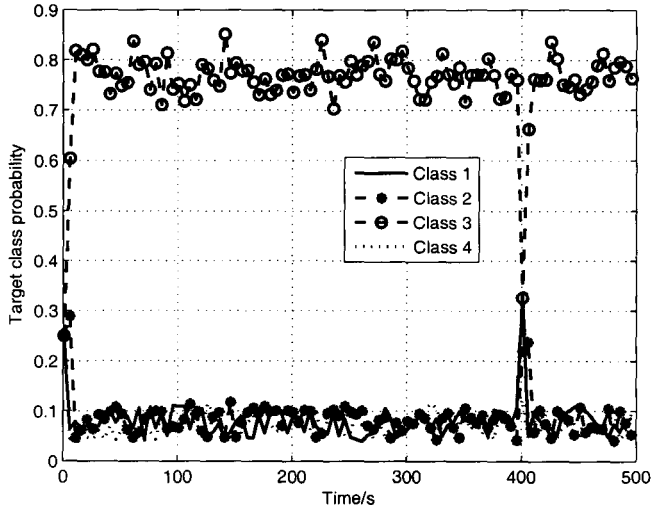



Figure 6.9: Target class estimate for target 5.

# Chapter 7

# Conclusions and Future Work

## 7.1 Conclusions

In this thesis, new joint target tracking and classification algorithms were presented. This work was motivated by the need for better classifier that incorporates multi-aspect sensor measurements. The Observable Operator Model (OOM) is used to capture unknown feature distribution of each target. Target classification and multi-sensor fusion using multiframe data association are integrated together, which ensures better classification and tracking results. When the targets are close to one another, separately extracting the measurements corresponding each target is difficult. That is, there is an uncertainty in the origin of the feature measurements. This problem is overcome by integrating the data association and the training based target classification together. A significant improvement in performance is achieved by using two-way exchange of useful information between the tracker and the classifier. The simulation studies explicitly show the superior performance of the proposed algorithm. Therefore, the fundamental relationship between the target classification and

tracking can be used to improve data association resulting in not only better target tracking performance but also in improved classification.

Further, this thesis discusses a target tracking system that provides improved estimates of target states using target orientation information in addition to standard kinematic measurements. The improved state estimation of highly maneuverable targets with noisy kinematic measurements, using a set of feature information from multiple sensors, is achieved. The OOM is used together with multiple sensor feature information to estimate the target orientation. It is shown that proposed OOM based target orientation information is helpful in accurately determining the target state. Also, a track initialization technique that incorporates the target orientation information is proposed, which improves the tracking performance. The experiments described in the thesis confirm that the OOM based target orientation estimation can be used successfully in determining the orientation of targets to a reasonable degree of accuracy. Also, it has been shown that incorporating estimated target orientation into tracking problem improves target tracking performance while reducing the computational complexity of OOM module by using the predicted target position from the tracker at every time step. In addition, the Posterior Cramer-Rao Lower Bound (PCRLB) that quantifies the achievable performance is derived.

Finally, a joint target tracking and classification algorithm using aspect dependent radar cross section for multistatic active sonar network is presented. The aspect dependent RCS is modeled using Rayleigh distribution. The aspect dependence is incorporated into the average signal-to-noise ratio of target using Butterworth approximation. A Sequential Monte Carlo (SMC) approach is used to perform joint tracking and classification as the aspect dependent RCS shows nonlinearity due to high signal fluctuations.

## 7.2 Future Work

The OOM based classifier does not provide good classification results when a new target when the target in the scene is not in the trained OOM set. An online learning of the OOM has to be developed so that the new OOM class can be obtained using the feature information obtained from the new type of target.

The target orientation estimates from OOM is approximated using a Gaussian distribution. This approximation provides improved state estimates when the number of sensors is greater than three. However, the state estimation performance degrades when very few number of sensors are used. Therefore, possible model mismatch between the OOM based orientation measurement estimator and the conditional Gaussian approximation has to be further addressed. Track initialization for low observable measurements can also be further addressed by incorporating the orientation and doppler information.

In this thesis, the coupling between target orientation and its course is taken into account with the assumption that the velocity of the target is nearly along the major axis of the target. This assumption may not be valid when the environmental or external forces influence the target motion. This problem can be overcome by incorporating a proper target drift models with the system model. Incorporation of target drift models into the system model has to be further investigated.

