Arab Academy for Science, Technology and Maritime Transport
College of Engineering and Technology
Electronics & Communications Engineering Department

MODIFIED KALMAN FILTER TRACKER BASED ON HIDDEN MARKOV MODEL

By
Eng. Nawal Aly Mahmoud Zaher

A dissertation submitted to AASTMT in partial fulfillment of the requirements for the award of the degree of

MASTER of SCIENCE

In
Electronics and Communications Engineering

Supervisors

Prof. Dr. Ashraf Mamdouh
Electronics & Communications Engineering Department
Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt.

Assoc. Prof. Dr. Hussien Hamed Ghouz
Electronics & Communications Engineering Department
Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt.

CAIRO
OCTOBER 2013

ACKNOWLEDGEMENTS

I am mostly grateful to **ALLAH,** the most gracious, for all the blessings he gave me; starting with my wonderful family, the privilege of education and the strength and will to complete my M.Sc. thesis.

I have my **parents** to thank for being the person I am now. They provided me with the best education and never saved an effort to try and make a better person out of me. The warm and loving environment they provided me during my childhood paved the way into my transformation into an engineer and a researcher.

Many thanks go to my husband, **Mohammad**, the most supportive person ever. His love, support and encouragement gave me the strength to finish my work. I also thank my son, **Adam**, whose smiles make my day. My husband's and son's mere existence makes the world a better place.

I thank my supervisors **Prof. Ashraf Mamdouh and Dr. Hussein Ghouz** for their guidance.

My deepest appreciation goes to **Prof. Ashraf Mamdouh** for his great contribution regarding my work. His fruitful discussions and explanations shed light onto things when problems seemed unsolvable. The editing, comments and improvements he provided me with, transformed simple ideas into fine research.

I would like to thank my beloved sisters and my brother, my extended family and my friends for their support. Each and every one of them helped me along the way.

I gratefully acknowledge the staff of the Electronics and Communications Engineering Department for providing me with support and convenience during my studies at the Arab Academy for Science, Technology and Maritime Transport.

Finally, I must thank my husband for proof-reading parts of the material.

ABSTRACT

Target Tracking is comprised of two stages; data association and position estimation. Data association in a noisy multi-target environment is one of the problems that need solving, for accurately tracking a target. In this thesis, a technique that utilizes Hidden Markov Model (HMM) is used for data association prior to tracking targets with a Kalman filter tracker. Then, multiple sensor data fusion is performed based on Bayesian Minimum Mean Square Error Criterion (MMSE). The fused estimates are considered to have correlated estimation error. Also feedback from the global estimate into local trackers is implemented to improve local tracking performance.

Examples for maneuvering and non-maneuvering crossing targets are simulated. Kalman Filter is used in the second stage of the algorithm to provide a state estimate for a target based on the measurement associated to the target from the first stage. The results show an enhancement in the error performance as compared to data association with the Nearest Neighbor Standard Filter (NNSF). Comparison to perfect association results also shows that the algorithm performs almost as good as perfect association performance. The association technique withstands high sensor error levels.

Multiple sensor data fusion also improves error performance. The effect of increasing the number of sensors is studied from two to five sensors. Also the effect of the variations in sensors' accuracies is simulated from moderate to extreme cases. In extreme cases of variations between sensors, performing data fusion is not recommended as it has lower performance than the track with the least estimation error.

TABLE OF CONTENTS

LIST OF SYMBOLS

A	State transition probability distribution
a_{ij}	Probability of transition
$a_i(t)$	Acceleration matrix at instant t
B	Observation probability distribution in state
C	Number of comparisons
d_{ij}	Distance between measurement j and target i
F	State transition matrix
G	Gain matrix
$g(x)$	Gaussian distribution of x
H	Measurement matrix
J_k	Mean square error
K_k	Kalman filter gain
$K_k^{(1)}$	Scaling factor matrix
K_i^f	Local Kalman filter gain with feedback
L	Sigma points
$l(O\|\lambda)$	Likelihood of observation O given model λ
M	Number of multiplications
M_o	Set of distinct alphabet
m	Number of mixtures in a GMM
m_k	Symbol k of the alphabet
N_O	Observation sequence length
N_c	Number of computations
N_s	Number of states of an HMM
n_m	Number of measurements
n_t	Number of targets
n_x	State dimension
O	Observation sequence
P_{ij}	Cross-correlation between two estimates
P_i^f	Local covariance matrix with feedback
P_k	Covariance of the state error

P_k^-	A priori covariance matrix
$P(O\|\lambda)$	Probability of observation O given model λ
Q_k	Process noise covariance
q_t	State of the model at time t
R_k	Measurement noise covariance
S_j	j^{th} state of an HMM
v_k	Measurement noise
$v_{x,y}(t)$	x or y velocity at time t
w_i	Weight of density function i
w_k	Process noise
X	Complete data
X_f	Fused estimate
$X(t)$	State vector
x_k	Discrete time state at instant k
$\hat{x}_k$	A posteriori estimate of the signal
$\tilde{x}_k$	Estimation error
$\hat{x}_k^-$	A priori estimate
$\hat{x}_i^f$	Local estimate of sensor i with feedback
$x(t)$	Actual state at time t
$x_i(t)$	State of target i at time t
Y	Observed data
Z	Unobserved data
z_k	Measurement at time k
$\tilde{z}_k$	Innovation
$z_i(t)$	Measurement of target i
β_i	Probability of association of the ith measurement
δ	Sampling interval
$\delta_T(j)$	Highest probability that a partial observation sequence matches a state sequence
λ	Hidden Markov Model
π	Initial state distribution
σ_i	Standard deviation of distribution i

σ_{sn}	Standard deviation of sensor n
σ_{xi}, σ_{yi}	Standard deviation of target i in the x or y direction
μ_i	Mean of distribution i

LIST OF ABBREVIATIONS

EM	Expectation Maximization
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
HTK	HMM Tool Kit
JPDA	Joint Probabilistic Data Association
LQG	Linear Quadratic Gaussian
MHT	Multiple Hypothesis Tracker
MMSE	Minimum Mean Square Error
MSE	Mean Square Error
MTMST	MultiTarget MultiSensor Tracking
NNSF	Nearest Neighbor Standard Filter
PDA	Probabilistic Data Association
PDF	Probability Density Function
VDA	Viterbi Data Association

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Target tracking is the process of determining the position of a target from various sensor data. It is a key part in many military and civilian applications. With the advancement in weapons systems, robotics and computer vision, the need for accurately tracking multiple targets has become prominent. Ballistic Missile defense and Airborne Surveillance require identification and tracking of hundreds of targets at a time. This encompasses maneuvering and non-maneuvering targets with noisy sensor measurements and noise from atmospheric disturbances. The Multitarget-Multisensor Tracking (MTMST) problem has a wide variety of applications such as satellite surveillance, battlefield surveillance, air defense, air traffic control and non-military vehicle tracking system. The problem also has application to pattern recognition problems and robotics [1].

The optimal solution in Bayesian sense of the MTMST problem is the Multiple Hypothesis Tracker (MHT), which involves calculating the probability of every possible track and selecting the most probable one [1,2]. The computational complexity of such approach makes its practical realization unfeasible using even the most powerful computers [1].

Data Association prior to tracking reduces the complexity of the problem. Various methods exist for assigning measurements to targets. A simple solution to the data association problem is the Nearest Neighbor Standard Filter (NNSF). In NNSF only one measurement is assigned to each target based on its proximity from the target estimate [3]. Other techniques, such as the probabilistic data association (PDA) and the joint PDA, use measurement-to-track association probabilities for the individual estimates as weights to combine innovations [4]. The all-neighbor fuzzy association approach uses the fuzzy clustering algorithm and possibility distribution to replace probability, thus reducing complexity [5].

Another category of solutions utilizes pattern recognition techniques such as neural networks and fuzzy logic techniques. However, neural networks require an unreasonably large number of neurons and thus are difficult to train [6]. Fuzzy logic techniques provide approximate solutions whose accuracy depends on the choice of variables [7].

On the other hand, multisensor data fusion can be implemented to enhance performance further and utilize all the available information from different sensors. The fusion of information from different sensors to improve performance can be implemented in various ways, with the assumption of either correlated or uncorrelated estimation errors for tracks from different sensors. Also feedback of the global estimate into local trackers may or may not be adopted. The effect of feedback onto the tracking performance should also be analyzed.

In order to find a solution to the MTMST problem; three stages have to be implemented. The assignment of measurements to targets needs to be resolved. Then the target estimate using a single measurement has to be determined. The last stage is combining various target estimates that originated from measurements obtained by various sensors.

1.2 Objective of Thesis

In this thesis, an approach is proposed that associates measurements to tracks based on likelihood calculated by projecting a short sequence of states on a hidden Markov model (HMM) that is previously trained to capture the dynamics of the target. The proposed approach is meant to provide a better metric than distance upon which measurements are associated. By selecting a measurement for each target rather than combining weighted measurements, complexity should be reduced. Tracking is then performed using a Kalman filter and then tracks from various sensors are combined based on minimum mean square error (MMSE) criterion and performance is simulated for different sensor conditions. Two types of tracks are simulated; maneuvering and non-maneuvering targets.

1.3 Organization of Thesis

The organization of this thesis is as follows: Chapter 2 presents the background of the topics relevant to the subject of the thesis such as HMM, Kalman filter and multisensor fusion. At the end of Chapter 2, the literature review is discussed. The suggested models of target tracks and the proposed data association and fusion methods are explained in Chapter 3. Chapter 4 starts with an overview of the used simulator. Then, it provides the obtained results and a comparison to NNSF. Also complexity analysis is included in this chapter. Chapter 5 summarizes the conclusions and suggests recommendations for future work. Finally, a list of references and an Arabic summary are provided.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW ON MULTITARGET MULTISENSOR TRACKING

2.1 Introduction

The MTMST problem involves tracking multiple targets when unassigned (unlabelled) measurements exist that originated from various similar or dissimilar sensors. The problem incorporates various aspects, such as; assigning measurements to targets, estimating the state of a target, minimizing the estimation error and possibly choosing from various sensor data or the choice of combining data. Those aspects can be quantified as; 1) data association, 2) state estimation, and 3) data fusion.

While tracking multiple objects, usually multiple measurements appear, e.g., both due to targets and measurement noise. The incorrect measurements are referred to as false measurements, clutter, or other target measurements. Data association deals with the problem of selecting the measurement(s) that most probably originated from the object to be tracked. If the wrong measurement is selected, or if the correct measurement is not detected at all, poor state estimates could be the result. Various techniques exist for data association, which are explained in the literature review later in this chapter.

After selecting a measurement for the target, this measurement along with any prior knowledge of the target model and any prediction of its current state, are combined together to form the state estimate of the target. This stage is known as the state estimation phase.

In the presence of multiple sensors that could each detect the same target, a method to make use of all the data and combine the different estimates from various sensors is needed to obtain a better estimate than form single sensors. In this stage data fusion algorithms are of great importance.

2.1.1 Target Model

The target state and measurement could be represented by eq.s' (2.1) and (2.2) [8];

$$x_{k+1} = Fx_k + w_k \tag{2.1}$$

where x_k is the discrete time state at instant k, F is the state transition matrix and w_k is the process noise. Eq. (2.1) is assumed to be a Markov process, i.e. contains all measurement information up to z_k the measurement at time k, given by the following measurement model [8]:

$$z_k = Hx_k + v_k \tag{2.2}$$

where H is the measurement matrix and v_k is the measurement noise.

In tracking, the goal will be to recursively estimate the states i.e. position and velocity of a target in eq. (2.1).

2.2 Hidden Markov Model (HMM)

2.2.1 Discrete Markov Process

A discrete Markov process is a system that can be described at any time as being in one of a set of N_s distinct states $S_1, S_2, \ldots\ldots, S_{N_s}$. At each sampling time the system undergoes a transition from one state to another according to a set of probabilities associated with the state called the state transition probabilities given by [9].

$$a_{ij} = P\big(x(t) = S_j \big| x(t-1) = S_i\big)\,, \quad 1 \le i,j \le N_s \tag{2.3}$$

where a_{ij} is the probability of transition from S_i to S_j and $x(t)$ is the actual state at time t, and a_{ij} obeys standard stochastic constraints such that [8, 10];

$$a_{ij} \le 1 \tag{2.4}$$

$$\textstyle\sum_{j=1}^{N_s} a_{ij} = 1 \tag{2.5}$$

The above model is observable, since the output is the set of states at each time instant t and each state corresponds to an observable event [11].

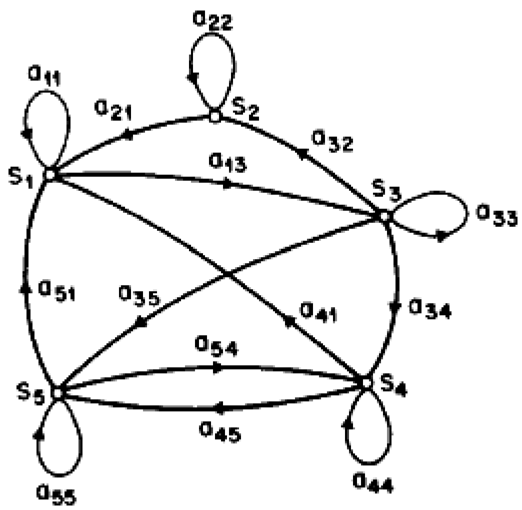

Figure 2.1: A Markov Chain with 5 states with selected state transitions [9].

2.2.2 Extension to Hidden Markov Model

In this section we extend the concept of Markov model to the case where the observation is a probabilistic function of the states. The resulting model is a doubly-embedded stochastic process with an underlying stochastic process that is not observable (hidden) but can only be observed through another set of stochastic processes that produce the sequence of observations [9].

The first stochastic process is a finite set of states, the transitions between the states are statistically defined by a set of transition probabilities. The second stochastic process is the distribution of observations over a particular state (since observations hold no certainty to which state they belong), usually the distribution of observable events over a state is a multidimensional probability distribution typically a Gaussian mixture model (GMM) [10].

Thus in an HMM when observations are made, no certainty is obtained about the state of the system, however there is a probability distribution for each state over possible observations. Figure 2.2 shows a depiction of an HMM with 4 states, where the sequence of states is chosen through observations.

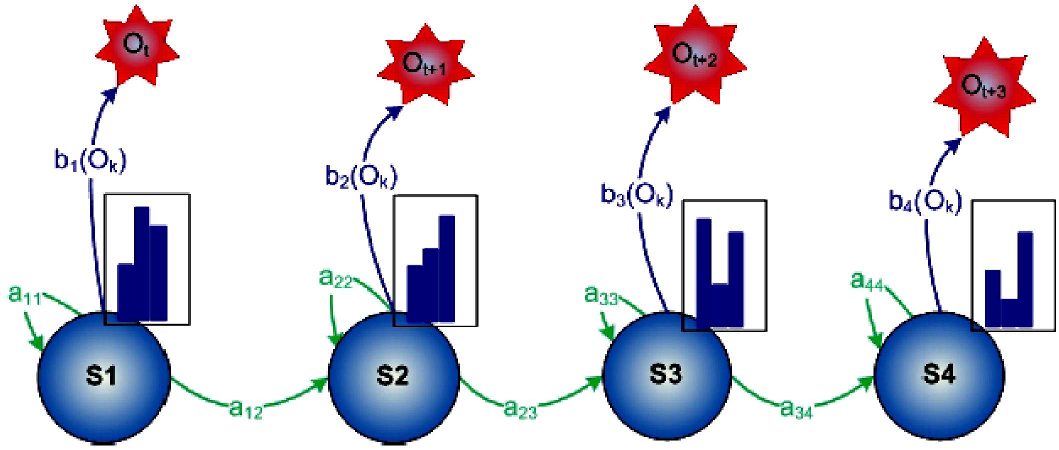

Figure 2.2: Example of an HMM [9].

2.2.3 Parameters of an HMM

An HMM is characterized by the following parameters [9, 10];

a) N_s, the number of states in the model. Generally the states are interconnected so that any state can be reached from any other state (an ergodic model). However, sometimes transitions between certain stages can be eliminated by setting the state transition probability between those two stages to zero.

b) Possible observation symbols per state; the observations could be discrete in nature, i.e. a set of distinct alphabet $M_o = \{m_1, m_2, \dots, m_k\}$ or continuous in nature like a physical phenomena being modeled.

c) The state transition probability distribution $A = \{a_{ij}\}$, where a_{ij} is given in (2.3). For an ergodic model we would have $a_{ij} > 0$ for all i, j.

d) The observation probability distribution in state j.

$$B = b_j(O) = P\left(O \middle| S_j\right) = P(x(t) = O | q_t = S_j) \tag{2.6}$$

where O is the observation and q_t is the state of the model at time t.

Usually B is a multidimensional distribution, typically a GMM.

e) The initial state distribution $\pi = \{\pi_i\}$, which describes the probability that the model is initially in state i.

$$\pi_i = P(x(0) = S_i) \qquad 1 \le i \le N \tag{2.7}$$

Thus the model λ can be described as;

$$\lambda = (A, B, \pi) \tag{2.8}$$

2.2.4 Gaussian Mixture Model

Each state is a GMM that is composed of a weighted sum of Gaussian density functions. The probability density function of the outputs over a particular state is given in eq.(5) [12, 13].

$$f_x(x) = \sum_{i=1}^{m} w_i g(x|\mu_i, \sigma_i) \tag{2.9}$$

where w_i is the weight given to density function i, m is the number of mixtures in the model and $g(x|\mu_i, \sigma_i)$ is a Gaussian distribution with a mean μ_i and a standard deviation σ_i.

2.2.5 Training

This phase is a crucial part for most of the applications of HMM, since it allows us to optimally adapt model parameters to observed training data, thus creating the most suitable model for a real phenomena [9].

The main problem of an HMM is to find a method to adjust the model parameters so as to maximize the probability of the training observation sequences. There is no known way to analytically solve for the model which maximizes the probability of the observations [14-16]. However, an iterative procedure can be used to solve for the model λ that locally maximizes $P(O|\lambda)$, such as the Baum-Welch Expectation Maximization (EM) algorithm [17].

2.2.6 Likelihood Calculation

Given an observation sequence $\boldsymbol{O}$ and a model $\boldsymbol{\lambda}$, the question is what is the most likely state sequence corresponding to this observation sequence. Using the Viterbi algorithm [18], the sequence with maximum likelihood can be chosen. An auxiliary variable is defined that gives the highest probability that a partial observation sequence matches a state sequence up to time $\boldsymbol{t}$ given the current state is $\boldsymbol{S_i}$ [10].

$$\delta_t(i) = max_{x(1),x(2),\dots x(t-1)} P(x(1), x(2), \dots x(t-1), x(t) = O_1, O_2, \dots, O_{t-1}, O_i | \lambda) \qquad (2.10)$$

$$\delta_{t+1}(j) = b_j(O_{t+1})\left[max_{1\le i\le N}\, \delta_t(i)\, a_{ij}\right],\ 1 \le t \le T-1 \qquad (2.11)$$

with;

$$\delta_1(j) = \pi_j b_j(O_1), 1 \le j \le N \qquad (2.12)$$

So $\delta_T(j)$ is calculated and we trace back through a trellis with the states of the model λ as its nodes, maximizing the probability as we go backwards uncovering the sequence of states.

