

**SAKARYA UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY**

**DECENTRALIZED KALMAN FILTER APPROACH
FOR MULTI-SENSOR MULTI-TARGET TRACKING
PROBLEMS**

M.Sc. THESIS

Fawzia ABDIEN ALI ABDULLA

**Department : ELECTRICAL AND ELECTRONICS
ENGINEERING**
Field of Science : ELECTRONICS
Supervisor : Assoc. Prof. Dr. Askin DEMIRKOL

June 2016

SAKARYA UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY

**DECENTRALIZED KALMAN FILTER APPROACH
FOR MULTI-SENSOR MULTI-TARGET TRACKING
PROBLEMS**

M.Sc. THESIS

Fawzia ABDIEN ALI ABDULLA

Department : ELECTRICAL AND ELECTRONICS
ENGINEERING
Field of Science : ELECTRONICS
Supervisor : Assoc. Prof. Dr. Askin DEMIRKOL

This thesis has been accepted unanimously / with majority of votes by the
examination committee on 27.06.2016



Assoc. Prof. Dr.
Askin DEMIRKOL
Head of Jury



Assoc. Prof. Dr.
Ali TANGEL
Jury Member



Assist. Prof. Dr.
Cenk YAVUZ
Jury Member

DECLARATION

I declare that all the data in this thesis was obtained by myself in academic rules, all visual and written information and results were presented in accordance with academic and ethical rules, there is no distortion in the presented data, in case of utilizing other people's works they were refereed properly to scientific norms, the data presented in this thesis has not been used in any other thesis in this university or in any other university.

Fawzia ABDIEN ALI ABDULLA

27.06.2016

ACKNOWLEDGEMENT

At the end of my thesis I would like to thank all those people who made this thesis possible and an unforgettable experience for me.

First of all, I would like to express my deepest sense of Gratitude to my supervisor Assoc.Prof.Dr. Askin DEMIRKOL, who offered his continuous advice and encouragement throughout the course of this thesis. I thank him for the systematic guidance and great effort he put into training me in the scientific field.

I would like to acknowledge the Turkish government and Turkish scholarship to give me this opportunity to study in Turkey. I would like to acknowledge also the Department of Electrical and Electronics Engineering at Sakarya University and all the Teaching staff there for the quality education given by them.

My very sincere thanks to my friends for their family-like support, generous care and the home feeling whenever I was in need during my stay in Turkey.

Finally, I take this opportunity to express the profound gratitude from my deep heart to my beloved parents, grandparents, and all my family for their love and continuous support.

TABLE OF CONTENTS

ACKNOWLEDGEMENT	i
TABLE OF CONTENTS	ii
LIST OF SYMBOLS AND ABBREVIATIONS	v
LIST OF FIGURES	vii
SUMMARY	x
CHAPTER 1.	
INTRODUCTION	1
1.1. Thesis Introduction	1
1.2. Problem Statement.....	4
1.3. Thesis Objective	4
1.4. Motivation	4
1.5. Thesis Organization	4
CHAPTER 2.	
BACKGROUND AND LITERATURE REVIEW	6
2.1. Why Tracking and Prediction are Needed In A Radar	6
2.2. Estimation Theory	9
2.3. Multi-Targets Tracking.....	12
2.4. Multi-sensor System	13
2.5. Sensor Data Fusion	14
2.5.1. Fusion architectures	15
2.5.1.1. Centralized architectures	15
2.5.1.2. Hierarchical architectures	16
2.5.1.3. Decentralized architectures.....	16

3.6. Kalman Filter	17
2.6.1. Reasons for using the Kalman Filter	18
2.6.2. The continuous time kalman filter	19
2.6.2. The discrete Kalman Filter	24
2.6.2.1. The process to be estimated.....	25
2.6.2.2. The computational origins of the filter	25
2.6.2.3. The probabilistic origins of the filter	27
2.6.2.4. The discrete Kalman Filter algorithm.....	28
2.7. Literature Review	30

CHAPTER 3.

MATERIALS AND METHODS	34
3.1. Decentralized Kalman Filter.....	34
3.2. Problem Formulation.....	36
3.2.1. Fusion Algorithm	39
3.3. MATLAB Implementation	40
3.3.1. Two sensors tracking the same target	40
3.3.2. Two sensors tracking two different targets	41
3.3.3. Two sensors tracking same two targets.....	42
3.3.4. Two sensors tracking three different targets	44

CHAPTER 4.

MATERIALS AND METHODS	34
4.1. The Results Of Two Sensors Tracking The Same Target	46
4.2. The Results Of Two Sensors Tracking Two Different Targets.....	48
4.3. The Results Of Two Sensors Tracking Same Two Targets.....	50
4.4. The Results Of Two Sensors Tracking Three Different Targets.....	52

CHAPTER 5.

CONCLUSION AND SUGGESTED FUTURE WORK	55
5.1. Conclusions	55

5.2. Suggested Future Work	56
REFERENCES.....	54
RESUME	61



LIST OF SYMBOLS AND ABBREVIATIONS

$\hat{X}_i(k)$: Estimated state vector corresponding to the i-th sensor
Θ	: scan angle
$\hat{X}^-(k)$: A priori state estimate at step k
$\hat{X}(k)$: A posteriori state estimate at step k
$\Phi(t, t_0)$: state transition matrix
μ_x	: mean x
2D	: Two dimension
B_i	: Input control matrix gain
$C_i(k)$: The measurement or observation matrix for the global filter
$D_i(k)$: Matrix reflect which of the targets are observed by the i-th sensor
DKF	: Decentralized kalman filter
$e(k)$: A posteriori estimate errors
$\bar{e}(k)$: A priori estimate errors
$E\{ \}$: expectation operator
EM	: Expectation-maximization
$F(t)$: System and error model dynamics matrix
$F(t)$: System and error model dynamics matrix
$F_i(k)$: Transition matrix
$G(t)$: The noise gain matrix
$G_i(k)$: The noise gain matrix for the i-th
$H(t)$: Measurement or observation matrix
$H_i(k)$: Measurement or observation matrix corresponding to the i-th sensor
JPDA	: Joint probabilistic data association
$K(k)$: Kalman gain

KF	: Kalman filter
L	: Total number of the target in the surveillance region
L_i	: The targets seen by the i -th sensor
LQE	: Linear quadratic estimation
MCMC	: Markov Chain Monte Carlo
ML	: Maximum-likelihood
MLANS	: Maximum Likelihood Adaptive Neural System
N	: The total distributed sensors in the tracking system
N	: The order of state vector for each target
NASA	: National Aeronautics and Space Administration
NN	: Nearest neighbor
$P(k)$: A posteriori estimate error covariance
$P^-(k)$: A priori estimate error covariance
$P_i(k)$: The error covariance matrix corresponding to the i -th
$Q(t)$: The covariance matrix of the state model uncertainties.
$Q_i(k)$: White gaussian process noise covariance
$R(t)$: the covariance matrix of the observation noise at time t .
R_i	: White gaussian measurement noise covariance
T	: time component
$u(t)$: The system input vector at time t .
$U_i(k)$: Input control matrix
$V(k)$: The measurement noise at step k .
$V_i(k)$: White gaussian measurement noise corresponding to the i -th sensor
$W(k)$: The process noise at step k .
$W_i(k)$: White gaussian process noise corresponding to the i -th sensor
$x(t)$: The state vector at time t .
$X_i(k)$: The state vector corresponding to the i -th sensor at step k .
$y(t)$: System output vector at time t .
$z_i(k)$: The measurement or observation vector received by the i -th sensor
$\delta(t)$: Dirac delta function

LIST OF FIGURES

Figure 2.1. Example of fan-beam surveillance radar	7
Figure 2.2. Multifunction PATRIOT electronically scanned phased-array radar... 8	8
Figure 2.3. L-band fan-beam track-while-scan Pulse Acquisition Radar. 8	8
Figure 2.4. Tracking problem..... 9	9
Figure 2.5. Fusion Centralized Architecture 16	16
Figure 2.6. Fusion Hierarchical Architecture..... 17	17
Figure 2.7. the general composition of a linear, time- varying state model..... 20	20
Figure 2.8. The ongoing discrete Kalman filter cycle..... 29	29
Figure 3.1. The decentralized kalman filter architecture 35	35
Figure 3.2. The decentralized Kalman filter flow diagram..... 40	40
Figure 3.3. The two sensors tracking the same target. 41	41
Figure 3.4. The two sensors tracking two different targets..... 42	42
Figure 3.5. The two sensors tracking same two targets. 44	44
Figure 3.6. Two sensors tracking three different targets..... 45	45
Figure 4.1. Tracking results of sensor 1 for one target state. 47	47
Figure 4.2. Tracking results of sensor 2 for one target state. 47	47
Figure 4.3. Tracking results of the central processor for one target state. 48	48
Figure 4.4. Tracking results of sensor 1 when the two sensors are tracking two. different targets..... 49	49
Figure 4.5. Tracking results of sensor 2 when the two sensors are tracking two. different targets..... 49	49
Figure 4.6. Tracking results of the central processor when the two sensors are. tracking two different targets..... 50	50
Figure 4.7. Tracking results of sensor 1 when the two sensors are tracking same two. targets..... 51	51

Figure 4.8. Tracking results of sensor 2 when the two sensors are tracking same two targets..... 51

Figure 4.9. Tracking results of the central processor when the two sensors are tracking same two targets. 52

Figure 4.10. Tracking results of sensor 1 when the two sensors are tracking three different targets. 53

Figure 4.11. Tracking results of sensor 2 when the two sensors are tracking three. different targets. 53

Figure 4.12. Tracking results of the central processor when the two sensors are tracking three different targets. 54



ÇOKLU SENSÖR ÇOKLU HEDEF İZLEME PROBLEMLERİ ÜZERİNE BİR DAĞITILMIŞ KALMAN FİLTRESİ YAKLAŞIMI

ÖZET

Anahtar kelimeler: Dağıtık Kalman Filtresi, Çoklu sensör sistemler, Hedef takibi, çoklu hedef takibi sistemleri.

Doğru pozisyon ve hedeflerin sayısı hava trafik kontrol ve füze savunması için çok önemli bilgilerdir. Bu çalışma, çoklu sensörlü çoklu hedef takibi sistemlerindeki veri füzyonu ve durum tahmini problemleri için dağıtık Kalman Filtreleme Algoritması sunmaktadır. Problem, radar olarak her biri kendi veri işleme birimine sahip aktif sensörlerin hedef alanını gözlemlemesini esas almaktadır. Bu durumda her bir sistemin iz sayısı olacaktır. Çalışmada önerilen dağıtık Kalman Filtresi, başta füze sistemleri olmak üzere savunma sistemlerinde hareketli hedeflerin farklı sensörlerle izlerini kestirmek ve farklı hedefleri ayırd etmek için kullanılmaktadır. Önerilen teknik, çoklu sensör sisteminden gelen verileri işleyen iki aşamalı veri işleme yaklaşımını içermektedir. İlk aşamada, her yerel işlemci kendi verilerini ve standart Kalman filtresi ise en iyi kestirimi yapmak için kullanılmaktadır. Sonraki aşamada bu kestirimler en iyi küresel bir kestirimi yapmak amacıyla dağıtık işlem modunda elde edilir. Bu çalışmada iki radar sistemi iki yerel Kalman filtresi ile uçakların pozisyonunu kestirmek amacıyla kullanılmakta, ardından bu kestirimler merkez işlemciye iletilmektedir. Merkez işlemci doğrulama maksadıyla bu bilgileri birleştirip küresel bir kestirim üretmektedir. Önerilen model uygulama olarak dört senaryo üzerinde test edildi. İlk senaryoda, tek bir hedef iki sensor tarafından izlenirken, ikincisinde, iki hedeften oluşan uzay herhangi bir sensor tarafından izlenmekte, üçüncüsünde, iki hedefin de herhangi bir sensor tarafından aynı anda izlenmesi, son olarak ise iki sensörden her birinin toplam üç hedeften herhangi ikisini izlediği senaryo göz önüne alınmıştır. Önerilen tekniğin performansı hata kovaryans matrisi kullanılarak değerlendirildi ve yüksek doğruluk ve optimal kestirim elde edildi. Uygulama sonuçları önerilen tekniğin yeteneğinin, yerel sensörlerce belirlenen ortak hedeflerin merkezi sistem tarafından ayırd edilebildiğini göstermiştir.

SUMMARY

Keywords: Decentralized kalman filter, Estimation, Multi-sensor, Multi-target, Target tracking.

For air traffic control and missile defense, the accurate position and the numbers of targets are the most important information needed. This thesis presents a decentralized kalman filtering algorithm (DKF) for data fusion and state estimation problems in multi-sensor multi-target tracking system. The problem arises when several sensors carry out surveillance over a certain area and each sensor has its own data processing system. In this situation, each system has a number of tracks. The DKF is used to estimate and separate the tracks from different sensors represent the targets, when the ability to track targets is essential in missile defense. The proposed technique is a two stage data processing technique which processes data from multi sensor system. In the first stage, each local processor uses its own data to make the best local estimation using standard kalman filter and then these estimations are then obtained in parallel processing mode to make best global estimation. In this work, two radar systems are used as sensors with two local Kalman filters to estimate the position of an aircraft and then they transmit these estimations to a central processor, which combines this information to produce a global estimation. The proposed model is tested on four scenarios, firstly, when there is one target and the two sensors are tracking the same target, secondly, when there are two targets and any sensor is tracking one of them, thirdly, when there are two targets and any sensor is tracking both of them and finally, when two sensors are used to track three targets and any sensor tracks any two of them. The performance of the proposed technique is evaluated using measures such as the error covariance matrix and it gave high accuracy and optimal estimation. The experimental results showed that the proposed method has the ability to separate the joint targets detected by the local sensors.

CHAPTER 1. INTRODUCTION

1.1. Thesis Introduction

Tracking is the process of filtering noisy measurements from one or more sensors such as radar, sonar or video to achieve the best possible estimation of the state of the target given the state and measurement model with possibly uncertain target-measurement associations [1, 2]. The purpose of a tracking system [3-6] is to determine the location or direction of a target on a near-continuous basis. The ability to track targets is essential in many applications. Well-established military applications include missile defense and battlefield situational awareness. Civilian applications are ever-growing, ranging from traditional applications such as air traffic control and building surveillance to emerging applications like supply chain management and wildlife tracking [7]. Target tracking is a necessary part of systems that perform functions such as surveillance, guidance or obstacle avoidance [1, 8]. Often the data measured by tracking devices are not exact and must be first filtered to improve the estimations [9]. This occurs due to a factor called “noise”. There are two types of noise; the measurement noise is caused by inaccuracies in the tracking device, and state noise is caused by turbulence or human error and other environmental factors [10]. Noise results from multiple factors: atmospheric interference, impreciseness of the radar’s measurements, turbulence affecting the target’s movement, and human inability to navigate in a perfectly straight line. Measurements taken at irregular time intervals also complicate the estimation process. Data-filtering target trackers are utilized in many situations to provide an estimation of the position and velocity of a target at the time of measurement. They are commonly utilized in air traffic control, navigation, and collision avoidance systems. Such fields have always sought improved accuracy in their predictions [8].

The basic principle of tracking is to combine measurement data with mathematical model of how the target is likely to move [11]. The state model describes the evolution of the target between two consecutive time increments; typically a set of differential equations highlighting the mathematic model of the target was used for the target model. While the measurement model links the target state vector to the measurement vector. Such models allow the system to predict both the state vector of the target and the measurement based on current state of target. The prediction is then combined with sensor measurements to produce an estimation of the target state.

There are some fundamental limitations of environment-based descriptions in a single source of sensor information towards to multi-sensor systems. Tracking in multi-sensor networks has gained popularity for several major reasons: (1) As the cost of sensors and devices rapidly decrease, they can be deployed in large numbers to achieve wide area coverage and their increased density allows sensors to reside far closer to the objects being sensed, improving sensing quality and discrimination. (2) Dense sensors enable overlapping coverage, which may result in increased robustness and improved accuracy. (3) Diverse sensing modalities provide complementary information. This diversity in sensing modalities can be exploited to provide accurate and rich information about the target. (4) Spatial sensing diversity greatly mitigates the effects of obstructions on line-of-sight sensors [7, 12].

Multiple-target tracking is one such application that can benefit from multiple sensing modalities [13]. The multiple-target tracking problem [14] extends the scenario to a situation where the number of targets may not be known and varies with time, and the measurements which have originated from targets are not known since some of them may be due to false alarms [15]. Multiple target tracking plays an important role in many areas of engineering such as surveillance [16], computer vision, network and computer security, and sensor networks [17]. In order to deal with more than one target, the multiple targets tracking system requires handling the discrete uncertainty of measurement origin [1]. This is known as the data association problem. For this purpose, several algorithms have been put forward. This includes the nearest neighbor (NN) algorithm which associates the measurement to the closest predicted target [18].

The joint probabilistic data association (JPDA) algorithm forms a set of probabilistic hypotheses over every possible target-measurement association [1, 3], which constitute the weights of the weighted innovation expression over the set of measurements used for the filter update.

In a multi-sensor multi-target tracking system, observations obtained by multiple sensors are usually sent to a fusion center for processing [19]. Data fusion is the process of combining information from these different sources to provide a robust and complete description of an environment or process of interest [20, 21].