# Bibliography

[1] Alexander, M., Rick, S. and Leonard, C., "MIMO radar with widely separated antennas", *IEEE Signal Processing Magazine*, Vol. 41, No. 3, pp. 868–878, January 2008.

[2] Anderson, O. and Moore, J. B., *Optimal filtering.* Englewood Cliffs, NJ, Prentice-Hall, 1979.

[3] Andrieu, M., Davy, M. and Doucet, A., "Efficient particle filtering for jump Markov systems: application to time-varying autoregressions", *IEEE Transactions on Signal Processing*, Vol. 51, No. 7, pp. 1762–1770, July 2003.

[4] Arulampalam, M. S., Maskell, S., Gorden, N. and Clapp, T., "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking", *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, pp. 173–188, February 2002.

[5] Bar-Shalom,Y. and Fortmann, T. E., *Tracking and data association.* New York, NY: Academic, 1998.

[6] Bar-Shalom,Y., Li, X., and Kirubarajan, T., *Estimation and tracking: Principles techniques and software.* New York, NY: John Wiley, 2001.

[7] Bar-Shalom,Y., and Li, X., *Multitarget-multisensor tracking: principles techniques.* Storrs, CT: YBS Publishing, 1995.

[8] Bar-Shalom, Y., Kirubarajan, T., Gokberk, C., "Tracking with classification-aided multiframe data association", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 41, No. 3, pp. 868–878, July 2005.

[9] Bergnam, N., *Sequential Monte Carlo methods in practice*, Spring-Verlag, New York, USA, 2001.

[10] Berger, C.R., Daun, M. and Koch, W., "Track initialization from incomplete measurements", *10th International Conference on Information Fusion, 2007*, pp. 1–8, Quebec, Canada, July 2007.

[11] Bharadwaj, P., Runkle, P., Carin, L., Berrie, J. A. and Hughes, J. A., "Multiaspect classification of airborne target via physics-based HMMs and matching pursuits", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 37, No. 2, pp. 595–606, April 2001.

[12] Blackman, S. and Popoli, R., *Design and analysis of modern tracking systems*, Artech House, 1999.

[13] Blair, W. D., and Brandt-Pearce, M., "Estimation and discrimination for Swerling targets", *Twenty-Eighth Southeastern Symposium on System Theory*, pp. 280–284, Barton Rouge, LA, USA, April 1996.

[14] Boers, Y. and Driessen, H., "Integrated tracking and classification: An application of hybrid state estimation", *Proceedings of SPIE Signal and Data Processing of Small Targets*, Vol. 4473, pp. 198-209, 2001, San Diego, CA, July 2005.

[15] Bruno, M. G. S., "Bayesian methods for multiaspect target tracking in image sequences", *IEEE Transactions on Signal Processing*, Vol. 52, No. 7, pp. 1848–1861, July 2004.

[16] Challa, S. and Pulford, G. W., "Joint target tracking and classification using radar and ESM sensors", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 37, No. 3, pp. 1039–1055, July 2001.

[17] Challa, S. and Pulford, G. W., "Comments on joint target tracking and classification using radar and ESM sensors (Digest)", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 40, No. 2, pp. 765–768, April 2004.

[18] Chen, H., Li, X.R. and Bar-Shalom, Y., "On joint track initiation and parameter estimation under measurement origin uncertainty", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 40, No. 2, pp. 675–694, April 2004.

[19] Cox, H., *Fundamentals of bistatic active sonar in underwater acoustic data processing*, Kluwer academic publishers, 1989.

[20] Deb. S, Yeddanapudi. M, Pattipati, R. and Bar-Shalom, Y., "A generalized S-D assignment algorithm for multisensor-multitarget state estimation", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 37, No. 2, pp. 523–538, April 1997.

[21] Dong, Y. and Carin, L., "Quantization of multiaspect scattering data: Target classification and pose estimation", *IEEE Transactions on Signal Processing*, Vol. 51, No. 12, pp. 3105–3114, December 2003.