The likelihood of this sequence is given by eq. (2.13).

$$l(O|\lambda) = \prod_{k=1}^{N} P(\,O_k|S_j)a_{ij} \qquad (2.13)$$

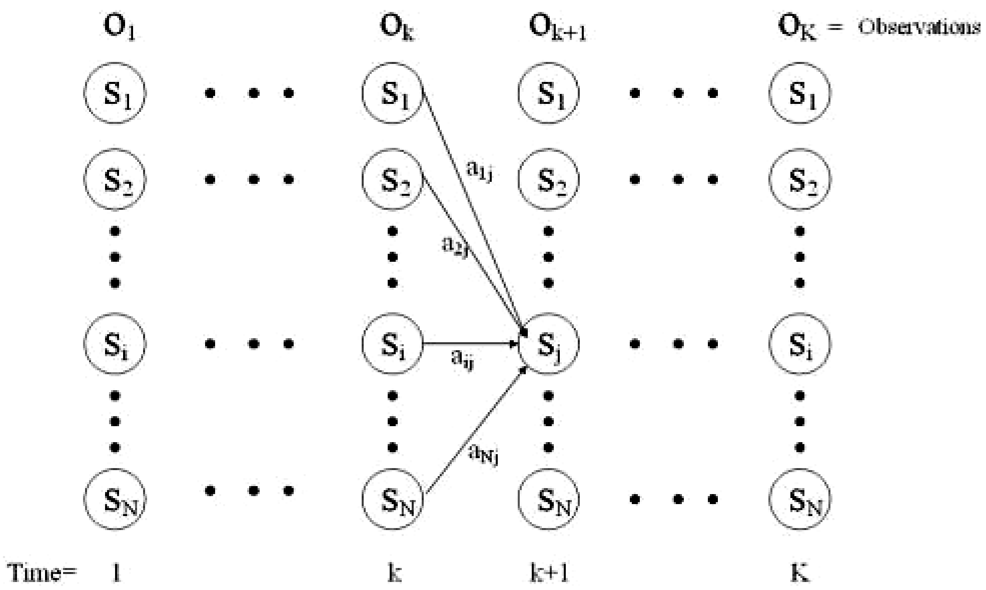

Figure 2.3: Trellis representation of an HMM

2.3 Kalman Filter

The Kalman filter is an optimum recursive data processing algorithm, used to estimate the state of a linear dynamic system perturbed by white Gaussian noise [19, 20]. One aspect of this optimality is that the Kalman filter incorporates all information available to it. It processes all available measurements regardless of their precision, to estimate the current value of the variables of interest with use of [19]:

a) Knowledge of the system and measurement device dynamics.

b) The statistical description of the system noises, measurement errors and uncertainty in the dynamics model.

c) Any available information about the initial conditions of the variables of interest.

The solution is recursive in that each updated estimate of the state is computed from the previous estimate and the new input data so the Kalman filter does not require all previous data to be stored and reprocessed every time a new measurement is taken [19, 21]. This is of vital importance so as to the practicality of the filter , also the Kalman filter is computationally more efficient than computing the estimate directly from the entire past observed data at each step of the filtering process [21]. Figure 2.4 shows a simple diagram to describe a system with the process and measurement noises and a Kalman filter used to estimate the system state.

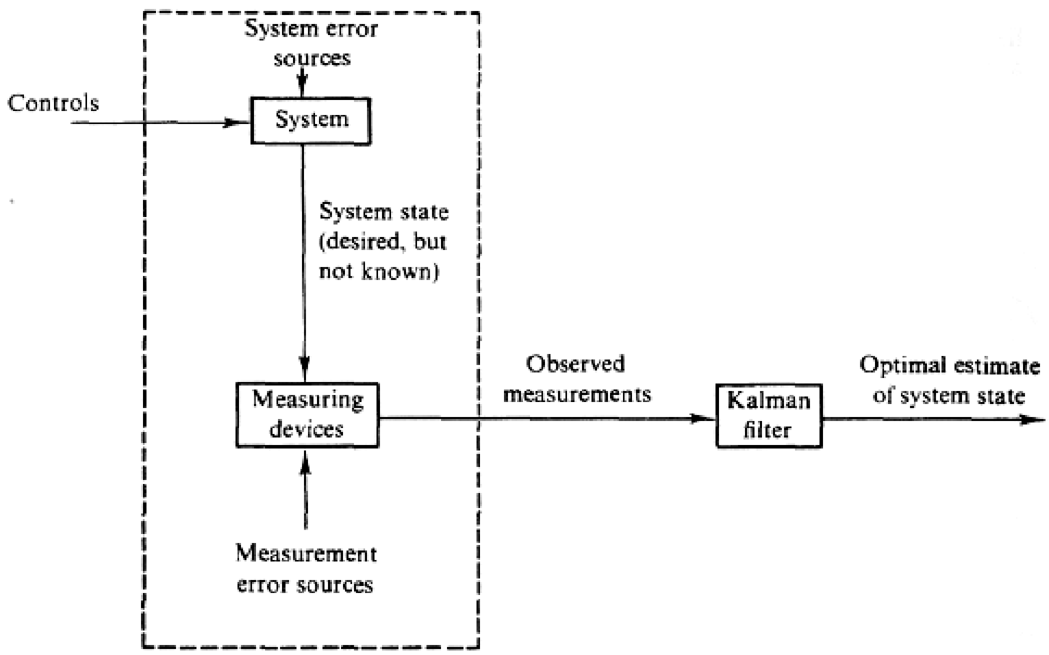

Figure 2.4: Typical Kalman filter application [19].

2.3.1 Basic Assumptions of a Kalman Filter

Theoretically, the Kalman filter is an estimator of the linear quadratic Gaussian (LQG) estimation problem [20]. The Kalman filter has three basic assumptions [19]:

a) A linear system model; which is adequate for many applications and mathematically more straightforward and developed than non-linear modeling.

b) Whiteness of noise; meaning that noise samples are uncorrelated in time and also have equal power over all frequencies. Figure 2.5 shows the equivalence between wideband noise and white noise within the pass band of the system.

c) Gaussialy distributed noise; this assumption can be justified by the fact that noise is a result of many random small sources combined together. It is mathematically proven that when a number of random variables are added up, the resulting distribution can be approximated by a Gaussian distribution.

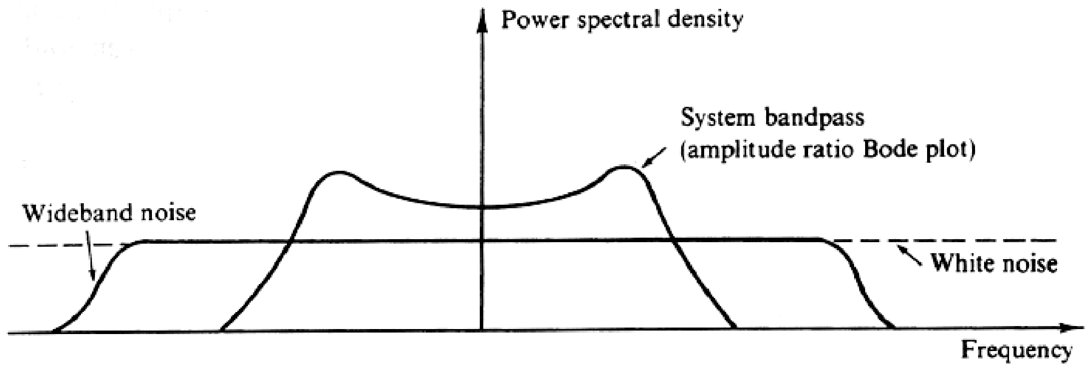

Figure 2.5: Wideband noise and white noise power spectral densities [19].

2.3.2 Mathematical Derivation

The derivation provided hereunder is from the work of Grewal and Andrews [20] and Haykin [21].
The system flow graph of a linear discrete time system is shown in Fig. 2.6 and the dynamical and measurement models are given by eq.s' (2.1) and (2.2).

The process noise w_k in (2.1) and the measurement noise v_k in (2.2) are assumed to be additive, white and Gaussian, with zero mean and with covariance matrices defined by;

$$Q_k = Cov(w_k) \tag{2.14}$$

$$R_k = Cov(v_k) \tag{2.15}$$

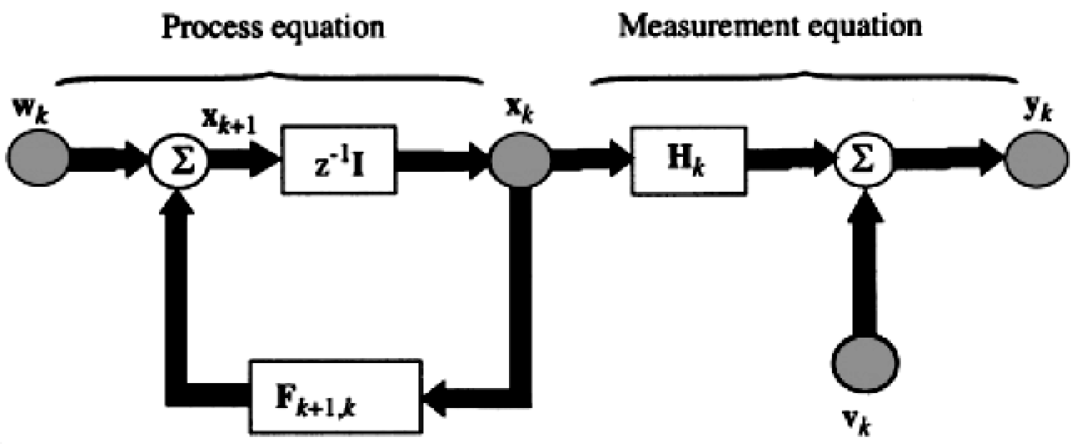

Figure 2.6: Signal-flow graph representation of a linear discrete-time system [21].

The problem of jointly solving the process and measurement equations for the unknown state in an optimum manner may now be formally stated as follows:

- Use the entire observed data, consisting of the vectors $z_1, z_2, \ldots, z_k$ to find for each k the minimum mean-square error estimate of the state x_i.
- The problem is called; filtering if $i = k$, prediction if $i > k$, and smoothing if $i < k$.

Let $\hat{x}_k$ denote the a posteriori estimate of the signal, given the observations $z_1, z_2, \ldots, z_k$. In general, the estimate $\hat{x}_k$ is different from the unknown signal x_k. To derive this estimate in an optimum manner, we need a cost function for incorrect estimates. The cost function should satisfy two requirements:

- The cost function is non-negative.
- The cost function is a non-decreasing function of the estimation error $\tilde{x}_k$ defined by;

$$\tilde{x}_k = x_k - \hat{x}_k \tag{2.16}$$

These two requirements are met by the mean-square error given by eq. (2.17).

$$\begin{aligned} J_k &= E[(x_k - \hat{x}_k)^2] \\ &= E[\tilde{x}_k^2] \end{aligned} \tag{2.17}$$

Principle of Orthogonality; if the stochastic processes $\{x_k\}$ and $\{z_k\}$ are zero mean and the estimate $\hat{x}_k$ is restricted to be a linear function of the measurements, and the cost function is the mean-square error,

Then:

The optimum estimate $\hat{x}_k$ given the observations $z_1, z_2, \dots, z_k$ is the orthogonal projection of x_k on the space spanned by these observations.

With a linear estimator the a posteriori estimate may be expressed as a function of the a priori estimate $\hat{x}_k^-$ and the measurement z_k , as follows;

$$\hat{x}_k = K_k^{(1)} \hat{x}_k^- + K_k z_k \tag{2.18}$$

Now it is required to determine the scaling factor matrices $K_k^{(1)}$ and K_k to get the optimum estimate. To find these two matrices we apply the principle of orthogonality, thus;

$$E[\tilde{x}_k z_i^T] = 0 \qquad for\ 1 \leq i \leq k-1 \tag{2.19}$$

Using eq.s' (2.16), (2.17), (2.18) and (2.19) we obtain;

$$E\left[(x_k - K_k^{(1)} \hat{x}_k^- - K_k H x_k - K_k w_k) z_i^T\right] = 0 \tag{2.20}$$

Since the process noise w_k and the measurement noise v_k are uncorrelated, then;

$$E[w_k z_i^T] = 0 \tag{2.21}$$

We may rewrite eq. (2.20) as;

$$E\left[\left(I - K_k H - K_k^{(1)}\right) x_k z_i^T + K_k^{(1)} (x_k - \hat{x}_k^-) z_i^T\right] = 0 \tag{2.22}$$

from the principle of orthogonality we know that;

$$E[(x_k - \hat{x}_k^-) z_i^T] = 0 \tag{2.23}$$

thus;

$$\left(I - K_k H - K_k^{(1)}\right) E[x_k z_i^T] = 0 \tag{2.24}$$

For arbitrary values of x_k and y_i this equation could only be satisfied if;

$$\left(I - K_k H - K_k^{(1)}\right) = 0 \tag{2.25}$$

which yields the following relationship between K_k and $K_k^{(1)}$:

$$K_k^{(1)} = I - K_k H \quad (2.26)$$

By substitution in eq. (2.18), the a posteriori estimate $\hat{x}_k$ can be obtained by eq. (2.27).

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (2.27)$$

The innovation process represents a measure of the new information contained in z_k and is given by;

$$\begin{aligned} \tilde{z}_k &= z_k - H\hat{x}_k^- \\ &= Hx_k + v_k - H\hat{x}_k^- \\ &= H\tilde{x}_k^- + v_k \end{aligned} \quad (2.28)$$

$$E[(x_k - \hat{x}_k)\tilde{z}_k^T] = 0 \quad (2.29)$$

Using eq. (2.27) and (2.28), we can express the state error as;

$$x_k - \hat{x}_k = (I - K_k H)\tilde{x}_k^- - K_k v_k \quad (2.30)$$

Substituting for eq. (2.28) and (2.30) into (2.29) we get;

$$E\left[\left((I - K_k H)\tilde{x}_k^- - K_k v_k\right)(H\tilde{x}_k^- + v_k)\right] = 0 \quad (2.31)$$

And since the measurement noise v_k is independent of the state x_k and thus of the state error $\tilde{x}_k^-$, then the above equation reduces to;

$$(I - K_k H)E\left[\tilde{x}_k^- \tilde{x}_k^{-T}\right]H^T - K_k E[v_k v_k^T] = 0 \quad (2.32)$$

Define the a priori covariance matrix P_k^- as;

$$P_k^- = E\left[\tilde{x}_k^- \tilde{x}_k^{-T}\right] \quad (2.33)$$

Then eq. (2.32) can be rewritten as;

$$(I - K_k H)P_k^- H^T - K_k R_k = 0 \quad (2.34)$$

Finally we get the Kalman filter gain K_k by solving the above equation;

$$K_k = P_k^- H^T [H P_k^- H^T + R_k]^{-1} \quad (2.35)$$

And the covariance of the state error P_k is given by;

$$P_k = (I - K_k H)P_k^- \quad (2.38)$$

2.4 Multisensor Estimate Fusion

Sensor fusion is the process of combining data from different sensors to form a better estimate than when using each sensor individually. Multisensor data fusion combines data from multiple sensors to perform inferences and achieve performance that may not be possible from a single sensor alone [22]. The concept of multisensor data fusion is hardly new. Humans and animals have evolved the capability to use multiple senses to improve their ability to survive. For example, it may not be possible to assess the quality of an edible substance based solely on the sense of vision or touch, but evaluation of edibility may be achieved using a combination of sight, touch, smell, and taste. Thus multisensory data fusion is naturally performed by animals and humans to achieve more accurate assessment of the surrounding environment and identification of threats, thereby improving their chances of survival [23].

Data fusion spans military and nonmilitary applications. Military applications include ocean surveillance, air-to-air and surface-to-air defense, battlefield intelligence, surveillance and target acquisition, and strategic warning and defense. Nonmilitary applications include medical diagnostic, robotics, remote sensing, and automated monitoring of equipments [23, 24].

2.4.1 Advantages of Multisensor Data Fusion

The advantage of a multisensor system over a single sensor system can be expressed in terms of the improvement in the system performance. The following are some performance measures that show the advantages of multisensor systems [25]:

a) Reliability: Multisensor systems have an inherent redundancy. If one or more sensors fail due to interference such as jamming, the system can continue to operate at a reduced performance level.

b) Coverage: Multiple sensors can observe a region larger than the one observable by a single sensor.

c) Confidence: Sensors can confirm each other’s inferences, thereby increasing confidence in the final system inference.

d) Response time: Since more data is collected by multiple sensors, a desired level of performance can be reached faster.

e) Resolution: The use of various sensors can result in an inference with better resolution than any of the sensors used.

2.4.2 Processing Levels in a Data Fusion Model

Data fusion incorporates various processing levels. Table 2.1 shows a summary of those levels along with their description.

Table 2.1: Processing levels of a data fusion system [23].

Level	**Description**
Level 0: Signal Refinement	Preprocessing of sensor data, e.g. amplification, de-noising, feature extraction, etc...
Level 1: Object Refinement	Combines locational, parametric and identity information to produce representatives of objects, e.g. position, identity, etc...
Level 2: Situation Refinement	Attempts to define a relationship between groups of entities, it incorporates environmental information, observations and a priori knowledge.
Level 3: Threat Refinement	Projects the current situation into the future to draw conclusions about enemy threats, it involves knowledge and analysis of enemy data.
Level 4: Process Refinement	A meta process (i.e. a process concerned about other processes), which involves assessing the performance of lower level processes and monitoring the performance of data fusion.

Those processing levels are incorporated in the data fusion process which is represented as a feedback closed loop structure shown in Figure 2.7. In this architecture the feedback through the sensor manager is responsible for process refinement.

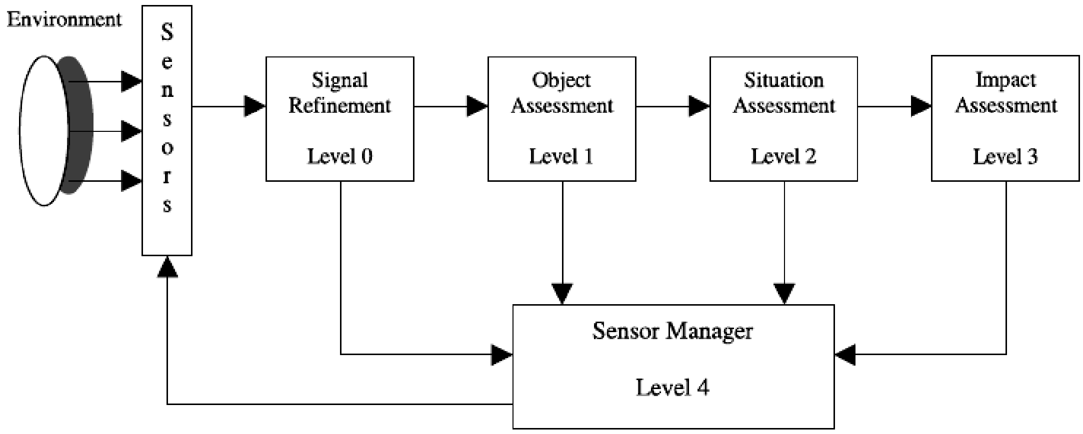

Figure 2.7: Feedback connection via sensor manager in a data fusion process [26].

2.4.3 Types of Inferences

According to the level of processing required, high or low level inferences can be made through a data fusion system. Figure 2.8 shows the hierarchy of inferences from lowest to highest levels of inference.

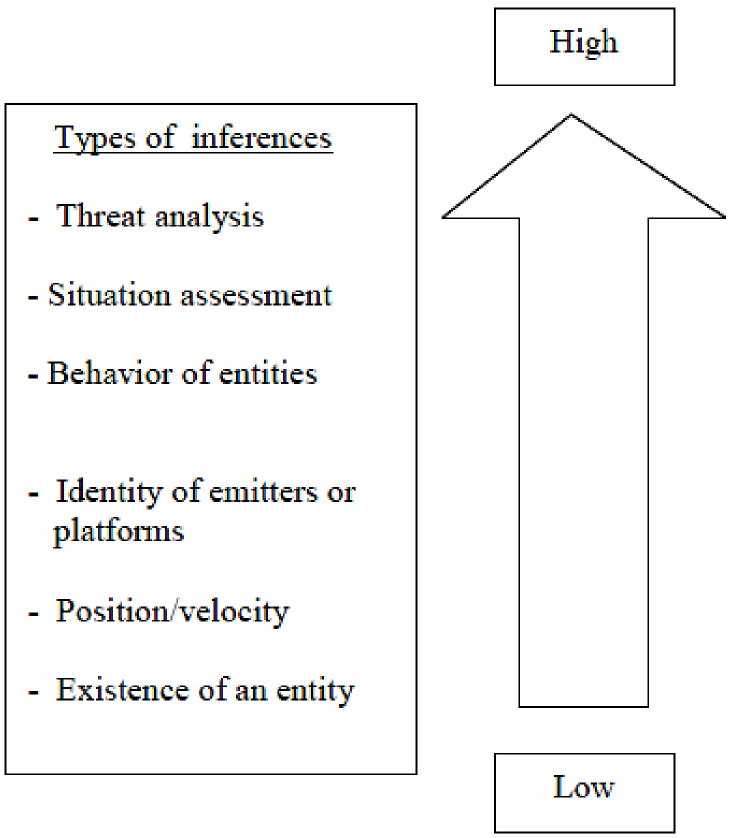

Figure 2.8: Hierarchy of inferences [22].