In last few decades, several researches have been developed various methods for target localization and tracking with different sensing modalities in sensor network. The Kalman Filter is a recursive process used to filter random inaccuracies in measurements to predict the most likely position and velocity (or any dimension based on position and time) of a moving target based on real-time position coordinate feeds. It considers the probability of a target's position and then updates its "belief" in the location of the target [22]. The advent and growth of computer science in the last half century allowed for application of the recursive linear filtering solution presented in Rudolf Emil Kalman's 1960 paper [23]. In this paper, Kalman described his new approach to linear filtering, a series of recursive equations that seek to minimize error by decreasing the covariance, increasing accuracy of the filter's prediction as each position coordinate is provided by target trackers such as radars.

Therefore, the aim of this study was to numerically design a system capable of effectively estimates the position of some targets (aircrafts) and determines if these targets data are representing same targets or different targets. The proposed system is composed of decentralized kalman filter (DKF) and multi-sensor system. The DKF is a two stage data processing technique which processes data from multi sensor system. In the first stage, each local processor uses its own data to make a best local estimation using standard kalman filter and then these estimations are then obtained in parallel processing mode to make the best global estimation.

1.2. Problem Statement

The problem arises when several sensors carry out surveillance over a certain area and each sensor has its own data processing system. Assume that at one time some targets are detected by sensor 1 and again some targets are detected by sensor2. The question arises as to whether these targets detected by the second sensor are the same targets detected by the first sensor or new targets. Futhermore, where these targets detected in the first time will be in the future.

1.3. Thesis Objective

The main objective of the thesis is to numerically design a tracking system capable of estimating the positions of multi-targets and determine whether these targets tracked by one sensor are the same targets tracked by another sensor or new targets in a multi-sensor system using decentralized kalman filter method.

1.4. Motivation

In the case of the air traffic control radar, correct knowledge of the number of targets present is important in preventing target collisions. In the case of the military radar it is important for properly assessing the number of targets in a threat and for target interception.

1.5. Thesis Organization

This thesis has been organized as follows:

Chapter 2: provides some background such as brief information on estimation theory, multi-sensor systems, data fusion and general introduction to kalman filtering and multi-targets tracking with some historical background.

Chapter 3: presents the material and methods used in this study.

Chapter 4: presents and discusses the obtained results.

Chapter 5: this chapter provides the conclusions drawn up from the thesis. It describes the main outcome of this thesis, and what more can be done in the future.



CHAPTER 2. BACKGROUND AND LITERATURE REVIEW

2.1. Why Tracking and Prediction are Needed in a Radar

Let first start by indicating why tracking and prediction are needed in radar. Assume fan-beam surveillance radar such as shown in Figure 2.1. For such radar the fan beam rotates continually through 360° , typically with a period of 10 sec. Such radar provides two-dimensional information about a target. The first dimension is the target range (i.e., the time it takes for a transmitted pulse to go from the transmitter to the target and back); the second dimension is the azimuth of the target, which is determined from the azimuth angle, the fan beam is pointing at when the target is detected [1]. Figure 2.2. and 2.3. show examples of fan-beam radars[1, 25].

Assume that at time $t=t_1$ the radar is pointing at scan angle Θ and two targets are detected at ranges R_1 and R_2 ; see Figure 2.4. Assume that on the next scan at time $t=t_1+T$, again two targets are detected; see Figure 2.4.

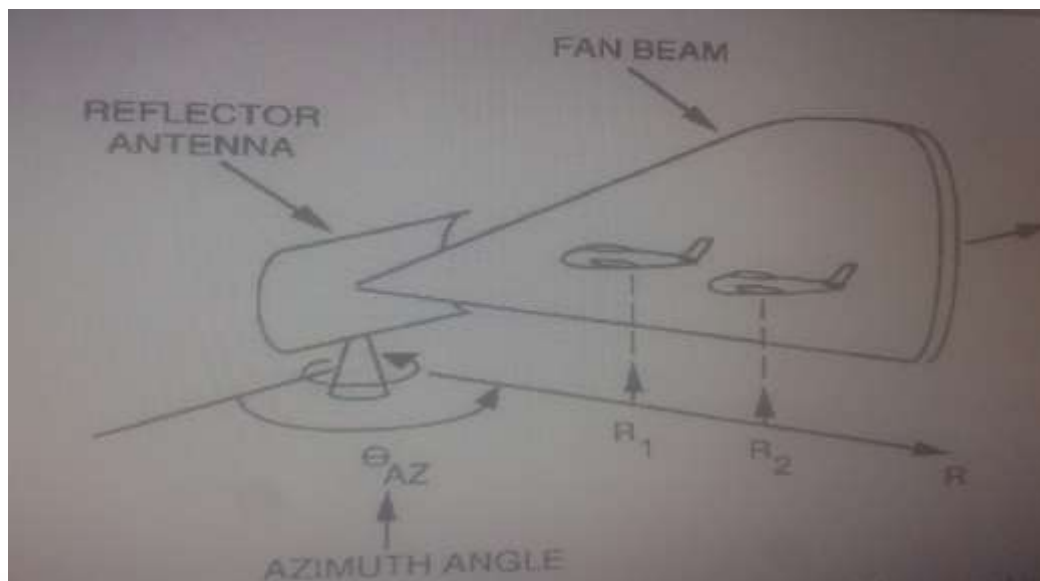


Figure 2.1. Example of fan-beam surveillance radar [11].

The question arises as to whether these two targets detected on the second scan are the same two targets or two new targets. The answer to this question is important for civilian air traffic control radars and for military radars. In the case of the air traffic control radar, correct knowledge of the number of targets present is important in preventing target collisions. In the case of the military radar, it is important for properly assessing the number of targets in a threat and for target interception [11].

Assume two echoes are detected on the second scan. Assume that it is correctly determined that these two echoes are from the same two targets as observed on the first scan. The question then arises as to how to achieve the proper association of the echo from target 1 on the second scan with the echo from target 1 on the first scan and correspondingly the echo of target 2 on the second scan with that of target 2 on the first scan.

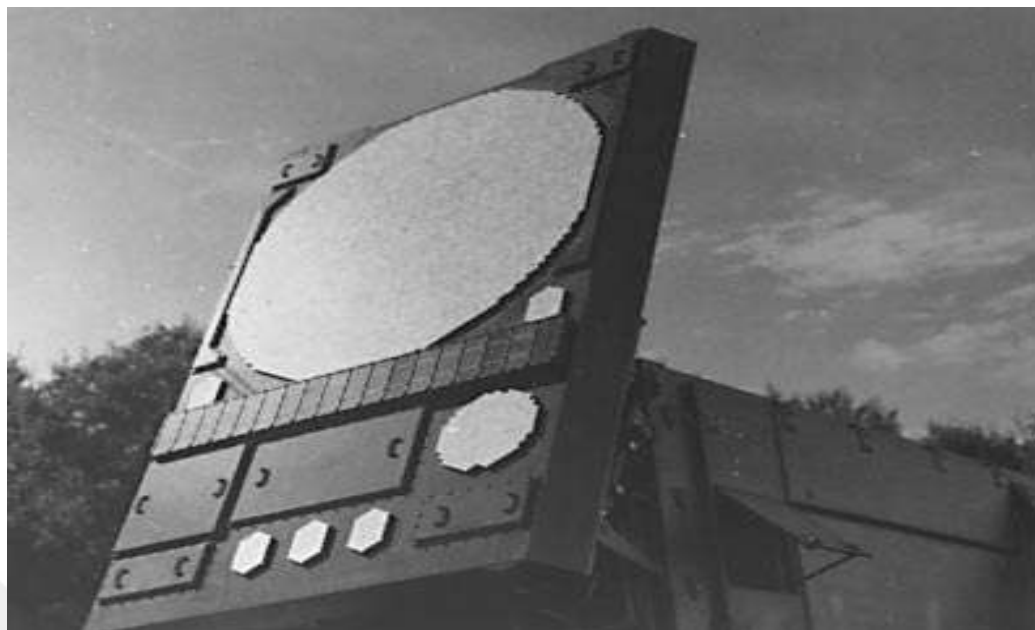


Figure 2.2. Multifunction PATRIOT electronically scanned phased-array radar used to do dedicated track on many targets while doing search on time-shared basis [25].

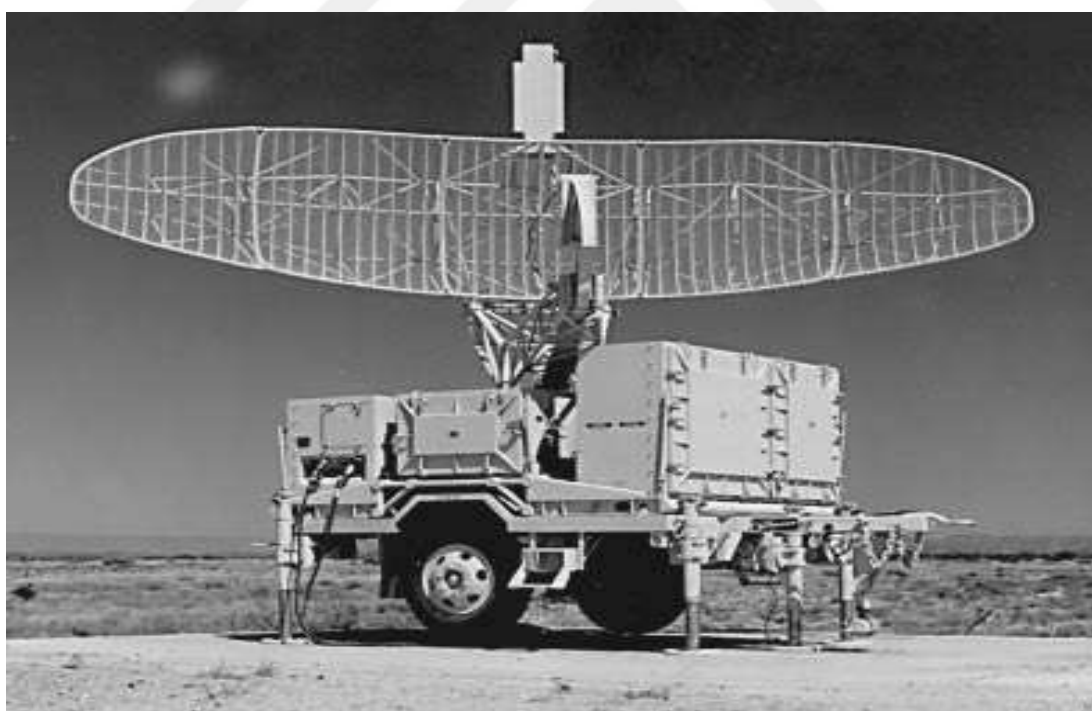


Figure 2.3. L-band fan-beam track-while-scan Pulse Acquisition Radar of HAWK system, which is used by 17 U.S. allied countries and was successfully used during Desert Storm [1].

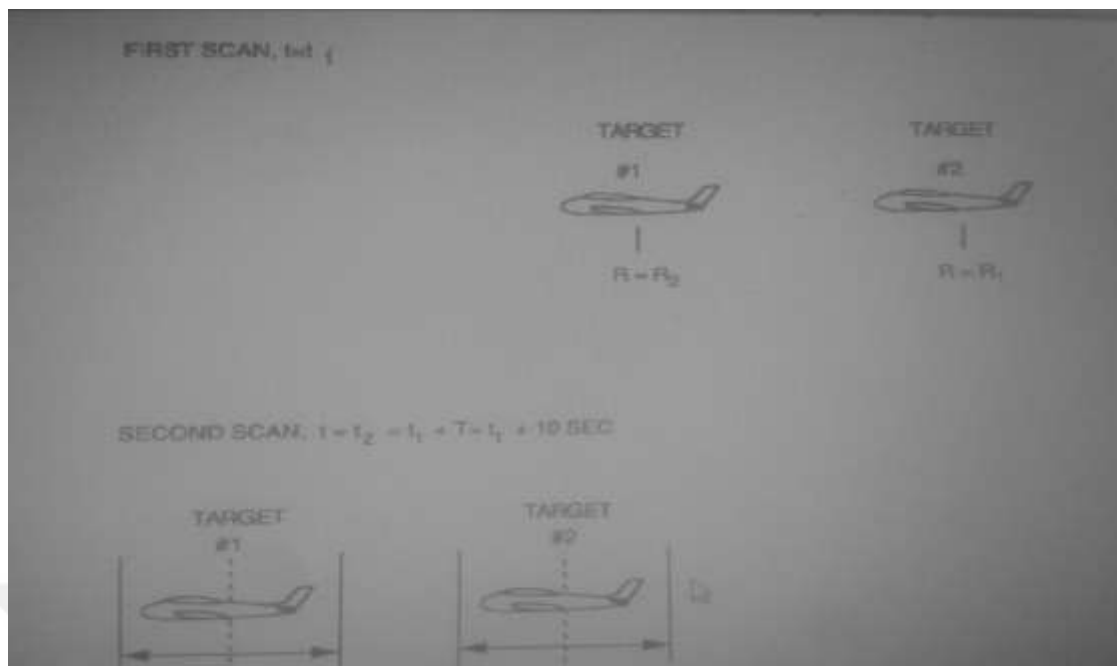


Figure 2.4. Tracking problem [1].

If an incorrect association is made, then an incorrect velocity is attached to a given target. For example, if the echo from target 1 on the second scan is associated with the echo from target 2 of the first scan, then target 2 is concluded to have a much faster velocity than it actually has. For the air traffic control radar this error in the target's speed could possibly lead to an aircraft collision; for a military radar, a missed target interception could occur.

The chances of incorrect association could be greatly reduced if we could accurately predict ahead of time where the echoes of targets 1 and 2 are to be expected on the second scan. Such a prediction is easily made if we had an estimate of the velocity and position of targets 1 and 2 at the time of the first scan. Then we could predict the distance target 1 would move during the scan to scan period and as a result have an estimate of the target's future position.

2.2. Estimation Theory

Many modern complex systems may be classified as estimation systems, combining several sources of (often redundant) data in order to arrive at an estimate of some

unknown parameters [27]. State estimation is virtually applicable to all areas of engineering and science. Any discipline that is concerned with the mathematical modeling of its systems is a likely (perhaps inevitable) candidate for state estimation. This includes electrical engineering, mechanical engineering, chemical engineering, aerospace engineering, robotics, economics, ecology, biology, and many others. The possible applications of state estimation theory are limited only by the engineer's imagination, which is why state estimation become such widely researched and applied discipline in the past few decades. State-space theory and state estimation was initially developed in the 1950s and 1960s, and since then there have been a huge number of applications [28].

State estimation is interesting to engineers for at least two reasons:

- Often, an engineer needs to estimate the system states in order to implement state-feedback controller. For example, the electrical engineer needs to estimate the winding current of a motor in order to estimate its position. The aerospace engineer needs to estimate the attitude of a satellite in order to control its velocity. The economist needs to estimate economic growth in order to try to control unemployment. The medical doctor needs to estimate blood sugar levels in order to control heart and respiration rates.
- Often, an engineer needs to estimate the system states because those states are interesting in their own rights. For example, if an engineer wants to measure the health of an engineering system, it may be necessary to estimate the internal condition of the system using a state estimation algorithm. An engineer might want to estimate satellite position in order to more intelligently schedule future satellite activities. An economist might want to estimate economic growth in order to make political point. A medical doctor might want to estimate blood sugar levels in order to evaluate the health of a patient.

The application contexts can be deterministic or probabilistic and the resulting estimations are required to have some optimality and reliability properties. Estimation is often characterized as prediction, filtering or smoothing, depending on the intended objectives and the available observational information. Prediction usually implies the

extension in some manner of the domain of validity of the information. Filtering usually refers to the extraction of the true signal from the observations. Smoothing usually implies the elimination of some noisy or useless component in the observed data. Optimal estimation always guarantees closed-loop system stability even in the event of high estimator gains [27].

State estimation is critical for a number of reasons: Accurate state estimates make control much easier, and allow better control actions to be selected [28]. In addition, state estimation is a super set of diagnosis, so faults and undersirable state can be detected to allow remedial actions to be taken. Finally, state estimation can provide prognostic information, identifying components or systems that are likely to fail soon and should be repaired or replaced. A key aspect of state estimation is that it is rarely certain. There is inevitably some ambiguity in the sensor data received from a system, and it is of great use to have a state estimation that represents the uncertainty explicitly [29, 30]. This is for several reasons: First, a probability distribution representing the uncertainty can summarize all the telemetry received by the state estimator so far, making it easier to keep the state estimation up to date. Secondly, this probabilistic representation is of use in decision making by allowing the effects of planned future actions to be evaluated in states that have low probability but potentially catastrophic outcomes, rather than only in the most likely state. Finally, probabilistic information is of use for prognostics and maintenance, providing information about components that could have failed, or are degraded but not yet faulty [26].

One of the prime contributing factors to the success of the present-day estimation and control theory is the ready availability of high-speed, large-memory digital computers for solving the equations [25].

2.3. Multi-Targets Tracking

Target tracking is the process of filtering noisy measurements from one or more sensors to achieve the best possible estimation of the state of the target [27]. The purpose of a tracking system is to determine the location or direction of a target on a near-continuous basis. The ability to track targets is essential in many applications [3]. Well-established military applications include missile defense and battlefield situational awareness. Civilian applications are ever-growing, ranging from traditional applications such as air traffic control and building surveillance to emerging applications like supply chain management and wildlife tracking.