[22] Doucet, A., Freitas, N. and Gordon, N.,"Sequential monte carlo methods in practice.", Spring-Verlag, New York, USA, 2001.

[23] Doucet, A., Andrieu, C. and Davy, M., "Particle filtering for multitarget tracking and sensor management", *In Proceedings of the Information Fusion*, Vol. 1, pp. 474-481, Queensland, Australia, July 2003.

[24] Ehrman, L.M., Burton, C. and Blair, W.D., "Using target RCS to aid measurement-to-track association in multitarget tracking", *Proceedings of the Thirty-Eighth Southeastern Symposium on System Theory*, vol. 7, pp. 89-93, Cookeville, TN, March 2006.

[25] Erdinc, O., Willet, P. and Coraluppi, S., "Multistatic sensor placement: A tracking approach", *Journals of Advances in Information Fusion*, Vol. 2, No. 1, pp. 22–34, June 2007.

[26] Glaser, J. I., "Some results in the bistatic radar cross section (RCS) of complex targets", *Proc. IEEE*, Vol. 77, No. 5, pp. 639–648, May 1998.

[27] Gordon, N., Maskell, S. and Kirubarajan, T., "Efficient particle filters for joint tracking and classification", *Proceedings of SPIE Signal and Data Processing of Small Targets*, vol. 4728, pp. 439-449, Orlando, FL, April 2002.

[28] Gordon, N., Salmond,D. J. and Smith, A. F. M., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", *IEE Proceedings Part F: Radar and Signal Processing*, Vol. 140, No. 2, pp. 107-113, April 1993.

[29] Grimmett, D. and Coraluppi, S., "Contact-level multistatic sonar data simulator for tracker performance' assessment", *9th International Conference on Information Fusion, 2006*, pp. 1–7, Florence, Italy, July 2006.

[30] Grimmett, D., Coraluppi, S., La Cour, B.R., Hempel, C.G., Lang, T., de Theije, P.A.M. and Willett, P., " MSTWG multistatic tracker evaluation using simulated

scenario data sets", *11th International Conference on Information Fusion, 2008*, pp. 1–8, Cologne, Germany, July 2008.

[31] Grocholsky, B., 'Information-theoretic control of multiple sensor platforms", *PhD thesis*, Department of Aerospace, Mechatronic and Mechanical Engineering, University of Sydney, Australia, 2002.

[32] Papanicolopoulos, C.D., Blair, W.D., Sherman, D.L. and Brandt-Pearce, M., "Use of a Rician distribution for modelling aspect-dependent RCS amplitude and scintillation", *IEEE Radar Conference, 2007*, pp. 218-223, Boston, MA, April 2007.

[33] Hempel, C. G., "Track initialization for multi-static active Sonar systems", *OCEANS 2007 - Europe*, pp. 1–6, Aberdeen, Scotland, June 2007.

[34] Horridge, P. R. and Hernandez, M. L., "Performance bounds for angle-only filtering with application to sensor network management", *Proceedings of the Sixth International Conference of Information Fusion*, Vol. 1, pp. 695–703, July 2003.

[35] Hue, C., Cardre, J. P. L. and Perez, P., "Sequential Monte Carlo methods for multitarget tracking and data fusion", *IEEE Transactions on Signal Processing*, vol. 50, No. 2, pp. 309-325, February 2002.

[36] Herman, S. and Moulin, P., "A particle filtering approach to FM-band passive radar tracking and automatic target recognition", *Proceedings of the IEEE Aerospace Conference*, vol. 4, pp. 1789-1808, Bigsky, Mont, USA, March 2002.

[37] Jaeger, H., "A short introduction to observable operator models of stochastic processes", *Fourteenth European Meeting on Cybernetics and Systems Research*, Vienna, pp. 3105–3114, April 1998.

[38] Jaeger, H. Zhao, K. Krettzhmar, T. Obesrtein, D. Popovinci and Kolling, A., "Learning observable operator models via the ES algorithm", In *New Directions in Statistical Signal Processing: from Systems to Brains*, To be published by MIT Press, editors: S.Haykin, J. C. Principe, T. Sejnowski, and J. McWhirter.

[39] Ji, S. and Carin, L., "Adaptive multiaspect target classification and detection with hidden markov models", *IEEE Transactions on Sensors*, Vol. 5, No. 5, pp. 1035–1042, October 2005.

