TARGET TRACKING WITH PHASED ARRAY RADAR BY USING ADAPTIVE UPDATE RATE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÖZLEM İPEK

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2010

Approval of the thesis:

TARGET TRACKING WITH PHASED ARRAY RADAR BY USING ADAPTIVE UPDATE RATE

submitted by **ÖZLEM İPEK** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen __________

Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen __________

Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Mustafa Kuzuoğlu __________

Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Mübeccel Demirekler ____________________

Electrical and Electronics Engineering Dept., METU

Prof. Dr. Mustafa Kuzuoğlu ____________________

Electrical and Electronics Engineering Dept., METU

Prof. Dr. Kemal Leblebicioğlu ____________________

Electrical and Electronics Engineering Dept., METU

Assist. Prof. Dr. Çağatay Candan ____________________

Electrical and Electronics Engineering Dept., METU

Assist. Prof. Dr. Asım Egemen Yılmaz ____________________

Electronics Engineering Dept., Ankara University

Date: 04.02.2010

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Özlem İpek

Signature :

ABSTRACT

TARGET TRACKING WITH PHASED ARRAY RADAR BY USING ADAPTIVE UPDATE RATE

İpek, Özlem

M. Sc., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Mustafa Kuzuoğlu

February 2010, 166 pages

In radar target tracking problems, it may be required to use adaptive update rate in order to maintain the tracking accuracy while allowing the radar to use its resources economically at the same time. This is generally the case if the target trajectory has maneuvering segments and in such a case the use of adaptive update time interval algorithms for estimation of the target state may enhance the tracking accuracy. Conventionally, fixed track update time interval is used in radar target tracking due to the traditional nature of mechanically steerable radars. In this thesis, as an application to phased array radar, the adaptive update rate algorithm approach developed in literature for Alpha-Beta filter is extended to Kalman filter. A survey over relevant adaptive update rate algorithms used previously in literature on radar target tracking is presented including aspects related to the flexibility of these algorithms for the tracking filter. The investigation of the adaptive update rate algorithms is carried out for the Kalman filter for the single target tracking problem where the target has a 90° maneuvering segment in its trajectory. In this trajectory, the starting and final time instants of the single maneuver are specified clearly, which is important in the assessment of the algorithm performances. The effects of incorporating the variable update time interval into target tracking problem are presented and compared for several different test cases.

Keywords: Adaptive update rate, adaptive update time interval, target tracking, phased array radar, Kalman filter, Alpha-Beta filter

ÖZ

DEĞİŞKEN GÜNCELLEME HIZI KULLANILARAK FAZ DİZİLİ RADAR İLE HEDEF TAKİBİ

İpek, Özlem

Yüksek Lisans, Elektrik Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Mustafa Kuzuoğlu

Şubat 2010, 166 sayfa

Radar hedef takip problemlerinde aynı zamanda hem radar kaynaklarını ekonomik kullanıp hem de takip doğruluğunu sağlayabilmek için değişken güncelleme hızı tekniğini kullanmak gerekebilir. Eğer hedef yörüngesinde manevra yapılan bölümler varsa bu husus önemlidir ve böyle bir durumda hedefin yerini tahmin etmek için değişken güncelleme hızı algoritmalarını kullanmak hedef takip doğruluğunu artırabilir. Alışılagelmiş şekilde, radar hedef takibinde sabit güncelleme hız aralığı, hareketi mekanik olarak yönetilen radarların doğasından ileri gelir. Bu çalışmada, bir faz dizili radar uygulaması olarak, literatürde Alfa-Beta filtresi için geliştirilmiş değişken güncelleme hızı uygulaması, Kalman filtresi uygulamalarına genişletilmiştir. Radar hedef takibi için konu ile ilgili literatürde yer alan değişken güncelleme hızı algoritmaları, bu algoritmaların takip filtreleri bakımından uygunluk yönlerini de içerecek şekilde sunulmuştur. Hedefin yörüngesi dahilinde 90°'lik manevra bölümüne sahip olduğu tek hedef takip probleminde Kalman filtresi için değişken güncelleme hızı algoritmaları kullanımı incelenmiştir.

Algoritma performans değerlendirmesinde önemli olan, manevra bölümünün başlangıç ve bitiş anları bu hedef yörüngesinde net olarak belirlenmiştir. Değişken güncelleme hızı özelliğini hedef takip algoritmalarının içine dahil etmenin etkileri sunulmuş ve çeşitli farklı test durumları için karşılaştırmalar yapılmıştır.

Anahtar Kelimeler: Değişken güncelleme hızı, değişken güncelleme zaman aralığı, hedef takibi, faz dizili radar, Kalman filtresi, Alfa-Beta filtresi

To My Family

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor Prof. Dr. Mustafa KUZUOĞLU for his encouragement, guidance, advice and support throughout this study.

I would like to thank to my colleagues and friends for their continuous encouragement and valuable advice during this thesis. I am also grateful to ASELSAN Inc. for the facilities provided for the completion of this thesis.

I would like to express my profound appreciation to my parents, Suna İPEK and Ahmet İPEK and my brother, Özgür İPEK for their continuous encouragement and support they have given me not only throughout my thesis but also throughout my life. I am also grateful to them for their continuous love and endless patience. Moreover, I want to thank them for everything.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

CV	:	Constant Velocity
IMM	:	Interacting Multiple Models
KF	:	Kalman Filter
m	:	Meter
MC	:	Monte Carlo
MMSE	:	Minimum Mean Square Error
MSE	:	Mean Square Error
M^3	:	Multiple Maneuver Model
rad	:	Radian
RMSE	:	Root Mean Square Error
sec	:	Second
2D	:	Two Dimensional

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND AND SCOPE OF THESIS

Time is one of the most important resource parameters that must be used efficiently for each task of a multi-function radar. It is desired to keep the tracking accuracy of a target as high as possible without allocating the time of radar unnecessarily for tracking of the target [5, 20].

An airborne target is engaged in straight-line constant-velocity motion rather than maneuvering motion for most of the time. An airborne target that flies generally with a constant velocity sometimes makes maneuver. In order to keep the track of the target at maneuvering segments, it is required to choose update time intervals (time interval between two consecutive track updates) relatively small [5, 20]. On the other hand, while the target moves with constant velocity, it is unnecessary to still use relatively small update time intervals for a multi-function radar.

Short update time intervals means allocating the radar resources for tracking task due to the interruption of the other radar tasks like surveillance and a lengthening of the time required to complete the other radar tasks [2].

Thus, there is a trade off between keeping the tracking error small and not to lose the track and using longer update time intervals in order to allocate radar resource economically for the tracking task [5].

The next update time should be calculated, according to the following criteria:

- "To maintain track in maneuvering and non-maneuvering segments of the trajectory, the update time interval should be small enough so that, at the next illumination, the target will be within the predicted region, scanned by the radar beam, with a high enough probability."

- "The update time should be not too small, to minimize the use of the radar resources. This will allow the radar to do more within a given time. Hence, a large update time should be used for tracking a non-maneuvering target and a faster update is needed to track a maneuvering target or fast targets, which accelerate harder or change range and which may possibly escape from the beam of the antenna. In this case, the update time choice must guarantee an operator specified track accuracy" [21].

In this thesis, as an application to phased array radars, adaptive update time interval algorithms are studied. Adaptive update rate algorithms will be implemented by using both Kalman filter and Alpha-Beta filter (a particular case of Kalman filter).

In this study, the application of adaptive update rate on the single target tracking problem is considered. The aim is to keep the tracking accuracy of the target as high as possible while allocating the time of radar economically for tracking task.

For such a target tracking problem, classical Alpha-Beta filter and classical Kalman filter equations which are generally based on fixed update time interval can not be used without any modifications due to the following reasons given below:

- Time interval between consecutive track updates is not fixed due to adaptive control.

- To find the update time interval for the next track update (i.e. the time interval from the current track data point up to the next track data point) an "Update Time Interval Determination Algorithm" must be developed.

- In the development of “Update Time Interval Determination Algorithm”, the nature of the filter must be considered and how the target’s motion characteristics (for example target maneuver) reflect to the filter operation must be taken into account.

- “Update Time Interval Determination Algorithm” must give feedback to filter so that the filter can accomplish the prediction accordingly.

The degree of adaptability of the target motion model in the implemented filter plays an important role for the filter to detect whether or not the target makes a maneuver.

Historically, the problem of adaptive update rate tracking has attracted the attention of researchers. Generally, the algorithms developed so far can be classified in two main groups based on the control parameter of the algorithm to detect the target maneuver. These are:

- Measurement residual

- Predicted error covariance

In the algorithms which are based on the measurement residual, by using the fact that the increase on the measurement residual (difference between the actual measured position and the predicted measurement) means tracking accuracy is decreasing, the algorithms to vary the track update time interval by an amount that tends to maintain a constant residual error are developed. Many researchers used this approach in the development of adaptive update rate algorithms [2, 5, 4].

In this group of algorithms, in which the adaptive control of update time interval is based on measurement residual, the update time interval is selected by using the following considerations:

- The update time interval is chosen with respect to the magnitude of the difference between the measured and the predicted position.

- The update interval is inversely proportional to the measurement residual.

On the other hand, in the algorithms which are based on predicted error covariance, by using the fact that the increase on the predicted error covariance means tracking accuracy is decreasing, the algorithms to vary the track update time interval by an amount that tends to maintain the predicted error covariance below a selected threshold value are developed [16, 17, 18, 19].

In this group of algorithms, in which adaptive control of update time interval is based on predicted error covariance, the update time interval is selected by using the following considerations:

- The update time is chosen according to the magnitude of the predicted error covariance by comparing with the selected threshold value (the threshold value is usually selected relative to the measurement error covariance)

- The next update time interval is selected when the predicted error covariance exceeds a given threshold.

As it was mentioned above previously, the degree of adaptability of the target motion to the model in the implemented filter may differ depending on the nature of the implemented filter.

In the Kalman filter and Alpha-Beta filters, predicted error covariance based adaptive update time interval algorithm is not realized in literature. The reason for this is the degree of adaptability of the target motion to the model is not linked to the Kalman filter's predicted error covariance matrix. Therefore, it is not possible to decide whether or not the target deviates from the motion model by observing the covariance matrix in Kalman filter [18]. Due to the nature of the Kalman filter, covariance matrix converges and mainly considered to be constant during filter operation. Moreover, since the covariance matrix update is carried out offline and it does not depend on the measurements or target motion, it seems logical that the degree of adaptability of the target motion to the model is not reflected on the covariance matrix of Kalman filter.

On the other hand, in the filters which are based on the multiple motion model, tracking process attempts to ensure accurate tracking performance simultaneously for various kinds of motion by setting several motion models in parallel. The tracking schemes based on the multiple motion model include M^3 (Multiple Maneuver Model) and IMM (Interacting Multiple Models) [18].

Predicted error covariance matrix reflects the degree of adaptability of each model when the predicted error covariance matrices of individual models are combined into a single predicted error covariance matrix. Thus, when the target starts to maneuver, this causes an increase in predicted error covariance matrix [18].

In this study, since the application of adaptive update rate on the single target tracking problem by using Kalman filter is focused on, measurement residual based approach will be considered for controlling of the next track update time interval. Several algorithms based on the measurement residual approach will be applied on Kalman and Alpha-Beta filters for adaptive update rate case by utilizing the previously used algorithms proposed historically.

The studies in the literature on adaptive update rate target tracking are summarized below:

Over the last four to five decades, phased-array radars have increased the flexibility of using adaptive update rate algorithms rather than fixed update rate algorithms for target tracking. The advance in phased array radars makes the adaptive update rate target tracking algorithms more useful by also increasing the demand for them [6].

Initially, the concept of adaptive update time interval has been reported by van Keuk in 1975 [2, 19]. He has proposed maintaining constant track sharpness by means of parallel processing of Kalman filters. He has mentioned that the next update time interval should be selected so that the predicted error covariance in position is kept under a given threshold. Based on this criterion an empirical formula for calculating the next update time interval has been derived where, measurement error covariance, maneuver correlation time and covariance of target acceleration parameters are used.

In 1977, Navarro has proposed an algorithm which controls the sampling interval in such a manner that the dynamic errors are kept constant. However, Navarro's concern about losing a highly maneuverable target resulted in an implementation consisting of a two-step procedure with slow convergence to an equilibrium condition [2].

In 1980, Castella [1] has presented an adaptive two-dimensional Kalman tracking filter which adaptively controls only weight of filtering and does not control update time interval. When the target flies with constant-velocity, along a straight-line trajectory, it filters heavily so that the filter may make much account of the past trajectory and inertial conditions; on the other hand, when the target is maneuvering, it filters lightly so that the filter may depend more on the recently acquired data than on the past data. However, there remains a problem. The light filtering causes a loss of information, and consequently the same accuracy obtained during heavy filtering cannot be guaranteed. The reason is that the adaptive algorithm of Castella [1] is based on the assumption of fixed sampling frequency.

After the investigation of the problem in [1], the attention to the adaptive tracking filters based on the adaptive update time interval has increased and many researchers have dealt with the problem of adaptive update rate.

After the algorithmic implementation of Navarro, which consists of a two-step procedure with slow convergence to an equilibrium condition not to lose a highly maneuverable target [2], in 1986 Cohen [2] proposed an algorithm for adaptive update rate similar to that of Navarro; however, the implementation in [2] is based on a one step procedure and still effective.

Cohen [2] has introduced a modification to the classical Alpha-Beta filter (with fixed update time intervals) which automatically varies the update time interval according to the maneuver of the target. The next update time interval varies inversely proportional to the inverse square root of the position residual.

Also, in 1986, Nomoto et al. [3] have proposed a Kalman filter algorithm, with reference filter and maneuver detector. The proposed filter adaptively controls its sampling frequency as well as weight of filtering according to the target's motion. When the target flies with constant-velocity, along a straight-line trajectory, the proposed filter operates heavy filtering at low sampling frequency in order to curtail the sampling. On the other hand, when the target maneuvers, it filters lightly at high sampling frequency in order to make up for the lack of information and to maintain the accurate track.

In 1988, Gardner and Mullen [4] have extended Cohen's algorithm [2] to implement the measurement residual based adaptive rate approach on Alpha-Beta-Gamma filter. They have extended the basic Alpha-Beta-Gamma filter to incorporate a variable update time interval; the application here is in tracking a maneuvering target using passive sonar. In their modification to the algorithm of [2], the update time interval varies as the inverse cube root of the position residual (measurement residual).