2.4.4 Positional Fusion

Positional fusion is combining data from various similar or dissimilar sensors to obtain an estimate of the position of a target. Dissimilar sensors means sensors with different accuracies. Positional fusion is divided into two stages: (1) parametric association and (2) estimation techniques. Parametric association correlates data from multiple sensors to multiple targets in MTMST problem. It is an important part of the tracking algorithm, as incorrect association might result in poor performance. Estimation techniques are then used to obtain a better estimate of the state vector [22, 27].

2.5 Data Association and Estimation Techniques

This section contains brief description of former data association techniques, state estimation algorithms and data fusion approaches that guided the flow of work in this thesis.

2.5.1 Multiple Hypothesis Trackers

This technique typically works with a set of detections comprised of both noisy measurements of the target position, in Cartesian or polar coordinates, and false alarms due to clutter. The detections are then either associated with existing tracks, used to create new tracks, or deemed false alarms. The first step of the MHF is the formulation of all feasible hypotheses. Then, when new data comes available, each hypothesis is expanded into a set of new hypotheses. This way a tree of hypothesis can be generated. With the formulation of each new hypothesis the compatibility constraint is maintained, i.e., only feasible hypothesis are considered. The track score is used to assess the validity of the track. As more data is measured, the size of the hypothesis tree can grow exponentially, which makes the solution computationally unfeasible [8, 28].

2.5.2 Nearest Neighbor Standard Filter

The idea of the NNSF is to select the measurement that is closest to the predicted estimate, i.e., it is an optimal solution in the sense that it minimizes the distance between predicted states and measured points. In order to do so, for each of the measurements the distance d_{ij} between measurement j and target i is calculated using eq. (2.37) and the measurement with the shortest distance is believed to be the correct one [29, 30].

$$d_{ij} = \sqrt{(\tilde{z}_i(t+1))^T \tilde{z}_i(t+1)} \tag{2.37}$$

where $\tilde{z}_{ij}(t+1)$ is the innovation calculated by eq. (2.28).

This filter can be implemented for tracking any number of known tracks and an advantage is the low computational complexity. An obvious drawback is that, with some probability, the nearest neighbor is not the correct measurement.

2.5.3 Measurement Gates

A common first step in solving the data association problem is the selection of a validation region, sometimes called (measurement) gate. The gate is a region in which the next measurement is highly probable to appear. In order to define the gate, the target is assumed to be on a track, such that a predicted measurement and the measurement prediction covariance matrix are available. It is assumed that the true measurement at time k, conditioned on the old measurements up to time $k+1$, is normally distributed. Measurements that fall within the gate are called validated measurements. The problem of single target data association with $h = 4$ is summarized in Figure 2.9 , where the size and shape of the ellipse are determined by the covariance of the innovation [29].

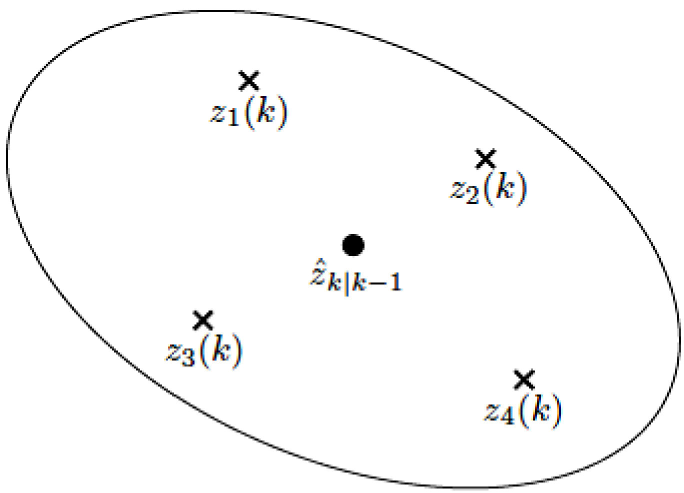

Figure 2.9: A single predicted target measurement with four validated measurements [8].

When the number of objects to be tracked exceeds one, it has to be decided which measurement originated from which target. A data association algorithm has to determine whether a measurement is correct or incorrect. A more complex situation is summarized in Figure 2.10, where the predicted measurements for two are validated using measurements gates. Data association algorithms based on gates, clearly should include a strategy that is able to deal with the appearance and the disappearance of tracks [29].

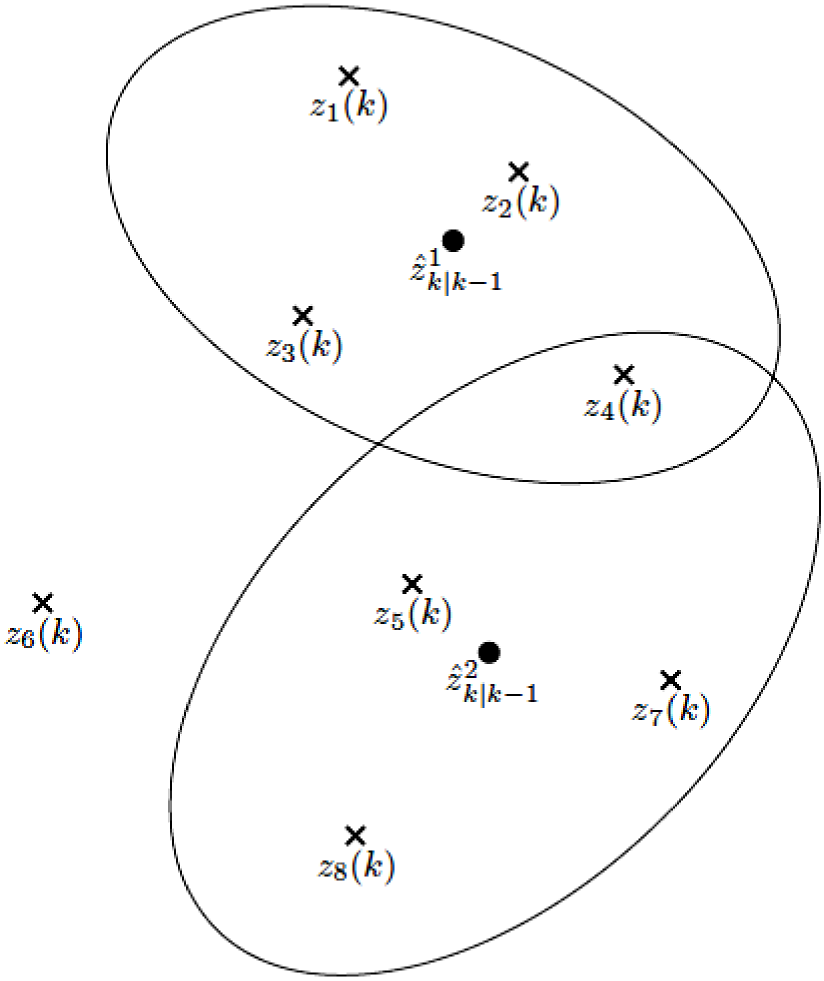

Figure 2.10: Two targets and the corresponding validated measurements [8].

2.5.4 Probabilistic Data Association Filter

The Probabilistic Data Association (PDA) Filter uses a Bayesian approach to the problem of data association or how to update the state when there is a single target and possibly no measurements or multiple measurements due to noise. Rather than possibly erring by choosing the nearest neighbor or data closest to what is expected in order to update the state, the PDA filter hedges its bets by weighting the influence of the various candidate measurements based on two assumptions. First, it assumes that there is exactly one target giving rise to one true measurement. Second, the PDA filter assumes that all other measurements are false and arise from a uniform noise process. The relevant step in the Kalman filter is the computation of the innovation. The PDA filter introduces a notion of the combined innovation, computed over the n measurements detected at a given time step as the weighted sum of the individual innovations [31, 32]:

$$\tilde{z} = \sum_{i=1}^{n} \beta_i \tilde{z}_i \tag{2.38}$$

Each β_i is the probability of the association that the ith measurement is target-originated.

2.5.5 Joint Probabilistic Data Association Filter

The Joint Probabilistic Data Association (JPDA) filter; an extension to the PDA filter, enforces a kind of exclusion principle that prevents two or more trackers from latching onto the same target by calculating target-measurement association probabilities jointly. Suppose that we are tracking T objects, for which a total of n measurements have been generated. A key notion in the JPDA filter is that of a joint event or conjunction of association events. More specifically, the difference is that the measurement to target association probabilities are calculated jointly across targets. The probability of a particular event depends, as with the PDA filter, on the distances between each target's predicted state and the measurements. However, an additional influence on the probability of an event stems from the interaction of the various association events. The JPDA filter disregards infeasible joint events and, thus, avoids inappropriate state convergence [29, 31].

2.5.6 All-Neighbor Fuzzy Association

Data association is performed by updating the predicted target state estimate using a fuzzy weighted sum of innovations. Unlike the joint probabilistic data association filter, in which the similarity measures are determined in terms of the conditional probability for all feasible data association hypothesis, the proposed fuzzy association approach determines the similarity measures between measurements and tracks in terms of possibility weights based on a partition matrix. The possibility weights are determined according to the fuzzy clustering algorithm. This approach has a lower computational complexity in the expense of a little lower performance compared to the standard JPDA filter [5].

2.5.7 Viterbi Data Association

If the sequence of measurements is set onto a trellis, the Viterbi algorithm can be used to solve the data association problem by finding an optimal sequence, i.e., an optimal track. In order to be able to find any optimal solution, some cost function has to be defined. The cost function that is used for Viterbi Data Association (VDA) can be based on similar ideas as the NNSF, explained earlier. The VDA starts with a set of validated measurements $z_i(k)$, with $i = 1, 2, \ldots, n$. Then, when a new set of measurements $z_i(k+1)$, becomes available, the lengths of the existing tracks from the starting point to the new measurement $z_i(k+1)$ through the old measurements $z_i(k)$, is calculated. As a result, for each measurement, $z_i(k+1)$ the length of the shortest path to this measurement is obtained. The difference with the NNSF lies in the fact that the NNSF only calculates one path each recursion step [33].

2.5.8 Track Splitting

The track splitting filter described in this section is a batch method, i.e., it uses a sequence of data obtained at multiple time instants. The main assumptions are linear dynamic and measurement models and Gaussian process and measurement noise. After the initialization, at time $k = 1$, the track is split up into m_k tracks, one for each validated measurement. Then m_k validation regions are calculated and at time $k = 2$, the procedure is repeated. If two measurements at successive times are close to each other and not associated, this could be used to initiate a new track. Clearly, this strategy has to deal with a rapidly increasing number of tracks and for that reason, likelihoods of all tracks are calculated. If the likelihood is lower than a predefined threshold, it will be eliminated.

Once a track has a long history, the likelihood of the track will be dominated by this long history. As a result, the response will be slow and the memory and computation requirements will rapidly increase. This is a major drawback and, therefore, a sliding window can be included to ensure that only the last measurements are taken into account. Another disadvantage is that the track splitting filter allows shared measurements between tracks which may result in non-physiological tracks [34].

2.5.9 Expectation Maximization

Estimating the state of a number of unknown targets under uncertain measurement origin is a non-classical filtering problem, the classical filtering problem arising when the measurements origins are known. The non-classical filtering problem can also be considered an incomplete data problem. To develop the idea of complete data let Y be the observed or incomplete data and Z represent some unobserved data which, if available, simplifies the estimation problem. Then the complete data can be represented by X, where X = (Y, Z). In the above state estimation problem, the observed data, Y, are the measurement returns from sensors over the observation time while the unobserved data, Z, are the associations between the measurements and the set of possible classes from which the measurements can originate. Looking at the non-classical filtering problem as an incomplete data problem, we can draw upon solution techniques for parameter estimation from this domain. Recently, there has been much interest in the literature regarding the maximum-likelihood (ML) estimation of parameters from incomplete data by use of the Expectation-Maximization (EM) algorithm. The EM algorithm is an iterative procedure that estimates both the parameters and the missing or unobservable data during an iteration. The approach first computes an approximation to the expectation of the log-likelihood functional of the complete data conditioned on the current parameter estimate. This is called the expectation step (E-step) and here the current incomplete data estimate is calculated. Next, a new parameter estimate is computed by finding the value of the parameter that maximizes the functional found in the E-step. This is called the maximization step (M-step). The EM algorithm has been found to have the advantages of reliable global convergence properties in most instances, although it can exhibit seeming slow convergence in some applications [35].

2.5.10 Bayesian Tracking Approach

In the Bayesian approach, the states are assumed to be random variables with a posterior probability density function (pdf) $p(x_k|z_{1:k})$, which is approximated using (1) a prediction and (2) an update step.

1. In the prediction step, the Chapman-Kolmogorov equation will be used to calculate

the prior pdf $p(x_k|z_{1:k-1})$, i.e., the predicted pdf of x_k on the basis of measurements up to time $k-1$. If the system in eq. (2.1) is assumed to be Markov, then the Chapman-Kolmogorov equation is:

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})\, dx_{k-1} \qquad (2.39)$$

2. At time step k, a new measurement z_k becomes available and Bayes' rule can be used to update the prior pdf to the posteriori probability, using the conditional probability:

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \qquad (2.40)$$

If this problem of recursively calculating the posterior pdf is solved exactly, the optimal Bayesian solution is obtained. Unfortunately, this optimal solution only exists in a restricted set of cases since it involves the evaluation of complex high-dimensional integrals [8].

2.5.11 Linear Kalman Filter

The Linear Kalman filter gives the optimal Bayesian solution to the state estimation problem, if the posterior pdf at every time step is Gaussian. Furthermore the state transition matrix F and the measurement matrix H should be known and linear. The noise vectors w_k and v_k should be drawn from zero mean Gaussian distributions with known covariance. The main advantages are then optimal Bayesian solution and the low computational complexity and memory requirements. An obvious drawback is the above mentioned assumptions [36].

2.5.12 Extended Kalman Filter

In many practical situations, dynamical and measurement models are non-linear functions of the states, thus applying the linear Kalman filter directly would fail. One obvious sub-optimal Bayesian method is using a local linear approximation at each time step and then applying the linear Kalman filter. In the extended Kalman filter, the state transition and observation models need not be linear functions of the state but may instead be differentiable functions. The state transition matrix F can be used to compute the predicted state from the previous estimate and similarly measurement matrix H can be used to compute the predicted measurement from the predicted state. However, F and H cannot be applied to the covariance directly. Instead a matrix of partial derivatives is computed. At each time step, the Jacobian is evaluated with current predicted states. These matrices can be used in the Kalman filter equations. This process essentially linearizes the non-linear function around the current estimate. If the system under consideration has weak nonlinearities, this suboptimal algorithm can be very effective. There are a few major drawbacks using this approach; if the nonlinearities become severe, the performance of the filter can decrease rapidly [37, 38].

2.5.13 Unscented Kalman Filter

A second sub-optimal Bayesian method that is based on the linear Kalman filter is the unscented Kalman filter, that falls within the group of Sigma-Point Kalman filters. The basic idea of this filter is that it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function or transformation. While considering the spread of a random variable the unscented Kalman filter tends to be more accurate than the first order Taylor series linearization used in the extended Kalman filter. A brief overview of the algorithm is as follows [39]:

1. Select a minimum number of L points, called sigma points, where $L = 2n_x + 1$ and n_x is the state dimension. Since this selection is made deterministically, the sigma points can be chosen from a Gaussian distribution with a desired mean or covariance, which limits the required number of sigma points.

2. The nonlinear equation is used to transform the sigma points, leading to a set of transformed points.

3. The transformed points are used to re-approximate the (nonlinearly transformed) mean and covariance of the Gaussian distribution.

Contrary to the extended Kalman filter, the unscented Kalman filter can deal with severe nonlinearities while its computational complexity has the same order of magnitude. Disadvantage is that again a Gaussian distribution is assumed. If the true density is non-Gaussian, it is very likely that neither the unscented Kalman filter nor the linear Kalman filter and the extended Kalman filter are able to describe it well [8].

2.5.14 Grid-Based Method

Another way to find the optimal pdf $p(x_k|z_{1:k})$, is using a grid-based method. This method assumes a discrete state space with a finite number of states. The posterior pdf is written as;

$$p(x_{k-1}|z_{1:k-1}) = \sum_{i=1}^{N_s} w_{k-1|k-1}^{i} \delta(x_{k-1} - x_{k-1}^{i}) \qquad (2.41)$$

where N_s is the number of states and $w_{k-1|k-1}^{i}$ is the conditional probability.

One of the disadvantages of this method is that the computational cost increases rapidly with the dimensionality of the state space, since the grid should be sufficiently dense to represent the continuous state space. Furthermore, the state space should be defined in advance and the grid should have a high constant resolution over the whole domain, or prior knowledge about regions with a high probability has to be used. Probably due to these reasons, grid-based methods are hardly used in recent literature [40].

2.5.15 Kalman Filtering Fusion

They use a test statistic to determine whether or not two tracks are the same and solve the problem of track fusion assuming independent estimation errors. The fused estimate, which minimizes the expected mean square error, and the corresponding covariance are given by [22, 41]:

$$X_f = P \sum_{i=1}^{n} P_i^{-1} \hat{x}_i, \qquad (2.42)$$

$$P^{-1} = \sum_{i=1}^{n} P_i^{-1}. \qquad (2.43)$$

2.5.16 Fusion with Correlated Noise

Track fusion is performed under the assumption that the estimation errors of different sensors are correlated. The measurement noises of two different sensors can be assumed independent but is not sufficient to yield the independence of their estimation errors. This is because the same process noise in the dynamic model makes the two estimation errors correlated [42, 43]. P_{ij} represents the cross-correlation between the two estimates and is given by:

$$P_{ij} = E\left[(\hat{x}_i - x)(\hat{x}_j - x)\right] = E\left[\tilde{x}_i \tilde{x}_j\right] = P_{ji} \qquad (2.44)$$

2.5.17 Fusion with Feedback

Kalman filtering track fusion formula with feedback is, like the track fusion without feedback, exactly equivalent to the corresponding centralized Kalman filtering formula. Moreover, the P matrices in the feedback Kalman filtering at both local trackers and the fusion center are still the covariance matrices of tracking errors. Although the feedback here cannot improve the performance at the fusion center, the feedback does reduce the covariance of each local tracking error i.e., the feedback improved local tracking performance [44].

It is obvious from surveying the various techniques for data association that the techniques that give the best performance are the ones that rely on putting more than a single measurement under study or combining various measurements. Where the techniques that select a measurement based on metrics as simple as distance have a much degraded performance. It is also noticed that most state estimation methods rely on the linear Kalman filter as linear estimation is mathematically more established and computationally more feasible than non-linear techniques. Were it necessary to have a non-linear model than locally linear techniques that linearize about a certain point at each time step are used. The last observation is that fusion can be implemented in a simple manner using Kalman filter equations or factors like correlated estimation errors and feedback may be considered.

CHAPTER 3

BASIC PROPOSED MODEL AND ANALYSIS

3.1 Introduction

In this chapter, the overall proposed tracking algorithm is discussed. The HMM-based data association is explained, then the Kalman filter tracker with full description of both maneuvering and non-maneuvering target models is shown, and, finally, Kalman filter data fusion with correlated estimation error and feedback is discussed.

3.2 The Proposed System

The proposed tracking system is composed of three main stages. The first stage is data association based on an HMM, which associates a single measurement to each target in a multitarget environment. The second stage is state estimation using a Kalman filter tracker to estimate x- and y- position and velocity. The final stage is combining estimates from various trackers through fusion based on MMSE criterion. Figure 3.1 shows an overall view of the proposed tracking system.