[40] Jie, C. Sinha, A. and Kirubarajan, T., "EM-ML algorithm for track initialization using possibly onoinformative data", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 41, No. 3, pp. 1030–1048, July 2005.

[41] Kendrick, J., Maybeck, P. and Reid, J., "Estimation of aircraft target motion using orientation measurements", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 17, No. 2, pp. 254–260, March 1981.

[42] Kirubarajan, T., Wang, H., Bar-Shalom, Y., Pattipathi, K.R., "Efficient multisensor fusion using multidimensional data association", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 37, No. 2, pp. 386–398, April 2001.

[43] Kirubarajan, T., Bar-Shalom, Y., "Low observable target motion analysis using amplitude information", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 32, No. 4, pp. 1367–1383, October 1996.

[44] Kostakis, J., Cooper, M., Green, T. J., Miller, M. I., O'Sullivan, J. A., Shapiro, J. H. and Snyder, D. L., "Multispectral Active-Passive Sensor Fusion for Ground-Based Target Orientation Estimation", *Proceedings of SPIE on Automatic Target Recognition*, Vol. 3371, pp. 500–5007, April 1998.

[45] La Cour, B., "Bayesian multistatic tracking: results on simulated data from the multistatic tracking working group", *9th International Conference on Information Fusion, 2006*, pp. 1–7, Florence, Italy, July 2006.

[46] La Cour, B., Collins, C. and Landry, J., "Multi-everything Sonar Simulator (MESS)", *9th International Conference on Information Fusion, 2006*, pp. 1–6, Florence, Italy, July 2006.

[47] Lanterman, A. D., "Tracking and recognition of airborne targets via commercial television and FM radio signals", *Proceedings of SPIE Acquisition, Tracking, and Pointing XIII*, vol. 3692, pp. 189-198, Orlando, FL, April 1999.

[48] Lang, T. and Hayes, G., "Evaluation of an MHT-Enabled Tracker with Simulated Multistatic Sonar Data", *OCEANS 2007–Europe*, pp. 1–6, Aberdeen, Scotland, June 2007.

[49] Lauberts, A., Karlsson, M., Näsström, F. and Aljasmi, R. , "Ground target classification using combined radar and IR with simulated data", *Conf*, Vol. 5, No. 5, pp. 1035–1042, October 2005.

[50] Layne, J., "Automatic target recognition and tracking filter", *In Proceedings of SPIE Aerosence*, Vol. 4473, pp. 198-209, Orlando, FL, April 1998.

[51] Layne, R. F., Krishnamurthy, A. K. and Ahalt, S. C. , "Multi-aperture SAR target detection using hidden Markov models", *Proceedings of the SPIE International Symposium on Algorithms for Synthetic Aperture Radar Imagery II*, Vol. 2487, pp. 198-209, Orlando, FL, April 1995.

[52] Li, J. and Stoica, P., "Efficient mixed-spectrum estimation with application to

target feature extraction", *IEEE Transactions on Signal Processing*, Vol. 44, No. 2, pp. 281–295, February 1996.

[53] Liao, X., Bao, Z. and Xing, M., "On the aspect sensitivity of high resolution range profiles and its reduction methods", *IEEE International Radar Conference*, Vol. 44, No. 2, pp. 281–295, May 2000.

[54] Linde, Y., Buzo, A. and Gray, R., "An algorithm for vector quantizer design", *IEEE Transactions on Communications*, Vol. 28, No. 1, pp. 84–95, January 1980.

[55] Liu, J. and West, M., "Combined parameter and state estimation in simulation-based filtering", *Sequential Monte Carlo Methods in Practice*, New York: Springer-Verlag, 2001.

[56] Malick, M., Maskell, S., Kirubarakan, T. and Gordon, N., "Littoral tracking using particle filter", *In Proceedings of the Information Fusion*, Vol. 2, pp. 935-942, USA, July 2002.

[57] Marc, R., Mohmood, R. and Jaime, S., "Multi-aspect target discrimination using hidden Markov models and nural network", *IEEE Transactions on Nural Network*, Vol. 16, No. 2, pp. 447–459, March 2005.