In 1991, the work of Cohen [2] has been extended in several ways by Wilkin et al. [5]. Firstly, the measurement noise has been assumed to be additive in the range and bearing directions rather than in the x and y directions as presumed in [2]. Secondly, Monte Carlo simulations were used. Cohen [2] has suggested that the maximum of the x and y residuals be used in the choice of update time interval. The benefits of using an alternative procedure have been investigated in [5], where the residual in the position is given by mean square estimated errors of the two coordinates.

In 1992, Munu et al. [6] have developed the work carried out in [5]. The adaptive update time interval algorithm used by Cohen [2], which varies as the inverse square root of the position residual, was extended to the basic Alpha-Beta-Gamma filter in [6]. The adaptive update time interval algorithm proposed by Gardner and Mullen [4], which uses an inverse cube root relationship, was also employed as a tool of comparison. The tracking performance of these algorithms was assessed by simulation over several different trajectories. A comparison of the performance figures, the mean square errors, mean update time interval and the mean number of updates, is presented.

Moreover, starting from 1993 many researchers of [16, 17, 18, 19] implemented algorithms for variable update time interval on IMM or M^3 filters. These adaptive update rate algorithms are mainly based on the change in predicted error covariance matrix. Since Multiple Model filters reflect the degree of adaptability of each model when the predicted error covariance matrices of individual models are combined into a single predicted error covariance matrix, the deviation of the target motion from the model can be observed.

Also, in 2007 a variable update algorithm based on IMM estimator is proposed in [19]. The update interval is proportional to the inverse square root of the position residual as in [2]. A controllable parameter is included to balance the tracking precision and the system load which is similar to the algorithm based on the predicted error covariance. In [19], it has been shown that the adaptive update rate approach based on measurement residual is also suitable for IMM filter and besides it has been claimed that the computational complexity is lower compared with predicted covariance based approaches.

As seen from the literature survey given above, for adaptive update rate, there are several algorithms implemented on the Alpha-Beta filter based on measurement residual approach. Moreover, many algorithms implemented on multiple mode filters based on the trace of predicted covariance approach. However, as far as we know, there is not a direct application of adaptive update rate algorithms on a single Kalman filter.

1.2 THESIS OUTLINE

In this thesis, the application of adaptive track update time interval (adaptive control of update rate) algorithms on the single target tracking problem as an application for phased array radar is studied. The adaptive update rate algorithm approach, which is based on keeping the track update time interval by an amount that tends to maintain a constant measurement residual, developed in literature for Alpha-Beta filter will be applied on the Kalman filter. The behavior of the adaptive update rate algorithms implemented on Kalman filter will be tested for tracking of a target which has a 90° maneuvering segment in its trajectory. In this trajectory, the starting and final time instants of the single maneuver are specified clearly, which is important in the assessment of the algorithm performances. Moreover, the effectiveness of adaptive update rate algorithms for a target trajectory which includes two maneuvering segments consisting of 90° maneuvers will be investigated. The effects of incorporating the variable update time interval into target tracking problem are presented and compared for several different test cases. The aim is to carry out the track maintenance along the target trajectory under economical resource allocation on radar for tracking task by using adaptive update rate algorithms.

In Chapter 2, the theoretical bases of single target tracking are explained by specifying the measurement and target motion models. The constant velocity model, which will be used for modeling the target motion throughout this study, is presented. Furthermore, target tracking problem with adaptive update time interval is presented by giving the general properties of phased array radar which provides the flexibility for using variable update time interval in target tracking.

In Chapter 3, a brief summary of Alpha-Beta filtering and the necessary theoretical bases are provided. Application of adaptive update time interval algorithm to Alpha-Beta filter is explained. Moreover, the methods that exist in literature for adaptive update time interval algorithms are presented.

In Chapter 4, a brief summary of Kalman filtering and necessary theoretical bases are provided. Application of adaptive update time interval algorithm to the Kalman filter equations is proposed and explained in a detailed way. Moreover, the methods that exist in literature for adaptive update time interval algorithms are investigated in terms of compatibility for Kalman filtering nature. The methods, which will be used in this thesis throughout adaptive Kalman filter simulations, are introduced.

Chapter 5 includes the implementations of the algorithms for adaptive update time interval algorithms on Alpha-Beta filter and Kalman filter, based on the algorithms described in Chapter 3 and Chapter 4. The assumptions that are used during this study are given and the comparison parameters for simulations are introduced. Besides, application of Monte Carlo Simulations for adaptive time interval algorithms is explained. The simulation results are compared and evaluated for different simulation cases.

Chapter 6 provides the summary and the concluding remarks of this study. Moreover, possible future studies are given in this part.

CHAPTER 2

RADAR TARGET TRACKING AND ADAPTIVE UPDATE TIME INTERVAL CASE

In this chapter, theoretical bases of single target tracking are explained by specifying the measurement and target motion models. The constant velocity model, which will be used for modeling the target motion throughout this study, is presented. Furthermore, target tracking problem with adaptive update time interval is introduced by giving the general properties of phased array radar.

2.1 GENERAL INFORMATION ON SINGLE TARGET TRACKING

Radar basically transmits a signal and detects the echo signal from the reflecting target in its domain. The main task of the radar is to find the location of the target [15, 14].

However, the task of the radar is not very simple due to various types of noise in the environment and lack of information about the target like the target's motion, the target's initial position, the velocity of the target etc. Due to the aforementioned reasons, one can easily defend the fact that radar does not only send the signal and then measures its return to find the target's location. In order to find the location of the target by making accurate estimate of the position of the target, radar executes the appropriate algorithms to analyze the returned signal [15, 14].

In order to find the target's location, radar should be successful in the estimation process. Estimation is the process of inferring the value of a quantity of interest from available observations. In radar target tracking, the aim of the estimation process is to obtain the states (position, velocity and acceleration) of the target accurately. By making use of filtering algorithms, the state of the target is estimated at each track data point.

Radar is one of the most frequently used sensors for target tracking applications. The aim of target tracking is to track the motions of the target relative to the observer by making accurate estimates of the target's position. Target tracking is basically the estimation of the states (position, velocity and acceleration) of the moving object (target) by making use of observations.

Even though the radar can only measure the position of the target, the filter implemented in the radar must estimate both position, velocity and acceleration of the target using the available measurements (noise-corrupted observations related to the position of the target) for achieving the tracking task of radar [14].

There are many possible target tracking cases depending on whether the observer is stationary or moving and the target is stationary or moving.

In this study, the radar is assumed to be stationary and the target to be tracked is assumed to be an airborne target like an aircraft.

Radar can measure the position of the target by noisy measurements. The measurement noise represents the inability of the radar to measure the target's position precisely. The measurement noise can originate from several technical limitations in the equipments of the radar.

In real-time applications of the target tracking, the measured data of the radar are taken in polar coordinates.

Mainly, radar measures the position of the target by the azimuth angle, elevation angle, and the range. For two dimensional target tracking applications, since the height is ignored, only azimuth and range parameters are used. During this study two dimensional target tracking model is used.

Target tracking with radar also deals with the non-linearity problem since the radar measurements are not in Cartesian coordinates, but they are in polar coordinates.

The below mentioned coordinate systems are used to investigate the effect of different ways of representing the nonlinearity in the equations:

- Spherical (or Polar) Coordinate System
- Cartesian Coordinate System

In this study, Cartesian Coordinate System will be used in tracking. So, Spherical Coordinate System will not be considered, but the use of Cartesian Coordinate System in tracking will be highlighted.

2.2 CARTESIAN COORDINATE SYSTEM IN TRACKING

The motion of the target is easily modeled in Cartesian coordinates. On the other hand, radar measurements are taken in polar coordinates.

The derivation of classical Kalman filter equations is based on the measurements to be in Cartesian coordinates.

Due to the fact that the measurements and the target motion are generally in different coordinates, a nonlinear transformation is required to relate the measurements to the Cartesian states of target motion.

There are two approaches for using the measurements in polar coordinate to track the target motion in Cartesian coordinates:

a) Transform the measurements from polar coordinate to Cartesian coordinates before feeding them to the tracking filter

b) Feed polar measurements to the tracking filter through the use of a nonlinear measurement in the form of an extended Kalman filter

In this study, the target motion is modeled in Cartesian coordinates and in order to estimate the states of the target, coordinate transformation of the measurements from polar to Cartesian is necessary.

Moreover, since measurement matrix measures the covariance of measurement error, which is given in polar coordinates, a new measurement matrix is required which is formed by use of noise in *x* and *y* positions. Thus, measurement noise in *x* and *y* positions must be calculated by using the measurement noise in polar coordinates.

2.3 TARGET MOTION MODELS IN CARTESIAN COORDINATE SYSTEM

When the Cartesian coordinates are used for modeling the motion of the target, the basic states of the target are the position and the velocity.

For 2D system:

$$x_k = \begin{bmatrix} d_{x,k} \\ d_{y,k} \\ V_{x,k} \\ V_{y,k} \end{bmatrix} \tag{2-1}$$

where,

t_k : the update time at the k^{th} track update data point

x_k : the state vector that represents the positions and velocities of the target in x and y Cartesian coordinates at data point t_k

$d_{x,k}$: the position of the target in x Cartesian coordinate at data point t_k

$d_{y,k}$: the position of the target in y Cartesian coordinate at data point t_k

$V_{x,k}$: x component of the velocity of the target at data point t_k

$V_{y,k}$: y component of the velocity of the target at data point t_k

2.3.1 Constant Velocity (CV) Model for Adaptive Update Rate Case

The target motion is modeled with the below state equation in CV Model:

$$x_{k+1} = A_k x_k + G_k w_k \tag{2-2}$$

where,

$$A_k = \begin{bmatrix} 1 & 0 & T(k) & 0 \\ 0 & 1 & 0 & T(k) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; G_k = \begin{bmatrix} T^2(k)/2 & 0 \\ 0 & T^2(k)/2 \\ T(k) & 0 \\ 0 & T(k) \end{bmatrix}, \; w_k = \begin{bmatrix} w_{x,k} \\ w_{y,k} \end{bmatrix} \tag{2-3}$$

The detailed explanations and meanings of the terms in Equation (2-3) are given below:

A_k, the state transition model matrix, updates the position and velocity components of the target's state vector at each time step of tracking. Update of the target's position is performed by adding the time interval between each radar measurement multiplied by the velocity along the same dimension [14]. Since the motion model is based on the constant velocity assumption, target's velocity is kept fixed at each time step of tracking.

w_k, the process noise, which represents the unknown acceleration of the target

G_k, the process noise control matrix, which defines the ambiguity in the target motion, by assuming that process noise is modeling the unknown acceleration of the target

Also, let Q_k denotes the covariance matrix of the process noise w_k. Then, the process noise covariance values in Q_k matrix allow tracking the target motion in the presence of ambiguity in the motion of the target which moves mainly with constant velocity [23].

t_k, the time at which k^{th} track update is done

$T(k)$, the update time interval between data points t_k and t_{k+1}, i.e.

$$T(k) = t_{k+1} - t_k \tag{2-4}$$

Note that, $T(k)$ parameter in A_k and G_k matrices allow to use adaptive update rate during tracking of the target.

2.4 GENERAL INFORMATION ON TARGET TRACKING WITH ADAPTIVE UPDATE TIME INTERVAL

In tracking problems, during maneuvering segment of the trajectory, it may be required to take measurements more frequently and consequently performing the track updates of the target's state estimates with a relatively higher rate in order to keep the track accuracy in a desired level. However, during the straight line segments of the target trajectory, in which the target moves with constant velocity, it is desirable to take measurements less frequently and consequently performing the track updates of the target's state estimates with a relatively lower rate in order to allow the radar to use of its resources for other radar tasks.

In such a case, where constant-velocity target has also maneuvering segments, the use of adaptive update time interval algorithm for estimation of the target state may overcome the problem.

By using the flexibility of phased array radar for using variable update time interval, adaptive update time interval algorithms can be implemented on the tracking filter for estimation of the target's state.

In this study, by means of the approach developed in the literature for adaptive update rate algorithms, the performance of the adaptive update rate algorithms will be investigated for tracking of a maneuvering target which can make 90° turn in its trajectory.

2.5 OVERVIEW OF PHASED ARRAY RADAR

The majority of radar devices move their antennas mechanically. This means that changing the direction of the antenna beam is relatively slow for such mechanically steerable radars.

Recent developments in radar technology include also the phased array radar. Phased array radar, which can direct the transmitted waveform electronically, is an important development in radar technology. The ability of phased array radar to steer the beam electronically, and therefore quickly, is of great interest in radar systems. Whereas the time taken to physically position the radar beam is a major problem for the radar devices which move their antennas mechanically, phased array radars allow achieving a rapid reaction time which is vital in a radar system [15].

The ability of the phased array radar to steer the beam electronically solves the problem of necessity of relatively large time taken to position the radar beam physically. The phased array radar antenna can instantaneously direct the antenna beam into a desired direction quickly without the need to physically rotate the antenna [18, 15]. In addition to fast beam steering property, the phased array radar allows to obtain longer dwell time which is the duration of the target's remaining in the beam of the radar during each scan.

In order to understand the principle of the operation of the phased array radar, the concept of a phased array antenna, which provides to the phased array radar the ability of steering the beam electronically, should be introduced. "A phased array antenna is a directive antenna consisting of an array face, which has many elements (or cells), where each element is operated separately. Each radiating element generates a radiation pattern, whose shape and direction is governed by the relative phases and amplitudes of the currents at each element. It is possible to steer the direction of the radiation by varying the relative phases. The radiating elements can be arranged in a number of ways. A two-dimensional planar array is of most use for radar applications since it is the most versatile of all antennas" [15].

The phased array radar, with its electronically steerable property by means of phased array antenna, makes it possible to combine the multifunction radar tasks of surveillance, tracking and weapon guidance by efficiently using radar resources.

One aspect of the added flexibility of electronically steerable beam property of the phased array radar is the ability to vary the rate of updating of a target's state according to the performance requirements. The main reason for this is that the tracking subsystem in the phased array radar is free within wide limits to decide when the target should be illuminated again for the next track update to continue the tracking.