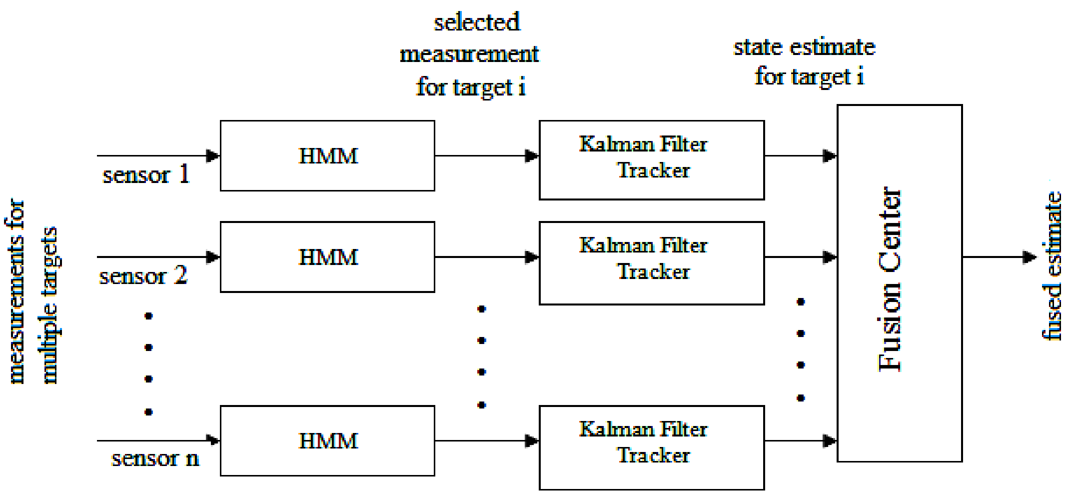

Figure 3.1: An overall view of the tracking system

3.3 Data Association Based on HMM

3.3.1 Constructing the model

As mentioned previously there is no optimal way for choosing an HMM's parameters, thus the initial parameters such as; model size and initial weights and variances of each mixture where chosen based upon simulation results for different model parameters.

The following values are chosen for the parameters:

a) The number of states in the model N_s. This number is of great importance, as a model with a small N_s might not capture all the statistical details of the problem, and a large value of N_s would result in a model that is practically impossible to train. From simulation results, it is found that $N_s = 5$ is adequate.

b) The state transition probability distribution $A = \{a_{ij}\}$; in our case an ergodic model is adopted, i.e., any state can transition to any other state.

$$a_{ij} > 0 \qquad \forall\, i, j \tag{3.1}$$

c) The observation probability distribution in state S_j of the model; the probability distribution chosen for each state is a GMM, where the number of mixtures is $m = 5$ of initial weights, means and variances given as:

$$w_i = \frac{1}{m} = 0.2,\ \mu_i = 0,\ \sigma_i = 1. \tag{3.2}$$

A large number of tracks are used to train the above HMM. Through training the state transition probabilities as well as the weights, means and variances of the GMMs' of each state are calculated. The training is performed by the Baum-Welch algorithm [16, 19]. The training phase is not part of the data association algorithm, rather an initial step to construct a model to capture the dynamics of the targets.

3.3.2 Data Association Metric

After the model is constructed it is used to calculate data association metrics as follows:

When a number of measurements h exist for a target, each measurement z_j is tested for $1 < j < h$. First a sequence of length N_O is created from the previously determined target states and a measurement as follows:

$$O = x(k - N_O + 2), x(k - N_O + 3), \dots, x(k-1), x(t), z_j \qquad (3.3)$$

The sequence length N_O is determined experimentally, the longer the sequence the better the performance is of the data association algorithm. However, a long sequence would result in higher complexity and more memory being used. The sequence length of the work presented in this thesis is taken as $N_O = 5$. A sliding window of length $N_O - 1$ is used with its center at $x\left(k - \frac{N_O}{2}\right)$ to obtain a state sequence. Then each measurement is added to constitute the last observation of the observation sequence O. The likelihood that an observation sequence was generated by our model λ is given by;

$$l(O|\lambda) = \prod_{k=1}^{N_O} P(\,O_k|S_j)a_{ij} \qquad (3.4)$$

Usually the likelihood is replaced by calculation of the log-likelihood given in eq. (3.5).

$$\log l(O|\lambda) = \sum_{k=1}^{N_O} \log\,(\,P(O_k|S_j)\;a_{ij}) \qquad (3.5)$$

While the state sequence through the model that maximizes the likelihood of an observation sequence is obtained using the Viterbi algorithm [18]. Figure 3.2 shows an illustration of the data association algorithm.

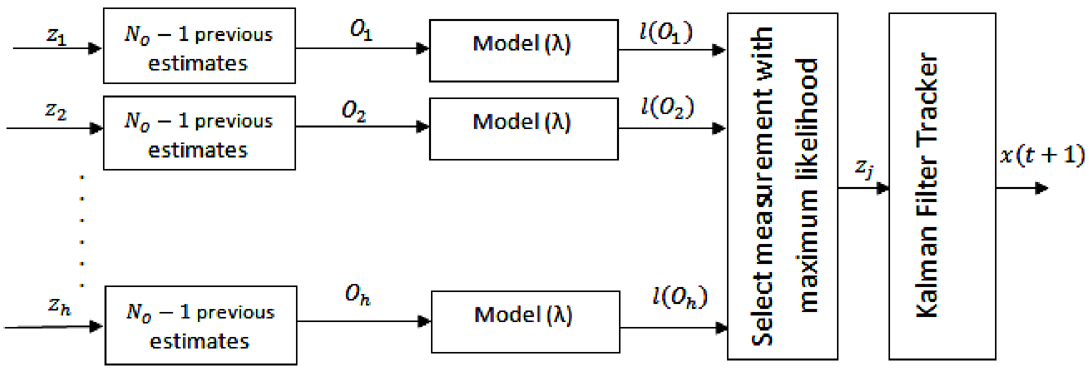

Figure 3.2: Data association algorithm.

3.3.3 Complexity Analysis

In this section the complexity of the algorithm; in terms of required additions, multiplications and comparisons, is analyzed as a function of the number of targets and measurements under consideration.

Let's assume that the number of targets under consideration is n_t and the number of measurements is n_m. The path that maximizes likelihood is determined using Viterbi decoding with a trellis of N_s nodes at each time step (the number of nodes is equivalent to the number of states of the model λ).

For a single target and a single measurement at any time step $2N_s$ multiplications are required to calculate the likelihood of all transitions. $N_s - 1$ comparisons are required to determine the maximum likelihood. However, as likelihood rather than path length is used for comparison, when the decoder terminates the data association metric would be already calculated.

This procedure is repeated for all targets with all measurements, thus the total number of multiplications M and comparisons C required at each time step is given by:

$$M = 2n_t n_m N_s, \tag{3.6}$$

$$C = n_t n_m (N_s - 1). \tag{3.7}$$

Note that in this implementation no additions are required at all. The total number of computations required for data association as a function of the number of targets and measurements is given by:

$$N_c = M + C = n_t n_m (3N_s - 1). \tag{3.8}$$

3.4 Kalman Filter Tracker

3.4.1 State Estimation

The target state and measurement follow the dynamical and measurement models in (3.9) and (3.10).

$$x_i(t+1) = Fx_i(t) + v_i(t) \tag{3.9}$$

$$z_i(t) = Hx_i(t) + w_i(t) \tag{3.10}$$

The Kalman filter is used in target tracking to estimate the position of a target through an assumed dynamical model and noisy measurement data. This is achieved through three stages [27, 29]:

a) Prediction: The state estimate of target i at $t+1$ is predicted from the old state estimate at t using the state transition matrix F that is constructed in accordance with the dynamical model.

$$\hat{x}_i(t+1/t) = F\hat{x}_i(t/t) \tag{3.11}$$

$$P_i(t+1/t) = FP_i(t/t)F^T + Q_i(t) \tag{3.12}$$

b) Correction: In this stage the Kalman filter gain and the innovation are calculated.

$$K_i(t+1) = P_i(t+1/t)H^T[HP_i(t+1/t)\,H^T + R_i(t+1)]^{-1} \tag{3.13}$$

$$\tilde{z}_i(t+1) = z_i(t+1) - H\hat{x}_i(t+1/t) \tag{3.14}$$

c) Update: Finally the state estimate and the error covariance matrix are updated.

$$\hat{x}_i(t+1/t+1) = \hat{x}_i(t+1/t) + K_i(t+1)\tilde{z}_i(t+1) \tag{3.15}$$

$$P_i(t+1/t+1) = [1 - K_i(t+1)H]P_i(t+1/t) \tag{3.16}$$

3.4.2 Non-Maneuvering Target Model

A non-maneuvering target is a target moving in a straight line with constant velocity. The state vector is composed of the x- and y- positions and x- and y- velocities as follows:

$$X(t) = \begin{bmatrix} x(t) \\ v_x(t) \\ y(t) \\ v_y(t) \end{bmatrix} \tag{3.17}$$

White Gaussian noise is added to the true trajectory to compose the target measurements. The state transition matrix F in such case is given by:

$$F = \begin{bmatrix} 1 & \delta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \delta \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.18}$$

where δ is the sampling interval.

The measurements are the x- and y- positions for a target and thus the measurement matrix H is given by:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.19}$$

The noise covariance matrix is given by:

$$R = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \tag{3.20}$$

Where $\sigma_x$and σ_y are the standard deviation of the noise in the x- and y- direction, respectively.

3.4.3 Maneuvering Target Model

The turning motion model is adopted in such case. The state estimate in this case is given by:

$$\hat{x}_i(t+1/t) = F\hat{x}_i(t/t) + Ga_i(t+1) \tag{3.21}$$

where G is the gain matrix and $a_i(t+1)$ is the acceleration matrix at instant $t+1$.

$$a_i = \begin{bmatrix} a_x \\ a_y \end{bmatrix} \tag{3.22}$$

$$G = \begin{bmatrix} \delta^2/2 & 0 \\ \delta & 0 \\ 0 & \delta^2/2 \\ 0 & \delta \end{bmatrix} \tag{3.23}$$

The process noise covariance matrix Q is given by [42]:

$$Q = q^2 \begin{bmatrix} \delta^3/3 & \delta^2/2 & 0 & 0 \\ \delta^2/2 & \delta & 0 & 0 \\ 0 & 0 & \delta^3/3 & \delta^2/2 \\ 0 & 0 & \delta^2/2 & \delta \end{bmatrix} \tag{3.24}$$

Where q is equal to $a\sqrt{\delta}$.

The acceleration in the x- and y- direction is calculated by:

$$a_{x,y}(t+1) = \frac{v_{x,y}(t+1) - v_{x,y}(t)}{\delta} \tag{3.25}$$

3.5 Multisensor Estimate Fusion Based on Bayesian MMSE

The implemented estimate fusion algorithm combines tracks from various sensors to obtain a global estimate of the state of a target. In our implementation, the cross-correlation between estimation errors from different systems is taken into consideration. Also, feedback of the global estimate into local trackers is implemented and its effect is analyzed in section 3.5.2. Figure 3.3 shows an overview of the estimate fusion system.

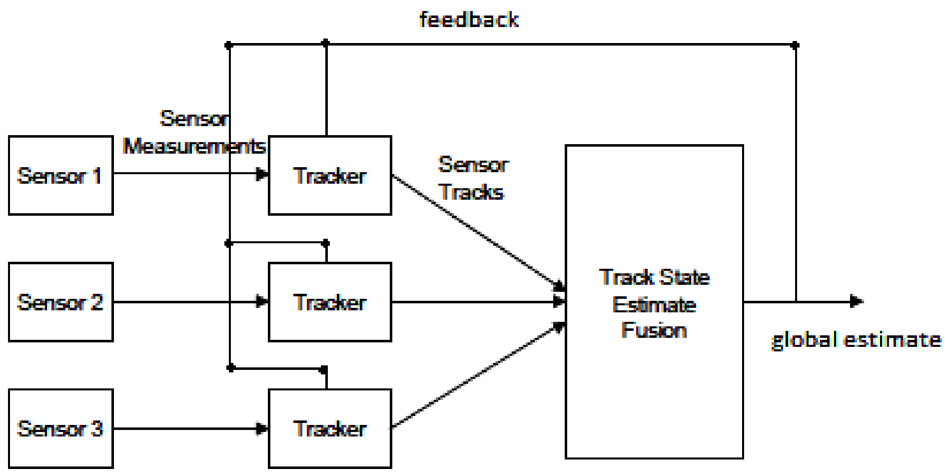

Figure 3.3: Overview of the estimate fusion system.

3.5.1 Cross-Covariance

As established in [42, 45], the estimation errors of two tracks of the same target but from different sensors are correlated, because while the measurement noise of the two sensors can be safely assumed independent, the same process noise in the dynamic model yields the correlation of the estimation errors.

The covariance of the difference d_{ij} between two estimates form sensors i and j is given by:

$$E\left[d_{ij}d_{ij}^T\right] = E[\left(\hat{x}_i - \hat{x}_j\right)\left(\hat{x}_i - \hat{x}_j\right)^T] \tag{3.26}$$

The covariance can be rewritten as;

$$E\left[d_{ij}d_{ij}^T\right] = E[\left(\hat{x}_i - x - (\hat{x}_j - x)\right)\left(\hat{x}_i - x - (\hat{x}_j - x)\right)^T]$$

$$= P_i + P_j - P_{ij} - P_{ji} \tag{3.27}$$

where P_{ij} represents the cross-correlation between the two estimates and is given by:

$$P_{ij} = E\left[(\hat{x}_i - x)\left(\hat{x}_j - x\right)\right] = E\left[\tilde{x}_i\tilde{x}_j\right] = P_{ji} \tag{3.28}$$

In this case, the results of the fused estimate and the corresponding covariance which minimize the mean square error (MSE) will be [22, 42, 46-46]:

$$X_f = \hat{x}_i + \left(P_i - P_{ij}\right)\left(P_i + P_j - P_{ij} - P_{ij}^T\right)^{-1}\left(\hat{x}_j - \hat{x}_i\right) \tag{3.29}$$

$$P = P_i - \left(P_i - P_{ij}\right)\left(P_i + P_j - P_{ij} - P_{ij}^T\right)\left(P_i - P_{ij}^T\right) \tag{3.30}$$

where P_{ij} is determined by the following recursive equation:

$$P_{ij} = \left(1 - K^i H\right)\left(F P_{ij} F^T + Q\right)(1 - K^j H) \tag{3.31}$$

3.5.2 Feedback

First the effect of feedback on the fused estimate and the global estimation error is presented as follows:

When feedback is performed, the local sensor predictions are given by:

$$\hat{x}_i^f(t+1/t) = F X_f \tag{3.32}$$

$$P_i^f(t+1/t) = F P(t/t) F^T + Q = P(t+1/t) \tag{3.33}$$

The Kalman filter gain and the innovation can be easily computed as:

$$K_i^f(t+1) = P(t+1/t) H^T [H P(t+1/t) H^T + R_i]^{-1} \tag{3.34}$$

$$\tilde{z}_i^f(t+1) = z_i(t+1) - H F X_f \tag{3.35}$$

The local state estimate and error covariance are given by:

$$\hat{x}_i^f(t+1/t+1) = F X_f + (P(t+1/t) H^T [H P(t+1/t) H^T + R_i]^{-1}) \tag{3.36}$$

$$P_i^f(t+1/t+1) = [1 - P(t+1/t) H^T [H P(t+1/t) H^T + R_i]^{-1}] P(t+1/t) \tag{3.37}$$

It is logical to state that the covariance of the global estimate is lower than the covariance of the local estimate without feedback because more information is contained in the global estimate, i.e. $P(t+1/t) < P_i(t+1/t)$. This statement yields;

$$P_i^f(t+1/t+1) < P_i(t+1/t+1) \tag{3.38}$$

Thus, feedback improves local tracking performance which in turn affects the global fused estimate without altering any of the fusion formulae at the fusion center.

3.5.3 Sensors

Two important aspects in multisensor estimate fusion are: (1) sensors' accuracies, and (2) number of sensors.

It is safe to say that as a single sensor's accuracy increases, the overall performance of the tracking algorithm improves. However, the problem is whether it is more adequate to use sensors with similar or close accuracies, or variations in sensor accuracies have no severe effect on overall performance. Another problem is how to determine the number of sensors to be used, as a large number will result in a huge processing load in the fusion center and also a vast amount of communications into and out of the fusion center from local trackers. Nevertheless, a small number of sensors might not improve performance significantly to justify carrying out the fusion process at all. The answers to these questions are presented in the next chapter, where different cases are simulated to show the effect of the number and accuracies of sensors on the error performance.

3.6 Concluding Remarks

The proposed system is in three stages. The first stage is the HMM-based data association, the second stage is state estimation using linear Kalman filter. Finally, estimate fusion among similar and dissimilar sensors is implemented using Kalman filter fusion with the assumption of correlated estimation errors and with feedback of the global estimate into local trackers. The performance of the proposed algorithm is evaluated using two different target models; non-maneuvering and maneuvering target models.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

Based on the described model in Chapter 3, MATLAB version 7.14 (R2012a) and HTK version 2.2 were used to simulate the proposed tracking algorithm.

All the displayed results are based on a 300 run Monte Carlo simulation. Two examples are considered, two crossing non-maneuvering targets and two crossing maneuvering targets. The performance of the data association algorithm is illustrated in the second section. The third section focuses on results due multisensor fusion and the effect of the number of sensors and sensor accuracies on the results.

4.2 HMM-Based Data Association

4.2.1 Non-Maneuvering Targets

The first case of two non-maneuvering crossing targets is simulated, the target trajectories and the measured tracks are shown in Figure 4.1. The standard deviations in eq. (3.20) of both target measurements are taken as $\sigma_{x1} = \sigma_{y1} = 100\ m$, $\sigma_{x2} = \sigma_{y2} = 150\ m$.

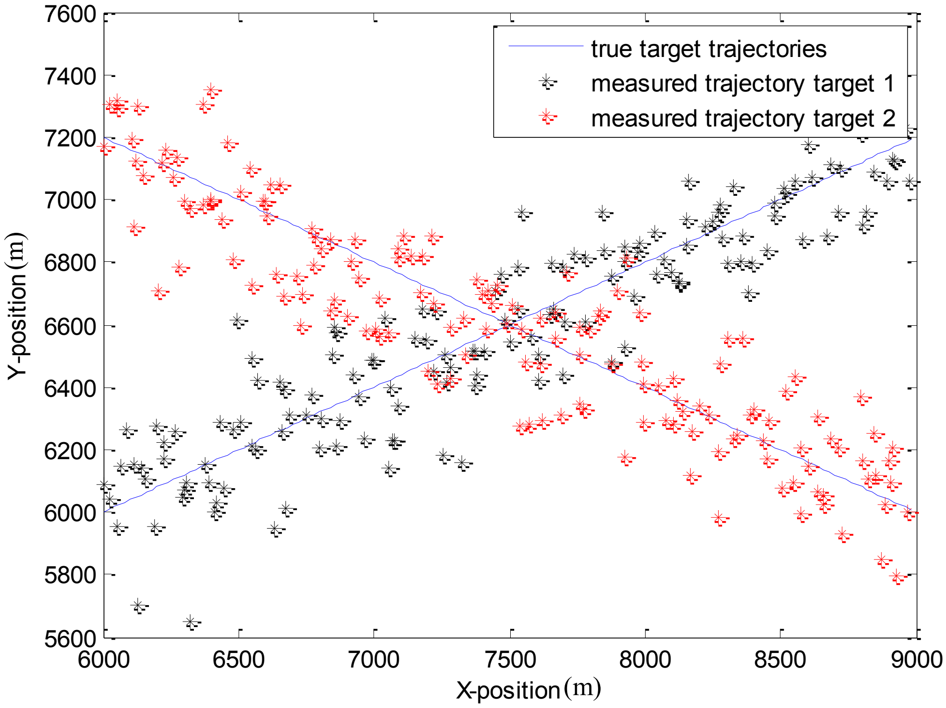

Figure 4.1: True and measured target trajectories for non-maneuvering targets.

Data association is performed based on the proposed HMM approach, followed by Kalman filter tracking. NNSF is also simulated and the results of both methods are shown in Figures 4.2 and 4.3. The results show better tracking performance in case of data association based on the proposed HMM approach. The estimated tracks in the case of HMM-based association are much closer to the true target trajectories.