The basic principle of tracking is to combine measurement data with mathematical model of how the target is likely to move. A model is a set of differential equations that describes, for instance, the relations between position, velocity and acceleration. The model is used to calculate, to predict, where the target will be in the future. The prediction is then combined with sensor measurements to produce an estimate of the target state. The reason of using a model is to reduce the influence of the measurement noise. It also makes it possible to predict where the target will be in between measurements and thereby creating more of a smooth track rather than just estimates at discrete time points [31]. A complete tracking system must be able to deal with more than one target. This fact adds to the complexity of the system. Since function dealing with associating measurements to the right tracks, track initiation and deletion has to be implemented. Multiple-target tracking is one such application that can benefit from multiple sensing modalities [32]. Multiple-target tracking play an important role in many areas of engineering such as surveillance, computer vision [33], network and computer security [34], and sensor network [35]. The multiple-target tracking problem extends the scenario to a situation where the number of targets may not be known and varies with time. The measurements which have originated from targets are not known since some of them may be due to false alarms. We are now required to estimate the positions of an unknown number of targets, based on observations of the targets corrupted by noise, with the possibilities that there may be missed detections and that observations may be false alarms due to clutter [36].

2.4. Multi-sensor System

In a single sensor system, one sensor is selected to monitor the system or its surrounding environment [37, 38]. However, many advanced and complex applications require large number of sensors, rendering single sensor systems inadequate. A multi-sensor system employs several sensors to obtain information in a real world environment full of uncertainty and change [39]. This means various types of sensors and different sensor technologies are employed, where some of these sensors have overlapping measurement domains. Multiple-sensors provide more information and hence a better and more precise understanding of a system. Moreover, a single sensor is not capable of obtaining all the required information reliably at all times in varying environments. Furthermore, as the size and complexity of a system increases, so does the number and diversity of sensors required to capture its description. These are the primary motivating issues behind multi-sensor systems. There is a considerable amount of literature on the limitations of single sensor systems and the merits of multi-sensor systems [40-42]. Multi-sensor systems have found applications in process control, robotics, navigation, aerospace and defense systems.

The advantages of multi-sensor systems include the followings [43]:

- Failure of a single sensor does not mean complete failure of the entire system because the other sensors can continue to be used. Overlap between sensor domains gives the system some degree of redundancy. Consequently, when sensor failure occurs, the system undergoes graceful degradation and not catastrophic failure.
- Different types of sensors can be used to give a more complete picture of the environment. Thus, different sensor technologies are utilized in the same application to provide improved system performance.
- Erroneous readings from a single sensor do not necessarily have a drastic effect on the system since information about the same environment can be obtained

from other sensors. This property is particularly reinforced when there is extensive overlapping of sensor domains.

- Geographical diversity is provided by information from sensors placed at different positions in the sensed environment.
- Sensor selection is more flexible and effective as several sensors can be selected to monitor one specific task in a system. Thus, cheaper, redundant and complimentary sensors can be chosen as opposed to a single expensive sensor, while retaining the same reliability and increasing survivability [44].

The disadvantages of multi-sensor systems:

- Need to solve the data association problem where we determine the correspondence between targets in different sensors [45].
- Need to determine how the data is going to be sent between sensors. Will the data be collected at a centralized node and processed there? Processed at each sensor individually and results transmitted to centralized node? Distributed fusion to remove the need for a centralized node? [46].

2.5. Sensor Data Fusion

Target tracking addresses the problem of combining sensed data and target history to provide accurate and timely knowledge of the location of one or more moving object [7].

In order for the advantage of multi-sensor systems to be realized, it is essential that the information provided by the sensors is interpreted and combined in such a way that a reliable, complete and coherent description of the system is obtained. This is the data fusion problem. Multi-sensor fusion is the process by which information from many sensors is combined to yield an improved description of the observed system [47]. Fusion methods can either be quantitative, qualitative or hybrid of both. Quantitative methods are based on numerical techniques while qualitative ones are based on symbolic representation of information. Examples of quantitative methods include statistical decision theory, identification techniques and probabilistic theory.

Qualitative methods include expert systems, heuristics, behavioral and structural modeling [48].

2.5.1. Fusion architectures

The taxonomy of fusion architectures corresponding to different fusion algorithms can be reduced to three general categories: centralized, hierarchical and decentralized [49]. In this section only introductory notes and illustrative diagrams are presented.

2.5.1.1. Centralized architectures

A fully centralized multi-sensor system comprises a control processor with direct connections to all sensor devices. Each of these devices obtains data about the environment which is forwarded to the central processor. The central processor is responsible for collecting readings from the sensor devices and processing the information obtained. Figure 2.1. illustrates a centralized fusion system.

Conceptually, the algorithms used are similar to those for single sensor systems and hence relatively simple. Resource allocation is easy because the central processor has an over all view of the system. The central processor makes decisions based on the maximum possible information from the system. Since the central processor is fully aware of the information from each sensor and its activities there should be no possibility of task or fusion duplication. Although centralized multi-sensor systems are an improvement on single sensor systems, they have a number of disadvantages. This include severe computational loads imposed on the central processor, the possibility of catastrophic failure(due to failure of the central node), high communication overheads and inflexibility to changes of application or sensor technology [27].

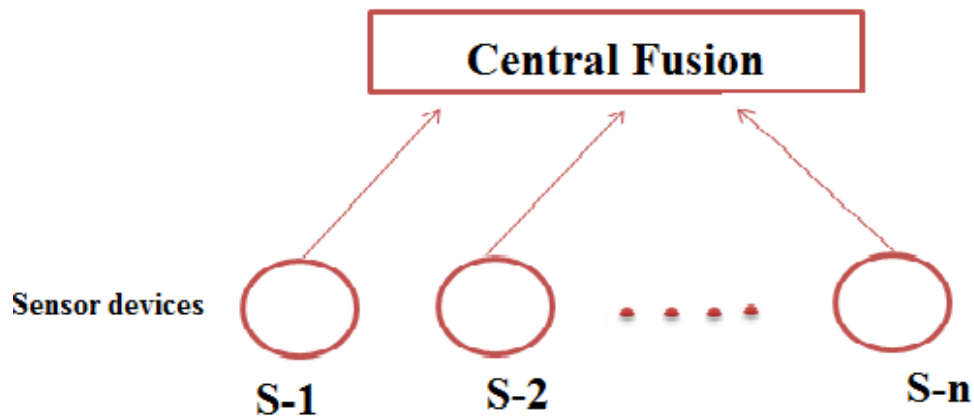


Figure 2.5. Fusion Centralized Architecture

2.5.1.2. Hierarchical architectures

A typical hierarchical architecture is shown in Figure 2.2. the principle of a hierarchy is to reduce the communication and computational problems of centralized systems by distributing data fusion tasks amongst a hierarchy of processors. In a hierarchy there is still a central processor acting as a fusion center. Processors constituting local fusion centers, locally process information and send it to the central processor. Extensive use of such systems has been made in robotics and surveillance applications. In fact, most advanced system today are generally variants of hierarchical structures. Although these systems have the advantage of distributing the computational load, they still retain some of the disadvantages associated with the centralized model. In addition to these problems, they have more implementational drawbacks which include algorithm requirements for sensor level tracking and data fusion, and vulnerability to communication bottlenecks [50,51].

2.5.1.3. Decentralized architectures

Most of the drawbacks of centralized and hierarchical architectures are resolved by using a fully decentralized architecture[52,52]. The advantage of fully decentralized systems provide the basis and motivation for the estimation and tracking algorithm

developed in this thesis. Consequently, Such systems are formally defined and discussed in chapter (3).

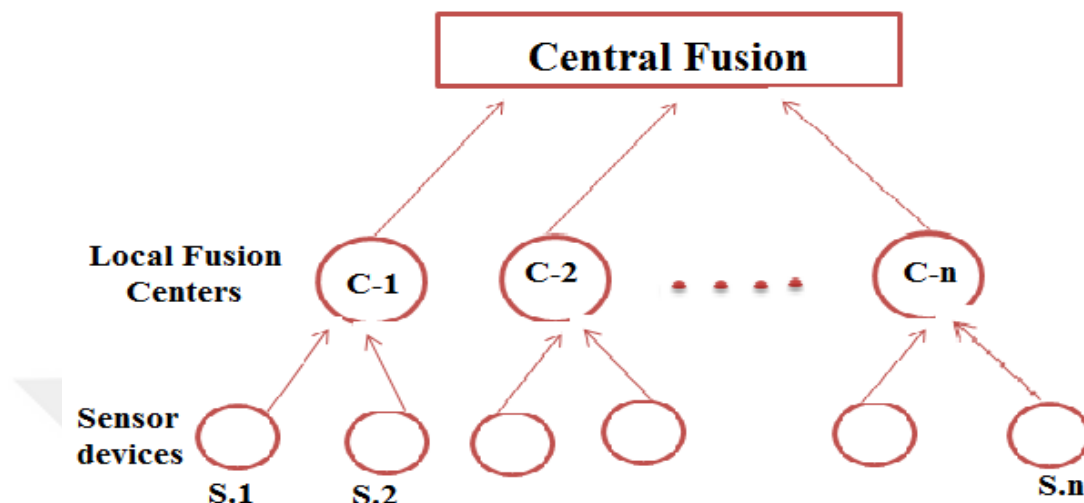


Figure 2.6. Fusion Hierarchical Architecture.

3.6. Kalman Filter

Kalman filtering, also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone, by using Bayesian inference and estimating a joint probability distribution over the variables for each timeframe. The filter is named after Hungarian Emigre Rudolf E. Kalman, although Thorvald Nicolai Thiele [54, 55] and Peter Swerling developed a similar algorithm earlier. Richard S. Bucy of the University of Southern California contributed to the theory, leading to it often being called the Kalman–Bucy filter. Stanley F. Schmidt is generally credited with developing the first implementation of a Kalman filter. He realized that the filter could be divided into two distinct parts, with one part for time periods between sensor outputs and another part for trajectory estimation for the Apollo program leading to its incorporation in the Apollo navigation computer. This Kalman filter was first described and partially developed in technical papers by Swerling (1958), Kalman (1960) [56] and Kalman and Bucy (1961).

The Kalman filter has numerous applications in technology. A common application is for guidance, navigation and control of vehicles, particularly aircraft and spacecraft. Furthermore, the Kalman filter is a widely applied concept in time series analysis used in fields such as signal processing and econometrics. Kalman filters also are one of the main topics in the field of robotic motion planning and control, and they are sometimes included in trajectory optimization. The Kalman filter has also found use in modeling the central nervous system's control of movement. Due to the time delay between issuing motor commands and receiving sensory feedback, use of the Kalman filter provides the needed model for making estimates of the current state of the motor system and issuing updated commands [57]. Kalman filters have been vital in the implementation of the navigation systems of U.S. Navy nuclear ballistic missile submarines, and in the guidance and navigation systems of cruise missiles such as the U.S. Navy's Tomahawk missile and the U.S. Air Force's Air Launched Cruise Missile. It is also used in the guidance and navigation systems of the NASA Space Shuttle and the attitude control and navigation systems of the International Space Station[58-60].

The kalman filter theory, in its various forms, has become a fundamental tool for analyzing and solving a broad class of estimation problems.

2.6.1. Reasons for using the Kalman Filter

Since the steady-state Kalman filter is identical to the Benedict–Bordner filter, the question arises as to why we should use the Kalman filter. The benefits accrued by using the Kalman filter are summarized as [61-63]:

- Provides running measure of accuracy of predicted position needed for weapon kill probability calculations; impact point prediction calculation
- Permits optimum handling of measurements of accuracy that varies with n ; missed measurements; nonequal times between measurements.
- Allows optimum use of a priori information if available.
- Permits target dynamics to be used directly to optimize filter parameters.

- Addition of random-velocity variable, which forces Kalman filter to be always stable.

In the real world the target will not have a constant velocity for all time. There is actually uncertainty in the target trajectory, the target accelerating or turning at any given time. Kalman allowed for this uncertainty in the target motion by adding a random component to the target dynamics.

2.6.2. The continuous time kalman filter

The continuous time Kalman filter is used when the measurements are continuous functions of time. Linear, time-varying state models are commonly expressed through state-space methods. Intrinsic to any state model are three types of variables: (1) input variables, (2) state variables, and (3) output variables, all generally expressed as vectors. The state model identifies the dynamic and interaction of these variables. The aforementioned variables will now be defined more formally [27].

Input-Output Variables: The input and output variables characterize the interface between the physical system and the external world. The input reflects the excitations delivered to the physical system, whereas the output reflects the signal returned to the external world.

State Variables: The state variables represent meaningful physical variables or linear combinations of such variables. For example, the state vector is a set of n variables, whose values describe the system behavior completely.

The diagram in Figure 4.1. illustrates the general composition of a linear, time-varying state model.

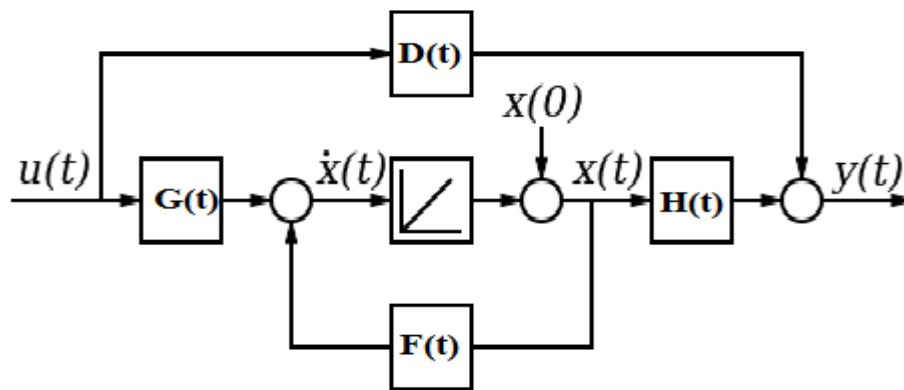


Figure 2.7. the general composition of a linear, time- varying state model [27].

From the figure, the following first-order, degree- n vector differential equation can be written [27]:

$$\dot{x}(t) = F(t) x(t) + G(t) u(t) \quad (2.1)$$

$$y(t) = H(t) x(t) + D(t) u(t) \quad (2.2)$$

with initial state:

$$x(t_0) = x_0$$

where $x(t) \in \mathbb{R}^m$ is the state vector (vector of the state variables), $u(t) \in \mathbb{R}^m$ is the system input vector, and $y(t) \in \mathbb{R}^m$ is system output vector. The matrices $F(t)$, $G(t)$, $H(t)$, and $D(t)$ have entries which are piecewise continuous, real valued functions of time.

Consider now an n -dimensional signal $x(t)$, which is referred to as the state of the system or simply the state vector. (The state vector is not a measurable quantity, analogous to the one-dimensional input signal of the Wiener problem.). Then, the following time-varying linear system driven by white noise will be assumed to represent the process [27, 42, 87]:

$$\frac{d x(t)}{dt} = F(t)x(t) + G(t)w(t) \quad (2.3)$$

where $x(t)$ = a state vector of dimension $n \times 1$, which represents the error model states, $F(t)$ = an $n \times n$ matrix, which describes the system and error model dynamics, $G(t)$ = an $n \times r$ matrix, often called the noise gain matrix [this matrix scales the white-noise inputs and sums them with the desired combinations of the states $x(t)$], which represents the effect of the input dynamics and $w(t)$ = a vector of stochastic inputs of dimensions $r \times 1$ (or zero-mean white-noise process).

The solution of the first-order time-varying vector differential equation (2.3) is given by:

$$x(t, t_0) = \Phi(t, t_0)x(t_0) + \int_{t_0}^t \Phi(t, \tau)G(\tau)w(\tau)d\tau \quad (2.4)$$

where the state transition matrix $\Phi(t, t_0)$ is a solution of the homogeneous matrix linear differential equation

$$\frac{d\Phi(t, t_0)}{dt} = F(t)\Phi(t, t_0) \quad (2.5)$$

with the initial condition

$$\Phi(t, t_0) = I$$

where I is the identity matrix.

Suppose now there are available m measurements that are linearly related to the state and are corrupted by additive white noise:

$$z(t) = H(t)x(t) + v(t) \quad (2.6)$$

Equation (2.5) states that the output, which is a measurable quantity, is an m -dimensional vector $z(t)$. This observation vector is composed of a known linear combination of the states vector with an m -dimensional noise vector $v(t)$. The $m \times n$ observation matrix $H(t)$ represents the linear relationship that exists between the state and the observation vector. Assuming the prior statistics of the noise processes to be white, Gaussian, zero mean processes, then the following apply:

$$E\{w(t)\}=E\{v(t)\}=0, E\{x(0)\}=\mu_x(0)$$

$$E\{w(t)w^T(\tau)\}=Q(t)\delta(t-\tau)$$

$$E\{v(t)v^T(\tau)\}=R(t)\delta(t-\tau)$$

$$E\{v(t)w^T(\tau)\}=0$$

Where $E\{\cdot\}$ = expectation operator,

$\mu_x(0)$ = mean of $x(0)$,

$\delta(t)$ = Dirac delta function,

$Q(t)$ = an $r \times r$ matrix, known as the covariance matrix of the state model uncertainties (system noise strength).

$R(t)$ = an $m \times m$ matrix, known as the covariance matrix of the observation noise (measurement noise strength).

It should be noted from the above equations that the white-noise sequences $w(t)$ and $v(t)$ are uncorrelated.