Due to the flexibility of the tracking subsystem of the phased array radar to decide the next track update time instant, the phased array radar allows using variable update time intervals rather than simply updating the track of the target (updating the target estimated state) at constant rate.

By the flexibility of changing the beam quickly, the phased array radar has the ability to vary the rate of updating the target's estimated state according to performance requirements depending on the behavior of the target movement. When the phased array radar detects that the target begins a maneuver, it can use a relatively higher track update rate to track the target accurately and to avoid track loss by changing the beam quickly and take measurements more frequently. Conversely, when the target moves with constant velocity, phased array radar can use a relatively low update rate sufficient to keep track of the target. The flexibility of reducing the update time interval in maneuvering segments prevents the degradation of the tracking accuracy. In constant-velocity segments of the target, the update time interval is increased so that the number of observations is reduced for an efficient operation of radar resources. Thus, by the adaptive update rate of phased array radar, the track accuracy can be maintained with less number of updates and the radar resources can be used economically for the maintenance of the tracks. The economical use of the radar resources can increase not only the number of tracks that can be maintained but also increase the resources that can be allocated to the searching task for new targets [18, 17, 5, 2, 19].

CHAPTER 3

ALPHA-BETA FILTER AND APPLICATION OF ADAPTIVE UPDATE TIME INTERVAL ALGORITHMS

In this chapter, a brief summary of Alpha-Beta filtering and necessary theoretical bases are provided. Application of adaptive update time interval algorithm to Alpha-Beta filter is explained. Moreover, the methods that exist in literature for adaptive update time interval algorithms are presented.

3.1 OVERVIEW OF ALPHA-BETA FILTER

In the steady state of Kalman filter, the Kalman gain and the covariance can be assumed to converge to some constant values. This particular case of Kalman filter is called Alpha-Beta filter for constant velocity model and Alpha-Beta-Gamma filter for constant acceleration model.

In this study, since the target motion is modeled with constant velocity, Alpha-Beta filter will be implemented.

The steady-state form of the Kalman filter is often used in order to reduce the computations required to maintain each track. In the steady-state form of the Kalman filter, the gains are constant and often denoted by α, β and γ in target tracking [25]. These constant gains are used in the measurement update step of filtering.

When the measurement errors in each coordinate are independent, the Kalman filter may be de-coupled into two optimal tracking filters, known as Alpha-Beta filters. This filter simplifies the computational requirements, because the states relating to each of the two coordinates can be estimated independently [25, 2, 13].

The equations for an Alpha-Beta filter to estimate position and velocity in x coordinate are given below:

$$x_e(k) = x_p(k) + \alpha\left(x_m(k) - x_p(k)\right) \tag{3-1}$$

$$v_e(k) = v_p(k) + \frac{\beta}{T(k-1)}\left[x_m(k) - x_p(k)\right] \tag{3-2}$$

The equations for an Alpha-Beta filter to predict the position and velocity in x coordinate are given below:

$$x_p(k) = x_e(k-1) + T(k-1)v_e(k-1) \tag{3-3}$$

$$v_p(k) = v_e(k-1) \tag{3-4}$$

where,

$x_p(k)$: the predicted position in x direction at data point t_k

$v_p(k)$: x component of the predicted velocity of the target at data point t_k

$x_e(k)$: the estimated position in x direction at data point t_k

$v_e(k)$: x component of the estimated velocity of the target at data point t_k

$x_m(k)$: the measured position in x direction at data point t_k

α : filter coefficient for position

β : filter coefficient for velocity

$T(k-1)$: the update time interval between track update data points t_k and t_{k-1}, i.e.

$$T(k-1) = t_k - t_{k-1} \tag{3-5}$$

Note that in the above state equations of Alpha-Beta filter, Kalman gain is replaced with:

$$\begin{bmatrix} \alpha \\ \beta / T(k-1) \end{bmatrix} \tag{3-6}$$

In the steady state of the Kalman filter, the Kalman gain and the covariance matrix are assumed to converge to constant values. This particular case of Kalman filter is called Alpha-Beta filter for constant-velocity model. The symbols α and β represent the gain coefficients of the position and velocity components respectively. These gains in the filter's update equations are taken as constant values.

The Alpha-Beta filter is based on the assumption that the target has constant velocity plus zero-mean, white Gaussian measurement noise in the dimension to be filtered. Given this assumption, the filter gains α and β are chosen as the steady state Kalman gains that minimize the mean-square error in the position and velocity estimates [25].

The condition $\beta = \dfrac{\alpha^2}{2-\alpha}$ is the optimum gain relationship, in the sense that the mean-square errors for both range and velocity are minimized when the input is a ramp function (velocity step) and the random noise is independent from sample to sample. This relationship results in a slightly underdamped system [2].

3.2 ALGORITHM FOR ALPHA-BETA FILTER WITH ADAPTIVE UPDATE TIME INTERVAL

Algorithm for Alpha-Beta filter with Adaptive Update Time Interval is given below:

1- Create noisy measurements (polar noise is assumed) with appropriate samples throughout the true target trajectory.

2- Initialize time update intervals $T(k)$ at data points t_0 and t_1 where $T(k)$ denotes the update time interval between track update data points t_{k+1} and t_k with appropriate values.

3- Select the values of the alpha and beta filter coefficients as proposed in [2]:

$$\alpha = 0.5 \tag{3-7}$$

$$\beta = 0.167 \tag{3-8}$$

4- Initialize first two phased array radar measurements at data points t_0 and t_1 by using $T(0)$ which is the initial time update interval value.

i.e. Assign values to $x_m(0)$ and $x_m(1)$, where $x_m(k)$ denotes the position measurement in *x* coordinate at data point t_k.

5- Initialize the estimated position and velocity at data points t_0 and t_1:

To initialize the estimated position $x_e(0)$, the first measurement, $x_m(0)$ of radar at data point t_0 is used:

$$x_e(0) = x_m(0) \tag{3-9}$$

$$v_e(0) = 0 \tag{3-10}$$

To initialize the position components of $x_e(1)$, the measurement of radar at data point t_1 is used. To initialize the velocity components $v_e(1)$, the measurements of radar at data point t_0 and t_1 (the measurements of first two consecutive scans) are used with the information of initial update time interval $T(0)$:

$$x_e(1) = x_m(1) \tag{3-11}$$

$$v_e(1) = [x_m(1) - x_m(0)]/T(0) \tag{3-12}$$

6- Initialize state predictions of position and velocity:

Since the filter starts to make state predictions starting from the data point t_2; $x_p(0)$, $x_p(1)$ and $v_p(0)$, $v_p(1)$ are initialized to be zero:

$$x_p(0)=0,\ \ x_p(1)=0,\ \ v_p(0)=0,\ \ v_p(1)=0 \tag{3-13}$$

7- Find phased array radar measurement at data point t_2, by using the time update interval value of $T(1)$ with the information of phased array radar's previous observation data point t_1. (The measurement at data point t_2 will be then used in calculation of $x_e(2)$ and $v_e(2)$)

FOR k=2:N (for each track update time step)

(Iteration starts at the data point $t_k = t_2$. Note that the filter starts to make state prediction and state estimate starting from the data point t_2. Iteration continues until the end of the target trajectory.)

8- Predict the position and the velocity in *x* direction at data point t_k:

   i. Project the position component of the state ahead by using previously calculated $x_e(k-1)$, $v_e(k-1)$ and $T(k-1)$:

$$x_p(k) = x_e(k-1) + T(k-1)v_e(k-1) \tag{3-14}$$

   ii. Project the velocity component of the state ahead. (Since the velocity is presumed constant, its projected value at the current data point equals to the previous value.)

$$v_p(k) = v_e(k-1) \tag{3-15}$$

9- Apply correction (measurement update) step to find position and velocity estimate at the current data point t_k:

   i. Find measurement residual $e_{k,x}$ (difference between the actual measured positions and the predicted measurements) in *x* direction at the current data point t_k. (by subtracting the predictions of the measurement from actual measurement at data point t_k:

$$e_{k,x} = x_m(k) - x_p(k) \tag{3-16}$$

   ii. By using the predicted position and the measured position, estimate the position component of state vector:

      1. Update the estimated position at data point t_k by using the calculated $x_p(k)$, α and the measurement residual $e_{k,x}$:

$$x_e(k) = x_p(k) + \alpha\left(x_m(k) - x_p(k)\right) \tag{3-17}$$

where,

$$e_{k,x} = x_m(k) - x_p(k) \tag{3-18}$$

iii. By using the predicted velocity, predicted position and the measured position, estimate the velocity component of state vector:

1. Update the estimated velocity at data point t_k by using the calculated $v_p(k)$, β, $T(k-1)$ and the measurement residual $e_{k,x}$:

$$v_e(k) = v_p(k) + \frac{\beta}{T(k-1)}\left[x_m(k) - x_p(k)\right] \tag{3-19}$$

10- Find $T(k)$, which is the update time interval for the next track update (i.e. the time interval from the current track data point t_k up to the next track data point t_{k+1}). To find $T(k)$, an "Update Time Interval Determination Method" is used:

i. The methods to be used in this step are presented in Section 3.3 in a detailed manner.

11- Find the next measurement at the data point t_{k+1} in order to use this measurement in the calculation of state estimate at the next data point t_{k+1}.

END for

(Iteration finishes by the end of the target trajectory.)

3.3 METHODS FOR ADAPTIVE UPDATE TIME INTERVAL ALGORITHMS

In the literature there are two update time interval determination algorithms proposed in [2] and [5] which are implemented on the Alpha-Beta filter. In this part, these algorithms will be presented.

3.3.1 Update Time Interval Determination Method-1 [2]

General Properties of Method:

$$T(k) = \frac{T(k-1)}{\sqrt{e_{0,k}}} \tag{3-20}$$

where,

$$e_{0,k} = \frac{|e_k|}{\sigma_n} \tag{3-21}$$

e_k : the measurement residual at data point t_k

$e_{0,k}$: the normalized measurement residual at data point t_k

σ_n : the standard deviation of measurement noise

Cohen [2] proposed an alternative to the Equation (3-20).

In this alternative method, next time interval $T(k)$ is selected by comparing the calculated measurement residual $e_{0,k}$ with the standard deviation of measurement noise σ_n at each data point.

In this method, $T(k)$ is chosen from a discrete set of values depending on the maximum of the magnitudes of the measurement residuals in *x* and *y* directions.

Upper and lower limits are placed on update time interval $T(k)$ to avoid the value of $T(k)$ taking impractically large or small values. Upper limit of update time interval $T(k)$ is selected as 4.00 sec and lower limit of $T(k)$ is selected as 0.25 sec. The set of update time intervals that can be used are 0.25, 0.50, 1.00, 2.00 and 4.00 sec.

To find the update time interval for next track update i.e. $T(k+1)$, by using "Update Time Interval Determination Algorithm" given in [2], which decides the value of next update time interval by comparing measurement residual at current data point with standard deviation of measurement noise, is summarized below:

Let,

$T(k+1)$: the update time interval for the next track update (i.e. the time interval from the current track data point t_{k+1} up to the next track data point t_{k+2})

$\sigma_{n,x}$: the standard deviation of measurement noise in *x* direction

$\sigma_{n,y}$: the standard deviation of measurement noise in *y* direction

$e_{k+1,x}$: the measurement residual in *x* direction at the current data point t_{k+1}

$e_{k+1,y}$: the measurement residual in *y* direction at the current data point t_{k+1}

Start of "Update Time Interval Determination Algorithm"

If either $\left|e_{k+1,x}\right| > 256 \times \sigma_{n,x}$ or $\left|e_{k+1,y}\right| > 256 \times \sigma_{n,y}$,

then $T(k+1) = 0.25$ sec;

else if either $\left|e_{k+1,x}\right| > 64 \times \sigma_{n,x}$ or $\left|e_{k+1,y}\right| > 64 \times \sigma_{n,y}$,

then $T(k+1) = 0.50$ sec;

else if either $\left|e_{k+1,x}\right| > 16 \times \sigma_{n,x}$ or $\left|e_{k+1,y}\right| > 16 \times \sigma_{n,y}$,

then $T(k+1) = 1.00$ sec;

else if either $\left|e_{k+1,x}\right| > 4 \times \sigma_{n,x}$ or $\left|e_{k+1,y}\right| > 4 \times \sigma_{n,y}$,

then $T(k+1) = 2.00$ sec;

else if both $\left|e_{k+1,x}\right| < \sigma_{n,x}$ and $\left|e_{k+1,y}\right| < \sigma_{n,y}$, and $T(k) \le 2$ sec,

then $T(k+1) = 2 \times T(k)$ sec;

else

$T(k+1) = T(k)$

end of "Update Time Interval Determination Algorithm"

Basically, the algorithm of Update Time Interval Determination Method-1 is summarized in the following simplified flow chart:

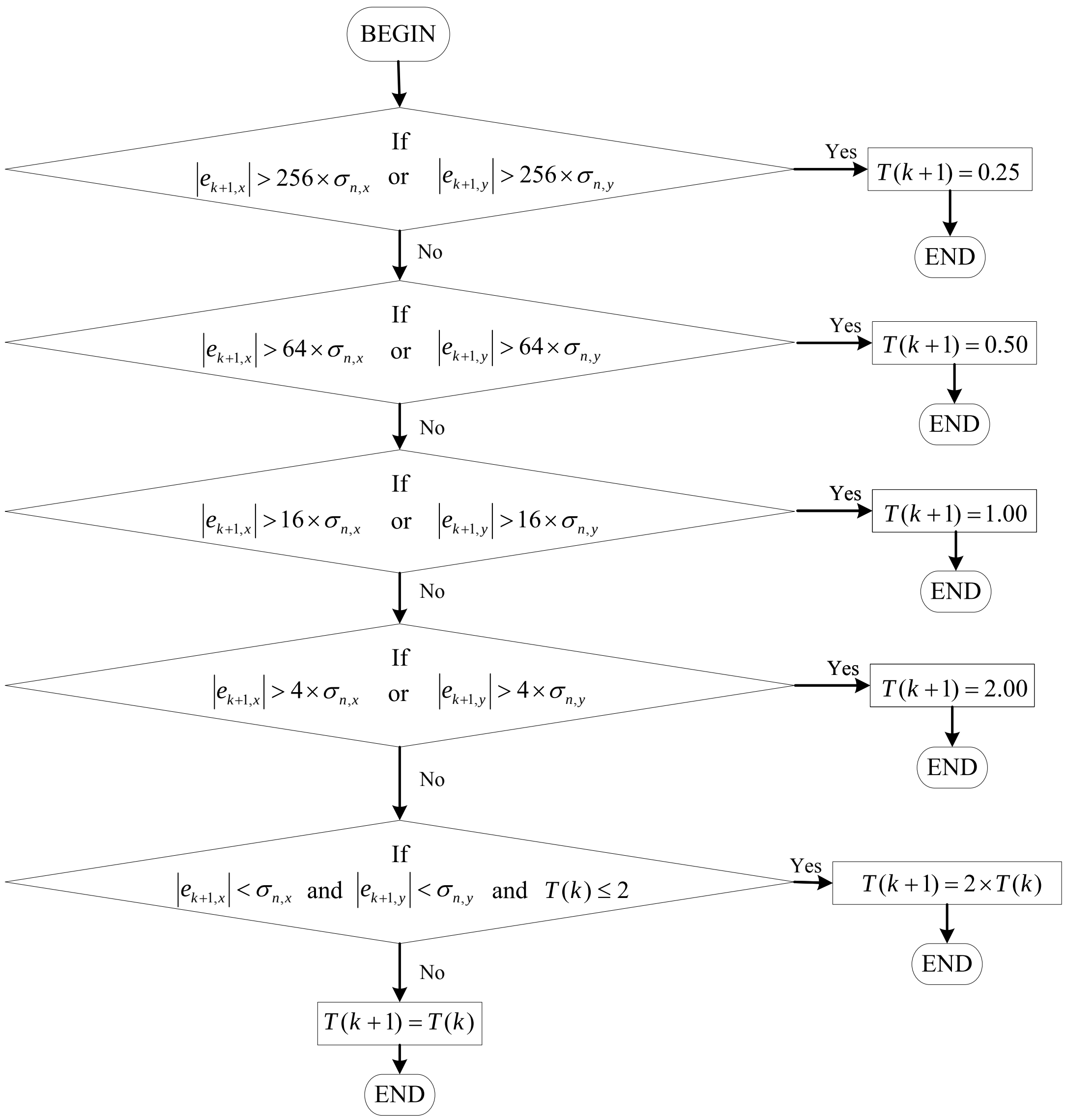

Figure 3-1: Algorithm of Update Time Interval Determination Method-1

3.3.2 Update Time Interval Determination Method-2 [5]

General Properties of Method:

The next update time interval has been chosen from a discrete set of values, the value chosen being dependent on the residual until another method for choosing next update time interval has been proposed in [5].

Cohen [2] has introduced a procedure where $T(k)$ is chosen from a discrete set of values depending on the maximum of the magnitudes of the measurement residuals in *x* and *y* directions, as a simplification to the Equation (3-20).

Cohen [2] has suggested to use the maximum of the magnitudes of the measurement residuals in *x* and *y* directions in the choice of update time interval. An alternative procedure has been investigated in [5] where the measurement residual to be used in the choice of next update time interval is computed with the below formula:

$$e_k = \sqrt{(x_m(k) - x_p(k))^2 + (y_m(k) - y_p(k))^2} \tag{3-22}$$

where,

$e_{k,x} = x_m(k) - x_p(k)$: the difference between the actual measured positions and the predicted measurements in *x* direction at data point t_k

$e_{k,y} = y_m(k) - y_p(k)$: the difference between the actual measured positions and the predicted measurements in *y* direction at data point t_k

In the alternative method of [5], the next time interval $T(k)$ is selected by computing $T(k)$ directly with Equation (3-20) and then selecting the direct continuous result of this formula or selecting 4.00 sec or selecting 0.25 sec after comparing the calculated value of $T(k)$ with the threshold U at each data point.

In this method, $T(k)$ may take the maximum or minimum values which are previously defined for $T(k)$ or it may take the continuous value between 0.25 and threshold U which is found to be the result of the formula given in Equation (3-20).

Direct usage of formula given in Equation (3-20) causes a problem, because this equation allows $T(k)$ to increase or decrease without limit. In order to avoid this problem, a minimum and a maximum limit for the update time interval are defined. Moreover, the algorithm given in [5] may be considered as an investigation whether or not the direct use of the equation leads to any improvement in the tracking performance.

The normalized measurement residual, $e_{0,k}$, will vary even in the constant velocity segments of the trajectory because of the effects of the measurement noise. Hence, $T(k)$ will occasionally be reduced below its maximum value in these segments, and this could lead to an unacceptably large number of updates.

To alleviate this problem, the following procedure is adopted in [5] for choosing $T(k)$. The update time interval is limited to a minimum value of 0.25 sec and a maximum value of 4.00 sec as in [2]. A threshold U is defined, and $T(k)$ is chosen according to the following algorithm.

"Update Time Interval Determination Algorithm" for determination of $T(k)$ by using the algorithm proposed in [5] is given below:

Let,

$T(k)$: the update time interval for the next track update (i.e. the time interval from the current track data point t_k up to the next track data point t_{k+1})

$\sigma_{n,x}$: the standard deviation of measurement noise in x direction

$\sigma_{n,y}$: the standard deviation of measurement noise in y direction

$e_{k,x}$: the measurement residual in *x* direction at the current data point t_k

$e_{k,y}$: the measurement residual in *y* direction at the current data point t_k

1- Select threshold value U

2- Compute the resultant measurement residual e_k, to be used in the choice of next update time interval by using the measurement residuals in *x* and *y* directions with the below formula:

$$e_k = \sqrt{(x_m(k) - x_p(k))^2 + (y_m(k) - y_p(k))^2} \quad (3\text{-}23)$$

where,

$$e_{k,x} = x_m(k) - x_p(k) \text{ and } e_{k,y} = y_m(k) - y_p(k) \quad (3\text{-}24)$$

3- Select the minimum of the standard deviation of measurement noise in *x* and *y* directions in order to be used then for computation of the normalized residual.

$$\text{i.e. Select } \sigma_n = \min(\sigma_{n,x}, \sigma_{n,y}) \quad (3\text{-}25)$$

4- Compute the normalized residual:

$$e_{0,k} = \frac{|e_k|}{\sigma_n} \quad (3\text{-}26)$$

5- Compute $T(k)$ by using the update time interval value at the previous data point and the normalized residual at the current data point:

$$T(k) = \frac{T(k-1)}{\sqrt{e_{0,k}}} \quad (3\text{-}27)$$

6- Choose $T(k)$ according to following approach:

If $U < T(k) < 4\,\text{sec}$, then

$T(k) = 4$

If $0.25\,\text{sec} < T(k) < U$, then

$T(k) = T(k)$

If $T(k) < 0.25\,\text{sec}$, then

$T(k) = 0.25$

In this way, with an appropriate selection of U, $T(k)$ is only altered if there is a considerable change in the measurement residual, i.e., in a maneuver.

Basically, the algorithm of Update Time Interval Determination Method-2 is summarized in the following simplified flow chart:

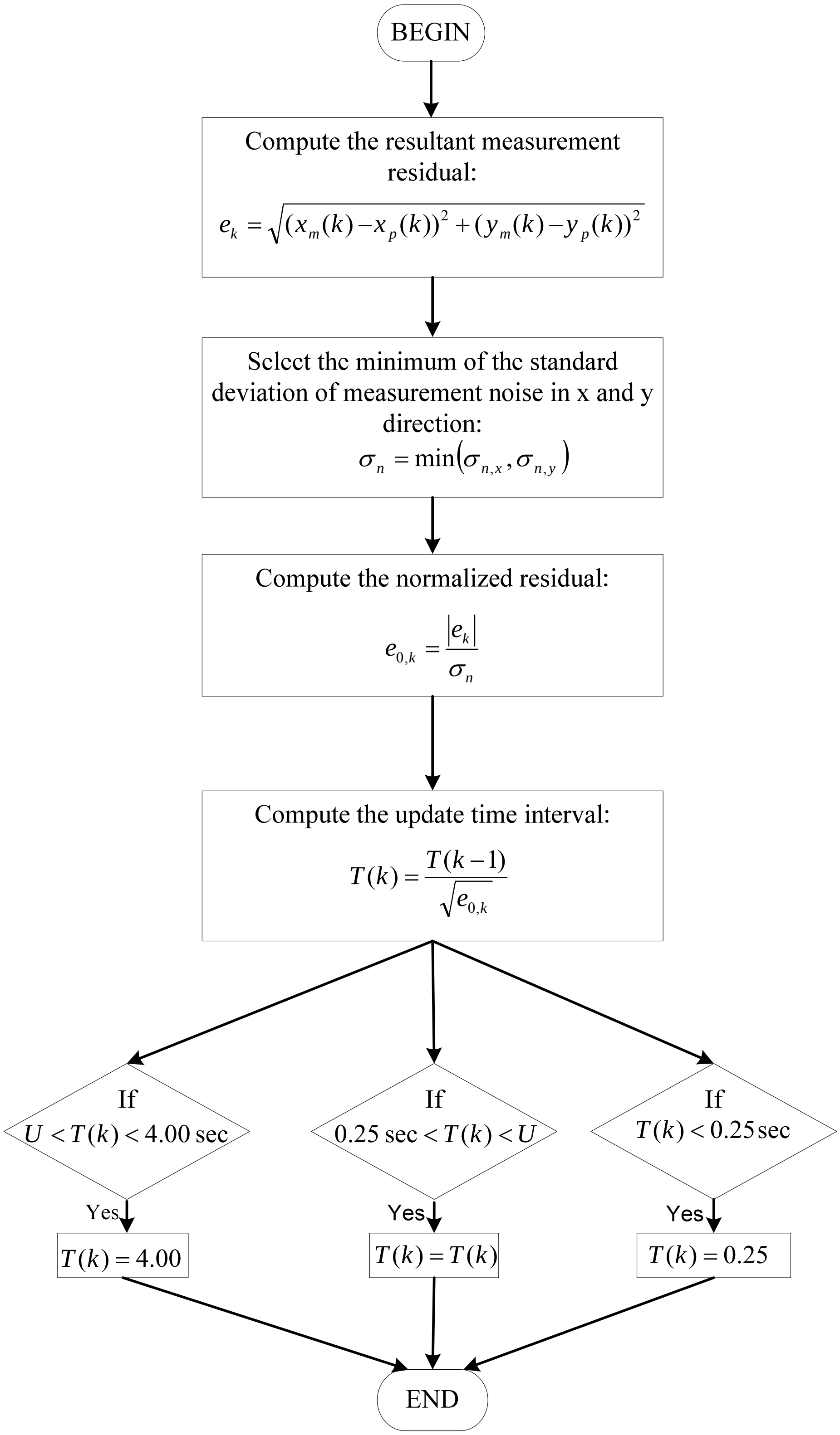

Figure 3-2: Algorithm of Update Time Interval Determination Method-2

CHAPTER 4

KALMAN FILTER AND APPLICATION OF ADAPTIVE UPDATE TIME INTERVAL ALGORITHMS

In this chapter, a brief summary of Kalman filtering and necessary theoretical bases are provided. Application of adaptive update time interval algorithm to the Kalman filter equations is proposed and explained in a detailed way. Moreover, the methods that exist in literature for adaptive update time interval algorithms are investigated in terms of compatibility for Kalman filtering nature. The methods which will be used in this thesis throughout adaptive Kalman filter implementations are introduced.

4.1 OVERVIEW OF KALMAN FILTER

In order to use Kalman filter for state estimation in tracking problems the below main assumptions/conditions must be satisfied:

- State and measurement models must be linear.
- Process and measurement noises must be independent.
- Process and measurement noises must have Gaussian distribution.

Kalman filter is widely used for the state estimation of a target. Kalman filter is the optimal state estimator (provides optimal MMSE, minimum mean square estimate) [7, 8, 9, 10] for the linear stochastic system that can be described by following state and measurement equations:

$$x_{k+1} = A_k x_k + B_k u_k + G_k w_k \tag{4-1}$$

$$y_k = C_k x_k + H_k v_k \tag{4-2}$$

Equation (4-1) is the state model equation for a discrete time linear stochastic system [7] and Equation (4-2) is the measurement model equation for a discrete time linear stochastic system [7], where:

x_k : the state vector at time t_k

y_k : the measurement vector at time t_k

u_k : the known input at time t_k

w_k : the white Gaussian process noise, with zero mean and covariance matrix Q_k at time t_k

v_k : the white Gaussian measurement noise, with zero mean and covariance matrix R_k at time t_k

Moreover, the initial state x_0 is a Gaussian random variable with mean 0 and covariance matrix Σ_0. $\{x_0, w_0, w_1,...w_k, v_0, v_1,..., v_k\}$ are all assumed to be independent random variables.

The matrices A_k, B_k, G_k, C_k, H_k are time varying matrices at data point t_k.

Also, for no input case $(u_k = 0)$, the state and measurement equations are described as given below:

$$x_{k+1} = A_k x_k + G_k w_k \tag{4-3}$$

$$y_k = C_k x_k + H_k v_k \tag{4-4}$$

A_k : the state transition model matrix at time t_k

G_k : the process noise control matrix at time t_k

C_k : the measurement model matrix at time t_k

H_k : the measurement noise matrix at time t_k

The detailed explanations and meanings of some of the matrices are given below:

A_k, the state transition model matrix, updates the state vector at each time step of the tracking. The state transition model matrix updates each position by adding the update time interval between each radar measurement multiplied by the velocity in the same dimension [14].

G_k, the process noise control matrix

C_k, the measurement model matrix, converts the state vector into a measurement vector by eliminating all the velocities since they cannot be measured. This is accomplished by multiplication with the measurement model matrix, which removes every velocity in state vector [14].

Q_k, the covariance matrix of the process noise w_k, describes the ambiguity in the target's motion. Although CV model is primarily used to track the non-maneuvering segments of the target motion, using higher process noise covariance values in Q matrix allows tracking the maneuvering segments of the target motion [23]. For an accurate estimation of the target's state during the whole trajectory, the process noise Q_k must be selected by taking into account the possible maneuvering motion of the target.

Since the Kalman filter depends on recursive state prediction and state estimation steps, there is no need to store all the previous measurements and there is no need to reprocess all the previous data at each update time step.