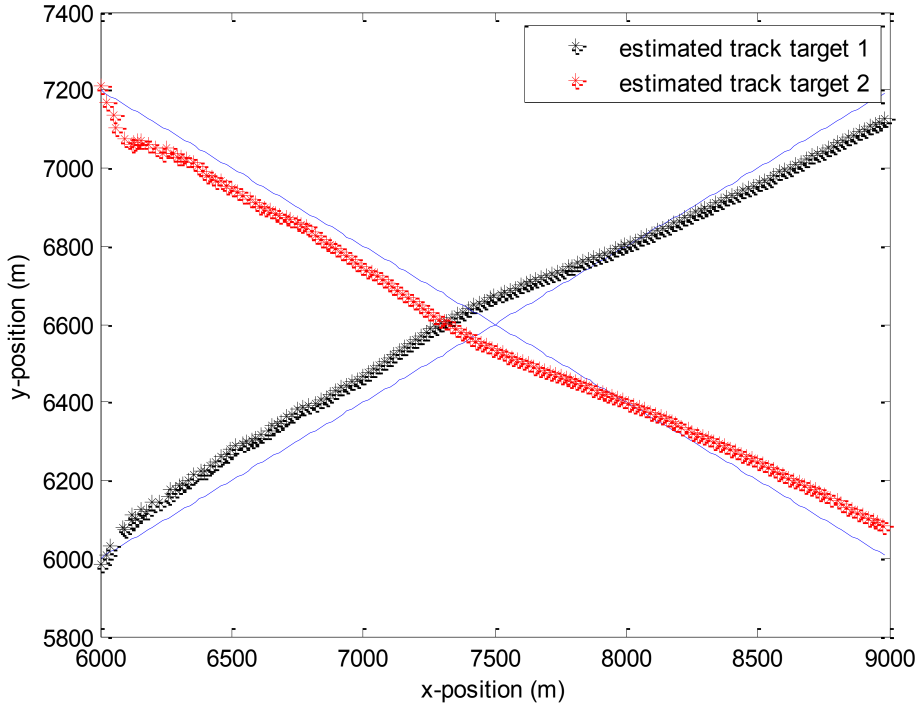

Figure 4.2: Estimated target tracks with NNSF.

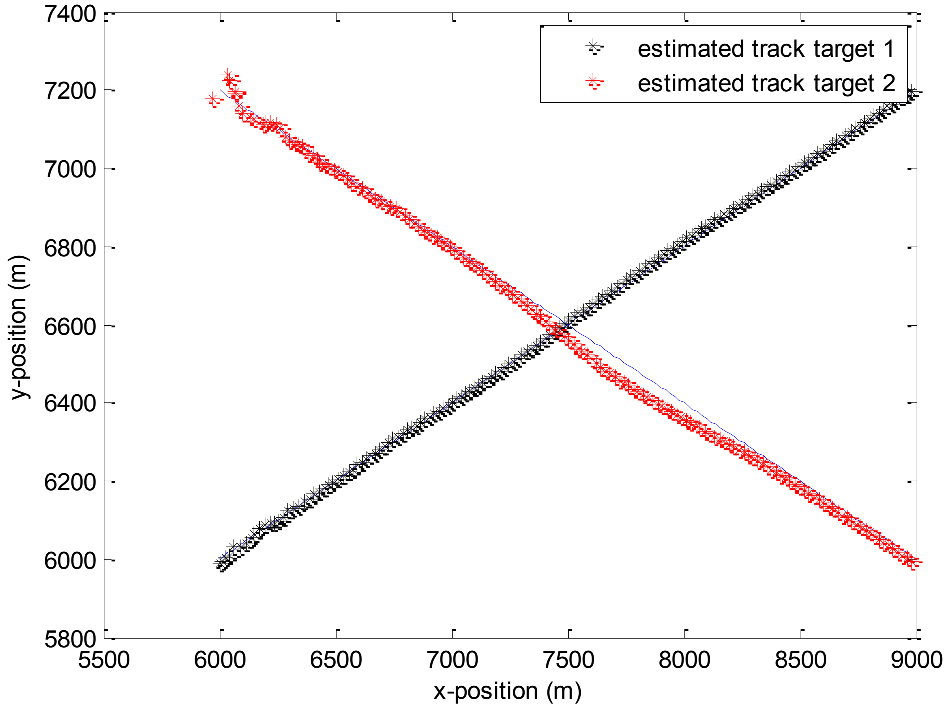

Figure 4.3: Estimated target tracks based on HMM data association.

Performance of both methods based on error calculation using eq. (4.1) is shown in Figure 4.4.

$$e = \sqrt{e_x^2 + e_y^2} = \sqrt{(x_{true} - x_{estimate}) + (y_{true} - y_{estimate})} \quad (4.1)$$

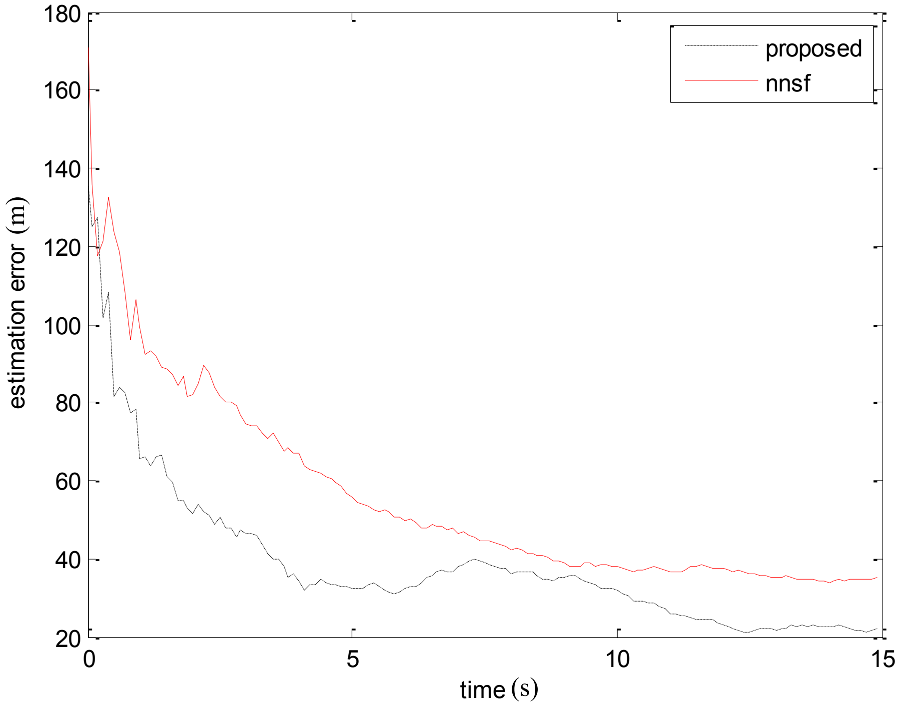

Figure 4.4: Estimation error for NNSF and HMM-based association for target 1.

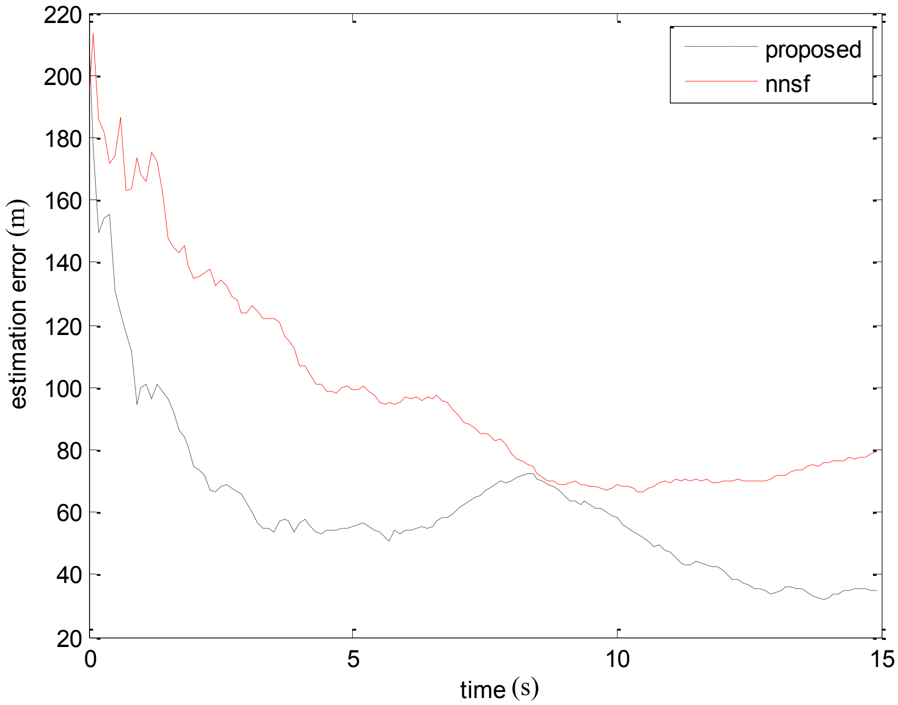

Figure 4.5: Estimation error for NNSF and HMM-based association for target 2.

From Figures 4.2-4.5 it is apparent that the proposed data association approach based on a HMM provides better performance than NNSF. The estimation error in the case of HMM-based association are lower than in the case of NNSF-based association for both targets at all times. At the crossing point of the two targets the HMM-based approach's performance deteriorates due to the targets being closer to each other, however, it decreases again after the crossing point.

4.2.2 Maneuvering Targets

The second example considers the case of two moving targets with acceleration, i.e. maneuvering targets with turn. The dynamical and measurement models are given by eq.s' (3.8) and (3.9), respectively. Figure 4.6 shows the actual and measured target trajectories. The standard deviations of both targets noisy measurements are $\sigma_{x1} = \sigma_{y1} = 50\ m$, $\sigma_{x2} = \sigma_{y2} = 75\ m$.

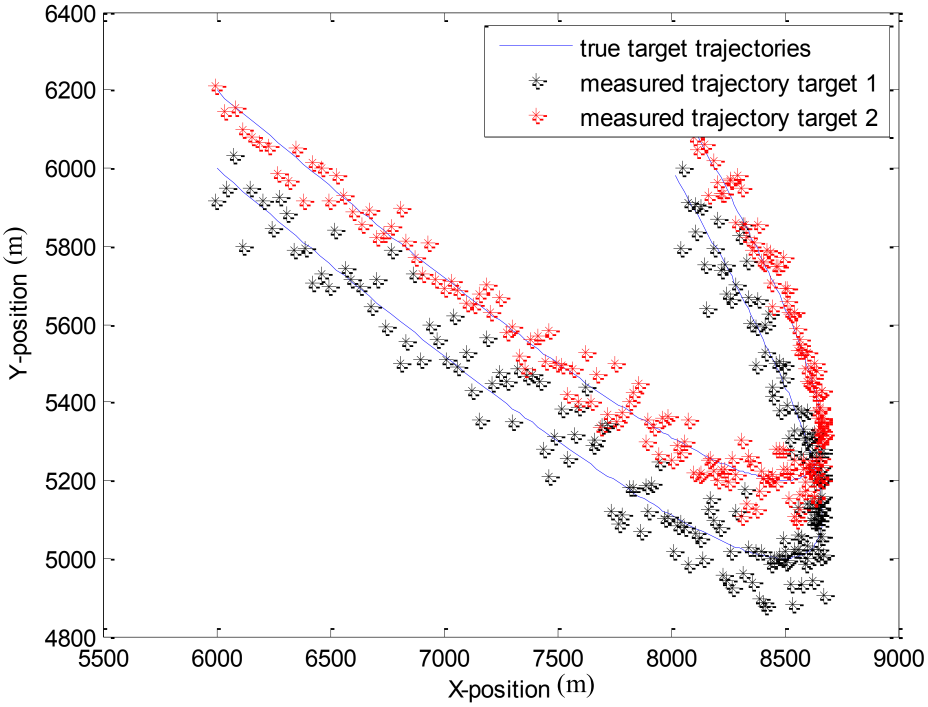

Figure 4.6: True and measured target trajectories for maneuvering targets.

The estimated tracks with NNSF and HMM-based association are shown in Figures 4.7 and 4.8, respectively. A comparison of the estimation errors of both methods is illustrated in Figure 4.9.

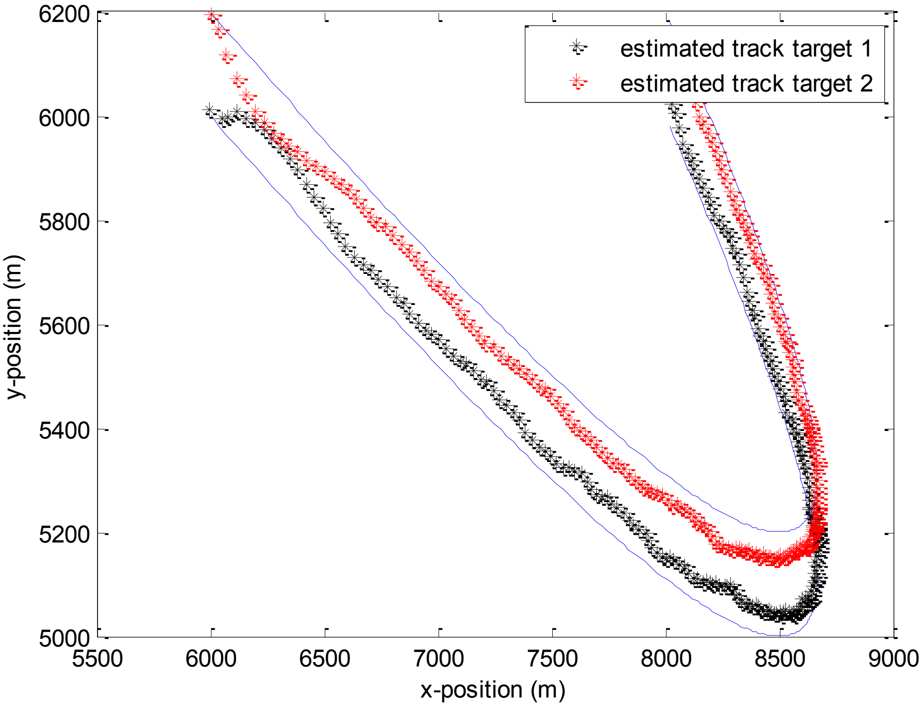

Figure 4.7: Estimated tracks based on NNSF.

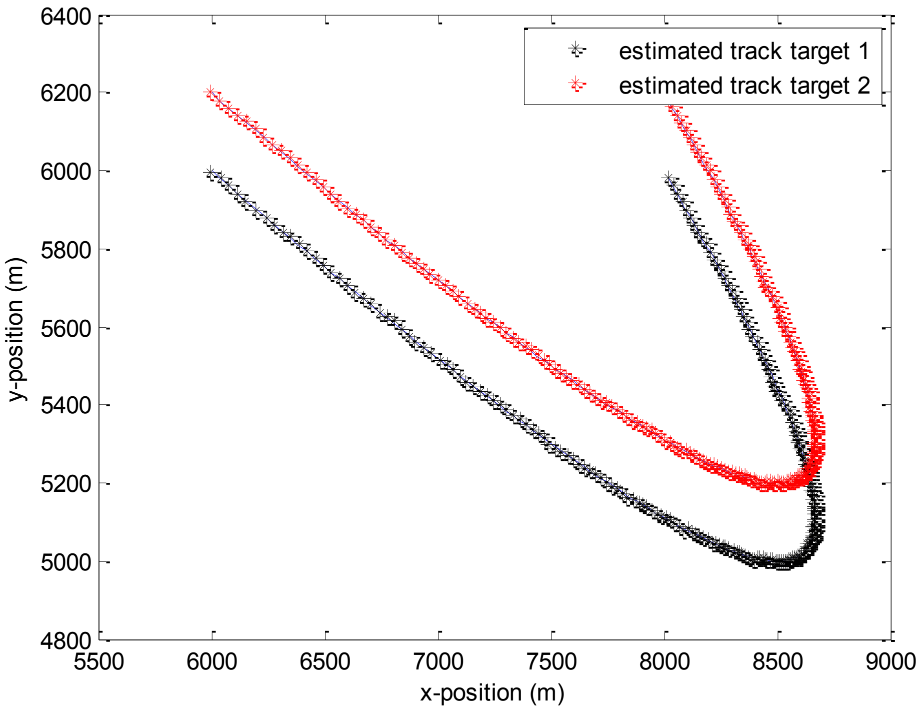

Figure 4.8: Estimated tracks based on HMM.

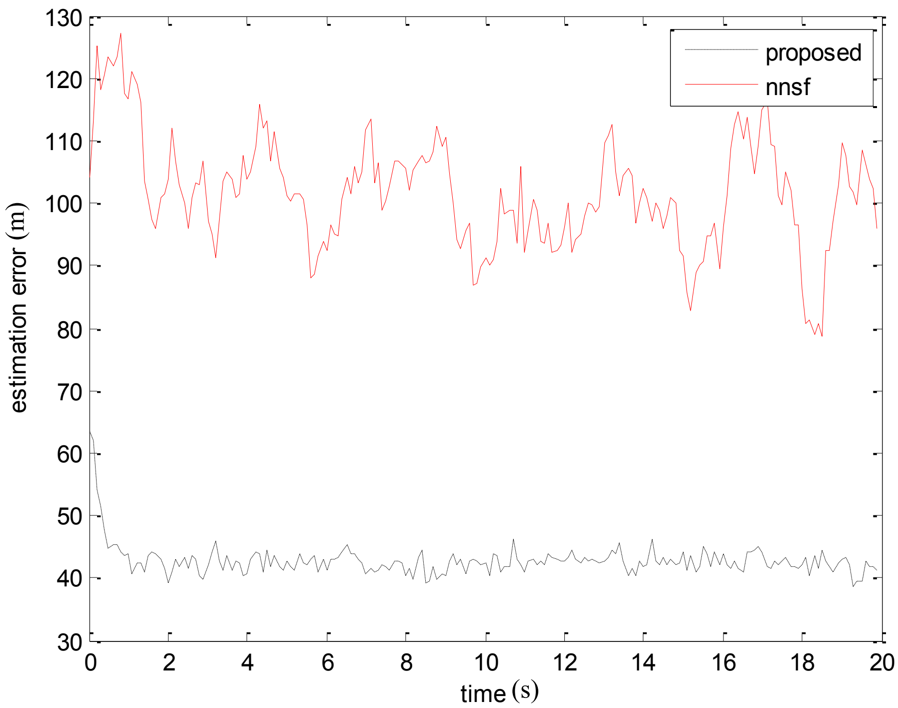

Figure 4.9: Estimation error for maneuvering targets with both methods.

Figures 4.7-4.9 show the superior performance of the proposed data association technique over the NNSF in the case of maneuvering targets. As opposed to the NNSF, the performance of the HMM-based association improves from non-maneuvering to maneuvering targets.

An apparent advantage is the error being nearly constant at all times as it reaches its steady state performance quicker than other techniques due to the low complexity of the algorithm.

4.2.3 Comparison with Perfect Association

In order to further upraise performance, tracking with perfect data association is simulated in Figures 4.10 and 4.11, and comparisons between estimation errors with perfect association and HMM-based association for non-maneuvering and maneuvering targets are shown in Figures 4.12 and 4.13, respectively.

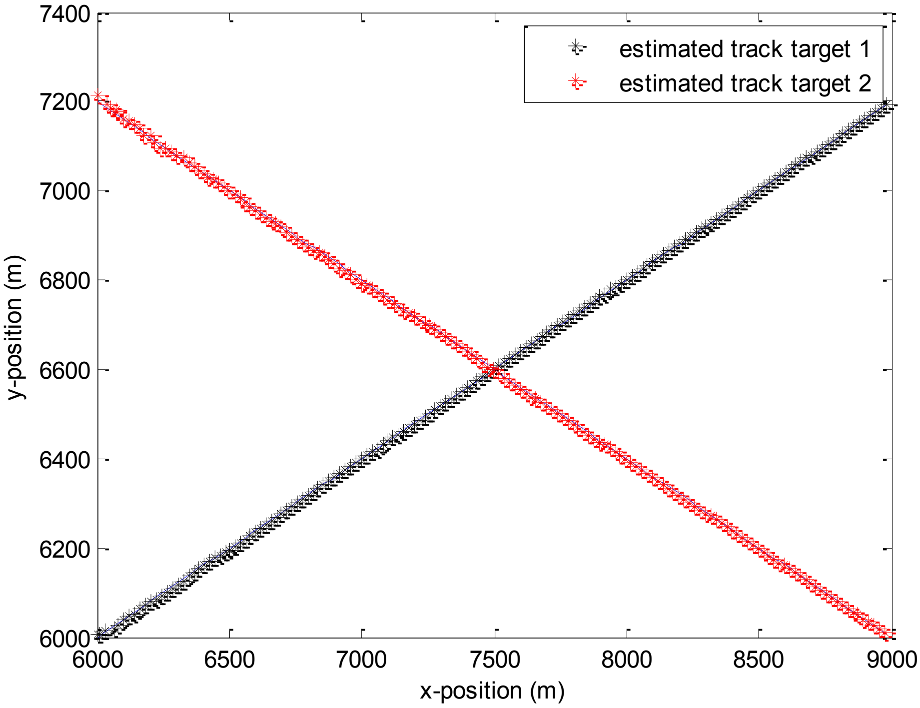

Figure 4.10: Estimated tracks with perfect association for non-maneuvering targets.