Given the above model, the best estimate of the state vector $x(t)$ according to a linear combination of the measurements $z(t)$ and the present state estimates $\hat{x}(t)$ so as to minimize the performance index

$$E\{[x(t)-\hat{x}(t)]^T[x(t)-\hat{x}(t)]\}=\text{minimum} \quad (2.7)$$

The solution to this problem is the well-known Kalman-Bucy filter [1, 41, 42]. The equation for the optimal estimator:

$$\frac{d\hat{x}(t)}{dt} = F(t)\hat{x}(t) + K(t)[z(t) - H(t)\hat{x}(t)] \quad (2.8)$$

The optimal Kalman filter estimates of the state $\hat{x}(t)$ are obtained from a weighted combination of predictions based upon the system model and corrections based upon the measurements. This optimal estimator, Equation (2.8) is based on the correct information of initial conditions, noise covariance and system.

The Kalman gain matrix $K(t)$ is an $n \times m$ matrix of the coefficients, which is determined by solving a nonlinear differential equation of the Riccati type. Now, define an error covariance matrix $P(t)$ by [27,42]:

$$P(t) = E\{[x(t) - \hat{x}(t)][x(t) - \hat{x}(t)]^T\} \quad (2.9)$$

The nonlinear matrix Riccati differential equation, also called the covariance equation, is [27]:

$$\frac{dP(t)}{dt} = F(t)P(t) + P(t)F^T(t) - P(t)H^T(t)R^{-1}(t)H(t)P(t) + G(t)Q(t)G^T(t) \quad (2.10)$$

Where:

$$P(t_0) = \text{cov}\{x(t_0), \hat{x}(t_0)\}$$

The propagation of the error covariance matrix $P(t)$ is independent of measurements; it depends only on the system dynamics $F(t)$ and system noise $Q(t)$. Consequently, if $H(t) = [0]$, no measurements are available, and Equation (4.10) reduces to the linear covariance equation:

$$\dot{P}(t) = F(t)P(t) + P(t)F^T(t) + G(t)Q(t)G^T(t)$$

Furthermore, if $v(t)$ in Equation (4.6) were identically zero, then the covariance matrix equation (4.10) would be singular, since $R(t)$ would be a null matrix.

The filter generates an x (estimate of the state) which minimizes the variances on the diagonal of $P(t)$. The optimal Kalman gain matrix is determined from the auxiliary relation:

$$K(k) = P(t)H^T(t)R^{-1}(t)$$

The properties of the filter that make it useful as an estimation model can be summarized as follows [27]:

- At a given time t , the filter generates an unbiased estimate \hat{x} of the state vector x ; that is, the expected value of the estimate is the value of the state vector at time t .
- The estimate is a minimum-variance estimate.
- The filter is recursive, meaning it does not store past data.
- The filter is linear or it must be linearized
- In applying the Kalman filtering theory, we make the following model assumptions:
 - The state vector $x(t)$ exists at the time t in a random environment (i.e., system dynamics) that is Gaussian with zero mean and covariance matrix $Q(t)$.
 - The state vector, which is unknown, can be estimated using observations or data samples that are functions of the state vector.
 - An observation made at a point in time t is corrupted by uncorrelated, Gaussian noise, having a zero mean and covariance matrix $R(t)$.

2.6.2. The discrete Kalman Filter

In 1960, R.E. Kalman published his famous paper describing a recursive solution to the discrete-data linear filtering problem [56]. Since that time, due in large part to advances in digital computing; the Kalman filter has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation. A very "friendly" introduction to the general idea of the Kalman filter can be found in Chapter 1 of [64], while a more complete introductory discussion can be found in [65], which also contains some interesting historical narrative. More extensive references include [66], [64], [67], [68], and [69].

2.6.2.1. The process to be estimated

The Kalman filter addresses the general problem of trying to estimate the state $X \in \mathbb{R}^n$ of a first-order, discrete-time controlled process that is governed by the linear difference equation:

$$X(k+1) = F(k)X(k) + BU(k) + W(k) \quad (2.1)$$

with a measurement $z \in \mathbb{R}^m$ that is:

$$z(k) = H(k)X(k) + V(k) \quad (2.2)$$

The random variables $W(k)$ and $V(k)$ represent the process and measurement noise (respectively). They are assumed to be independent (of each other), white, and with normal probability distributions

$$P(W) = N(0, Q) \quad (2.3)$$

$$P(V) = N(0, R) \quad (2.4)$$

The $n \times n$ matrix F in the difference equation (2.1) relates the state at time step k to the state at step $k+1$, in the absence of either a driving function or process noise. The $n \times 1$ matrix B relates the control input $U \in \mathbb{R}^1$ to the state X . The $m \times n$ matrix H in the measurement equation (2.2) relates the state to the measurement $z(k)$.

2.6.2.2. The computational origins of the filter

We define $\hat{X}^-(k) \in \mathbb{R}^n$ (note the "super minus") to be our a priori state estimate at step k given knowledge of the process prior to step k , and $\hat{X}(k) \in \mathbb{R}^n$ to be our a posteriori state estimate at step k given measurement $z(k)$. We can then define a priori and a posteriori estimate errors as

$$e^-(k) = X(k) - \hat{X}^-(k) \text{ and}$$

$$e(k) = X(k) - \hat{X}(k)$$

The a priori estimate error covariance is then

$$P^-(k) = E[e^-(k)e^-(k)^T] \quad (2.5)$$

and the a posteriori estimate error covariance is

$$P(k) = E[e(k)e(k)^T] \quad (2.6)$$

In deriving the equations for the Kalman filter, we begin with the goal of finding an equation that computes an a posteriori state estimate $\hat{X}(k)$ as a linear combination of an a priori estimate $\hat{X}^-(k)$ and a weighted difference between an actual measurement $z(k)$ and a measurement prediction $H(k)\hat{X}^-(k)$ as shown below in (2.7).

$$\hat{X}(k) = \hat{X}^-(k) + K(z(k) - H(k)\hat{X}^-(k)) \quad (2.7)$$

The difference $(z(k) - H(k)\hat{X}^-(k))$ in Equation (2.7) is called the measurement innovation, or the residual. The residual reflects the discrepancy between the predicted measurement $H(k)\hat{X}^-(k)$ and the actual measurement $z(k)$. A residual of zero means that the two are in complete agreement.

The $n \times m$ matrix K in Equation (2.7) is chosen to be the gain or blending factor that minimizes the a posteriori error covariance (2.6). This minimization can be accomplished by first substituting (2.7) into the above definition for $e(k)$, substituting that into Equation (2.6), performing the indicated expectations, taking the derivative of the trace of the result with respect to K , setting that result equal to zero, and then solving for K . One form of the resulting K that minimizes (2.6) is given by Equation (2.8).

$$K(k) = P^-(k)H(k)^T(H(k)P^-(k)H(k)^T + R)^{-1} \quad (2.8)$$

$$= \frac{P^-(k)H(k)^T}{H(k)P^-(k)H(k)^T + R(k)}$$

Looking at Equation (2.8) we see that as the measurement error covariance $R(k)$ approaches zero, the gain K weights the residual more heavily. Specifically,

$$\lim_{R(k) \rightarrow 0} K(k) = H^{-1}(k)$$

On the other hand, as the a priori estimate error covariance $P^-(k)$ approaches zero, the gain K weights the residual less heavily. Specifically,

$$\lim_{P(k) \rightarrow 0} K(k) = 0$$

Another way of thinking about the weighting by K is that as the measurement error covariance $R(k)$ approaches zero, the actual measurement $z(k)$ is "trusted" more and more, while the predicted measurement $H(k)\hat{X}^-(k)$ is trusted less and less. On the other hand, as the a priori estimate error covariance $P^-(k)$ approaches zero the actual measurement $z(k)$ is trusted less and less, while the predicted measurement $H(k)\hat{X}^-(k)$ is trusted more and more.

2.6.2.3. The probabilistic origins of the filter

The justification for Equation (2.7) is rooted in the probability of the a priori estimate $\hat{X}^-(k)$ conditioned on all prior measurements $z(k)$ (Baye's rule). For now let it suffice to point out that the Kalman filter maintains the first two moments of the state distribution,

$$E[X(k)] = \hat{X}(k)$$

$$E[(X(k) - \hat{X}(k))(X(k) - \hat{X}(k))^T] = P(k)$$

The a posteriori state estimate (2.7) reflects the mean (the first moment) of the state distribution it is normally distributed if the conditions of (2.3) and (2.4) are met. The a posteriori estimate error covariance (2.6) reflects the variance of the state distribution (the second non-central moment). In other words,

$$\begin{aligned} P(X(k)|z(k)) &= N(E[X(k)], E[(X(k) - \hat{X}(k))(X(k) - \hat{X}(k))^T]) \\ &= N(\hat{X}(k), P(k)) \end{aligned}$$

2.6.2.4. The discrete Kalman Filter algorithm

The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. As such, the equations for the Kalman filter fall into two groups: time update equations and measurement update equations. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the a priori estimates for the next time step. The measurement update equations are responsible for the feedback i.e. for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate.

The time update equations can also be thought of as predictor equations, while the measurement update equations can be thought of as corrector equations. Indeed the final estimation algorithm resembles that of a predictor-corrector algorithm for solving numerical problems as shown below in Figure 2.7.

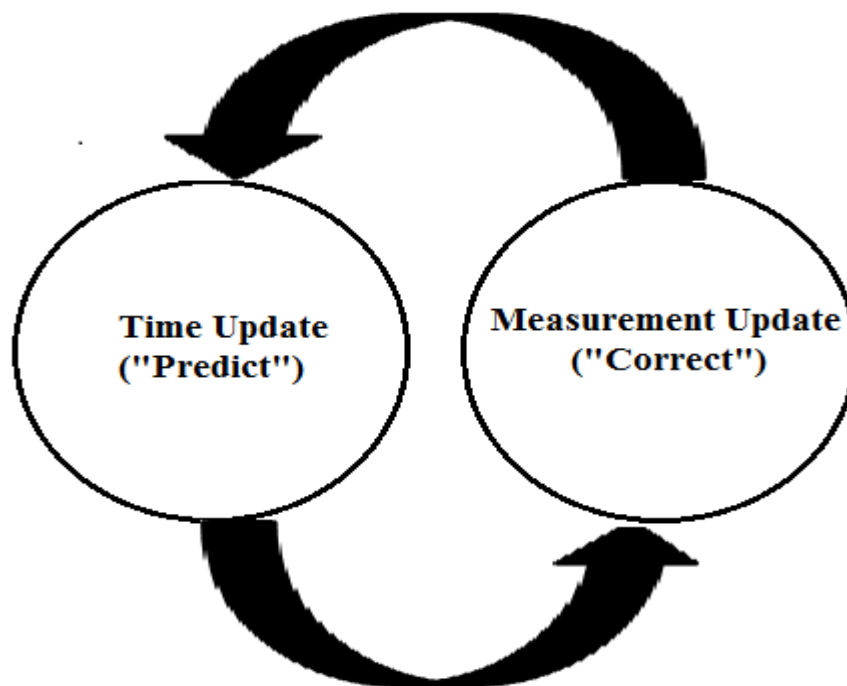


Figure 2.8. The ongoing discrete Kalman filter cycle.

The time update projects the current state estimate ahead in time. The measurement update adjusts the projected estimate by an actual measurement at that time. Notice the resemblance to a predictor- corrector algorithm.

The specific equations for the time and measurement updates are presented below in Table 2.1. and Table 2.2.

Table 2.1. Time updates equations

$$\hat{\mathbf{X}}(k+1) = \mathbf{F}(k)\hat{\mathbf{X}}(k) + \mathbf{B}\mathbf{U}(k) \quad (2.9)$$

$$\mathbf{P}^-(k+1) = \mathbf{F}(k)\mathbf{P}(k)\mathbf{F}^T(k) + \mathbf{Q}(k) \quad (2.10)$$

Table 2.2. Measurement updates equations

$$\mathbf{K}(k) = \mathbf{P}^-(k)\mathbf{H}^T(k)(\mathbf{H}(k)\mathbf{P}^-(k)\mathbf{H}^T(k) + \mathbf{R}(k))^{-1} \quad (2.11)$$

$$\hat{\mathbf{X}}(k) = \hat{\mathbf{X}}^-(k) + \mathbf{K}(z(k) - \mathbf{H}(k)\hat{\mathbf{X}}^-(k)) \quad (2.12)$$

$$\mathbf{P}(k) = (\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k))\mathbf{P}^-(k) \quad (2.13)$$

The first task during the measurement update is to compute the Kalman gain, $K(k)$. Notice that the equation given here as (2.11) is the same as (2.8). The next step is to actually measure the process to obtain $z(k)$, and then to generate an a posteriori state estimate by incorporating the measurement as in (2.12). Again (2.12) is simply (2.7) repeated here for completeness. The final step is to obtain an a posteriori error covariance estimate via equation (2.13).

After each time and measurement update pair, the process is repeated with the previous a posteriori estimates used to project or predict the new a priori estimates. This recursive nature is one of the very appealing features of the Kalman filter it makes practical implementations much more feasible than (for example) an implementation of a Weiner filter which is designed to operate on all of the data directly for each estimate. The Kalman filter instead recursively conditions the current estimate on all of the past measurements.

2.7. Literature Review

In last few decades, several researches have been developed for target localization and tracking with different sensing modalities in sensor network.

In 1986, Kuo-Chu Chang, Chee-Yee Chong and Yaakou Bar-Shalom from the Department of Electrical Engineering and Computer Science in Connecticut university in United States [70] used the joint probabilistic data association (JPDA) algorithm to track multiple targets in a cluttered environment. In their work, each node first performs the tracking functions with the JPDA algorithm using the local sensor measurements and sends the processed results to other nodes. The receiving node then fuses the information from other nodes with its local information to arrive at a better estimate. A distributed version of the JPDA algorithm, which takes into account the fusion problem, is derived by adopting the linear fusion algorithm of Chong [71] and Speyer [72].

In 1990, Carlson N.A from Integrity Systems, a small aerospace engineering firm located in Winchester [73] introduced a decentralized scheme named federated kalman

filter based on the square root of the kalman filter. The federated filter yields estimates that are globally optimal, or conservatively suboptimal, depending upon the master filter processing rate.

In 1991 and 2001, Leonid I. Perlovsky from Nichols research corporation and Oxford university [3, 74] also, applied a previously developed a hierarchical Maximum Likelihood Adaptive Neural System (MLANS) to the problem of tracking multiple objects in heavy clutter. This is a type of neural network that incorporates a model-based concept, leading to greatly increased learning efficiency compared to conventional, nonparametric neural networks [75]. This neural network has a hierarchical, two-layer structure: the bottom, signal modeling layer and the top, classification layer. In this approach the MLANS performs a fuzzy classification of all objects in multiple frames into multiple classes of tracks and random clutter [74].

In 1992, Daniel Avitzour from ELTL electronics industries in Israel [76] developed maximum-likelihood (ML) procedures for multi-target tracking use expectation-maximization (EM) for data association. The algorithm is applied to multi-target trajectory estimation of constant-velocity targets from direction measurements taken by a moving sensor. He assumed that the targets are independent, their number and probability of detection are known, an unknown number of targets can be handled by performing ML estimation for different target numbers and selecting the number leading to greatest likelihood. The sensor moves in a known trajectory in the plane, taking a snapshot of the scene at prescribed instants. As a result, the measurement from a target is the direction from sensor to target, contaminated by additive Gaussian noise of zero mean and known variance. Using these measurements the estimates of the target position and velocities are achieved.

In 2001, Rickard Karlsson and Fredrik Gustafsson from the department of electrical engineering in Linkoping university in Sweden [77] introduced a Bayesian data association method (Monte Carlo data association) based on the particle filter idea and the joint probabilistic data association (JPDA) hypothesis calculations for multi-target tracking in a cluttered environment. A comparison between the JPDA and the

probabilistic multi-hypothesis tracking method is also made. They have assumed time-invariant target models, and they used the same Bayesian approach as in [78], for the estimation. In addition, they extend the idea and introduce hypothesis calculations according to the JPDA method. In the proposed method the clutter or false alarm model is assumed uniformly distributed in the volume and the number of false alarms for a given time is assumed to be Poisson distributed and the RMSE is used to describe the performance. As a result, they reached that for non-linear problems and problems where the noise distribution is highly non-Gaussian, the proposed algorithms may increase the overall tracking performance.

In 2006, N.Shrivastava, R.Mudumbai, U.Madhow and S.Suri from the department of computer science in California University [79] identified the fundamental limits of tracking performance that can be achieved with binary proximity sensors. They designed a geometric algorithm using an Occam's razor approach to compute a piecewise linear path that approximates the trajectory within this fundamental limit of accuracy.

In 2009, Songhwai Oh, Stuart Russell, and Shankar Sastry from the department of electrical engineering and computer sciences in California University [80] developed Markov Chain Monte Carlo (MCMC) data association method for solving a real-time multi-target tracking problems in a cluttered environment. In statistics, MCMC methods are a class of algorithms for sampling from a probability distribution based on constructing a Markov Chain that has the desired distribution as its equilibrium distribution. The state of the chain after a number of steps is then used as a sample of the desired distribution. The quality of the sample improves as a function of the number of steps [82]. The presented method is tested to achieve two technical results. The first is a theorem showing that, when the number of targets is fixed, single-scan MCMC data association is a fully polynomial randomized approximation scheme for joint probabilistic data association (JPDA). And the second technical result is the complete specification of the transition structure for a multi-scan version of MCMC that includes detection failure, false alarms, and track initiation and termination [80].