The Kalman filter being a recursive estimator, to compute the estimate of the current state, only the estimated state from the previous time step and the current measurement are needed [12].

Kalman filter operation is based on predicting the state of the target and then by using actual observations correcting the prediction and obtaining a more accurate state estimate of the target.

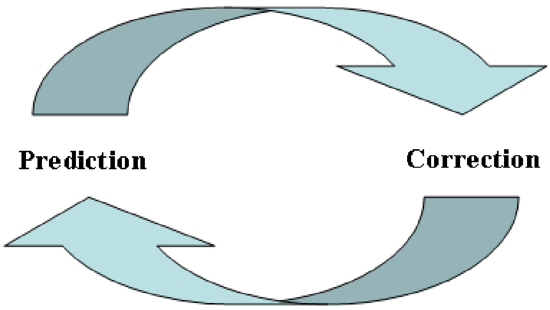

Figure 4-1: Kalman Filter's Recursive Cycle

Kalman filter equations can be derived in two phases namely: **Prediction** and **Correction** stages [7, 12]:

The prediction stage uses the state estimate from the previous time step to produce a prediction of the state at the current time step. The predicted state estimate is the state at the current time step, but it does not include observation information from the current time step.

In the correction stage, the current prediction is combined with current observation information to refine the state estimate [12].

First Stage: (Prediction stage to have the predicted state vector and the predicted error covariance matrix)

The first phase of each iteration of the Kalman filter is the prediction stage of state vector and error covariance matrix. In this stage, the predictions of the state vector and the error covariance matrix, which is a measure of how reliable the prediction is, are calculated.

By the prediction stage, the conditional probability density function of the state is found. Since the conditional probability density function of the state is a Gaussian process, this density is represented by its mean vector and covariance matrix. The mean vector and the covariance matrix, which measures the covariance between predicted and actual states, are updated according to the following equations:

$$E\left(x_{k+1} \middle| Y^k\right) = x_{k+1|k} = A_k x_{k|k} \tag{4-5}$$

$$\operatorname{cov}\left(x_{k+1} \middle| Y^k\right) = \Sigma_{k+1|k} = A_k \Sigma_{k|k} A_k^T + G_k Q G_k^T \tag{4-6}$$

where,

Y^k : the set of all the measurements up to time k i.e.

$$Y^k = \{y_0, \ldots, y_k\} \tag{4-7}$$

Second Stage: (Correction stage to have the state estimate and the error covariance estimate)

This stage includes the calculation of the measurement residual (innovation), the residual (innovation) covariance, Kalman gain, state estimate, error covariance estimate) [14].

In this stage, firstly the measurement is predicted using all the measurements up to time k:

$$E\left(y_{k+1}\middle|Y^{k}\right)=y_{k+1|k}=C_{k+1}x_{k+1|k} \tag{4-8}$$

Then, by subtracting the predictions of the measurement from actual measurement, the measurement residual (difference between the actual measured positions and the predicted positions) are obtained:

$$e_{k+1}=y_{k+1}-y_{k+1|k} \tag{4-9}$$

After that, the residual covariance, which measures covariance between predicted and measured position states, is found:

$$\operatorname{cov}(e_{k+1})=\Sigma_{k+1|k}^{y}=C_{k+1}\Sigma_{k+1|k}C_{k+1}^{T}+H_{k+1}RH_{k+1}^{T} \tag{4-10}$$

When updating state estimates, the Kalman filter includes a weighting factor known as the Kalman gain matrix into state estimate equation.

Basically, Kalman gain represents whether the measurements are reliable or not. In other words, if the measurement noise is high, Kalman gain is relatively low so that the estimated state vector and error covariance matrix are closer to the predicted state vector and covariance matrix.

Kalman gain is given with the below formula:

$$L_{k+1}=\Sigma_{k+1|k}C_{k+1}^{T}\left[C_{k+1}\Sigma_{k+1|k}C_{k+1}^{T}+H_{k+1}RH_{k+1}^{T}\right]^{-1} \tag{4-11}$$

where,

$$\Sigma_{k+1|k} C_{k+1}^{T} \left[C_{k+1} \Sigma_{k+1|k} C_{k+1}^{T} + H_{k+1} R H_{k+1}^{T} \right] = \mathrm{cov}(e_{k+1}) \tag{4-12}$$

Lastly, the estimations of the state and error covariances are performed:

By using the predicted state and the predicted measurements, estimation of the state vector and the error covariance matrix of the state vector are found by the following equations:

$$E\left(x_{k+1} \middle| Y^{k+1}\right) = x_{k+1|k+1} = x_{k+1|k} + L_{k+1}\left(y_{k+1} - y_{k+1|k}\right) \tag{4-13}$$

$$\mathrm{cov}\left(x_{k+1} \middle| Y^{k+1}\right) = \Sigma_{k+1|k+1} = \left(I - L_{k+1} C_{k+1}\right) \Sigma_{k+1|k} \tag{4-14}$$

Note that the covariance matrix does not depend on the measurements.

By using the following initial conditions for the state and the covariance matrix, the state and the covariance matrix are estimated recursively by Kalman filter.

$$x_{0|-1} = x_0 = E\left(x_0\right) \tag{4-15}$$

$$\Sigma_{0|-1} = \Sigma_0 \tag{4-16}$$

4.2 CONVENTIONAL USAGE OF KALMAN FILTER WITH FIXED UPDATE TIME INTERVAL

Conventionally, Kalman filtering is used in target tracking problems by fixed update time interval due to the traditional nature of track while scan radars. In this type of radars, the antenna system does not get a feedback of information from tracking subsystem of radar. Moreover, since these radars rotate mechanically, it takes relatively long time to change beam direction of the antenna [24].

4.3 USAGE OF KALMAN FILTER WITH ADAPTIVE UPDATE TIME INTERVAL

In contrast to mechanically steerable radars, Phased array radars offer shorter update time intervals by their electronic beam steering property. In order to use this capability of phased array radars efficiently, feedback information from tracking subsystem of radar to the antenna is provided. Depending on the target motion, phased array radars decide whether or not the target maneuvers and adjusts the next update time interval accordingly at each track update data point [24].

Application of variable update time interval on Kalman filter equations seems cumbersome at first look. The main reason for this is that classical Kalman filter equations (with fixed update rate) are already composed of relatively complex and recursive prediction and correction equations.

It seems to be a problem when the track update time interval between the two consecutive track update data points is not fixed. Since it is necessary to know the change in update time interval in order to find the change in position of the target, A_k and G_k matrix are needed to be computed at each track update by using the next update time interval [14].

Thus, an adaptive update time interval algorithm should be created to find the next update time interval $T(k)$, which is the track update time interval between data points t_k and t_{k+1}.

To find the time varying A_k and G_k matrix at each track update data point, A_k and G_k matrix based on the next update time interval should be obtained. Once A_k and G_k matrix are found, then running of the Kalman filter prediction and correction equations can be carried out [14].

4.3.1 State Model and Measurement Model for Adaptive Kalman Filter

In this thesis, CV model is used as a motion model in 2D Cartesian coordinates for adaptive Kalman filter. Target motion and measurements are modeled with the state and measurement equations as described below:

State model:

$$x_{k+1} = A_k x_k + G_k w_k \tag{4-17}$$

where,

$$A_k = \begin{bmatrix} 1 & 0 & T(k) & 0 \\ 0 & 1 & 0 & T(k) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; G_k = \begin{bmatrix} T^2(k)/2 & 0 \\ 0 & T^2(k)/2 \\ T(k) & 0 \\ 0 & T(k) \end{bmatrix} \tag{4-18}$$

$$x_k = \begin{bmatrix} d_{x,k} \\ d_{y,k} \\ V_{x,k} \\ V_{y,k} \end{bmatrix}, w_k = \begin{bmatrix} w_{x,k} \\ w_{y,k} \end{bmatrix} \tag{4-19}$$

x_k : the state vector that represents the position and velocities of the target at *x* and *y* Cartesian coordinates at data point t_k

$w_{x,k}$ and $w_{y,k}$: the Gaussian process noises, with zero mean and covariance matrix Q at data point t_k

A_k : the state transition model matrix at data point t_k, which updates the state vector each time step of tracking, taken as above, by using the fact that each position is updated by adding the time interval between each radar measurement multiplied by the velocity in the same dimension [14]

G_k : the process noise control matrix at data point t_k, which describes the ambiguity in the target motion, taken as above, by assuming that process noise is acting on the acceleration of the target

Here, A_k and G_k matrices are taken as time varying by using the fact that the phased array radar allows using adaptive update time intervals between consecutive track updates depending on the motion of the target [22, 23].

$T(k)$: the update time interval between the data points t_k and t_{k+1}

i.e.

$$T(k) = t_{k+1} - t_k \tag{4-20}$$

Measurement model:

$$y_k = Cx_k + Hv_k \tag{4-21}$$

where,

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \; H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; y_k = \begin{bmatrix} y_{x,k} \\ y_{y,k} \end{bmatrix}, \; v_k = \begin{bmatrix} v_{x,k} \\ v_{y,k} \end{bmatrix} \tag{4-22}$$

y_k : the measurement vector that represents position coordinates of the measurements at x and y directions at data point t_k

$v_{x,k}$ and $v_{y,k}$: the Gaussian measurement noises, with zero mean and covariance matrix R at data point t_k

C : the measurement model matrix

H : the measurement noise matrix

4.4 DEVELOPMENT OF KALMAN FILTER WITH ADAPTIVE UPDATE TIME INTERVAL

State and measurement model equations are described below:

$$x_{k+1} = A_k x_k + G_k w_k, \qquad \text{where, } p(w_k) \sim N(0, Q) \tag{4-23}$$

$$y_k = C x_k + H v_k, \qquad \text{where, } p(v_k) \sim N(0, R) \tag{4-24}$$

The following assumptions must be satisfied:

w_k, v_k are the Gaussian process noise and Gaussian measurement noise respectively.

Initial state x_0 is a Gaussian random variable.

$\{x_0, w_0, w_1, ..., w_k, v_0, v_1, ..., v_k\}$ are all independent random variables.

4.4.1 Algorithm for Kalman Filter with Adaptive Update Time Interval

Algorithm for Kalman filter with adaptive update time interval is given below:

1- Create noisy measurements (polar noise is assumed) with appropriate samples throughout the true target trajectory.

2- Initialize system matrices A_0, A_1, G_0, G_1, C, H, Q, R and update time intervals $T(k)$ at data points t_0 and t_1 where $T(k)$ denotes the update time interval between track update data points t_{k+1} and t_k :

i. Initialize the update time intervals at data points t_0 and t_1 i.e. assign to $T(0)$ and $T(1)$ appropriate values.

ii. Initialize A_k and G_k matrices at data points t_0 and t_1 i.e. initialize A_0, A_1 and G_0, G_1 in order to use in the prediction equations of Kalman filter at data points t_1 and t_2 (A_k is created by using the fact that the position of the target at data point t_{k+1} is updated by adding the time interval between two consecutive radar measurements (the time interval between track update data points of t_{k+1} and t_k) multiplied by the velocity in the same direction at data point t_k. G_k is created by assuming that the process noise w_k is acting on the acceleration of the target, not on the position or velocity):

$$A_0 = \begin{bmatrix} 1 & 0 & T(0) & 0 \\ 0 & 1 & 0 & T(0) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad G_0 = \begin{bmatrix} T^2(0)/2 & 0 \\ 0 & T^2(0)/2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4-25}$$

$$A_1 = \begin{bmatrix} 1 & 0 & T(1) & 0 \\ 0 & 1 & 0 & T(1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad G_1 = \begin{bmatrix} T^2(1)/2 & 0 \\ 0 & T^2(1)/2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4-26}$$

iii. Initialize constant C and H matrices. (C is created by the fact that C converts the state vector into a measurement vector by eliminating all the velocities since the velocities cannot be measured. H is created assuming that there is no coupling of measurement errors in different coordinates. C and H matrices are constant during the whole target tracking process) :

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4-27}$$

iv. Initialize Q matrix, the covariance matrix of process model noise (Q is created so that this matrix can describe the ambiguity in the target motion. Using relatively higher values for process noise variance in Q matrix allows to track the target motion in maneuvering segments. Q is assumed to be constant during the whole target tracking):

$$Q = \begin{bmatrix} \text{varProc} & 0 \\ 0 & \text{varProc} \end{bmatrix} \tag{4-28}$$

where,

varProc: the covariance of the assumed process noise in the filter

v. Initialize R, the covariance matrix of measurement model noise (R is created by using the same covariance values as the actual measurement covariance values):

$$R = \begin{bmatrix} \text{var}\,Meas_x & 0 \\ 0 & \text{var}\,Meas_y \end{bmatrix} \tag{4-29}$$

where,

$\text{var}\,Meas_x$: the covariance of the measurement noise in x coordinate

$\text{var}\,Meas_y$: the covariance of the measurement noise in y coordinate

3- Initialize Σ_0, the covariance matrix of initial state vector x_0 with an appropriate value.

4- Compute $\Sigma_{0|-1}$, L_0, $\Sigma_{0|0}$, $\Sigma_{1|0}$, L_1, $\Sigma_{1|1}$:

i. Calculate $\Sigma_{0|-1}$:

$$\Sigma_{0|-1} = \Sigma_0 \tag{4-30}$$

since there is no measurement before data point t_0

ii. Calculate L_0:

$$L_0 = \Sigma_{0|-1} C^T \left[C\, \Sigma_{0|-1} C^T + HRH^T \right]^{-1} \tag{4-31}$$

iii. Calculate $\Sigma_{0|0}$:

$$\Sigma_{0|0} = \left(I - L_0 C \right) \Sigma_{0|-1} \tag{4-32}$$

iv. Calculate $\Sigma_{1|0}$:

$$\Sigma_{1|0} = A_0 \Sigma_{0|0} A_0^T + G_0 Q G_0^T \tag{4-33}$$

v. Calculate L_1:

$$L_1 = \Sigma_{1|0} C^T \left[C\, \Sigma_{1|0} C^T + HRH^T \right]^{-1} \tag{4-34}$$

vi. Calculate $\Sigma_{1|1}$:

$$\Sigma_{1|1} = \left(I - L_1 C \right) \Sigma_{1|0} \tag{4-35}$$

5- Initialize first two phased array radar measurements at data points t_0 and t_1 by using $T(0)$ which is the initial time update interval value.

i.e. Assign values to $y_{x,0}$, $y_{y,0}$, $y_{x,1}$, $y_{x,1}$, where $y_{x,k}$ denotes the measurement in *x* coordinate at data point t_k.