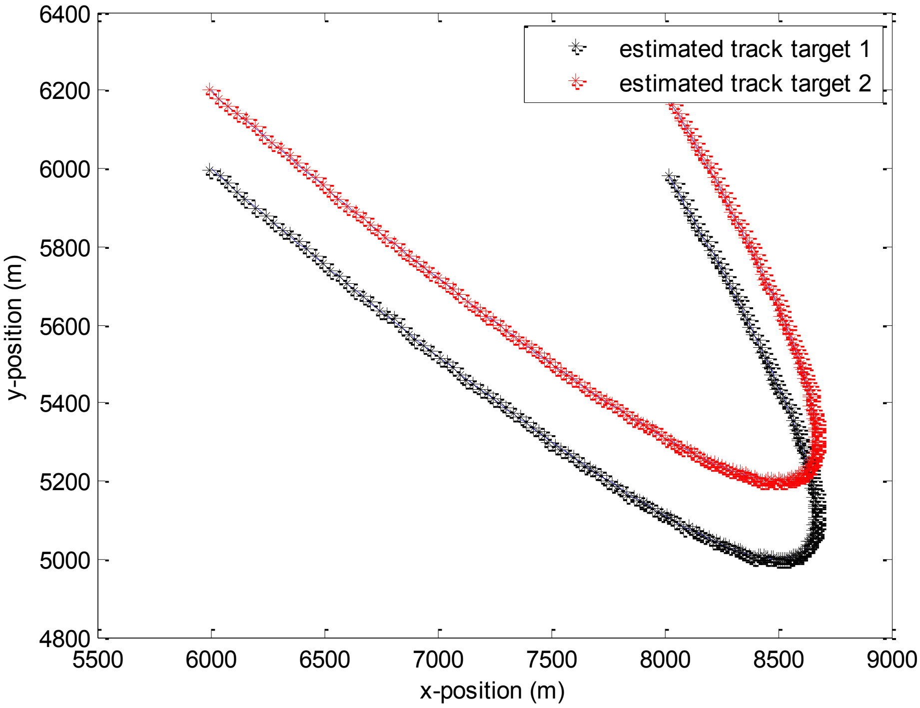

Figure 4.11: Estimated tracks with perfect association for maneuvering targets.

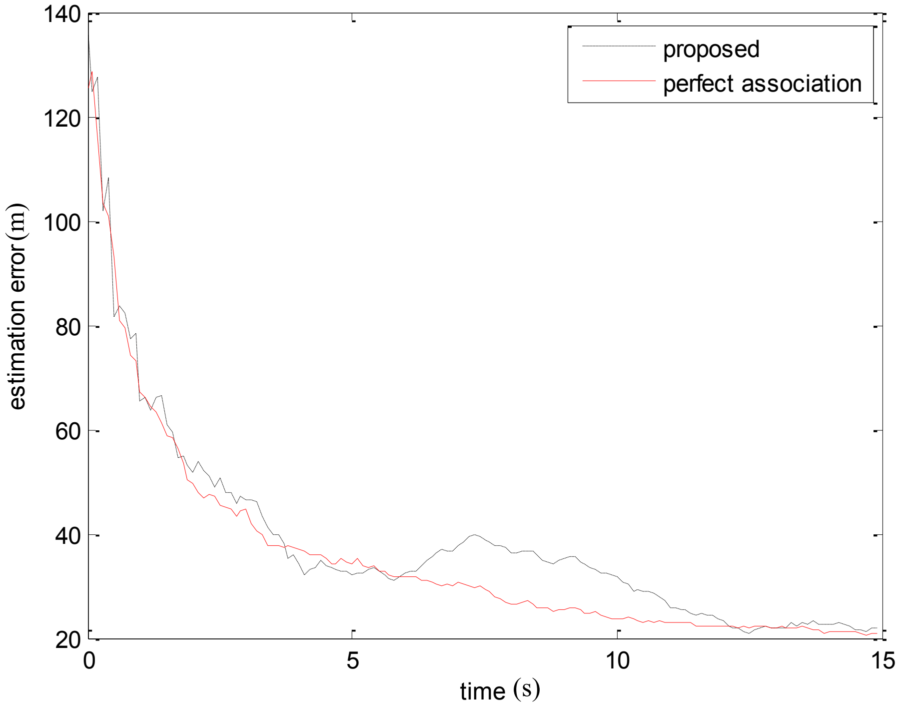

Figure 4.12: Estimation error for non-maneuvering target with perfect association and the proposed association approach.

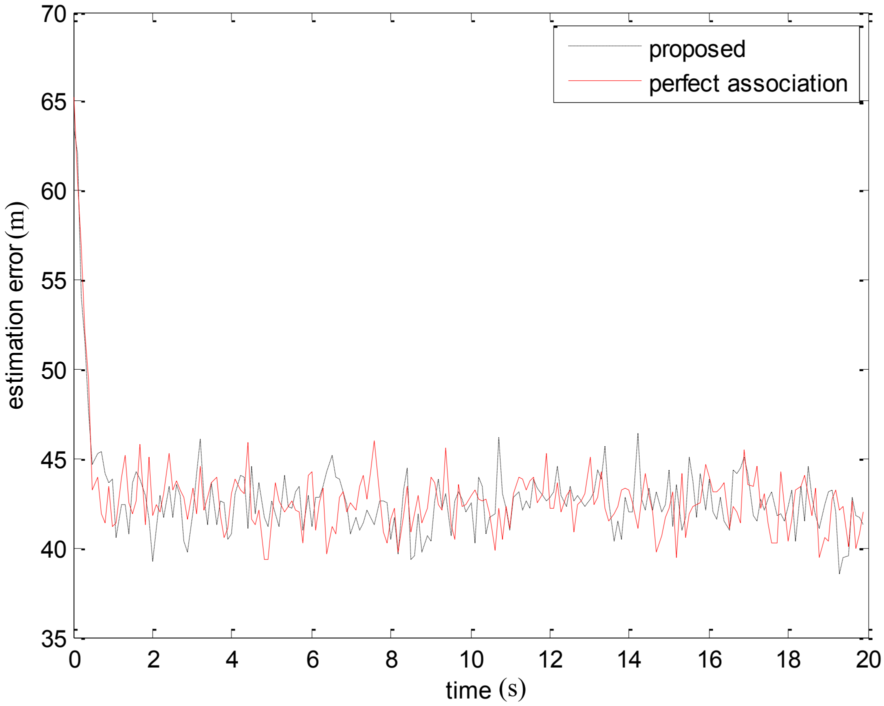

Figure 4.13: Estimation error for maneuvering targets with perfect association and the proposed association approach.

Figures 4.12 and 4.13 show the proximity of the performance of the proposed tracking algorithm to tracking based on perfect association. In terms of error performance the proposed association approach performs almost as good as perfect association especially in the case of maneuvering targets. Results obtained based on perfect data association exclude any factors related to wrong associations among targets and includes only the effect of the estimation method used. This provides a good reference line for comparison.

4.2.4 Effect of the Standard Deviation on Performance

It is intuitional that as the noise standard deviation increases, the estimation error increases due to higher noise levels. The effect of the noise appears both in association phase and also during tracking. Various simulations showing the drift in the error performance from that of the perfect association case when the noise standard deviation has increased are presented in this section. Considering the perfect association error performance as a reference for comparison, partially eliminates the tracking algorithm effect and highlights the data association performance under different conditions.

Figure 4.13 showed the case when the standard deviation is taken as $\sigma_{x1} = \sigma_{y1} = 50\ m$, $\sigma_{x2} = \sigma_{y2} = 75\ m$. In the next example shown in Figure 4.13 the values of the standard deviation are $\sigma_{x1} = \sigma_{y1} = 100\ m$, $\sigma_{x2} = \sigma_{y2} = 100\ m$. Increasing the noise even further so that $\sigma_{x1} = \sigma_{y1} = 200\ m$, $\sigma_{x2} = \sigma_{y2} = 200\ m$, results in the estimation error shown in Figure 4.14.

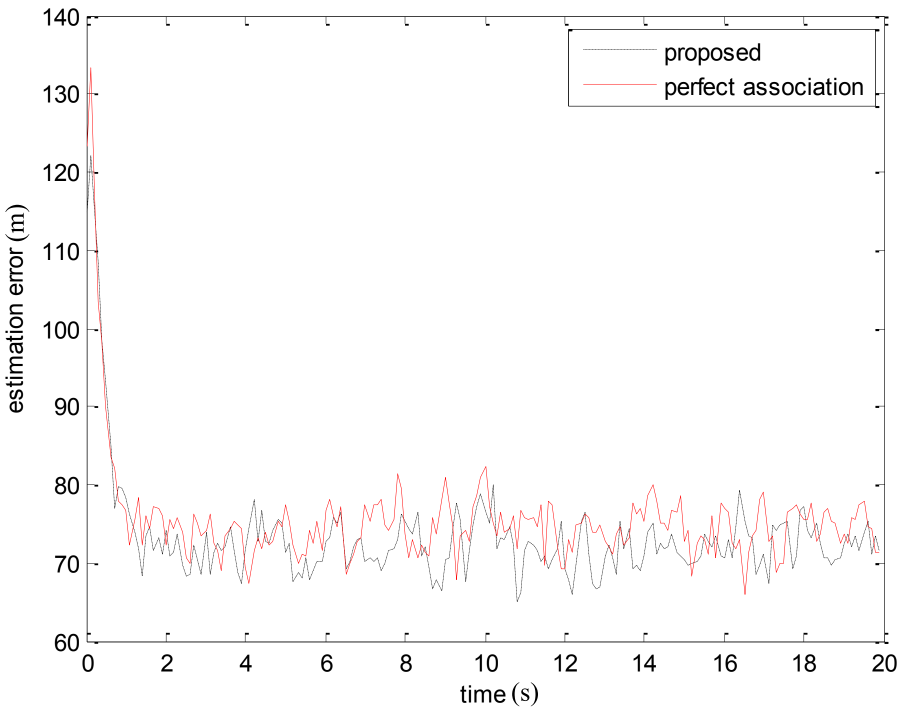

Figure 4.14: Estimation errors for perfect association and HMM-based association when $\sigma = 100$.

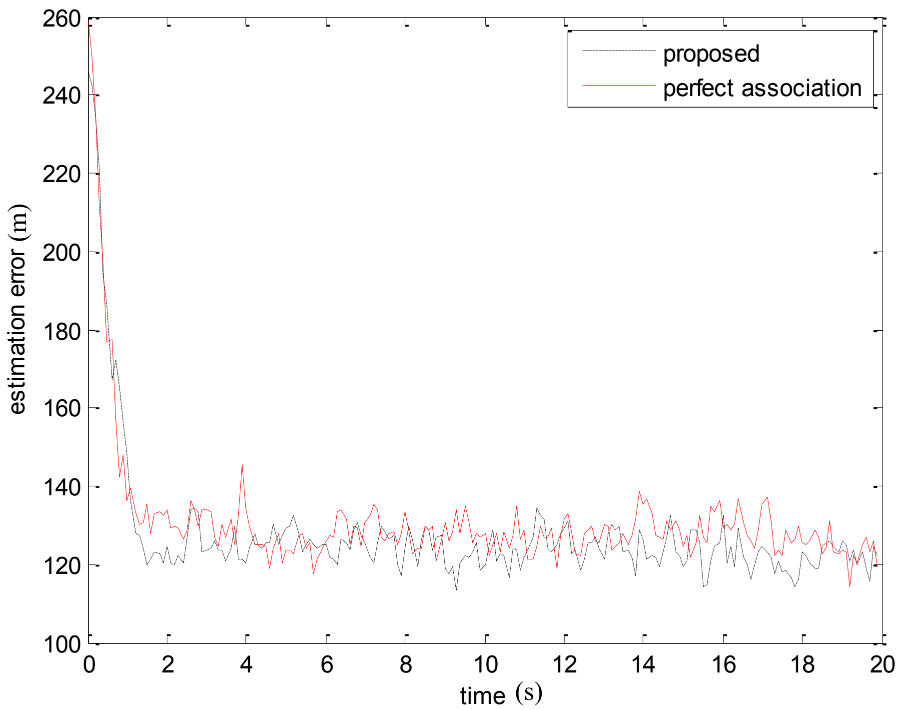

Figure 4.15: Estimation errors for perfect association and HMM-based association when $\sigma = 200$.

Figures 4.13-4.15 show the robustness of the HMM-based data association technique, as the drift from perfect association performance is insignificant with the increase in σ. It is clear that the estimation error increases as the noise level increases.

4.2.5 Computational Complexity

An expression for the computational complexity of the data association phase in terms of number of calculations performed as a function of the number of targets and measurements under consideration is given by eq. (3.7). Table 4.1 shows a comparison between the complexity of the algorithm and other methods reported in the literature [5].

Table 4.1: Number of operations required for data association for various techniques.

n_t	n_m	Standard JPDA	Cheap JPDA	All-Neighbor Fuzzy Association	HMM-based Association
3	6	998	546	216	252
3	7	1,598	804	252	294
4	4	1,155	564	240	224
4	6	6,375	1,824	360	336
5	6	31,204	4,200	540	420
6	7	358,265	14,646	882	588

From the above results, it is obvious that the HMM-based data association has a lower number of operations than other association approaches that are based on combining weighted measurements. This results in higher computational feasibility and faster performance.

For small values of n_t and n_m only the all neighbor fuzzy association has lower complexity than the HMM-based association and even in this case the difference in the number of operations required by both techniques is not big. Generally, the proposed data association approach has lower computational complexity than the cheap JPDA, and thus lower than the JPDA, for any number of targets and measurements.

For larger values of n_t and n_m the HMM-based technique requires a lower number of operations than all of the other techniques shown in the table. This is mainly because the increase in the number of operations is a linear function of n_t and $n_m$and thus increases less rapidly than in the other techniques. The closer the number of targets is to the number of measurements, the better the computational complexity is. The most efficient performance regarding complexity is achieved when the number of targets is equal to the number of measurements.

4.3 Multisensor Data Fusion Results

Multisensor data fusion is performed based on Bayesian MMSE criterion and assuming correlated estimation errors for local trackers. The results displayed for non-maneuvering and maneuvering targets with similar and dissimilar sensors. The effect of the variation of the sensors' accuracies and the number of sensors on performance is also presented. Note that at the local trackers data association is performed based on a HMM and tracking is carried out by the Kalman filter tracking algorithm.

4.3.1 Performance of the Multisensor Fusion

The first case simulated is that of five identical sensors with standard deviation $\sigma_{s1} = \sigma_{s2} = \cdots = \sigma_{s5} = 100m$. Figure 4.16 shows the estimation error in the case of a single sensor and multiple identical sensors. In this case, the multisensor estimate fusion improves performance significantly.

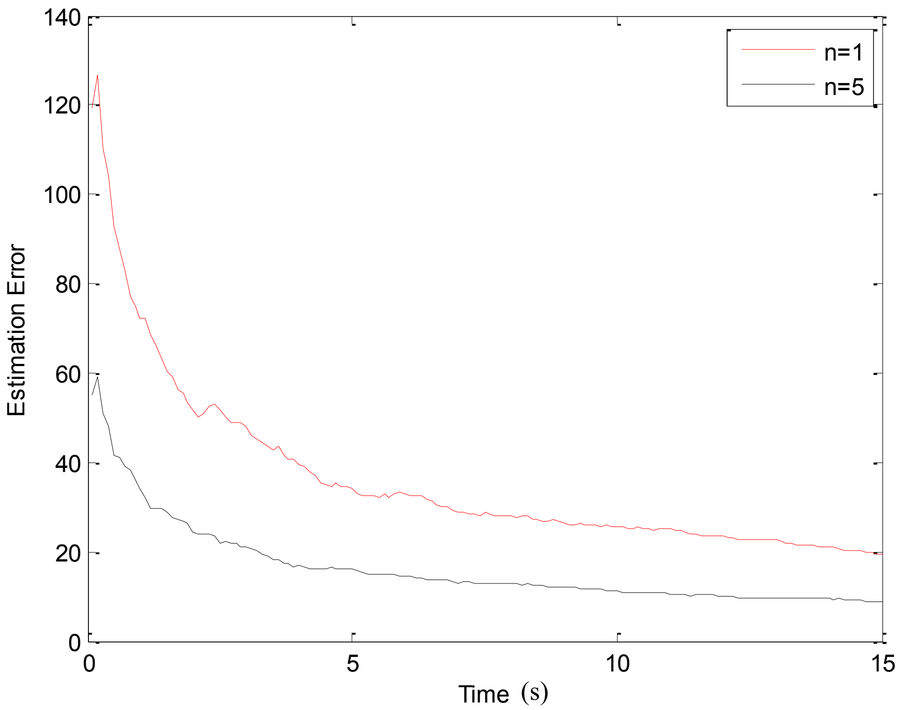

Figure 4.16: Estimation error in case of a single sensor and multiple identical sensors for a non-maneuvering target.

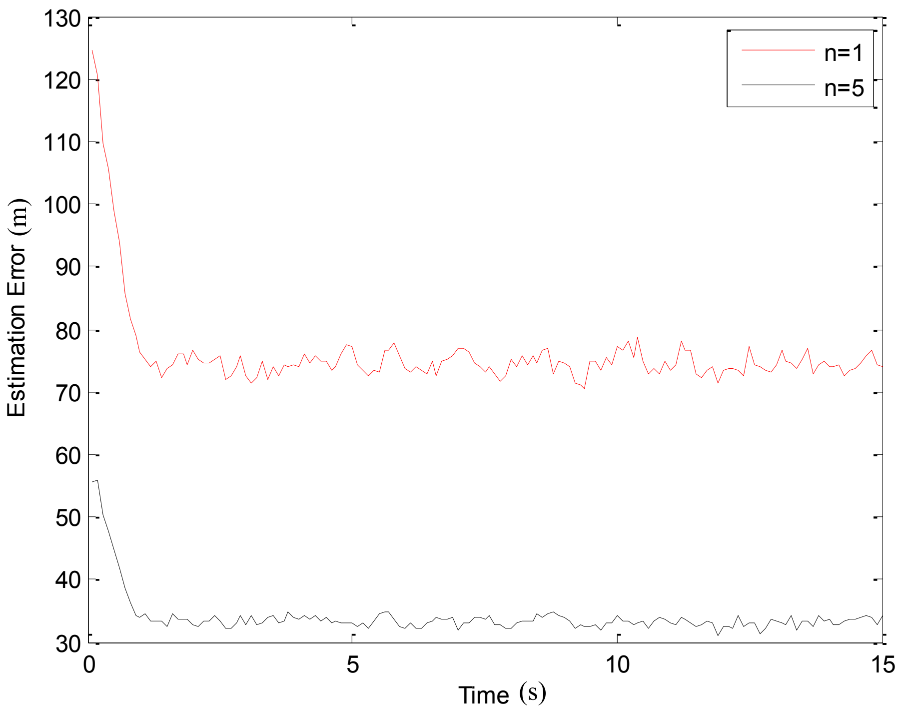

Figure 4.17: Estimation error in case of a single sensor and multiple identical sensors for a maneuvering target.

4.3.2 The Effect of the Number of Sensors on Performance

To deduce the effect of the number of sensors used to perform fusion, estimation errors for various number of sensors from $n = 1$ to $n = 5$ are shown in Figures 4.18 and 4.19 for both non-maneuvering and maneuvering targets, respectively. The sensors' accuracies in this case are identical $\sigma_{sn} = 100m$.