In 2011, Sheldon Xu and Anthony Chang from Cornell University [82] presented an approach for training a function for dynamically adjusting Kalman filter measurement error covariance, in attempt to tune a Kalman filter to favor better suited tracking algorithms, and penalize ill-suited ones, during runtime. Unfortunately, they have mixed results. Though their method increased tracking accuracy in one type of environment, it was matched or outperformed by the baseline CamShift algorithm in most cases. Thus, they were only partly successful in increasing robustness over CamShift.

In 2014 Michael Beard, Ba Tuong Vo and Ba-Ngu Vo from the defence science and technology organization and the department of electrical and computer engineering in Australia [83] developed a principled Bayesian solution to the multi-target tracking problem based on the theory of random finite sets and implementation based on the generalised labelled multi-Bernoulli filter.

In 2014, K. Meier and A. Desai from Ira A. Fulton College of Engineering and Technology [84] implemented a Kalman filter to track the range, range rate, bearing, and bearing rate of a maneuvering aircraft with unknown varying accelerations using sensors that only measure the bearing angle and the range of the aircraft

CHAPTER 3. MATERIALS AND METHODS

This chapter discusses the concepts of decentralized Kalman filtering with parallel processing capabilities, for use in distributed multi-sensor multi-target systems and how to use the DKF to estimate and separate the tracks from different sensors represent the targets. Also, we will explain our MATLAB implementation and the experiments we did.

3.1. Decentralized Kalman Filter

It is a well-known fact that the conventional kalman filter provides the best sequential linear, unbiased estimate, or globally optimal estimate, when noise processes are jointly Gaussian. However, in practice the use of a large centralized kalman filter may not be feasible due to such factors as: (1) computational burden, (2) high data rate and (3) accuracy specifications. As a result decentralized or parallel versions of standard kalman filter began to receive increasing attention in recent years [27,85-90].

The study of decentralized filters began as an attempt to reduce throughput. Figure.1 illustrates the decentralized or parallel filter architecture. As the figure shows the decentralized kalman filtering can be defined as two stage data processing techniques which process data from multi sensor system [91]. In the first stage, each local processor uses its own data to make a best local estimate. These estimates are then obtained in parallel processing mode. The local estimates are then fused by a master filter to make a best global estimate of the state vector of the master system. In this design, a number of sensor-dedicated local filters run in parallel, the outputs being fused into a master filter, yielding estimates that are globally optimal. As a result, the filtering duties are divided so that each measurement is processed in a local filter, which contains only those states that are directly relevant to the measurement.

The state estimates and covariances of a given local filter are then weighted and combined in the master filter, whose own state is the fusion of all the local filter states. This hierarchical filtering structure provides some important improvements: (1) increasing the data throughput by local parallel processing, (2) reducing the required bandwidth for information transmission to a central processor, and (3) allowing for a more fault-tolerant system design [27].

A decentralized Kalman filtering system consists of a bank of N local subsystems, each local filter generating estimates of the full state vector. All decentralized Kalman filtering mechanizations require some processing at the local level, that is, at the subsystem level [86]. Depending on the type of application, these local processed data may then be sent to a central processing unit (also referred to as a master filter, fusion center, or collating unit) that combines the local results in order to generate a global or central result. The decentralized mechanization discussed here is the most attractive one, since it require no inter processor communications and no communication from the collating filter back to the local filters (that is, bidirectional communication). In other words, the local Kalman filters are stand alone, in that each of them generates estimates based solely on its own available raw data. This allows the local filters to run in parallel at a faster rate [27].

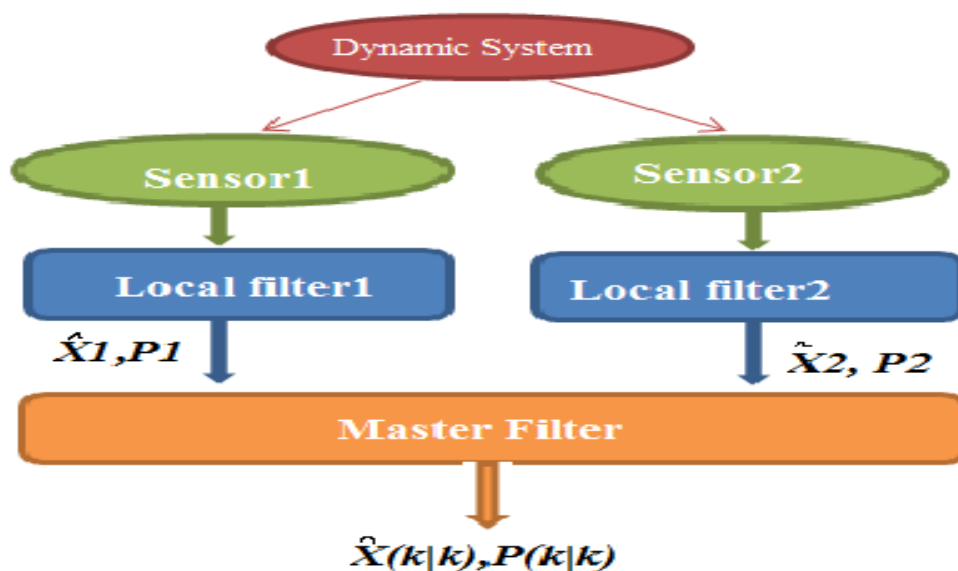


Figure 3.1. The decentralized kalman filter architecture

3.2. Problem Formulation

The standard discrete-time linear target dynamic models of the tracking systems associated with the i -th local sensor are defined as [27] (p.351):

$$X_i(k) = F_i(k)X_i(k-1) + B_i(k)U_i(k) + G_i(k)W_i(k); \quad i = 1, \dots, N \quad (3.1)$$

Where $X_i(k)$ is the state vector corresponding to the i -th sensor at time k ; it consist of all the parameters that are estimated by the filter, F_i is a known transition matrix; it transforms a given state at time $(k-1)$ to another state at time k , $U_i(k)$ is an input control matrix with gain B_i , $W_i(k)$ is the white gaussian process noise with zero mean and covariance $Q_i(k)$, $G_i(k)$ is the noise gain matrix for the i -th sensor and N is the total distributed sensors in the tracking system.

The measurements of sensor i from the target at time k can also be performed according to the model below [27] (p.351):

$$z_i(k) = H_i(k)X_i(k) + V_i(k) \quad ; \quad i = 1, \dots, N \quad (3.2)$$

Where $z_i(k)$ is the measurement or observation vector at time k received by the i -th sensor, $H_i(k)$ is the measurement or observation matrix corresponding to the i -th sensor. It relates the components of the true state vector to the measurements. Moreover, the measurement matrix is computed at each measurement time and contains the partial derivatives of the measurement equations and finally $V_i(k)$ is the measurement noise associated with zero mean and known covariance R_i .

The true state of the system $X_i(k)$ cannot be directly observed, and the kalman filter (KF) provides an algorithm to determine an estimate $\hat{X}_i(k)$. Kalman filter algorithm involves two stages: prediction and measurement update [92] (p.347). The standard local kalman filter equations for the prediction stage are given by equation (3.3) and (3.4) [27] (p.350).

$$\hat{X}_i(k|k-1) = F_i(k)\hat{X}_i(k-1|k-1) + B_i(k)U_i(k) \quad (3.3)$$

$$P_i(k) = F_i(k)P_i(k-1|k-1)F_i^T(k) + G_i(k)Q_i(k)G_i^T(k) \quad (3.4)$$

And the measurement update equations are given by equation (5) and (6) [27] (p.350).

$$\hat{X}_i(k|k) = \hat{X}_i(k|k-1) + K_i(k)[z_i(k) - H_i(k)\hat{X}_i(k|k-1)] \quad (3.5)$$

$$P_i(k|k) = P_i(k|k-1) - K_i(k)H_i(k)P_i(k|k-1) \quad (3.6)$$

Where $P_i(k)$ is the error covariance matrix corresponding to the i -th sensor at time k . that is, this matrix represents the covariance of the difference between the true state vector $X_i(k)$ and the estimated state vector $\hat{X}_i(k)$ [27] (p.97):

$$P_i(k) = E([X_i(k) - \hat{X}_i(k)][X_i(k) - \hat{X}_i(k)]^T) \quad (3.7)$$

In minimizing the diagonal elements of $P_i(k)$, each of the filter estimation rms errors is minimized. The error sources may be initial uncertainties in $\hat{X}_i(0)$, the process $W_i(k)$, or the measurements noise $V_{i(k)}$.

And $K(k)$ represent the kalman gain which can be calculated using equation (3.8) [27] (p.121):

$$K_i(k) = P_i(k|k-1)H_i(k) \cdot [H_i(k)P_i(k|k-1)H_i^T(k) + R_i(k)]^{-1} \quad (3.8)$$

The global target dynamic model and measurement model of the central tracking system are defined by equation (9) and (10) below [27] (p.352):

$$X(k) = F(k)X(k-1) + G(k)W(k) \quad (3.9)$$

$$Z_i = C_i(k)X(k) + V_i(k); \quad i = 1, \dots, N \quad (3.10)$$

Where :

$$X(k) = [X_1^T(k), X_2^T(k), \dots, X_L^T(k)]^T$$

$$F(k) = \text{block diag}(F_1(k), F_2(k), \dots, F_L(k))$$

$$G(k) = \text{block diag}(G_1(k), G_2(k), \dots, G_L(k))$$

$$W(k) = [W_1^T(k), W_2^T(k), \dots, W_L^T(k)]^T$$

$$Z_i(k) = [z_1^T(k), z_2^T(k), \dots, z_{L_i}^T(k)]^T$$

$$C_i(k) = [C_1^T(k), C_2^T(k), \dots, C_{L_i}^T(k)]^T$$

$$V_i(k) = [V_1^T(k), V_2^T(k), \dots, V_{L_i}^T(k)]^T$$

Where L is the total number of the target in the surveillance region, L_i are the targets seen by the i -th sensor and $C_i(k)$ is the measurement or observation matrix for the global filter which is calculated using equation (3.11).

$$C_i(k) = H_i(k)D_i(k) \tag{3.11}$$

$D_i(k)$ Is a time varying matrix which is used to associate the local kalman filter with the central processor and for track to track correlation in the fusion Algorithm [93]. $D_i(k)$ Reflect which of the targets are observed by the i -th sensor [27]. The sensor fusion algorithm based on track to track correlation has recently been considered in [93].

From all the above equations the central or global equations for the prediction stage and the measurement update can be defined as below:

$$P(k) = F(k)P(k-1|k-1)F^T(k) + G(k)Q(k)G^T(k) \tag{3.12}$$

$$\hat{X}(k|k-1) = F(k)\hat{X}(k-1|k-1) \tag{3.13}$$

$$K_i(k) = P(k|k-1)C_i(k) \cdot [C_i(k)P(k|k-1)C_i^T(k) + R_i(k)]^{-1} \quad (3.14)$$

$$M(k) = [\sum_{i=1}^N D_i^T(k) D_i(k)]^{-1} \quad (3.15)$$

$$P(k|k) = [I - \sum_{i=1}^N K_i(k)C_i(k) \cdot M(k)]P(k|k-1) \quad (3.16)$$

$$A(k) = \left[I - \sum_{i=1}^N K_i(k)C_i(k) \cdot M(k) \right] F(k) \quad (3.17)$$

$$PHI = P(k|k)D_i^T(k) P_i^{-1}(k|k) \quad (3.18)$$

$$SI(k|k) = A(k)\hat{X}(k-1|k-1) - \sum_{i=1}^N PHI(k) \hat{X}_i(k-1|k-1) \cdot M(k); \quad (3.19)$$

$$\hat{X}(k|k) = SI(k|k) + \sum_{i=1}^N PHI_i(k)\hat{X}_i(k|k) \cdot M(k) \quad (3.20)$$

3.2.1. Fusion Algorithm

In this approach, the local sensor performs its own data association and estimation for those targets seen by it. A time-varying matrix $D_i(k)$ used to associate the local processors with the central processor and for track to track correlation is defined in our fusion algorithm. The $D_i(k)$ arises in decentralized filtering which indicates the targets are tracked by each individual local sensor. The fusion algorithm in the central processor is applied to combine the local sensor track results with this matrix. Such a matrix is defined as follows:

$$D_i(k) = [D_i^{jk}]; \quad j = 1, 2, \dots, L_i, k = 1, 2, \dots, L \quad (3.21)$$

where is an n by n block that indicates which targets are seen by the i -th sensor, and the n is the order of state vector for each target. For example, the jk -th block is set to be the identity matrix if the k -th target is seen by this sensor. Figure 3.2. show the basic decentralized kalman filter flow diagram.

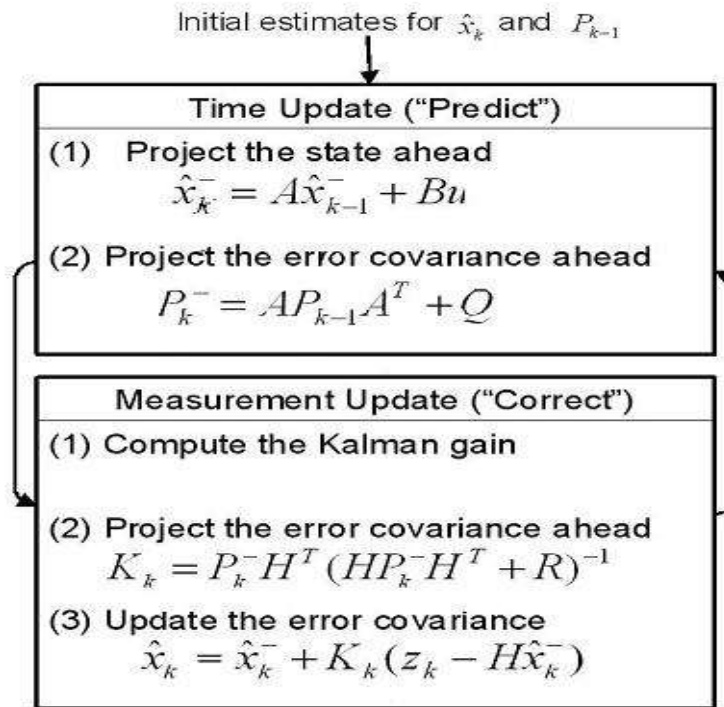


Figure 3.2. The decentralized Kalman filter flow diagram [94].

3.3. MATLAB Implementation

The algorithm is implemented using MATLAB program. In this work, two radar systems are used as sensors with two local kalman filters to estimate the position of an aircrafts and then they transmit these estimates to a central processor, which combines this information to produce an optimal estimate and separate the tracks from different sensors represent the targets. The proposed model is tested four times as follow:

3.3.1. Two sensors tracking the same target

Firstly, the probosed model are tested when their is one target in the sensors region, and any one of the sensors tracked the same target as shown in figure 3.3. when the two sensor are tracking the same constant-velocity target. The target state space is defined in terms of 2D cartesian position and velocity vectors.

$$X = [X \ Y \ \dot{X} \ \dot{Y}]^T$$

With initial conditions:

Target1# [0km 5km 0.4km/s -0.2km/s]

And the sampling interval is chosen to be 1 second

$$F_i(k) = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, H_i(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

and the standard deviation of system noise is 0.03km. The white Gaussian process and measurement noise is generated using a random number generator function where the standard deviation of measurement noise is 0.3km for position error and 0.03km/s for velocity error for both sensors.

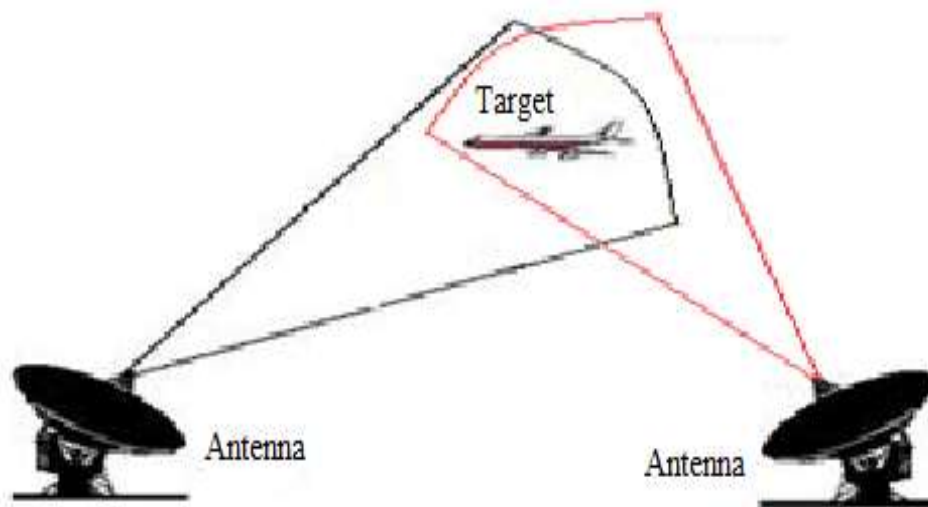


Figure 3.3. The two sensors tracking the same target.

3.3.2. Two sensors tracking two different targets

Secondly, the proposed model are tested when there are two targets in the sensors region, and any one of the sensors tracked one of the targets as shown in Figure 3.4. when the two sensors are tracking two constant-velocity targets. The targets state space is defined in terms of 2D cartesian position and velocity vectors.

$$X = [X \ Y \ \dot{X} \ \dot{Y}]^T$$

With initial conditions:

Target1# [0km 5km 0.4km/s -0.2km/s]

Target2# [0km 2km 0.4km/s 0.2km/s]

And the sampling interval is chosen to be 1 second

$$F_i(k) = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad H_i(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

and the standard deviation of system noise is 0.03km. The white Gaussian process and measurement noise is generated using a random number generator function where the standard deviation of measurement noise is 0.3km for position error and 0.03km/s for velocity error for both sensors.