6- Initialize the state estimates $x_{0|0}$ and $x_{1|1}$ at data points t_0 and t_1:

To initialize $x_{0|0}$, the first measurement $y_{x,0}$, $y_{y,0}$ of radar at data point t_0 is used:

$$x_{0|0} = \begin{bmatrix} y_{x,0} \\ y_{y,0} \\ 0 \\ 0 \end{bmatrix} \tag{4-36}$$

To initialize the position components of $x_{1|1}$, the measurement of radar at data point t_1 is used. To initialize the velocity components of $x_{1|1}$, the measurements of radar at data point t_0 and t_1 (the measurements of first two consecutive scans) are used with the information of initial update time interval $T(0)$:

$$x_{1|1} = \begin{bmatrix} y_{x,1} \\ y_{y,1} \\ (y_{x,1} - y_{x,0}) / T(0) \\ (y_{y,1} - y_{y,0}) / T(0) \end{bmatrix} \tag{4-37}$$

7- Initialize state predictions $x_{0|-1}$, $x_{1|0}$:

Since Kalman filter starts to make state predictions starting from the data point t_2, $x_{0|-1}$ and $x_{1|0}$ are initialized to be zero vectors (initial state vector) :

$$x_{0|-1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \; x_{1|0} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4-38}$$

8- Find phased array radar measurement at data point t_2, by using the time update interval value of $T(1)$ with the information of phased array radar's previous observation data point t_1. (The measurement at data point t_2 will be then used in calculation of $x_{2|2}$.)

FOR k=1:N

(Iteration starts at the data point $t_{k+1} = t_2$. Note that Kalman filter starts to make state prediction and state estimate starting from the data point t_2. Iteration continues until the end of the target trajectory.)

9- Apply prediction step of Kalman filter (Predict the mean vector and the covariance matrix of state x_k for data point t_{k+1})

i. Project the state ahead by using previously calculated $x_{k|k}$ and A_k:

$$E\left(x_{k+1} \middle| Y^k\right) = x_{k+1|k} = A_k x_{k|k} \tag{4-39}$$

ii. Project the predicted error covariance (i.e. state prediction covariance) ahead by using previously calculated $\Sigma_{k|k}$, A_k and G_k:

$$\mathrm{cov}\left(x_{k+1}\middle|Y^k\right)=\Sigma_{k+1|k} = A_k\Sigma_{k|k}A_k^T + G_kQG_k^T \tag{4-40}$$

10- Apply correction (measurement update) step of Kalman filter to find the updated state estimate and the updated error covariance matrix of the state at the current data point t_{k+1}:

i. Predict the measurement at data point t_{k+1}:

$$E\left(y_{k+1}\middle|Y^k\right)=y_{k+1|k}=Cx_{k+1|k} \tag{4-41}$$

ii. Find measurement residual e_{k+1} (difference between the actual measured positions and the predicted measurements) by subtracting the predictions of the measurement vector from actual measurement at data point t_{k+1}:

$$e_{k+1} = y_{k+1} - y_{k+1|k} \tag{4-42}$$

iii. Compute the residual covariance matrix (innovation covariance matrix) at the current data point t_{k+1} (This matrix measures covariance between predicted and measured position states):

$$\mathrm{cov}(e_{k+1}) = \Sigma_{k+1|k}^y = C\,\Sigma_{k+1|k}C^T + HRH^T \tag{4-43}$$

iv. Compute the Kalman gain at the current data point t_{k+1}:

$$L_{k+1} = \Sigma_{k+1|k}C_{k+1}^T\left[C_{k+1}\Sigma_{k+1|k}C_{k+1}^T + H_{k+1}RH_{k+1}^T\right]^{-1} \tag{4-44}$$

where,

$$C_{k+1}\Sigma_{k+1|k}C_{k+1}^T + H_{k+1}RH_{k+1}^T = \mathrm{cov}(e_{k+1}) \tag{4-45}$$

v. By using the predicted state and the predicted measurements, estimate state vector and the state error covariance matrix of the state vector:

Update the state estimate at data point t_{k+1} by using the calculated $x_{k+1/k}$, L_{k+1}, and the measurement residual e_{k+1}:

$$E\left(x_{k+1}\middle|Y^{k+1}\right)=x_{k+1|k+1}=x_{k+1|k}+L_{k+1}\left(y_{k+1}-y_{k+1|k}\right) \qquad (4\text{-}46)$$

where, $e_{k+1}=y_{k+1}-y_{k+1|k}$ (4-47)

Update the state error covariance matrix by using the calculated $\Sigma_{k+1|k}$ and the current Kalman gain L_{k+1}:

$$\operatorname{cov}\left(x_{k+1}\middle|Y^{k+1}\right)=\Sigma_{k+1|k+1}=\left(I-L_{k+1}C\right)\Sigma_{k+1|k} \qquad (4\text{-}48)$$

11- Find $T(k+1)$, which is the update time interval for the next track update (i.e. the time interval from the current track data point t_{k+1} up to the next track data point t_{k+2}). To find $T(k+1)$, an "Update Time Interval Determination Algorithm" will be used:

i. The methods to be used in this step are presented in Section 4.5 in a detailed way.

12- Find the next measurement at the data point t_{k+2} in order to use this measurement in the calculation of state estimate at the next data point t_{k+2}.

13- Compute the next values of A and G matrices i.e. A_{k+1} and G_{k+1} by using the update time interval found in step 11, in order to use these matrices in the prediction equations of the Kalman filter for next iteration,

i. Create A_{k+1} by using the fact that the position of the target is updated by adding the update time interval between current and next radar measurement multiplied by the velocity in the same dimension.

$$A_{k+1} = \begin{bmatrix} 1 & 0 & T(k+1) & 0 \\ 0 & 1 & 0 & T(k+1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4-49}$$

ii. Create G_{k+1} by assuming that the process noise w_k is acting on the acceleration) :

$$G_{k+1} = \begin{bmatrix} T^2(k+1)/2 & 0 \\ 0 & T^2(k+1)/2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4-50}$$

END for

(Iteration finishes by the end of the target trajectory.)

Basically, one cycle of Kalman filter with variable update time interval case can be summarized in the following simplified flow chart:

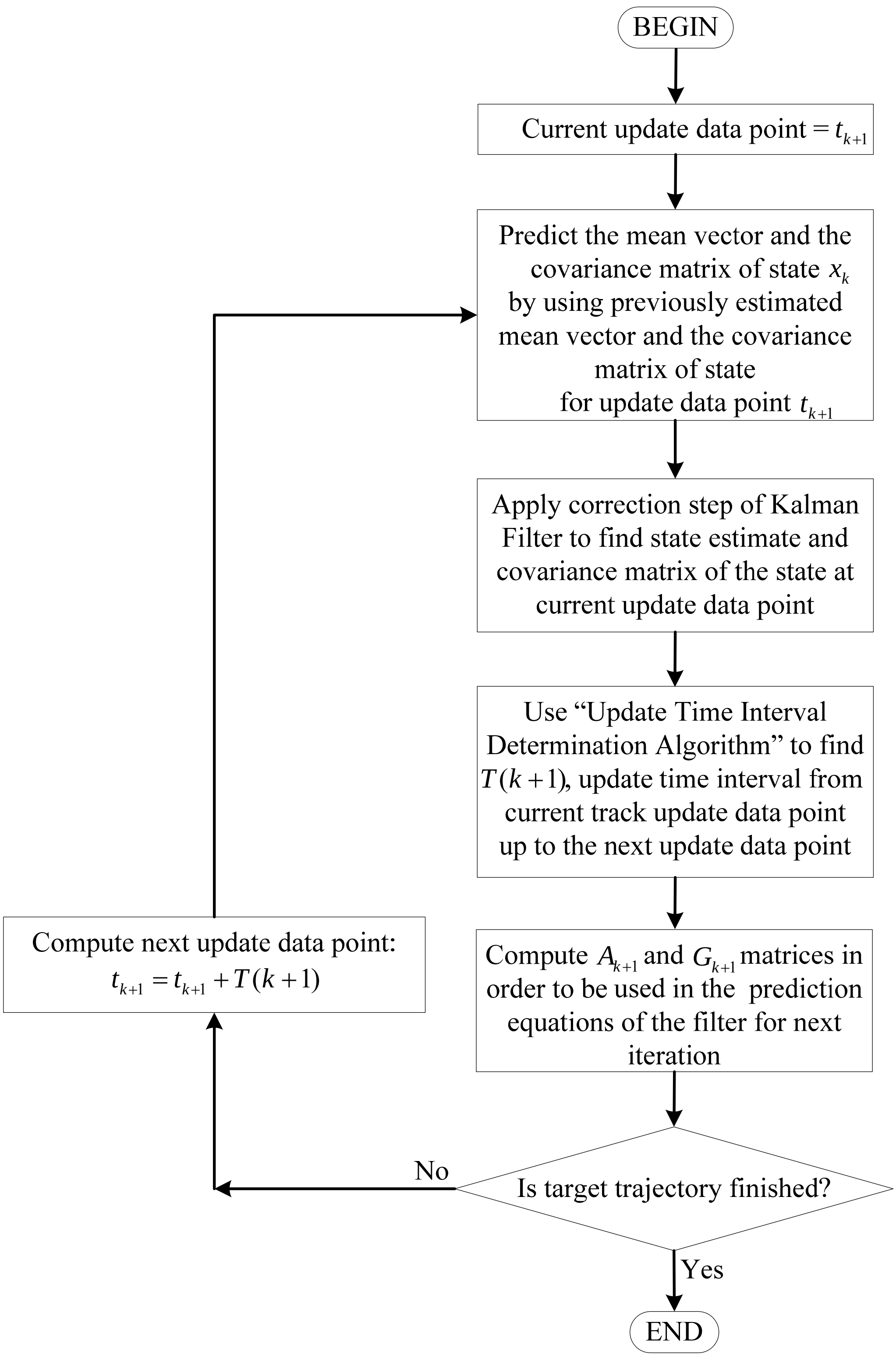

Figure 4-2: Algorithm for One Cycle of Kalman Filter with Variable Update Time Interval Case

4.5 INVESTIGATION OF METHODS FOR ADAPTIVE UPDATE TIME INTERVAL DETERMINATION ALGORITHMS ON KALMAN FILTER

As mentioned previously, the degree of adaptability of the target motion to the model in the implemented filter may differ depending on the nature of the implemented filter.

Within respect to the methods proposed for adaptive update rate algorithms in the literature which are based on measurement residual and predicted error covariance matrix, only the measurement residual based approach may be implemented on Kalman filter.

In Kalman filter, the covariance matrix converges and mainly considered to be constant during the filter operation. Moreover, the covariance matrix update is done offline and it doesn't depend on the measurements or target motion. Since the degree of adaptability of the target motion to the model is not linked to the Kalman filter's predicted error covariance matrix, predicted error covariance based adaptive update time interval algorithms cannot be applied on Kalman filter. The main reason for this is that it is not possible to decide whether or not the target deviates from the motion model by observing the covariance matrix in Kalman filter [18].

In this study, since the application of adaptive update rate on the single target tracking problem by using Kalman filter is focused on, measurement residual based approach will be considered for controlling of the next track update time interval.

Historically, adaptive update rate algorithms to vary the track update time interval by an amount that tends to maintain a constant measurement residual are developed for Alpha-Beta filter. This approach, which is based on keeping the track update time interval by an amount that tends to maintain a constant measurement residual (difference between the actual measured position and the predicted measurement), used in the development of adaptive update rate algorithms for Alpha-Beta filter will be extended to the Kalman filter.

In this thesis, the two update time interval determination algorithms of [2] and [5] proposed in literature will be implemented on Kalman filter. In this part, these algorithms will be summarized by concentrating on the details for the application on Kalman filter.

4.5.1 Update Time Interval Determination Method-1 [2]

This method is completely same with the method detailed for the application on Alpha-Beta filter. As described in Section 3.3.1, in this method next time interval $T(k)$ is selected by comparing the calculated measurement residual with the standard deviation of measurement noise at each data point.

As given in Section 3.3.1, the next update time interval is chosen from a discrete set of values depending on the maximum of the magnitudes of the measurement residuals in x and y directions. Upper and lower limits are placed on the update time interval to avoid impractically large or small update time interval values. The set of update time intervals that can be used are 0.25, 0.50, 1.00, 2.00 and 4.00 sec.

To find the update time interval for next track update, "Update Time Interval Determination Method-1" will be implemented on Kalman filter, by using the measurement residuals in x and y directions i.e. $e_{k+1,x}$ and $e_{k+1,y}$ which are computed based on the Kalman filter equations.

4.5.2 Update Time Interval Determination Method-2 [5]

This method is completely same with the method detailed for the application on Alpha-Beta filter. As described in Section 3.3.2, in this method the next update time interval is selected by computing $T(k)$ directly with the formula given in Equation

(3-20) and then selecting the direct continuous result of the formula or selecting 4.00 sec or selecting 0.25 sec after comparing the calculated value of $T(k)$ with the threshold U at each data point as described in Section 3.3.2.

As given in Section 3.3.2 in this method, the next update time interval may take the maximum or minimum values which are previously defined or it may take the continuous value between 0.25 and threshold U which is found to be the result of the formula given in Equation (3-20).

To find the update time interval for next track update, "Update Time Interval Determination Method-2" will be implemented on Kalman filter, by using the measurement residuals in x and y directions i.e. $e_{k+1,x}$ and $e_{k+1,y}$ which are computed based on the Kalman filter equations.

CHAPTER 5

SIMULATIONS AND EVALUATIONS

This chapter includes the implementations of the methods for adaptive update time interval algorithms on Alpha-Beta filter and Kalman filter to sample target trajectories, based on the methods described in Chapter 3 and Chapter 4 and the system model defined in Chapter 2. The assumptions that are used during this study are given and the comparison parameters for simulation results are introduced. Besides, the application of Monte Carlo simulations for adaptive update time interval case is explained. The simulation results are compared and evaluated for different simulation cases.