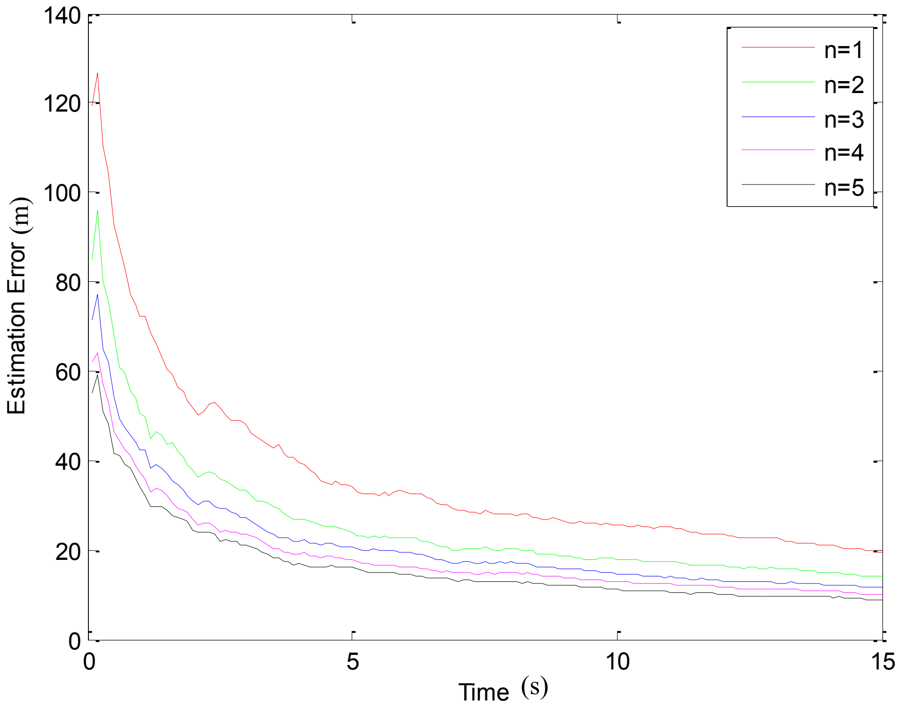

Figure 4.18: Estimation error in case of multiple identical sensors for a non-maneuvering target.

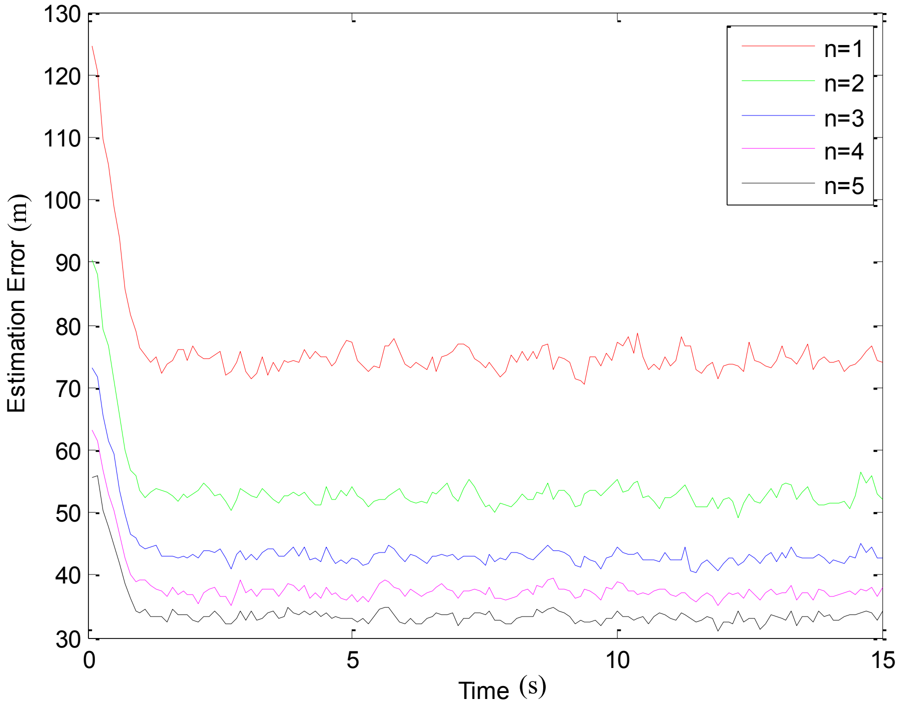

Figure 4.19: Estimation error in case of multiple identical sensors for a maneuvering target.

As the number of sensors increases the performance of multisensor fusion improves for both non-maneuvering and maneuvering target. However, the change is more significant between two to three sensors than when increasing the number of sensors from three to four sensors and so on. The improvement becomes less and less significant as we increase the number of sensors even further. As we can see from Figures 4.18-4.19 that increasing the number of sensors from four to five sensors resulted in a minor improvement in estimation error. At a certain point it would be unwisely to increase the number of sensors used.

4.3.3 The Effect of Sensor Accuracies on Performance

Figures 4.20-4.23 show estimation errors for various cases of non-identical sensor accuracies.

- In the first example the sensor accuracies are $\sigma_{s1} = 100,\ \sigma_{s2} = 125,\ \sigma_{s3} = 150,\ \sigma_{s4} = 200$ and $\sigma_{s5} = 300\ m$. The results are shown in Figure 4.20.
- In the second example the sensor accuracies are $\sigma_{s1} = 100,\ \sigma_{s2} = 150,\ \sigma_{s3} = 200,\ \sigma_{s4} = 300$ and $\sigma_{s5} = 500\ m$. The results are shown in Figure 4.21.

- In the third example the sensor accuracies are $\sigma_{s1} = 100$, $\sigma_{s2} = 175$, $\sigma_{s3} = 250$, $\sigma_{s4} = 400$ and $\sigma_{s5} = 700\ m$. The results are shown in Figure 4.22.
- In the fourth example the sensor accuracies are $\sigma_{s1} = 100$, $\sigma_{s2} = 200$, $\sigma_{s3} = 300$, $\sigma_{s4} = 500$ and $\sigma_{s5} = 900\ m$. The results are shown in Figure 4.23.

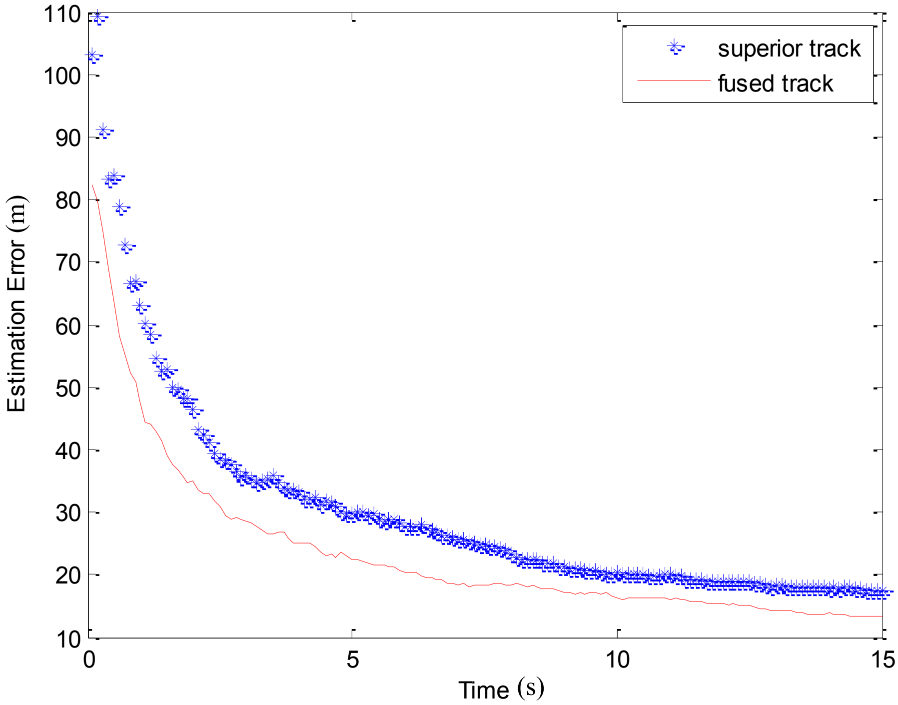

Figure 4.20: Estimation error for non-identical sensors and the superior track $\sigma_{s1} = 100, \sigma_{s2} = 125, \sigma_{s3} = 150, \sigma_{s4} = 200\ and\ \sigma_{s5} = 300\ m.$

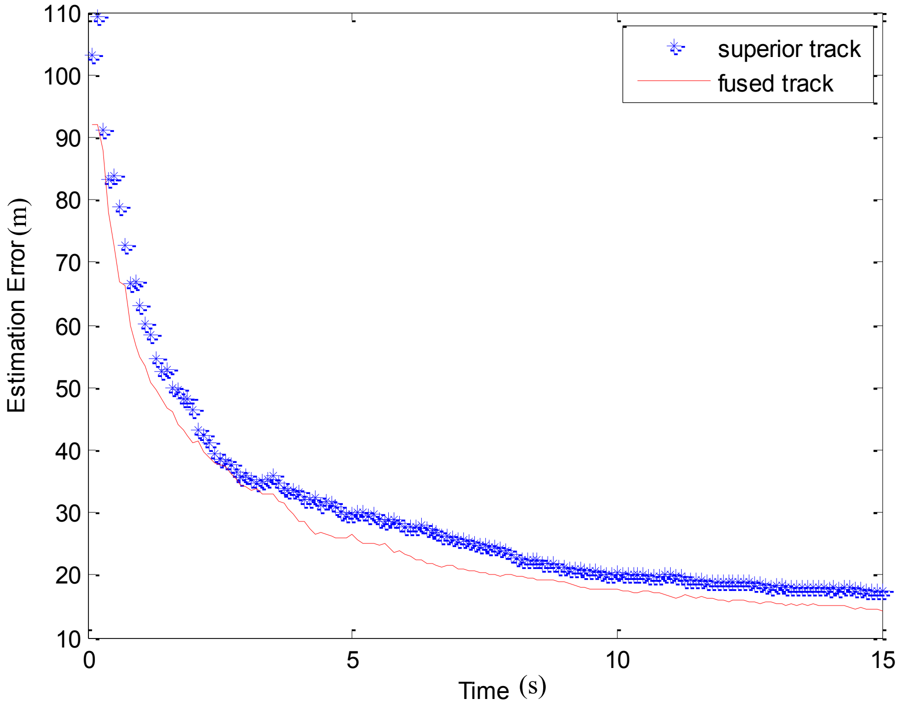

Figure 4.21: Estimation error for non-identical sensors and the superior track $\sigma_{s1} = 100$ $\sigma_{s2} = 150, \sigma_{s3} = 200, \sigma_{s4} = 300\ and\ \sigma_{s5} = 500\ m.$

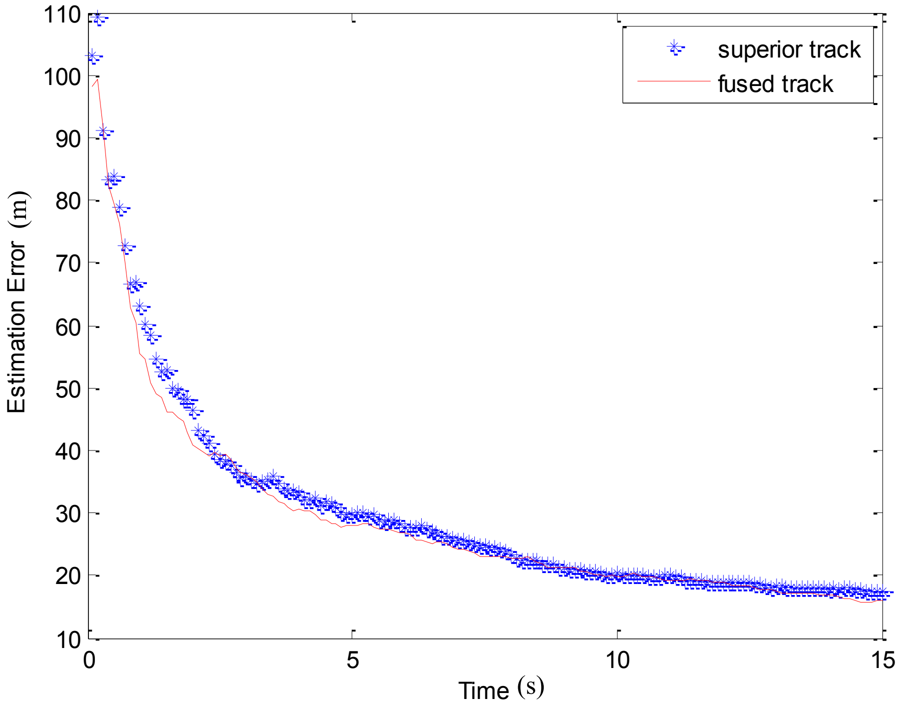

Figure 4.22: Estimation error for non-identical sensors and the superior track $\sigma_{s1} = 100$ $\sigma_{s2} = 175, \sigma_{s3} = 250, \sigma_{s4} = 400\ and\ \sigma_{s5} = 700\ m.$

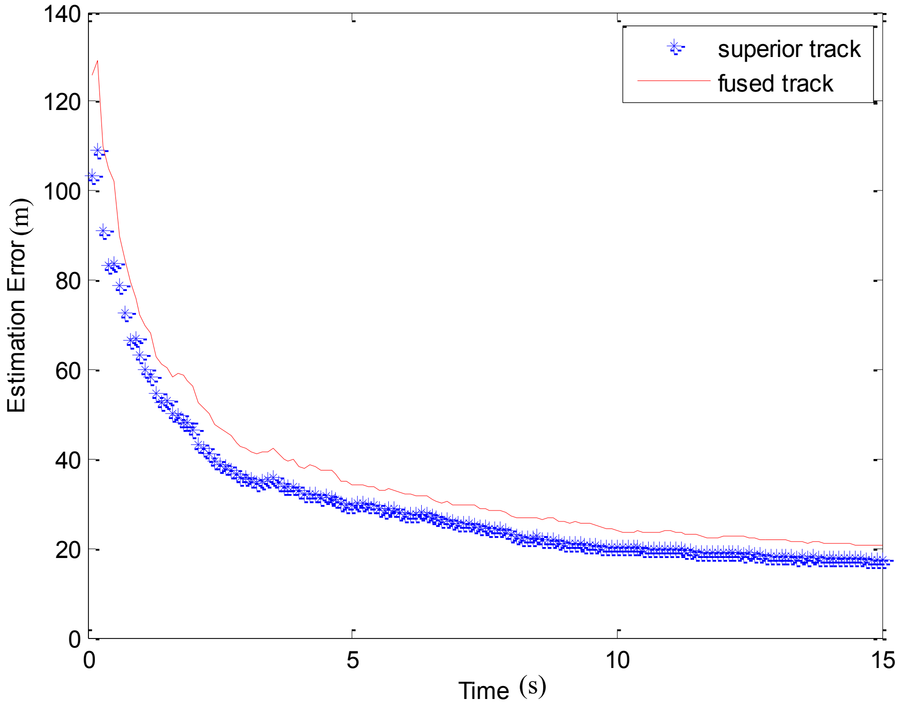

Figure 4.23: Estimation error for non-identical sensors and the superior track $\sigma_{s1} = 100$ $\sigma_{s2} = 200, \sigma_{s3} = 300, \sigma_{s4} = 500$ *and* $\sigma_{s5} = 900\ m$.

As illustrated by Figures 4.20-4.23, as variations between sensor accuracies increases, overall performance is degraded. In the first two cases the fused track performance is still better than the performance of the track with the highest accuracy (least σ_s). Increasing the variations even further in the third case, the fused track and the superior track have almost the same estimation errors. In the last case in Figure 4.23 the superior track performance beats the fused track performance.

4.4 Large Number of Targets

So far all the simulated examples contained only two targets. However, in real life situations the number of targets may vary. It is safe to say that a larger number of targets would be harder to handle and that performance will degrade as the number of targets increases. The question is whether the algorithm would still show acceptable performance in the presence of more than two targets. Figure 4.24 shows the true and measured trajectories of five non-maneuvering targets. The standard deviation of the measurement noise of the targets is shown in Table 4.2.

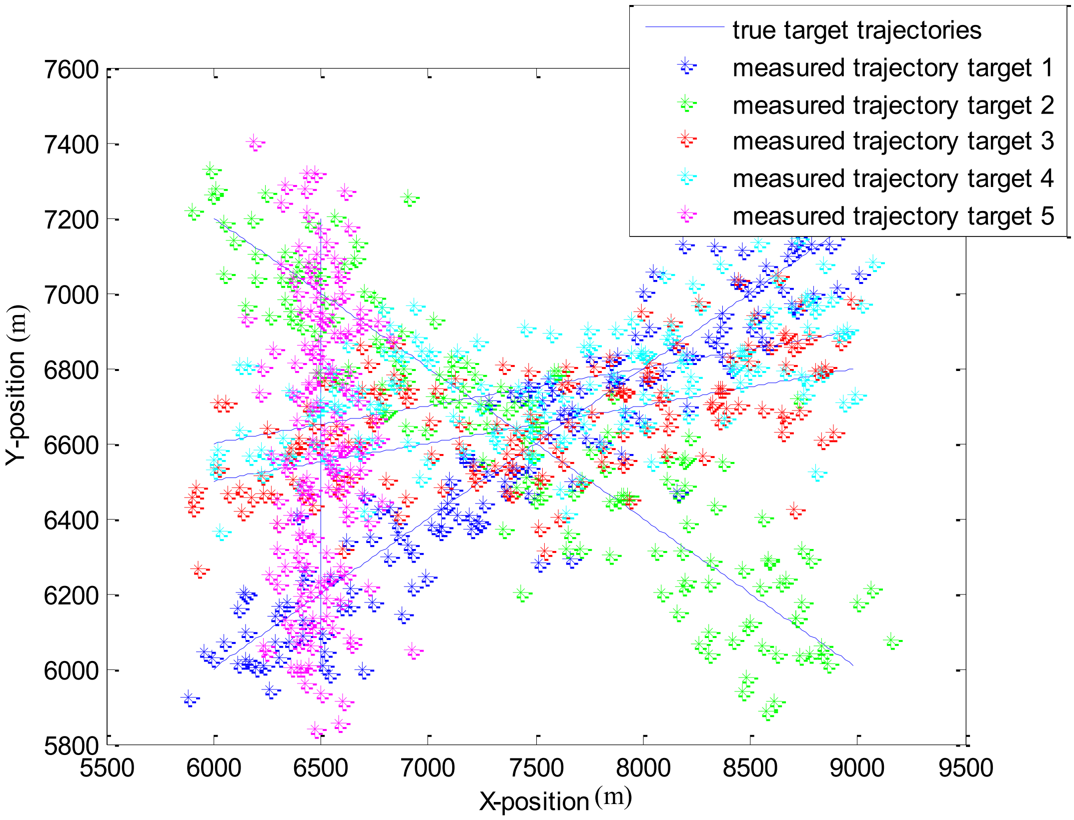

Figure 4.24:True and measured target trajectories for five non-maneuvering targets.

Table 4.2: Values of Standard Deviation

σ_{x1} , σ_{y1}	**100 m**
σ_{x2} , σ_{y2}	**150 m**
σ_{x3} , σ_{y3}	**130 m**
σ_{x4} , σ_{y4}	**130 m**
σ_{x5} , σ_{y5}	**100 m**

The estimated target tracks using the proposed algorithm are shown in Figure 4.25. From Figure 4.26 it is obvious that the estimation error for the five targets is slightly worse than the case of two targets only. The estimation error shown in Figure 4.26 is higher in the middle at about 7.5 seconds. This is due to the proximity of the targets to each other as we get close to crossing points between targets.

The performance especially degrades for the target moving in a vertical line. That is because the data association algorithm was mostly trained with tracks that are moving in both x- and y- directions. The performance for this target is slightly shaky in the beginning of tracking, but improves and saturates by time. The results of the fifth target are displayed for both perfect association and HMM-based association in Figure 4.27 for the sake of comparison.