[58] Masataka, H., Tetsuya, K., and Iwao, S., "Maneuver target tracking with an acceleration estimator using target past position", *Electronics and Communications in Japan, Part I*, Vol. 85, No. 12, pp. 29–37, November 1995.

[59] Miller, M. I., Srivastava, A. and Grenander, U., "Conditional-mean estimation via jump-diffusion processes in multiple target tracking/recognition", *IEEE Transactions on Signal Processing*, Vol. 43, No. 11, pp. 2678–2690, November 1995.

[60] Pattipati, K. R., Popp, R. L. and Kirubarajan, T., "Survey of assignment techniques for multitarget tracking", In Bar-Shalom, Y. and Blair, W. D., (EDs.), *Multisensor-Multitarget Tracking: Application and Advances*, Vol. 3, Boston, MA: Artech House, 2000.

[61] Pei, B. and Bao, Z., "Multi-aspect radar target recognition method based on scattering centers and HMMs classifiers", *IEEE Transactions on Signal Processing*, Vol. 41, No. 3, pp. 1067–1074, July 2005.

[62] Rabiner, L. R. and Juang, B. H., "An Introduction to hidden markov models", *IEEE ASSP Magazine*, Vol. 39, No. 1, pp. 4–16, January 1986.

[63] Register, A., Blair, W. D., Ehrman, L. and Willet, P. K., "Using measured RCS in a serial, decentralized fusion approach to radar-target classification", *IEEE Aerospace Conference*, pp. 1–8, March 2008.

[64] Runkle, P. R., Nguyen, L.H., McClellan, J. H. and Carin, L., "Multi-aspect target detection for SAR imagery using hidden markov models", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 39, No. 1, pp. 46–55, January 2001.

[65] Runkle, P. R. and Bharadwaj, P. K., "Hidden markov models for multiaspect target classification", *IEEE Transactions on Signal Processing*, Vol. 47, No. 7, pp. 2035–2040, July 1999.

[66] Sathyan, T., Sinha, A. and Kirubarajan, T., "Passive geolocation and tracking of an unknown number of emitters", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 42, No. 2, pp. 740–749, April 2006.

[67] Storvik, G., "Particle filters for state-space models with the presence ofunknown

static parameters", *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, pp. 281–289, February 2002.

[68] Sutharsan, S., Kirubarajan, T., Lang, T. and McDonald, M., "Observable Operator Model Based Joint Multitarget Tracking with Multiaspect Classification", *Submitted to IEEE Transactions on Aerospace and Electronic Systems*, 2009.

[69] Sworder, D. and Hutchins, R., "Maneuver estimation using measurements of orientation", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 26, No. 4, pp. 625–638, July 1990.

[70] Sworder, D. and Hutchins, R., "Image enhanced tracking", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 26, No. 4, pp. 701–710, July 1990.

[71] Tharmarasa, T., Kirubarajan, T. and Hernandez, M. L., "Large-scale optimal sensor array management for multitarget tracking", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 37, No. 5, pp. 803–814, September 2007.

[72] Tharmarasa, R., Sutharsan, S. and Kirubarajan, T., "Multiframe assignment tracker for MSTWG data", *12th International Conference on Information Fusion, 2008*, Seattle, Washington, USA, July 2009.

[73] Tichavská, P., Muravchik, C. H. and Nehorai, A., "Posterior Cramér-Rao bounds for discrete-time nonlinear filtering", *IEEE Transactions on Signal Processing*, Vol. 46, No. 5, pp. 1386–1396, May 1998.

[74] Urich, R., *Principles of underwater sounds.* 3rd ed., New York, NY: Mc-Graw-Hill, 1983.

[75] Van Trees, H. L., *Detection, estimation and modulation theory.* New York, NY: John Wiley, 1968.

[76] Vihola, M., "Rao-blackwellised particle filtering in random set multitarget tracking", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 43, No. 2. pp. 689–705, April 2007.

[77] Yaakov, O. and Davis, A., "Differential-game-based guidance law using target orientation observations", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 42, No. 1. pp. 316–326, January 2006.

[78] Zyweck, A. and Bogner, R. E., "Radar target classification of commercial aircraft", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 32, No. 2. pp. 598–606, April 1996.