5.1 AIM OF THE SIMULATIONS

In the simulations of this thesis, by the flexibility of phased array radar for using variable update time interval, adaptive update time interval determination algorithms are implemented on the tracking filter during estimation of the target's state under some assumptions.

The approach developed in the literature for adaptive update rate algorithms for the Alpha-Beta filter will be extended to the Kalman filter. The behavior of the adaptive update rate algorithms will be investigated for tracking a single maneuvering target which can make 90° turn in its trajectory by using a Kalman filter.

Since the aim of this study is to apply the approach developed in the literature for adaptive update rate algorithms on Alpha-Beta filter to the Kalman filter, the simulations are mainly implemented by using the Kalman filter. However, in order to demonstrate the application of adaptive update time interval algorithms developed for Alpha-Beta filter historically and to show clearly the performance of adaptive time interval algorithms on Alpha-Beta filter, firstly update time interval determination methods developed in literature will be implemented on Alpha-Beta filter for a target trajectory which includes a maneuvering segment consisting of a single 90° maneuver.

5.1.1 Assumptions for the Target and Environment throughout the Simulations

The assumptions that are used during the simulations are given below:

- 2-dimensional target motion (in x and y coordinates) is considered. The ellipsoid shape of the earth and the earth's rotation are ignored.
- The target is assumed to be an airborne target like an aircraft.
- In order to simplify the problem, a model is built with a single target and a single target tracker.
- Radar location is fixed and known.
- Single model filter is used in which the update time interval is varied.
- Measurements are noisy (white Gaussian measurement noise is considered).
- Measurements are the positions of the target in 2D polar coordinates i.e. the azimuth angle (in radian) and the range (in meter).

- Measurement noise is given in polar coordinates i.e. the azimuth angle (in radian) and the range (in meter) to make the simulations more realistic.

- The target focused on during this study moves with mainly constant velocity and sometimes makes maneuver which is a coordinated turn maneuver.

- The target moves with constant speed (magnitude of the velocity vector) along the trajectory consisting of maneuvering segment with 90° turn.

- There are no clutter and false alarms in the environment, so there is no need for data association step to track the target.

- No need for track initiation step of target tracking since track initiation is assumed to be completed.

- Only "estimation step" of target tracking is necessary to track the target.

- The CV motion model is used in the tracking filter.

- Discrete-time filter, where the measurements occur and the state is estimated at discrete data points in time, is used.

- The state equation does not contain any input terms.

- There are no missed detections.

- There are no track losses due to the terrain structure. The radar is in line of sight with the target throughout the trajectory.

When real time applications are considered, some of these assumptions may not be realistic. Nevertheless, in this study the aim is to demonstrate that the use of adaptive update time interval is effective in terms of maintaining tracking accuracy and resource allocation of radar. It is believed that this study can be easily extended in order to be used for real time applications.

5.1.2 Construction of Simulations

In order to demonstrate the properties of target tracking when adaptive update time interval is used, different scenarios are constructed for the target trajectory. Then, several adaptive update time interval algorithms are applied to these scenarios with different simulation parameters in order to clearly analyze the target tracking problem with adaptive update time interval case.

All the scenarios for the target trajectory in the simulations are carried out offline with artificial data. Artificial noise is added on the measurements. The target's states are tracked via noisy measurements. Moreover, the tracking filter uses the actual values of the measurement covariances. So, the measurement noise model is the same as the one used while creating the artificial measurement.

The tracks are generated by using manually selected initial conditions. Also, the tracking filter is initialized by using the first two measurements.

In this study, Alpha-Beta filter with CV Model in 2D Cartesian coordinates and KF with CV Model in 2D Cartesian coordinates are implemented during the simulations.

5.1.2.1 Radar Condition

The radar location is assumed to be fixed and known at (x_r, y_r). In principle, the phased array radar observes the target in terms of the range, azimuth angle and elevation angle. For simplicity, it is assumed that the observation is in the polar coordinates of the range and the azimuth angle [18].

5.1.2.2 Target Trajectories

The true target trajectories that are mainly used in the simulations are given in Figure 5-1 and Figure 5-2.

True Target Trajectory-1:

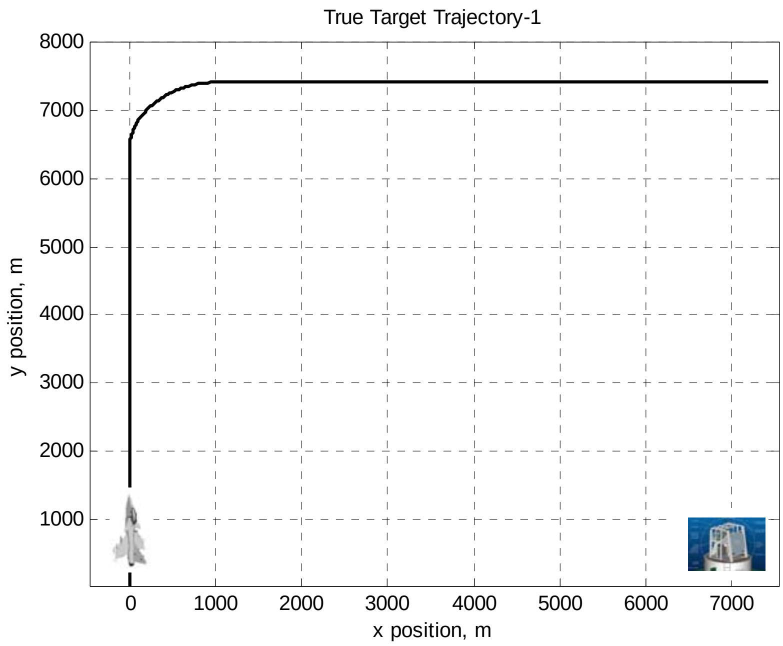

Figure 5-1: True Target Trajectory-1 in 2D

Details of True Target Trajectory-1:

Trajectory Duration: 144 sec

Target Speed: 100 m/sec (constant during trajectory)

Trajectory is composed of 3 segments:

Segment 1: A Non-maneuvering segment, constant velocity motion along *y* axis, lasting 64 sec, from 0 to 64 sec

Segment 2: A Maneuvering segment, 90° maneuver with constant speed, lasting 16 sec, from 64 to 80 sec

Segment 3: A Non-maneuvering segment, constant velocity motion along *x* axis, lasting 64 sec, from 80 to 144 sec

<u>True Target Trajectory-2:</u>

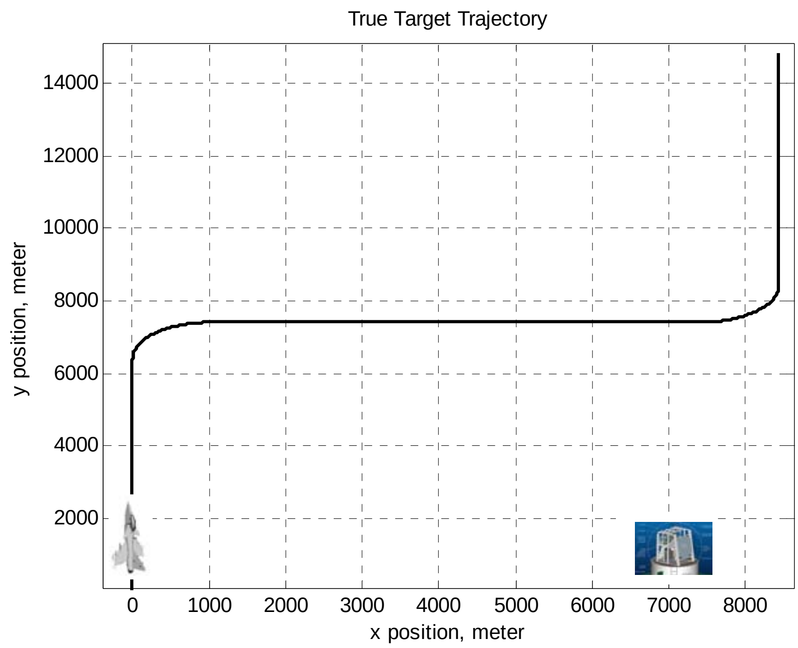

Figure 5-2: True Target Trajectory-2 in 2D

<u>Details of True Target Trajectory-2:</u>

Trajectory Duration: 224 sec

Target Speed: 100 m/sec (constant during trajectory)

Trajectory is composed of 5 segments:

Segment 1: A Non-maneuvering segment, constant velocity motion along *y* axis, lasting 64 sec, from 0 to 64 sec

Segment 2: A Maneuvering segment, 90° maneuver with constant speed, lasting 16 sec, from 64 to 80 sec

Segment 3: A Non-maneuvering segment, constant velocity motion along x axis, lasting 64 sec, from 80 to 144 sec

Segment 4: A Maneuvering segment, 90° maneuver with constant speed, lasting 16 sec, from 144 to 160 sec

Segment 5: A Non-maneuvering segment, constant velocity motion along y axis, lasting 64 sec, from 160 to 224 sec

5.2 EVALUATION CRITERIA FOR SIMULATION RESULTS

In order to analyze the application of adaptive update time interval algorithms to target tracking and to show clearly the performance of adaptive update time interval algorithm, "Estimated Trajectory", "Position Residual (measurement residual) as function of time", "Track Update Time Interval as function of time", "MSE as function of time" are demonstrated. Since the track updates may occur at different data points in adaptive update time interval case, it is not possible to show the track update time interval, the position residuals, the MSE graphs averaged over all Monte Carlo runs, at each track update data point. Thus, these graphs are demonstrated for one MC simulation.

However, in order to compare adaptive update time interval case with fixed update time interval case and to compare Update Time Interval Determination Method-1 and Method-2 in terms of RMSE, number of updates and update time interval, MC simulations are realized. The track updates may occur at different data points for adaptive update time interval case. Thus, it is not possible to calculate mean square estimation errors, averaged over Monte Carlo runs, at each data point for each Monte Carlo run and the application of MC simulations on adaptive time interval case should be different than the classical application of MC. In this study, comparison and performance analysis are performed by using 100 Monte Carlo simulations. The details of the MC simulations performed for adaptive update time interval case will be explained in detail in Section 5.2.2.

5.2.1 Overview of Monte Carlo Simulations

Monte Carlo Simulation is a technique which is often used in the analysis of tracking algorithms for the fixed update rate case. In this technique, a particular target track is corrupted with noise generated artificially by a random number generator. Then, the filtering algorithm under test is applied to estimate the states of the target by using the measurements for this particular target track [5, 6].

The square of the estimation error is computed at each track update data point. The same target track is then corrupted with noise using another sequence of random numbers, and the application of filtering algorithm is repeated by using the measurements for this particular target track. This procedure is repeated several times. Finally, at each data point, the square of the estimation error is averaged over all the MC runs, to form a mean square estimation error at each data point. The obtained mean square errors which are shown as a function of time give an indication of the time dependence of the filtering action of the target tracking algorithm. This technique is referred to as Monte Carlo simulation [5, 6].

5.2.2 Application of Monte Carlo Simulations for Adaptive Update Time Interval Case

In literature, Monte Carlo Simulations are also applied for the cases in which adaptive update time interval algorithms are used.

The tracking performance of adaptive update time interval algorithm is investigated in [2] by simulating the target trajectory corrupted with only one realization of the measurement noise. In the literature, firstly Wilkin et al. [5] have used Monte Carlo simulations when comparing the filtering action of the fixed update rate Alpha-Beta filter with the adaptive update rate case.

In this thesis, the tracking performance of the adaptive update time interval algorithms is also evaluated by using MC simulations, where the simulation results are averaged over many realizations of the measurement noise.

In adaptive update time interval algorithms, the track updates may occur at different data points. According to this characteristic of adaptive algorithms, it is not possible to calculate mean square estimation errors, averaged over Monte Carlo runs, at each data point for each Monte Carlo run.

In order to apply the Monte Carlo simulation technique to the adaptive update rate case, another procedure is needed to be adopted. In this procedure, the target trajectory is divided into smaller stages.

5.2.2.1 Comparison and Evaluation Parameters of Monte Carlo Simulations

5.2.2.1.1 Average Update Time Interval within Each Stage of the Trajectory over all MC Runs

In the analysis of the variable update algorithm, it is also of interest to compute the mean (average) update time interval since the mean update time interval provides an indication of the amount of the time that the radar is being used to track the target. The mean update time interval within each stage of the target track, averaged over all Monte Carlo runs is computed as given below [5, 6]:

$$T_{mean} = \frac{1}{M}\sum_{i=1}^{M} \frac{1}{m_i}\sum_{j=1}^{m_i} T_{ij} \tag{5-1}$$

where,

T_{ij} : the update time interval for the j^{th} measurement in the i^{th} Monte Carlo run

m_i : the total number of measurements for the i^{th} Monte Carlo run within the stage of interest.

M : the number of Monte Carlo runs

5.2.2.1.2 Average Number of Updates Within Each Stage of the Trajectory over All MC Runs

In the analysis of the variable update algorithm, another evaluation parameter which is one of the most important merits from the point of view of the efficiency of utilization of the radar, is the average number of updates. Although this parameter is related to the mean update time interval, no simple relationship exists between them.

The average number of updates within each stage of the target track, averaged over all Monte Carlo runs can be computed as given below [6]:

$$N_{mean} = \frac{1}{M} \sum_{i=1}^{M} m_i \tag{5-2}$$

where,

T_{ij} : the update time interval for the j^{th} measurement in the i^{th} Monte Carlo run

m_i : the total number of measurements for the i^{th} Monte Carlo run within the stage of interest.

M : the number of Monte Carlo runs

5.2.2.2 Implementation Details of MC Simulations

As explained in Part 5.2.2, there is a difficulty in applying MC simulations to variable update rate algorithms, since the track updates may occur at different time instants for each MC run.

In this study, in order to overcome this difficulty, for the evaluation of the simulation results, the whole target trajectory is divided into stages of equal length with 8 sec duration. Thus, the time step in the plots is equal to length of 8 sec. Each time step represents the corresponding small stage of the trajectory with duration of 8 sec. It is believed that by dividing the trajectory into small stages and obtaining the plots of average update time interval and average number of updates versus time step will give an indication of time dependence of these parameters for adaptive update rate target tracking.