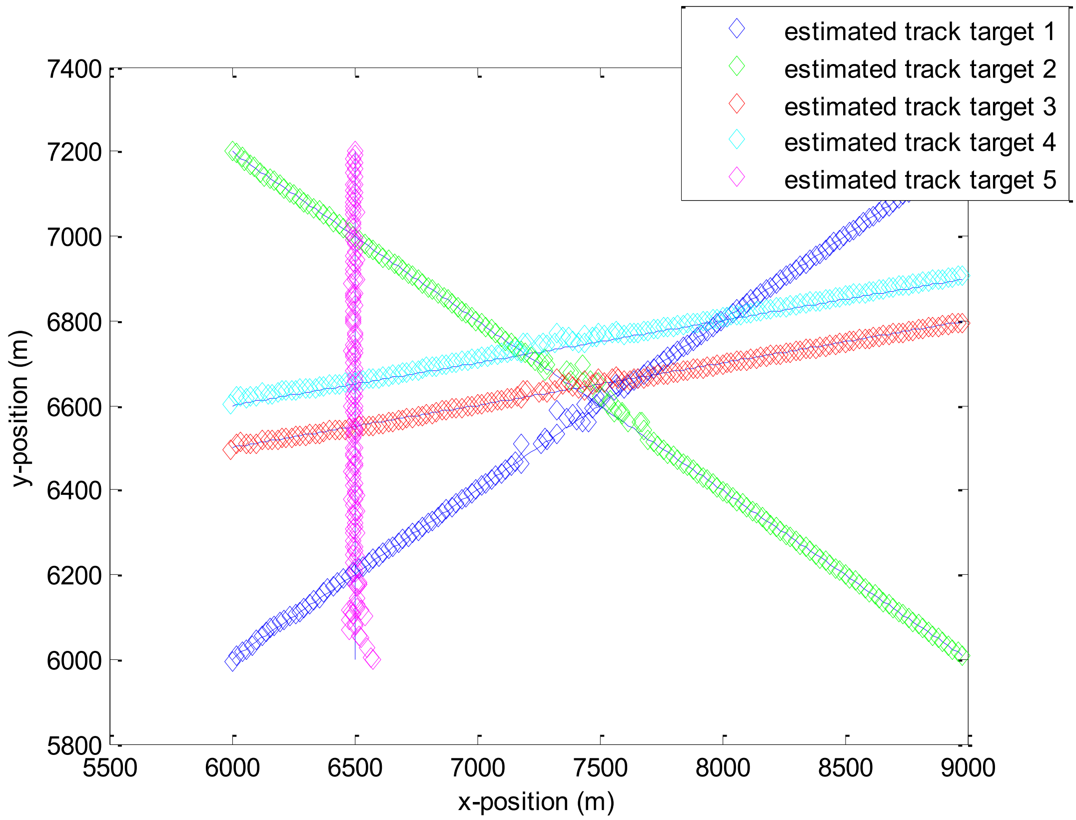

Figure 4.25: Estimated target tracks for five non-maneuvering targets using the proposed algorithm.

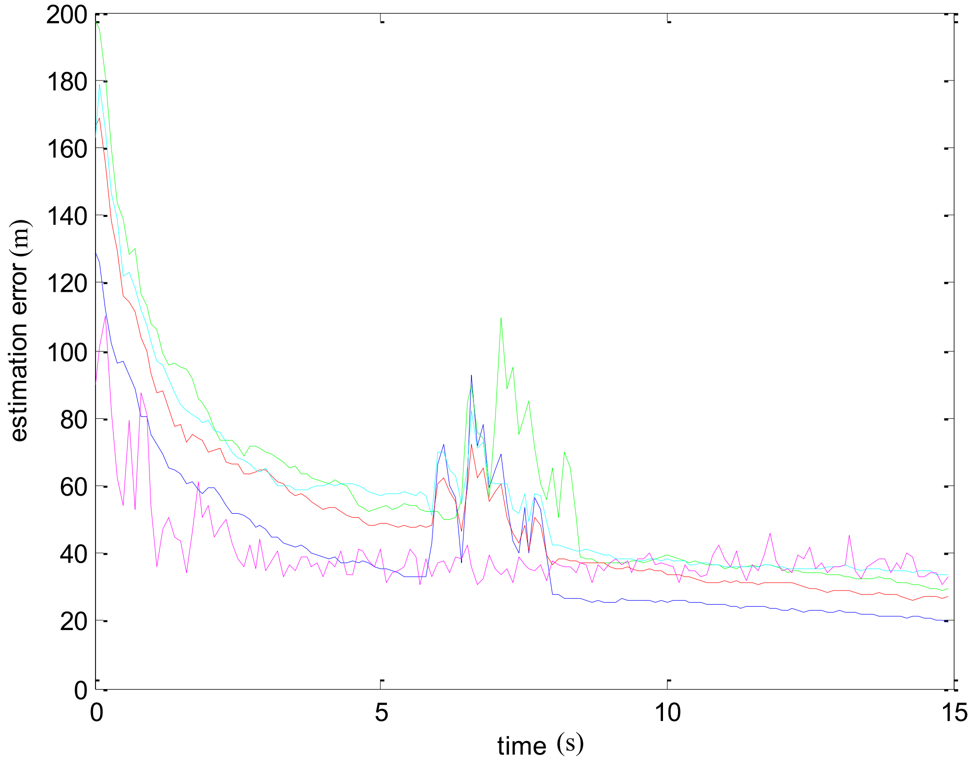

Figure 4.26: Estimation error for the five targets.

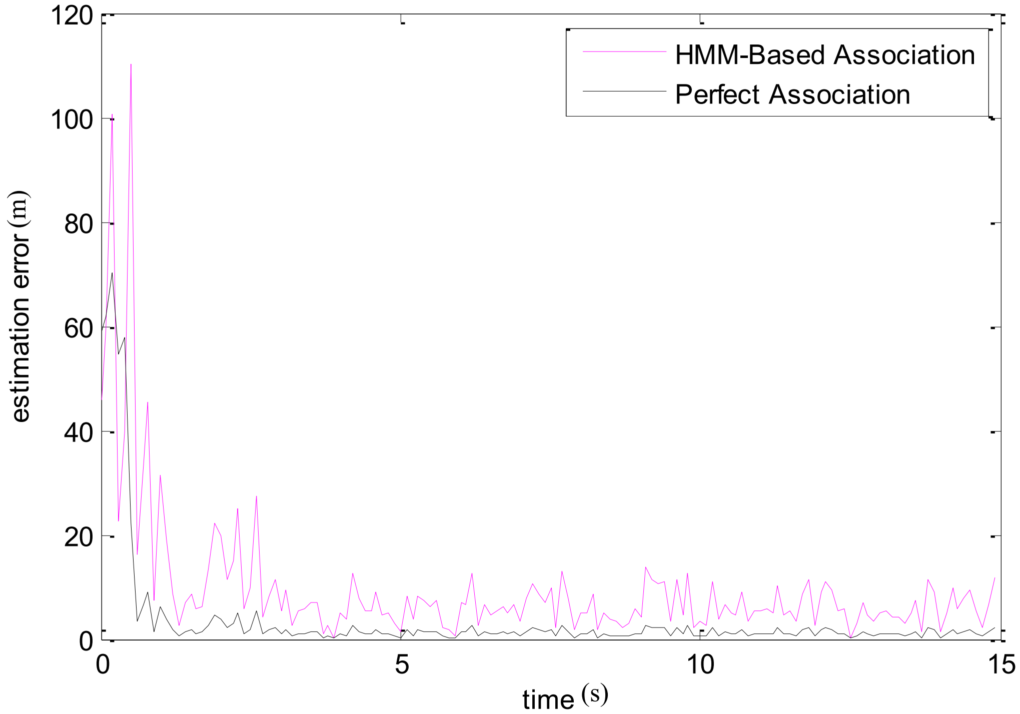

Figure 4.27: Estimation error for a vertically moving target using HMM-based association and perfect association.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this work, an approach to solve the MTMST problem is proposed, and the performance of the proposed system is analyzed. From the results that have been obtained in this study, the following conclusions can be made:

1. A trade-off exists between performance and computational complexity of data association techniques.
2. The proposed HMM-based data association technique outperforms the NNSF commonly used in data association.
3. HMM-based association significantly improves performance in maneuvering targets, as it is more tailored to the dynamical model than NNSF.
4. The performance of HMM-based data association approaches perfect association performance especially in the case of maneuvering targets.
5. The suggested association method withstands the increase in noise levels in measurements.
6. The data association approach presented has lower number of operations than most of the methods reported in the literature. Also as the number of targets and measurements increases, the number of computations does not increase as rapidly as in other methods.
7. Multisensor estimate fusion can be used to enhance error performance even further.
8. Increasing the number of sensors, improves the estimation error of the fused track. As the number of sensors increases further, the performance starts to reach a saturation level beyond which performance does not significantly improve.
9. Variations in sensors' accuracies affect the performance of the fused track. The best performance is obtained using identical sensors.

10. As variations in sensors' accuracies become more severe, the performance of the superior track beats the performance of the fused track, and thus it is unjustified to adopt fusion in such case.
11. As the number of targets increases the performance of the overall tracking algorithm degrades slightly. However, the performance is still within acceptable limits.
12. The performance of the data association approach deteriorates in the case of a vertically moving target, as examples for such target were not used during training.

5.2 Future Work

In the following, we suggest some research points that are recommended for further investigation:

1. Studying the problem of track initiation and track deletion using the proposed data association approach.
2. Applying maneuver detection to the proposed tracking system to switch between multiple dynamical models.
3. Examining more dynamical models for maneuvering targets, other than the turning motion model, using the proposed system.
4. Testing the efficiency of the system in image based tracking, using features other than solely depending on position.
5. Solving the MTMST problem for tracking targets in a three dimensional space.

LIST OF PUBLICATIONS

1. Ashraf M. Aziz , Nawal A. Zaher, “Multisensor Estimate Fusion Based on Bayesian Minimum Mean Square Error Criterion,” Proceeding of the 29th National Radio Science Conference, Apr. 2012, pp. 502-517.

2. Nawal A. Zaher, Ashraf M. Aziz, Hussein H. Ghouz, “A Data Association Approach for Multitarget Tracking Based on a Hidden Markov Model,” Accepted.

REFERENCES

[1] E. Mazor, A. Averbuch, Y. Bar Shalom, J. dayan, "Interacting multiple model methods in multitarget-multisensor tracking: survey", IEEE Transactions on Aerospace and Electronic Systems, vol. 34(1), Jan. 1998, pp. 103-123.

[2] Stefan J. Schwoegler et al., "Multiple hypothesis tracking", US patent #20120233097A1, Sep. 2012.

[3] S.S. Blackman, "Multiple-Target Tracking With Radar Applications", Artech House, Norwood, MA, 1986.

[4] J. Fisher, D. Casasent, "Fast JPDA multitarget tracking algorithm", Applied Optics, vol. 28 (2), 1989, pp. 371–376.

[5] Ashraf M. Aziz, "A novel all-neighbor fuzzy association approach for multitarget, tracking in a cluttered environment", Signal Processing, vol. 91(8), Mar. 2011, pp. 2001-2015.

[6] Yi-Nung Chung, Pao- Hua Chou, Maw-Rong Yang, "Multiple target tracking with competitive Hopfield neural network based data association", IEEE Transactions on Aerospace and Electronic Systems, vol. 43 (3), Nov. 2007, pp. 1180–1188.

[7] Ashraf M. Aziz, "Fuzzy track-to-track association and track fusion approach in distributed multisensor multitarget multiple-attribute environment", Signal Processing, vol. 87(6), Jun. 2007, pp. 1474–1492.

[8] Jos Alfring, Rob Jensen, Rene van de Molengraft, "Robots sharing a knowledge base for world modeling and learning of actions", Seventh Framework Programme, Apr. 2010.

[9] Lawrence R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition", Proceedings of the IEEE, vol. 77(2), Feb. 1989, pp. 257-286.

[10] Przemyslaw Dymarski, "Hidden Markov Models, Theory and Applications", Rijeka, Intech., 2011.

[11] A. W. Drake, "Discrete-State Markov Processes", McGraw-Hill, New York, 1967.

[12] S. Young et al., "HTK Book (for HTK version 2.2)", Entropic Ltd., California 1999.

[13] T. Shinozaki, T. Kawahara, "GMM and HMM training by aggregated EM algorithm with increased ensemble sizes for robust parameter estimation", Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Las Vegas, Sep. 2008, pp. 4405-4408.

[14] L. E. Baum, T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains", Ann. Math. Stat., vol. 37, 1966, pp. 1554-1563,1966.

[15] L. E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes", Inequalities, vol. 33, 1972, pp. 1-8.

[16] L. E. Baum, T. Petri, G. Soules, N. Weiss, "A maximization technique occuring in the statistical analysis of probabilistic functions of Markov chains", Ann. Math. Stat., vol. 41(1), 1970, pp. 164-171.

[17] A. P. Dempster, N. M. Laird, D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", J. Roy. Stat. Soc., vol. 39(1), 1977, pp. 1-38.

[18] G. D. Forney, Jr., "The Viterbi Algorithm", Proceedings of the IEEE, vol. 63(3), 1973, pp. 268-278.

[19] Peter S. Maybeck, "Stochastic Models, Estimation and Control", vol. (1), Academic Press, New York, 1979.

[20] Mohinder S. Grewal, Angus P. Andrews, "Kalman filtering: Theory and Practice Using MATLAB", 2nd edition, John Wiley & Sons, New York, 2001.

[21] Simon Haykin, "Kalman Filtering and Neural Networks", John Wiley & Sons, New York, 2001.

[22] Ashraf M. Aziz, Nawal A. Zaher, "Multisensor estimate fusion based on Bayesian minimum mean square error criterion", Proceedings of the 29th National Radio Science Conference, Egypt, Apr. 2012, pp. 503-514.

[23] David L. Hall, James Llinas, "An introduction to multisensor data fusion", Proceedings of the IEEE, vol. 85 (1), Jan. 1997, pp. 8-23.

[24] David L. Hall, "Mathematical Techniques in Multisensor Data Fusion", Norwood, MA, Artech House, 1992.

[25] P. K. Varshney, "Multisensor data fusion", Electronics and Communication Engineering Journal, Dec. 1997, pp. 245-253.

[26] N. Xiong, P. Svensson, "Multi-sensor management for information fusion: issues and approaches", Information Fusion, vol. 3, 2002, pp. 163-186.

[27] Y. Bar Shalom, "Multitarget Multisensor Tracking: Applications and Advances", vol. 1, Norwood, MA, Artech House, 1990.

[28] Mark R. Morelande, Christopher M. Kreucher, and Keith Kastella , "A Bayesian approach to multiple target detection and tracking", IEEE Transactions on Signal Processing, vol.55 (5), May 2007, pp. 1589-1604.

[29] Y. Bar Shalom, Thomas E. Fortmann, "Tracking and Data Association", Academic Press, New York, 1988.

[30] Taek L. Song, Dong G. Lee, Jonha Ryu, "A probabilistic nearest neighbor filter algorithm for tracking in a clutter environment", Signal Processing, vol. 85(10), pp. 2044-2053.

[31] Christopher Rasmussen, Gregory D. Hager, "Probabilistic data association methods for tracking complex visual objects", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23 (6), Jun. 2001, pp. 560-576.

[32] T. Kirubarajan, Y. Bar-Shalom, "Probabilistic data association techniques for target tracking in clutter", Proceedings of the IEEE, vol. 92(3), Mar. 2004, pp. 536-557.

[33] Andreas Krauβling, Frank E. Schneider, Dennis Wildermuth, "Tracking expanded objects using the viterbi algorithm", Proceedings of the Second IEEE International Conference on Intelligent Systems, 2004, pp. 566-571.

[34] P. Smith, G. Buechler, "A branching algorithm for discriminating and tracking multiple objects", IEEE Transactions on Automatic Control, vol. 20, 1975, pp. 101-104.

[35] K. J. Molner, J. W. Modestino, "Application of the EM algorithm for the multitarget/multisensor tracking problem", IEEE Transactions on Signal Processing, vol. 46(1), Jan 1998, pp. 115-129.

[36] R.E. Kalman, "A new approach to linear filtering and prediction problems", IEEE Transactions on Signal Processing, vol. 82(4), 1960, pp. 35-45.

[37] G. Bianchi, I. Tinnirello, "Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network" . Proceeding of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, 2003, pp. 844-852.

[38] Dan Simon, "Optimal State Estimation", John Wiley & Sons, Hoboken, NJ, 2006.

[39]Julier, Simon J., & Uhlmann, Jeffrey K, "Unscented filtering and nonlinear estimation", Proceedings of the IEEE, vol. 92(3), 2004, pp. 401-422.

[40] N. Zheng, J. Xue, "Statisical Learning and Pattern Analysis for Image and Video Processing", 1st edition, Advances in Pattern Recognition. Springer, London, 2009.

[41] D. Willner, B. Chang, and K. P. Dunn, "Kalman filter algorithms for a multi-sensor system," Proceedings of IEEE Conference on Decision and Control, Dec. 1976, pp. 570-574.

[42] Y. Bar-Shalom and X. Rong Li," Multitarget Multisensor Tracking: Principles and Techniques," IEEE Control Systems Magazine , Feb. 1996, pp. 93-96.

[43] Y. Bar-Shalom, "On the Track-to-Track Correlation Problem," IEEE Transactions on Automatic Control, vol. 26 (2), Apr. 1981, pp. 571-572.

[44] Yunmin Zhu, Zhisheng You, Juan Zhao, Keshu Zhang, X. Rong Li, "The optimality for the distributed Kalman filtering fusion with feedback", Automatica, vol. 37, 2001, pp. 1489-1493.

[45] Y. Bar-Shalom, and L. Campo, "The effect of the common process noise on the two-sensor fused track covariance," IEEE Transaction on Aerospace and Electronic Systems., vol. 22, Nov. 1986, pp. 803 - 805.

[46] Chee-Yee Chong, Kuo-Chu Chang, Shozo Mori, William H. Barker, "Architectures and algorithms for track association and fusion", IEEE Aerospace and Electronics Systems Magazine, vol. 15 (1), Jan 2000, pp. 5-13.

ARABIC SUMMARY

الملخص

تتبع الأهداف يتكون من مرحلتين و هما: ربط البيانات وتخمين الموقع. ربط البيانات المقاسة بالأهداف في بيئة مليئة بالتشويش و في وجود عدة أهداف هي مهمة صعبة تحتاج الي حل دقيق حتي يتسني لنا تتبع الأهداف بدقة. هذه الأطروحة تتناول طريقة لربط البيانات بالأهداف باستخدام نموذج ماركوف المستتر و ذلك قبل تتبع الأهداف باستخدام مرشح كالمان للتتبع. بعد ذلك يتم دمج البيانات من عدة أجهزة استشعار عن طريق أقل متوسط لمربع الخطأ. دمج البيانات قائم علي فرضية وجود رابط احصائي بين الخطأ الناتج عن الموقع المحدد من كل مستشعر علي حدة. يتم نقل الموقع الناتج عن دمج البيانات الي المستشعرات ليتم استخدامه في تحسين التنبؤ لكل مستشعر علي حدة في الدورة القادمة.

تم محاكاة امثلة لأهداف مناورة و غير مناورة تتشابك مساراتها في بعض النقاط. في المرحلة الثانية يستخدم مرشح كالمان الخطي لتحديد موقع الهدف بالعتماد علي القياس الذي تم ربطه بالهدف في المرحلة الأولي. تظهر النتائج تحسن كبير في تقليل الخطأ مقارنة باستخدام مرشح معتاد يعتمد علي اختيار القياس الأقرب مسافة من الهدف. الطريقة المقترحة تعطي نتائج بجودة مقاربة لجودة الربط المثالي للبيانات. تتحمل طريقة الربط المقترحة قيم كبيرة من قيمة الخطأ للمستشعرات.

دمج البيانات من أجهزة استشعار مختلفة يؤدي الي تحسن كبير في النتائج أيضا. تم دراسة تأثير زيادة عدد المستشعرات من اثنين الي خمسة. تم أيضا دراسة تأثير الاختلاف في دقة المستشعرات المستخدمة، بدءا باختلافات بسيطة و وصولا الي حالات أكثر تباينا. في حالة وجود اختلافات كبيرة في الدقة، يصبح أداء المستشعر الأقل خطأ افضل من الأداء في حالة دمج البيانات من كل أجهزة الاستشعار.

كلية الهندسة و التكنولوجيا – القاهرة

متتبع كالمان معدل باستخدام نموذج ماركوف المستتر

إعداد

نوال علي محمود زاهر

رسالة مقدمة للأكاديمية العربية للعلوم والتكنولوجيا والنقل البحري
لإستكمال متطلبات نيل الماجستير في

هندسة الالكترونيات و الاتصالات

المشرفين

الأستاذ الدكتور / أشرف ممدوح عزيز

قسم الالكترونيات و الاتصالات؛ كلية الهندسة

الأكاديمية العربية للعلوم و التكنولوجيا

و النقل البحري ؛ فرع القاهرة

أستاذ مساعد دكتور/ حسين حامد غز

قسم الالكترونيات و الاتصالات؛ كلية الهندسة

الأكاديمية العربية للعلوم و التكنولوجيا

و النقل البحري ؛ فرع القاهرة