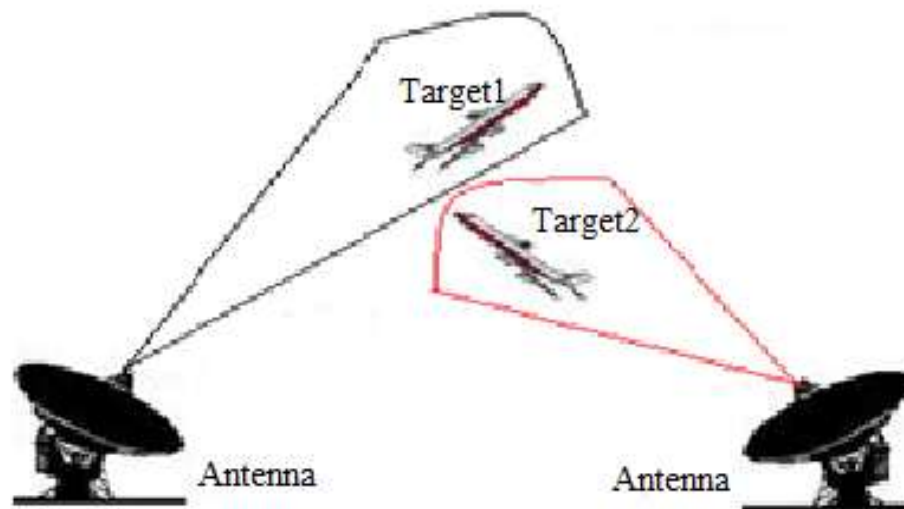


Figure 3.4. The two sensors tracking two different targets.

3.3.3. Two sensors tracking same two targets

Thirdly, the proposed model are tested when there are two targets in the sensors region, and any one of the sensors tracked both of them as shown in Figure 4.5. When

the two sensor are tracking two constant-velocity targets. The targets state space is defined in terms of 2D cartesian position and velocity vectors.

$$X = [X \ Y \ \dot{X} \ \dot{Y}]^T$$

With initial conditions:

Target1# [0km 5km 0.5km/s -0.2km/s]

Target2# [0km 1km 0.5km/s 0.2km/s]

And the sampling interval dt is chosen to be 1 second

$$F_i(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad H_i(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and the standard deviation of system noise is 0.03km. The white Gaussian process and measurement noise is generated using a random number generator function where the standard deviation of measurement noise is 0.3km for position error and 0.03km/s for velocity error for both sensors.

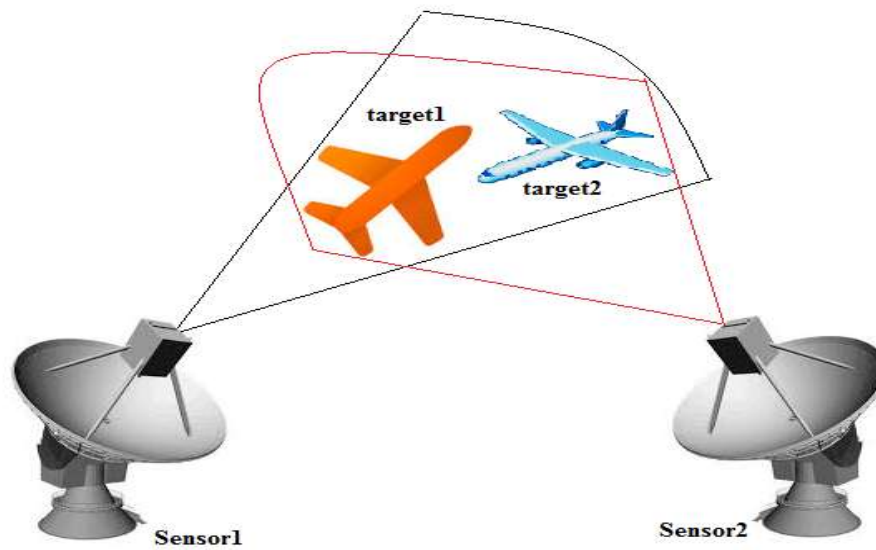


Figure 3.5. The two sensors tracking same two targets.

3.3.4. Two sensors tracking three different targets

Finally, the proposed model are tested when there is three targets in the sensors region, and any one of the sensors tracked two targets as shown in Figure 3.6. The target state space is defined in terms of 2D cartesian position and velocity vectors.

$$X = [X \ Y \ \dot{X} \ \dot{Y}]^T$$

With initial conditions:

Target1: [0km 5km 0.5km/s -0.1km/s]

Target2: [0km 1km 0.5km/s 0.1km/s]

Target3: [0km 3km 0.5km/s -0.1km/s]

The sampling interval dt is chosen to be 1 second.

$$F_i(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and the standard deviation of system noise is 0.03 km. The white Gaussian process and measurement noise is generated using a random number generator function where the standard deviation of measurement noise is 0.3 km for position error and 0.03 km/s for velocity error for both sensors.

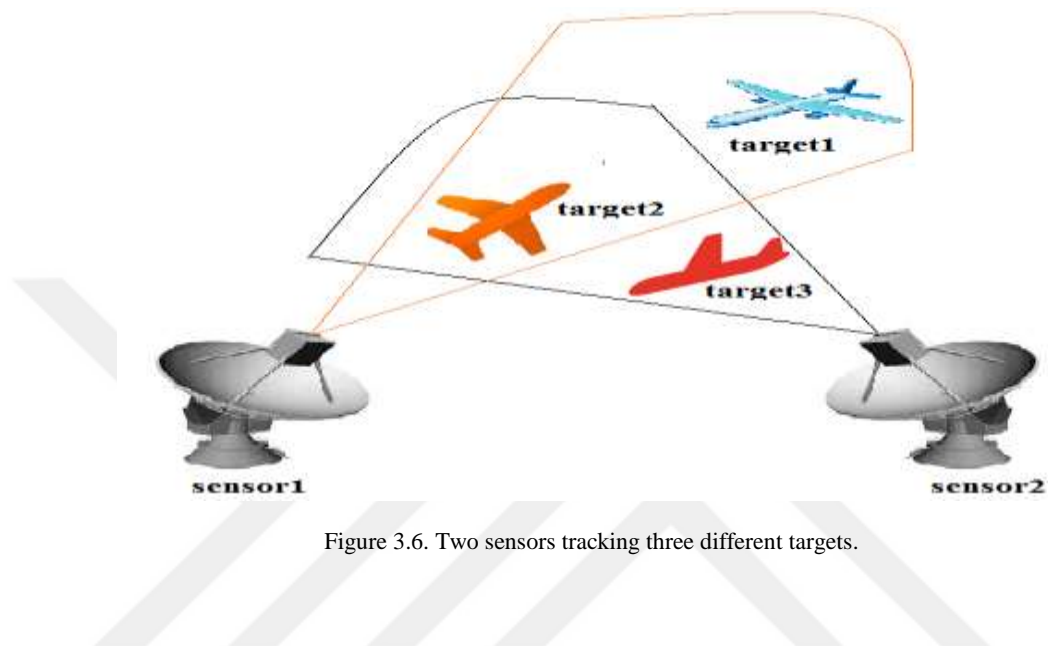


Figure 3.6. Two sensors tracking three different targets.

CHAPTER 4. RESULTS AND DISCUSSION

This chapter present and discusses the results of the experiments we explained in the previous chapter. We designed and analyzed a decentralized Kalman filter technique for use in estimating the position of an aircrafts and separate the tracks from different sensors represent the targets. The achieved numerical simulations results are as follow:

4.1. The Results Of Two Sensors Tracking The Same Target

The experiment results for the first model when the two sensors are tracking the same constant-velocity target as we discussed in (3.3.1) are shown in Figure 4.1.-4.2. and 4.3.

We can see that the central kalman filter estimates as shown on Figure 4.3. is more accurate than the local kalman filter estimates for sensor1 and sensor2 which are represented by green and red lines in Figure 4.1. and Figure 4.2. respectively. We can see also that the central processor detected that the data came from sensor1 and sensor2 representing same target which is our main problem.

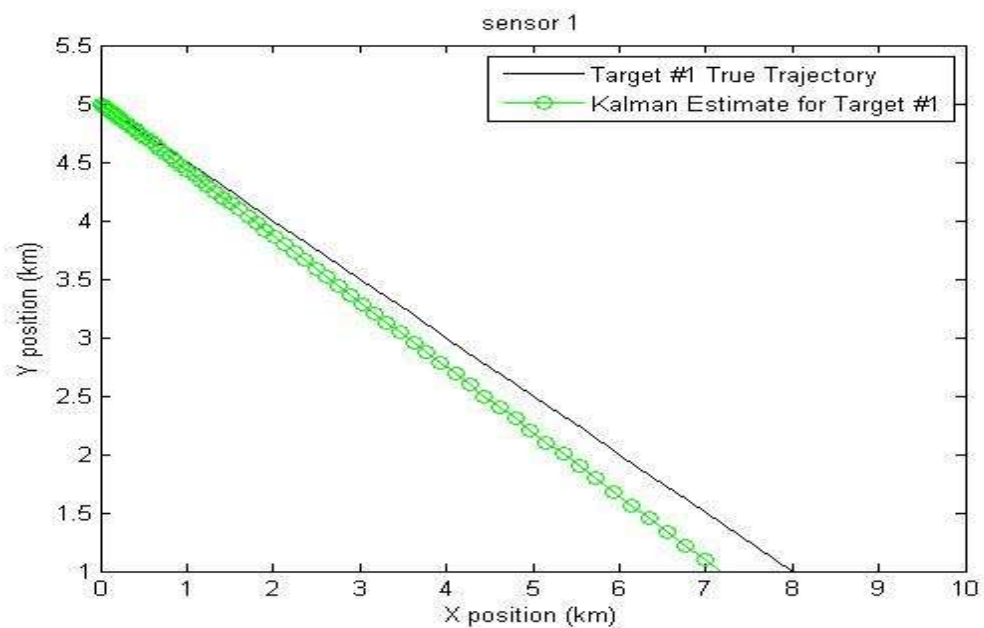


Figure 4.1. Tracking results of sensor 1 for one target state.

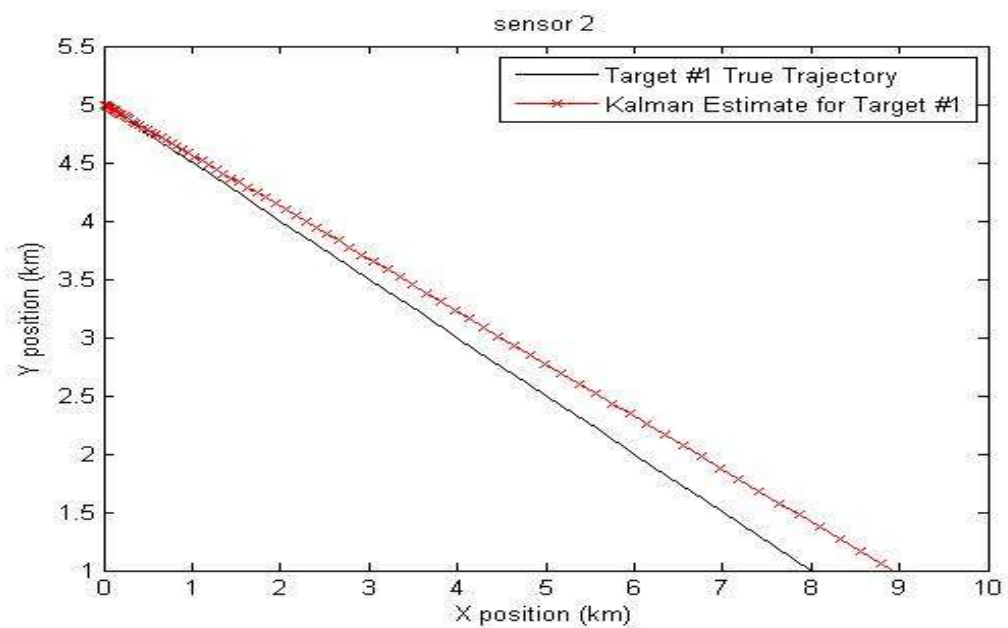


Figure 4.2. Tracking results of sensor 2 for one target state.

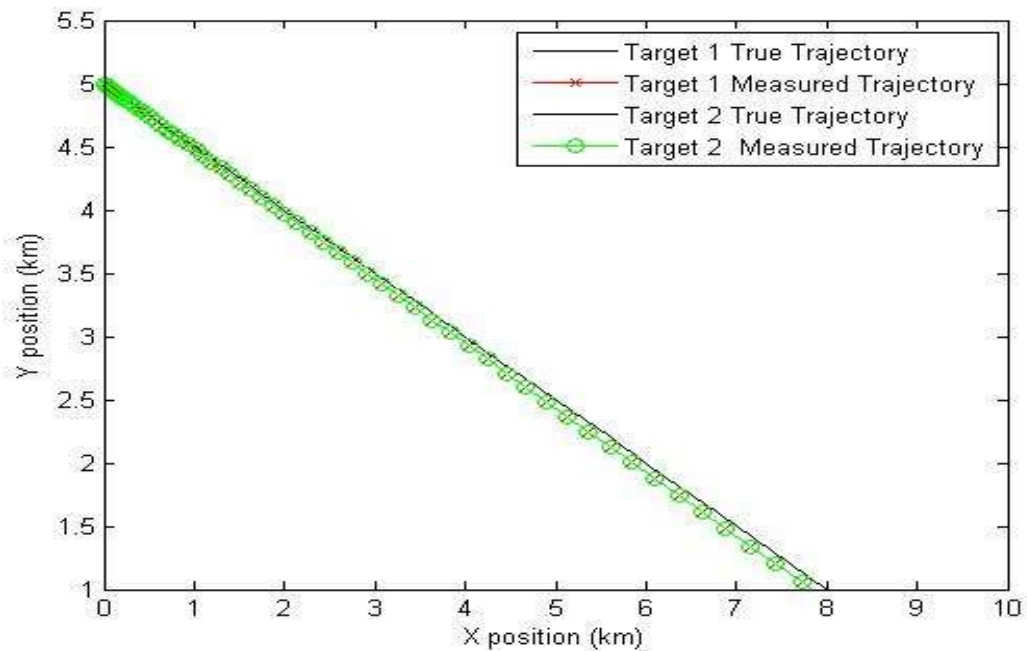


Figure 4.3. Tracking results of the central processor for one target state.

4.2. The Results Of Two Sensors Tracking Two Different Targets

The experiment results for the second model when the two sensors are tracking two constant-velocity targets as we discussed in (3.3.2) are shown on Figure 4.4-4.5 and 4.6. We can see that the central processor as shown on figure 4.6 has the ability to analyze and separate the data coming from sensor1 (target1) and sensor2 (target2) Figure 4.4. and Figure 4.5. to determine that it is representing two different targets.

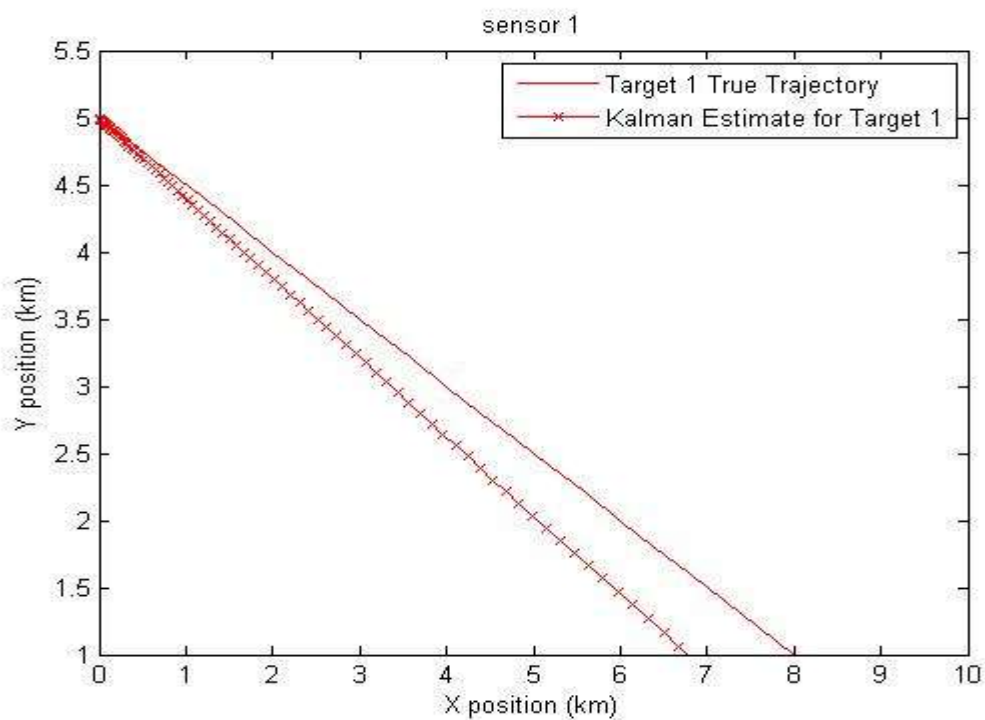


Figure 4.4. Tracking results of sensor 1 when the two sensors are tracking two different targets

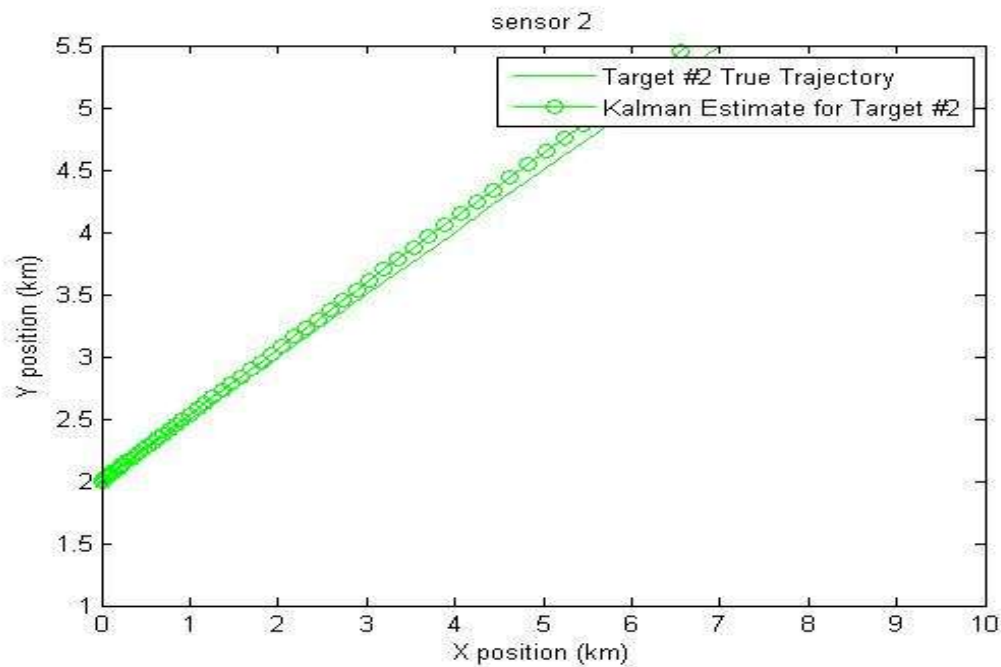


Figure 4.5. Tracking results of sensor 2 when the two sensors are tracking two different targets

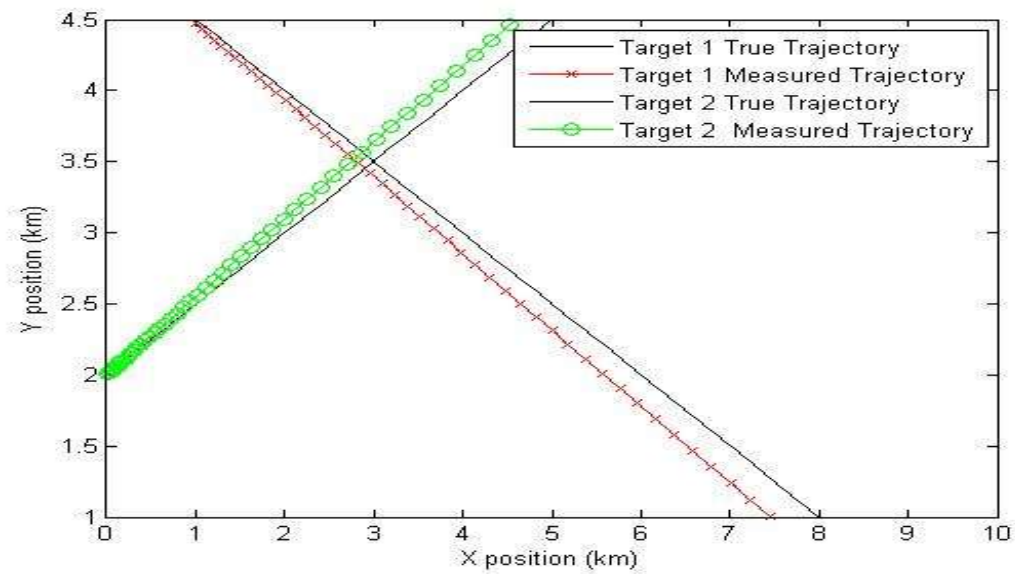


Figure 4.6. Tracking results of the central processor when the two sensors are tracking two different targets.

4.3. The Results Of Two Sensors Tracking Same Two Targets

Figure 4.7-4.8. and 4.9. shows the experiment results for the third model when the two sensors are tracking the same two constant-velocity targets as discussed on section (3.3.3). We can see that sensor1 tracked target1 and target2 as in Figure 4.7. and sensor2 also tracked target1 and target2 as in Figure 4.8. which mean that the two targets are tracked by the two sensors in the same time and that is detected by the central processor as shown on Figure 4.9.

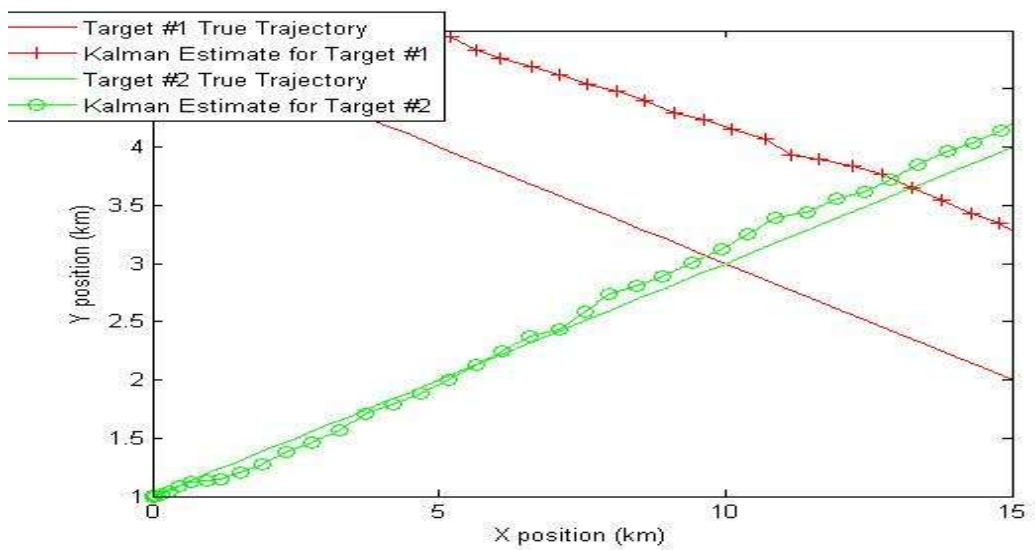


Figure 4.7. Tracking results of sensor 1 when the two sensors are tracking same two targets.

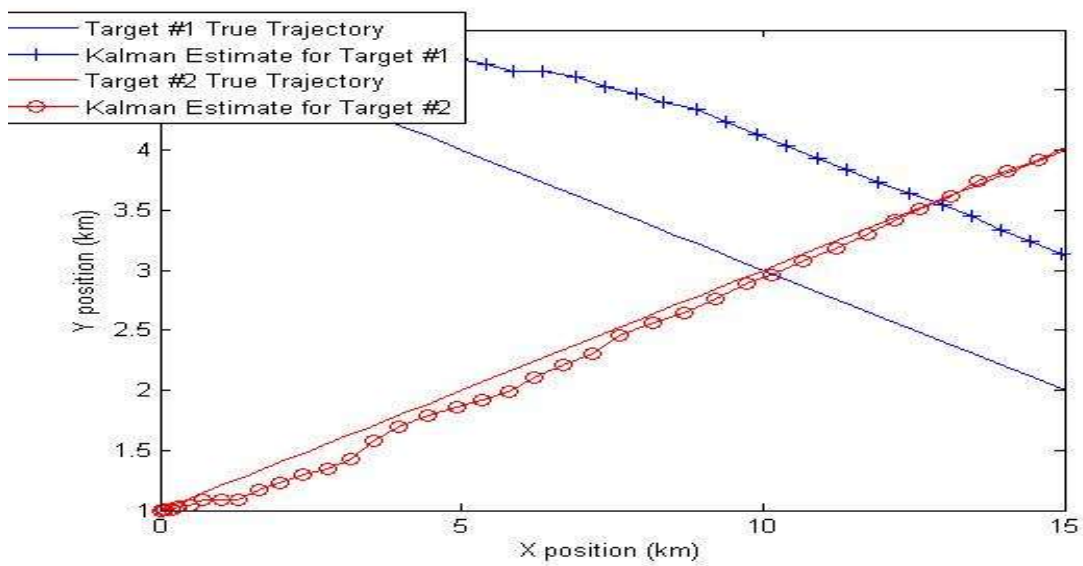


Figure 4.8. Tracking results of sensor 2 when the two sensors are tracking same two targets.

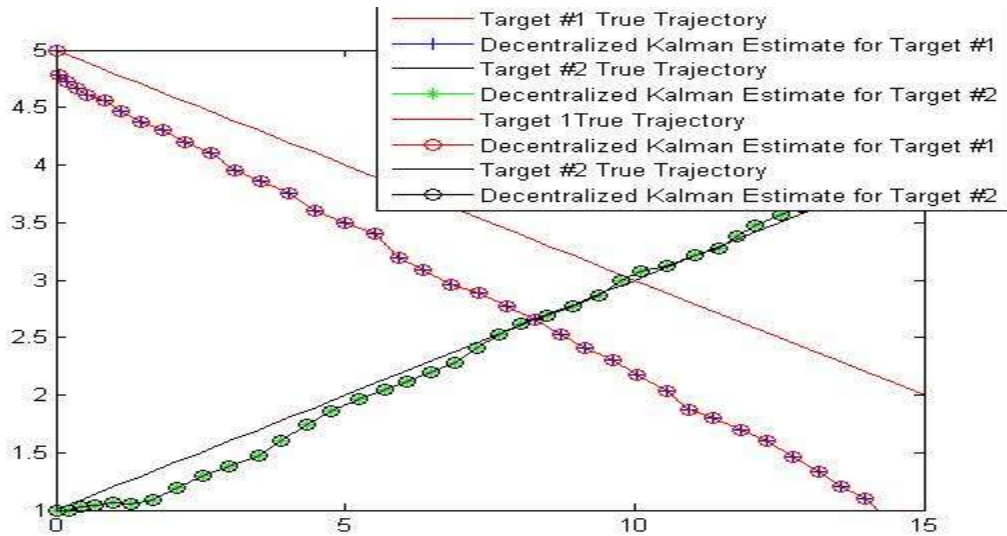


Figure 4.9. Tracking results of the central processor when the two sensors are tracking same two targets.

4.4. The Results Of Two Sensors Tracking Three Different Targets

The experimental results in Figure 4.10.-4.11. and 4.12. shows that the proposed algorithm gived global positions estimates as shown in Figure 4.12. comparing to the local filter estimates as in Figure 4.10. and Figure 4.11. and it have the ability to differentiate between targets. We can see that sensor1 tracked target1 and target2 when sensor2 tarcked each of target2 and target3 which meen that target2 are tracked by the two sensors in the same time and that is detected by the central processor as shown on Figure 4.12.

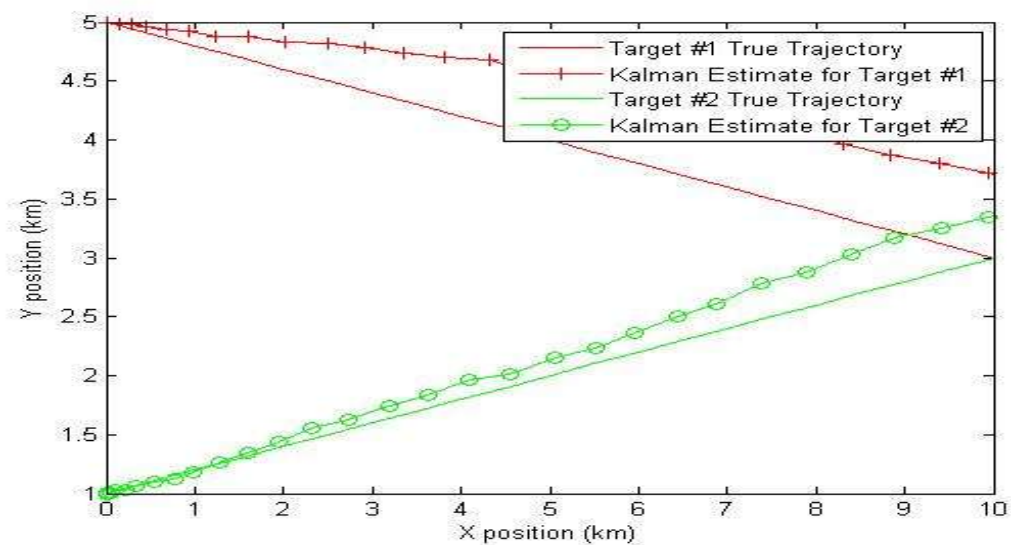


Figure 4.10. Tracking results of sensor 1 when the two sensors are tracking three different targets.

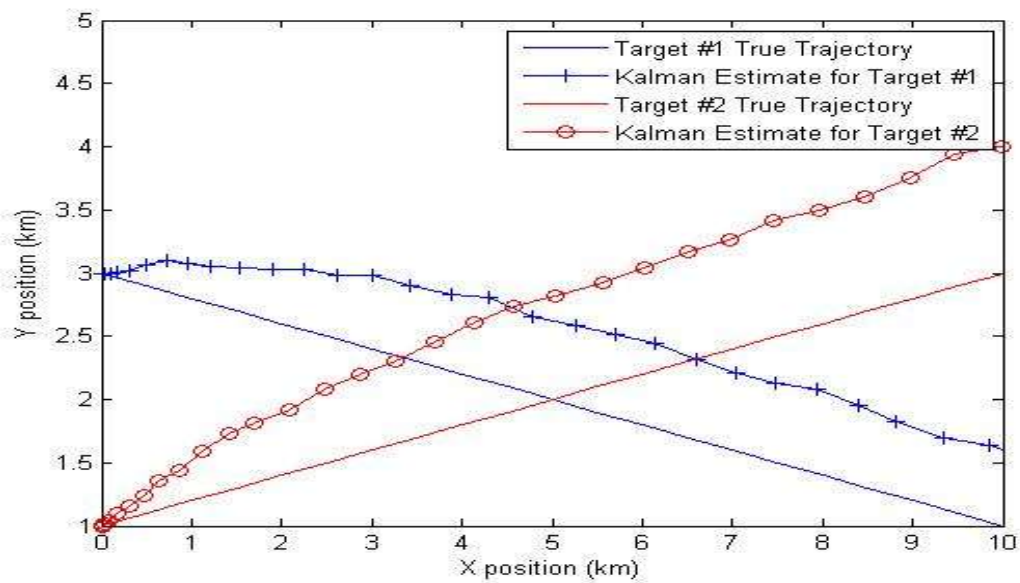


Figure 4.11. Tracking results of sensor 2 when the two sensors are tracking three different targets.

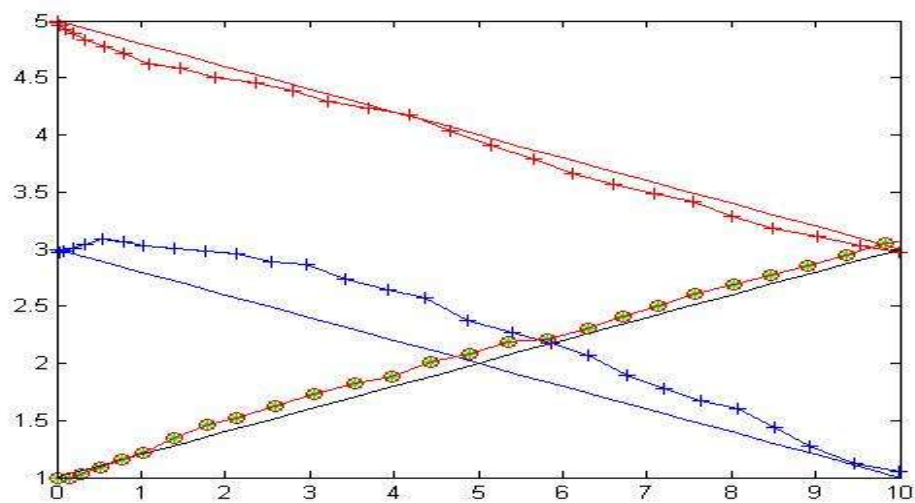


Figure 4.12. Tracking results of the central processor when the two sensors are tracking three different targets.

From all the above results we can see that the central processor has the ability to separate between the data representing same targets and the data representing different targets. In addition, when the target tracked by the two sensors at same time, the central processor give better estimate than the single track by one sensor.

CHAPTER 5. CONCLUSION AND SUGGESTED FUTURE WORK

5.1. Conclusions

This thesis has presented the decentralized kalman filter algorithm for multi-sensor multi-target tracking problems. It is used to differentiate the tracking objects whether it representing same target or different targets which is a real problem in many applications such as air traffic control and missile defense. Numerical simulations and previous results showed that:

- Decentralized kalman filter algorithm could be successfully used to help in tracking targets using multi-sensors systems.
- The proposed algorithm gived globel positions estimates comparing to the local filter estimates
- Proposed algorithm have the ability to differentiate between targets.
- Decentralized kalman filter algorithm succeeded to differentiate between targets in the states of single target, two different targets tracked by differnet sensors, two different targets tracked by both of the sensors and three targets tracked by two sensors with one target common between them.
- This decentralization make the ability to test a single sensor performance easy, since the output of each local sensor filter can be tested. The sensor have low performance can be removed from the sensor network befor it affects the total filter output.
- Also this algorithm distribute the computational load and targets information among many sensors.
- When the target tracked by the two sensors at same time the central processor give better estimate than the single track by one sensor.

5.2. Suggested Future Work

In the future work we can consider the following points:

- Design the hardware part of the proposed work and test it in the real life.
- Test the proposed algorithm when a random sensor is tracking multi-targets provided that some of the targets are common between all the sensors taking into account the speed and acceleration expectations.
- study the effect of increasing the number of sensors and targets in our system to the over all accurecy.
- Study the ability of the proposed algorithm to detect and classify the tracked targets type .

REFERENCES

- [1] M. Zahir, O. Mourad, and O. Abdelaziz, Multiple target tracking using cheap joint probabilistic data association multiple model particle filter in sensors array. *International Journal of Artificial Intelligence & Applications*, vol. 3, no. 4, pp. 1–19, 2012.
- [2] Alireza Toloei, And Saeid Niazi, State estimation for target tracking problems with nonlinear kalman filter algorithms. *International Journal of Computer Applications*, vol. 98, no.17, 2014.
- [3] L. I. Perlovsky, *Neural Networks and intellect: using model-based concepts*. vol. 16, no. 2. 2005.
- [4] R. Bale And G. Widmann, Navigated CT-guided interventions. *Minim. Invasive Ther. Allied Technol.*, vol. 16, no. 4, pp. 196–204, 2007.
- [5] N. Shrivastava, R. Mudumbai, U. Madhow, And S. Suri, Target tracking with binary proximity sensors: fundamental limits, minimal descriptions, and algorithms. *Proc. 4th Int. Conf. Embed. networked Sens. Syst.*, pp. 251–264, 2006.
- [6] W. Chen, J. C. Hou, And L. Sha, Dynamic clustering for acoustic target tracking in wireless sensor networks. 2003.
- [7] J. Liu, M. Chu, And J. E. Reich, Multitarget tracking in distributed sensor networks. *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 36–46, 2007.
- [8] W. Pedrycz, A. Sillitti, And G. Succi, Computational intelligence: An introduction. *Studies in Computational Intelligence*, vol. 617. pp. 13–31, 2016.
- [9] Moustapha Doumiati, Ali Charara, Alessandro Victorino, Daniel Lechner, Vehicle dynamics estimation using kalman filtering. 2012.
- [10] H. Wu, S. Chen, B. Yang, And K. Chen, Feedback robust cubature kalman filter for target tracking using an angle sensor. *Sensors*, vol. 16, no. 5, p. 629, 2016.
- [11] J. Wiley, *Tracking and kalman filtering made easy*. vol. 1. 1998.

- [12] B. Khaleghi, A. Khamis, F. O. Karray, And S. N. Razavi, Multisensor data fusion: A review of the state-of-the-art. *Inf. Fusion*, vol. 14, no. 1, pp. 28–44, 2013.
- [13] M. Kushwaha, S. Oh, I. Amundson, X. Koutsoukos, And A. Ledeczi, Target tracking in heterogeneous sensor networks using audio and video sensor fusion. *2008 IEEE Int. Conf. Multisens. Fusion Integr. Intell. Syst.*, pp. 14–19, 2008.
- [14] M. Pettersson, Extraction of moving ground targets by a bistatic ultra-wideband SAR. *Doktorsavhandlingar vid Chalmers Tek. Hogs.*, no. 1650, pp. 1–22, 2000.
- [15] M. Beard, B. T. Vo, And B. N. Vo, Bayesian multi-target tracking with merged measurements using labelled random finite sets. *IEEE Trans. Signal Process.*, vol. 63, no. 6, pp. 1433–1447, 2015.
- [16] H. Durrant-Whyte And T. C. Henderson, Multisensor data fusion. *Robotics*, pp. 585–610, 2008.
- [17] C. Soto, B. Song, And A. K. Roy-Chowdhur, Distributed multi-target tracking in a self-configuring camera network. *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009.
- [18] Constantine A. Balanis, *Antenna theory analysis and design*. Third edition.
- [19] K. Zhang, X. R. Li, And H. Chen, Multi-Sensor multi-target tracking with out-of-sequence measurements . pp. 672–679, 2003.
- [20] S. A. Hoseini And M. R. Ashraf, Computational complexity comparison of multi-sensor single target data fusion methods by Matlab. *arXiv Prepr. arXiv 1307.3005*, vol. 2, no. 2, pp. 1–8, 2013.
- [21] R. Anitha, S. Renuka, And A Abudhahir, Multi sensor data fusion algorithms for target tracking using multiple measurements. *Comput. Intell. Comput. Res. (ICCIC)*, 2013 *IEEE Int. Conf.*, pp. 1–4, 2013.
- [22] W. C. Selby, *Autonomous navigation and tracking of dynamic surface targets on-board a computationally impoverished aerial vehicle*. 2011.
- [23] P. Agarwal, D. Federman, K. Ge, T. Jan, C. Jones, Y. Lou, S. Modi, K. Shi, V. Tan, And M. Zhang, Target tracking: Implementing the kalman filter. pp. 1–17.
- [24] Brookner. E, *Aspects of modern radar*. Book. 1988.
- [25] Brookner. E. , *Phased-Array Radars*. *Scientific American*, Vol. 252, No. 2, pp. 94–102, 1985.
- [26] Brookner . E., *Major Advances in Phased Arrays: Part I*. *Microwave Journa*, pp. 288–294, May 1997.

- [27] George M.Siouris, An engineering approach to optimal control and estimation theory. John Wiley & Sons, Inc. 1996
- [28] J. Voros, J. Mikles, And L. Cirka, Parameter estimation of nonlinear systems. Food Technol., no. 1, pp. 309 – 320, 2008.
- [29] A. Srivastava, Fuzzy logic in software effort estimation : an exploration. vol. 5, no. 5, pp. 471–475, 2016.
- [30] L. Goodson And F. Rohani, Higher order thinking skills • definition • teaching strategies • assessment, thinking, vol. 18, p. 458, 1998.
- [31] A. . Fallis, Decentralized tracking with feedback, Adaptive sample rate and IMM, J. Chem. Inf. Model., vol. 53, no. 9, pp. 1689–1699, 2013.
- [32] M. Kushwaha, S. Oh, I. Amundson, X. Koutsoukos, And A. Ledeczi, Target tracking in heterogeneous sensor networks using audio and video sensor fusion. 2008 IEEE Int. Conf. Multisens. Fusion Integr. Intell. Syst., pp. 14–19, 2008.
- [33] Richard Szeliski, Computer vision: algorithms and applications. Springer.2010.
- [34] E. Lefebvre, Computer network security. pp. 765–778, 2002.
- [35] I. Ensor, Wireless sensor networks. commun. ACM, vol. 47, no. 6, pp. 53–57, 2004.
- [36] L. Uftring And E. Miller, A Multiple target range and range-rate tracker using an extended kalman filter and a multilayered association scheme. Electr. Eng., vol. Master of, no. May, 2008.
- [37] K. Shinghal, Intelligent humidity sensor for - wireless sensor network. Int. J. Wirel. Mob. Networks, vol. 3, no. 1, pp. 118–128, 2011.
- [38] X. Long, B. Yin, And R. M. Aarts, Single-accelerometer-based daily physical activity classification., Conf. Proc. IEEE Eng. Med. Biol. Soc., vol. 2009, pp. 6107–6110, 2009
- [39] T. K. Nayak, S. K. Dash, H. S. Sasmal, And S. Patanaik, Object detection by an autonomous robot through data fusion. vol. 3, no. 5, pp. 4951–4955, 2012.
- [40] Doyle, J. C., Guaranteed margins for LQG regulators. IEEE Trans. Automat. Control, Vol. AC-23, No. 4, pp. 756- 757, August 1978.
- [41] Thornton, C. L. And Bierman, G. J., Udu, Covariance factorization for kalman filtering, in control and dynamic systems. Academic Press, Inc., New York, pp. 177-248, 1980.
- [42] Bar- Shalom , Y. (Ed.) , Multitarget-Multisensor tracking: advanced applications. Artech House, Norwood, Massachusetts, 1990.

- [43] K. Camara, A visual query language served by a multi-sensor environment. no. 1333. 2007.
- [44] C. Albaladejo, F. Soto, R. Torres, P. Snchez, And J. A. Lpez, A low-cost sensor buoy system for monitoring shallow marine environments. *Sensors* (Switzerland), vol. 12, no. 7, pp. 9613–9634, 2012.
- [45] L. Chen, M. J. Wainwright, M. Cetin, And A. S. Willsky, Data association based on optimization in graphical models with application to sensor networks. *Math. Comput. Model.*, vol. 43, no. 9–10, pp. 1114–1135, 2006.
- [46] E. M. Saad, E. L. Bardawiny, H. I. Ali, And N. M. Shawky, Improving data association based on finding optimum innovation applied to nearest neighbor for multi-target tracking in dense clutter environment. *J. Circuits, Syst. Signal Process.*, vol. 5, no. 4, pp. 322–332, 2011.
- [47] H. Bostrom, S. F. Andler, M. Brohede, R. Johansson, A. Karlsson, J. Van Laere, L. Niklasson, M. Nilsson, A. Persson, And T. Ziemke, On the Definition of Information Fusion as a Field of Research. *IKI Tech. Reports*, pp. 1–8, 2007.
- [48] K. Laws, R. Mcleod, And K. Laws, Go Back Case study and grounded theory : Sharing some alternative qualitative research methodologies with systems professionals . *Faculty of Education The University of Sydney* . pp. 1–25, 2006.
- [49] B. Khaleghi, A. Khamis, F. O. Karray, And S. N. Razavi, Multisensor data fusion: A review of the state-of-the-art. *Inf. Fusion*, vol. 14, no. 1, pp. 28–44, 2013.
- [50] B. Bramwell, Centralized and Decentralized kalman filter techniques for tracking navigation and control . vol. 32, no. 4, pp. 859–886, 2005.
- [51] K. P. Murphy, Proposed design for gR , a graphical models toolkit for R. *Interface*, no. September, 2003.
- [52] S. Pinchinat And S. Riedweg, On the architectures in decentralized supervisory control. *Proc. 44th IEEE Conf. Decis. Control. Eur. Control Conf. CDC-ECC '05*, vol. 2005, pp. 12–17, 2005.
- [53] A. Narayanan, S. Barocas, V. Toubiana, H. Nissenbaum, And D. Boneh, A Critical look at decentralized personal data architectures, *DUMW Data Usage Manag. Web*, 2012.
- [54] Steffen, L. Lauritzen., Time series analysis in 1880. A discussion of contributions made by T.N. Thiele. *International Statistical Review* 49, 1981, 319–333.
- [55] Steffen, L. Lauritzen, Thiele, pioneer in statistics. *Oxford University Press*. ISBN 0-19-850972-3.2002.
- [56] Kalman, R. E. , A New approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 1960.

- [57] Einicke, G.A., smoothing, filtering and prediction: estimating the past, present and future. Rijeka, Croatia: Intech. ISBN 978-953-307-752-9.2012.
- [58] Wolpert, Daniel; Ghahramani, Zoubin, Computational principles of movement neuroscience". Nature Neuroscience.2000.
- [59] Anderson, Brian D. O.; Moore, John B., Optimal filtering. Englewood Cliffs, NJ: Prentice Hall, Inc. pp. 176–190. 1979.
- [60] Stratonovich, R. L, Conditional markov processes. theory of probability and its applications. 5, pp. 156–178. 1960.
- [61] P. Kalane, Target tracking using kalman filter . vol. 2, no. 2, pp. 16–24, 2012.
- [62] K. Kenda, Usage of the kalman filter for data cleaning of sensor data. pp. 3–6, 2013.
- [63] G. Welch And G. Bishop, An introduction to the kalman filter. In Pract., vol. 7, no. 1, pp. 1–16, 2006.
- [64] Maybeck , Peter S, Stochastic models, estimation, and control. Volume1, Academic Press, Inc. 1979.
- [65] Sorenson, H. W., Least- Squares estimation : from gauss to kalman. IEEE Spectrum, vol. 7, pp. 63-68, July 1970.
- [66] Gelb, A. Applied Optimal Estimation, MIT Press, Cambridge, MA.1974.
- [67] Lewis, Richard. Optimal estimation with an introduction to stochastic control theory. John Wiley & Sons, Inc. 1986.
- [68] Brown, R. G. And P. Y. C. Hwang, Introduction to random signals and applied kalman filtering. Second Edition, John Wiley & Sons, Inc. 1992.
- [69] Jacobs, O. L. R., Introduction to control theory. 2nd Edition. Oxford University Press. 1993.
- [70] Y. Bar-Shalom, Joint probabilistic data association in distributed sensor networks. IEEE Trans. Automat. Contr., vol. 31, no. 10, pp. 889–897, 1986.
- [71] B. D. O. Anderson And J. B. Moore, Optimal filtering. Englewood Cliffs, NJ: Prentice-Hall, 1979.
- [72] R. Mucci, J. Arnold, And Y. Bar-Shalom, Track segment association with a distributed field of sensors. J. Acoust. SOC. Amer., vol. 78, pp. 1317-1324, Oct. 1985.
- [73] Neal A. Carlson, Federated square root filter for decentralized parallel processes. IEEE Transactions on Aerospace and Electronic Systems, 26.3 (1990)

- [74] Marc Gregoire and Luc Beaudoin, Visualization for network situational awareness in computer network defence. 2005.
- [75] Blackman, S . S ., Multiple target tracking: with radar applications. Artech House, Norwood, MA. 1986.
- [76] Juan Liu, Maurice Chu and James Ell Reich., Multitarget Tracking in Distributed Sensor Networks. IEEE Signal Processing Magazine, 24.3 (2007).
- [77] R. Karlsson And F. Gustafsson, Monte Carlo data association for multiple target tracking. IEE Work. Target Track., no. 3, pp. 247–254, 2000.
- [78] D. Avitzour., Stochastic simulation Bayesian approach to multi target tracking. IEE Proc. on Radar, Sonar and Navigation, 142(2), 1995.
- [79] N.Shrivastava, R.Mudumbai, U.Madhow And S.Suri, Target tracking with binary proximity sensors: fundamental limits, minimal descriptions, and algorithms. Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, 2006
- [80] N. Bergman and A. Doucet , Markov Chain Monte Carlo data association for target tracking. IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100), 2 (2000).
- [81] Chen, S., Josef Dick, and Art B. Owen., Consistency of markov chain quasi-monte carlo on continuous state spaces. The Annals of Statistics 39.2 (2011)
- [82] Sheldon Xu And Anthony Chang, Robust Object Tracking Using Kalman Filters With Dynamic Covariance. Cornell University, 2011.
- [83] Michael Beard, Ba Tuong Vo And Ba Ngu Vo, Bayesian multi-target tracking with merged measurements using labelled random finite sets. IEEE Transactions on Signal Processing, 63.6 .2015.
- [84] K. Meier and A Desai, using the kalman filter to estimate the state of a maneuvering aircraft. 2008.
- [85] Chang, Kuo-Chu, Distributed estimation in distributed sensor networks. The University of Connecticut, Storrs, Connecticut, 1986.
- [86] Chong, C. Y., Hierarchical estimation, Proceedings of the 2nd MIT/ONR Workshop, Monterey, California, 1979.
- [87] Chong, C. Y., Mori, S., And Chang, K. C., distributed multitarget-multisensor tracking, chapter 8 in distributed multisensor-multitarget tracking: advanced applications, Artech House, Norwood, Massachusetts, 1990.
- [88] Hashemipour, H. R., Roy, S., And Laub, A. J., Decentralized structures for parallel kalman filtering. IEEE Trans. Automat. Control, Vol. 33, No. 1, pp. 88-94, January 1988.

- [89] Hong, L., Distributed filtering using set models. *IEEE Trans. Aerospace and Electronic Systems*, Vol. 28, No. 4, pp. 1144-1153, October 1992.
- [90] Speyer, J. L., Computation and transmission requirements for a decentralized linear-quadratic-gaussian control problem. *IEEE Trans. Automat. Control*, Vol. AC-24, No. 2, pp. 266-269, April 1979.
- [91] Wangyan Li And Others, A Survey on multisensor fusion and consensus filtering for sensor networks, *Discrete Dynamics in Nature and Society*, 2015
- [92] Kuo-Chu Chang, Chee-Yee Chong And Yaakou Bar-Shalom, Joint Probabilistic Data Association in Distributed Sensor Networks. *IEEE Transactions on Automatic Control*, 31 . 1986.
- [93] Yi-Nang Chung; Erol Emre; Donald L. Gustafson, Global modeling approach for multisensor problems. *Signal and Data Processing of Small Targets* , 1991.
- [94] <http://www.eeherald.com/images/formc.jpg> , Access Date: 28.05.2016.

RESUME

Fawzia Abdien Ali Abdulla was born in Sudan 1989. She received the B.Sc. degree from University of Gezira, Gezira, Sudan, in Department of Electronics Engineering in 2011. She got her Turkish language education from Sakarya University's Turkish Language Center between 2013 and 2014. She started studying M.Sc. in Sakarya University, Sakarya, Turkey, Department of Electrical and Electronics Engineering from 2014. She has currently been working as Teaching Assistant for Department of Electronics Engineering, University of Gezira, Sudan since 2012. Interesting in image and signal processing, Tracking, Programming with MATLAB, Antenna, Biomedical instrumentation, Radar systems.