The results of simulations obtained over 100 MC runs are presented in average number of updates and average update time interval graphs vs time step which represents the index of the stage of the trajectory with duration of 8 sec. As an example, if the duration of the whole trajectory is 144 sec, when the trajectory is divided into small stages with duration of 8 sec, there are 18 stages to be investigated. For example, let the index of the stage be equal to 3, then this index represents the time duration between 16 sec - 24 sec time instants of the trajectory.

Moreover, the average RMSE of position over 100 MC runs is obtained by dividing the sum of the MSE in x and y coordinates computed at each track update data point by the number of updates carried out at the corresponding update times among overall MC runs and calculating the square roots to get RMSE. It is believed that RMSE versus time graph obtained by this way will give an indication of the time dependence of RMSE for adaptive update rate case.

5.3 SIMULATION RESULTS WITH ALPHA-BETA FILTER

In this thesis, in order to apply the approach of adaptive update rate algorithms developed in literature for Alpha-Beta filter to the Kalman filter, the simulations are mainly implemented by using Kalman filter. Nevertheless, firstly the Update Time Interval Determination Methods developed in literature are implemented on Alpha-Beta filter to analyze the performance of adaptive update time interval algorithms implemented on Alpha-Beta filter.

In this part, the behavior of the adaptive update rate algorithms developed in literature for Alpha-Beta filter are investigated for target trajectory which includes maneuvering segment consisting of 90° turn maneuver. The scenario defined for Target Trajectory-1 is tracked by using an Alpha-Beta filter with CV Model in 2D Cartesian coordinates.

In order to analyze the application of adaptive time interval algorithms to target tracking with Alpha-Beta filter and to show clearly the performance of adaptive time interval algorithm, "Estimated Trajectory", "Position Residual as function of time", "Track Update Time Interval as function of time" and "MSE as function of time" graphs are demonstrated for one MC run.

5.3.1 Simulation Group 1 – Investigation of Update Time Interval Determination Method-1

Update Time Interval Determination Method-1 of [2] which is presented in Section 3.3.1 (which compares the measurement residual with the standard deviation of the measurement noise to decide the next update time interval) is used to control the update time interval by using Alpha-Beta filter with CV Model in 2D Cartesian coordinates throughout the simulations in this part. The aim is to observe the effectiveness of Update Time Interval Determination Method-1 for a target trajectory which includes maneuvering segment consisting of 90° turn maneuver.

The important parameters of the simulation are given in Table 5-1.

Table 5-1: Parameters of Simulation

Update Time Interval (sec)	Adaptive
Update Time Interval Determination Method	Update Time Interval Determination Method-1
Initial Update Time Interval (sec)	4.00 sec
Actual Measurement Noise	Range (m): Gaussian, $\sigma = 30$ Azimuth (rad): Gaussian, $\sigma = 0.003$
Radar Location	$x_r = 7000$ m, $y_r = 0$ m
Target Initial Location	x=0 m, y=0 m

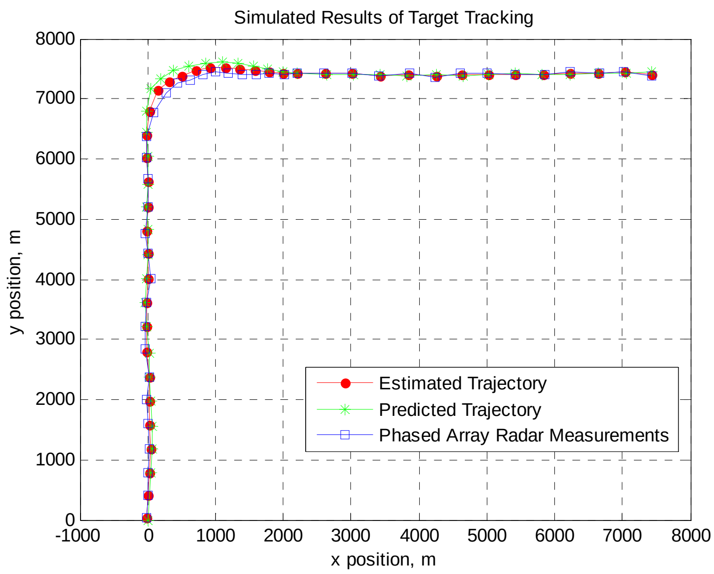

Figure 5-3: Predicted, Estimated Output of the Filter and Phased Array Radar Measurements in 2D, for one MC run

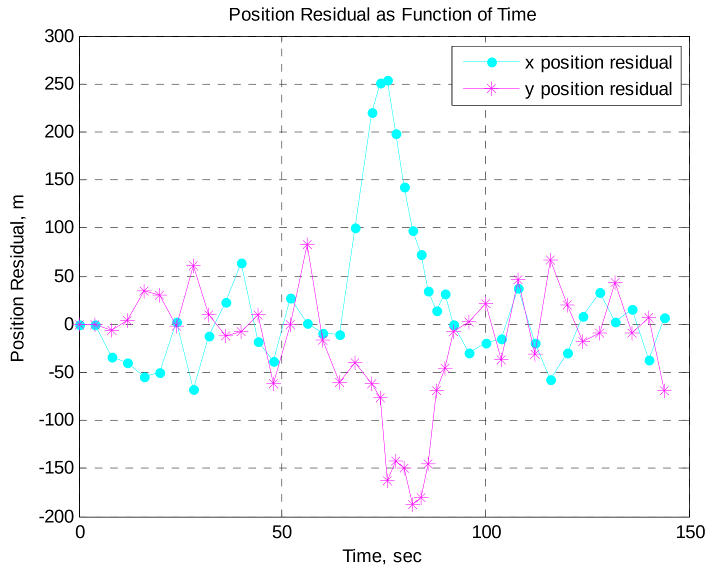

Figure 5-4: Position Residual in *x* and *y* coordinates versus Time, for one MC run

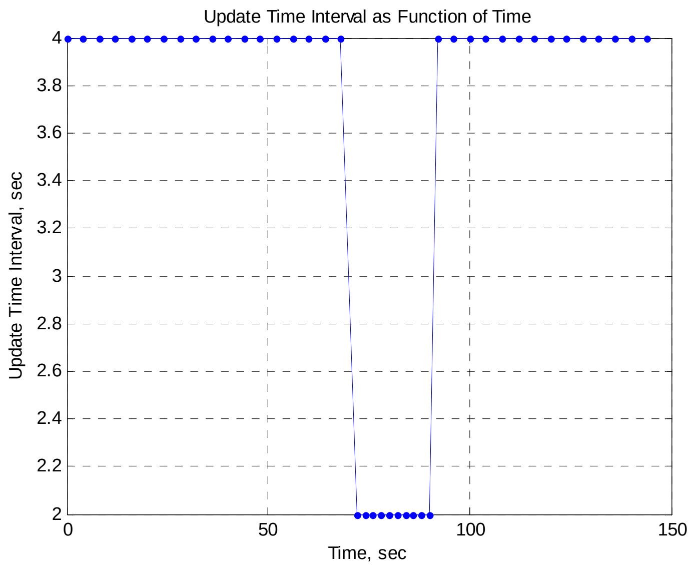

Figure 5-5: Track Update Time Interval versus Time, for one MC run

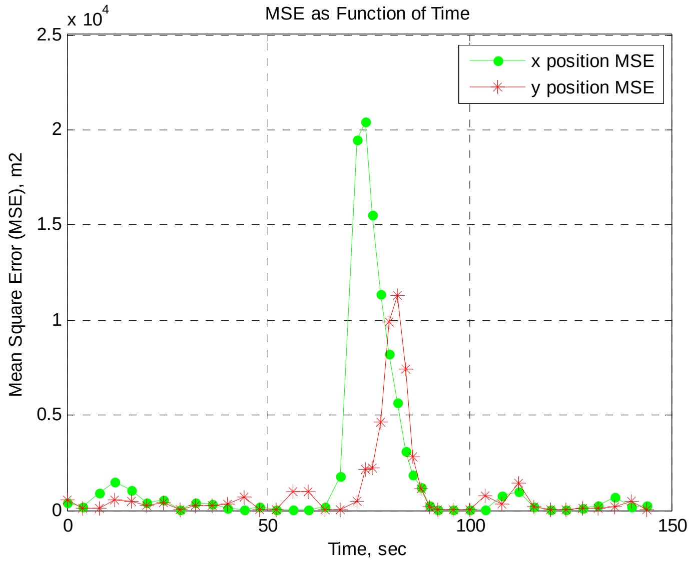

Figure 5-6: MSE in *x* and *y* coordinates versus Time, for one MC run

By implementing the Update Time Interval Determination Method-1 on Alpha-Beta for the control of the update time interval, the following remarks/comments are attained after the evaluation of the simulation results for tracking of Target Trajectory-1.

Remarks and Comments:

- After the maneuver begins, the measurement residuals (position residuals) and the estimation error increase, causing the update time interval to decrease.
- When the decrease of the update time interval occurs in the maneuvering segment, the tracking accuracy improves and the MSE in *x* and *y* coordinates decrease.
- The update time interval is reduced to a minimum of 2.00 sec during the trajectory.

- When the decrease on the update time interval occurs in the maneuvering segment, the decrease on the measurement residuals occurs which causes the update time interval to increase again.

- On termination of the maneuver, the CV model of the filter becomes valid again, and the measurement residual and estimation error decreases. By the considerable decrease on the measurement residual, the maximum update time interval (4.00 sec) is attained. Then, the restored maximum update time interval is kept up to the completion of the trajectory.

5.3.2 Simulation Group 2 – Investigation of Update Time Interval Determination Method-2

Update Time Interval Determination Method-2 proposed in [5] (which decides the value of the next update time interval after computing the next update time interval by the formula given in Equation (3-20) where the value of the next time interval is directly proportional with the previous update time interval and inversely proportional with the square root of the measurement residual) is used to control the update time interval in target tracking by using Alpha-Beta filter with CV Model in 2D Cartesian coordinates throughout the simulations in this part. The aim is to observe the effectiveness of Update Time Interval Determination Method-2 for a target trajectory which includes maneuvering segment consisting of 90° maneuver.

The parameter values of this simulation are same as the parameter values used in Simulation Group 1 given in Table 5-1 except the Update Time Interval Determination Method.

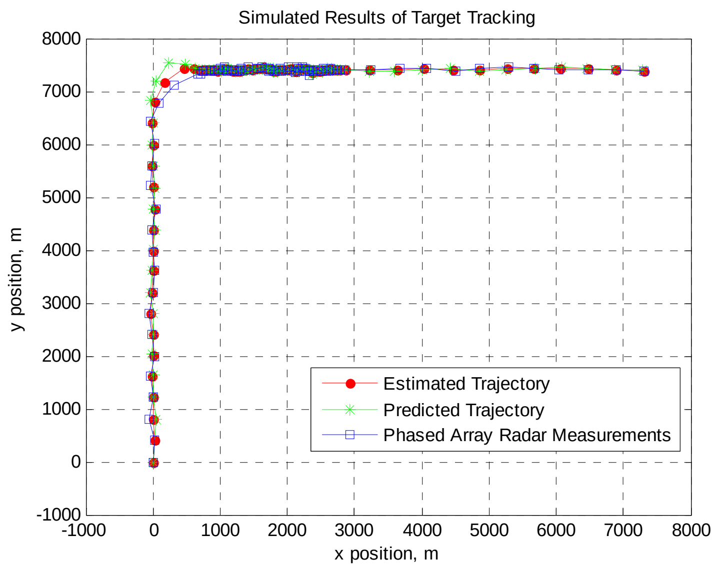

Figure 5-7: Predicted, Estimated Output of the Filter and Phased Array Radar Measurements in 2D, for one MC run

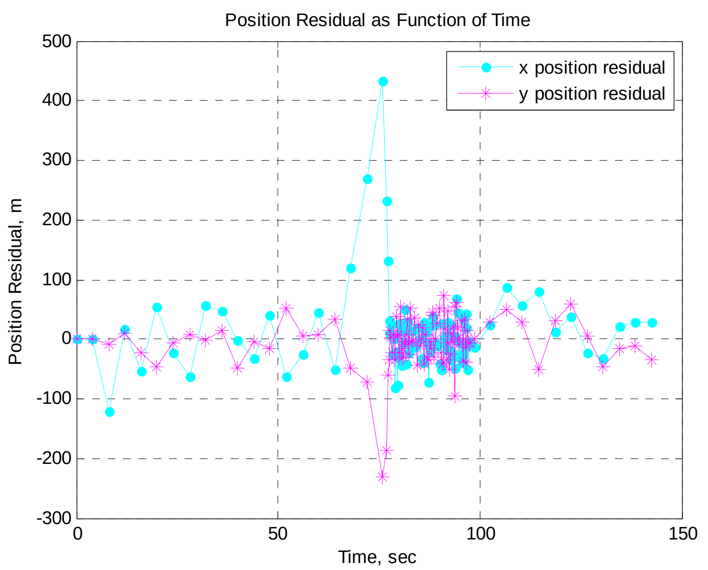

Figure 5-8: Position Residual in *x* and *y* coordinates versus Time, for one MC run

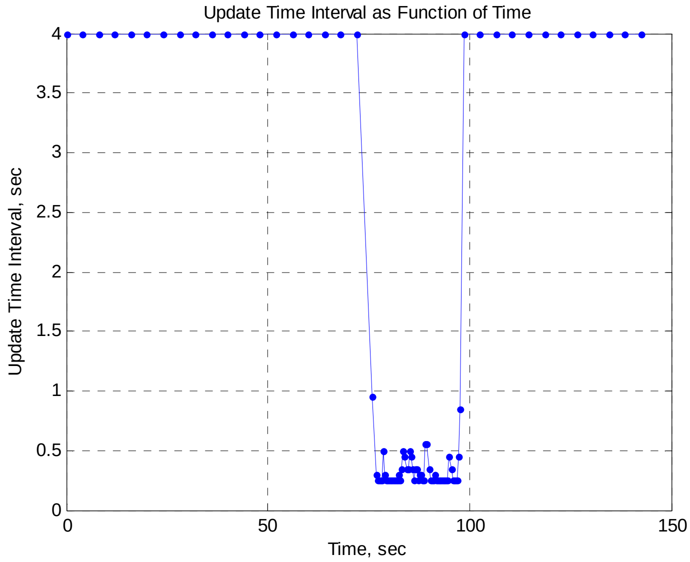

Figure 5-9: Track Update Time Interval versus Time, for one MC run

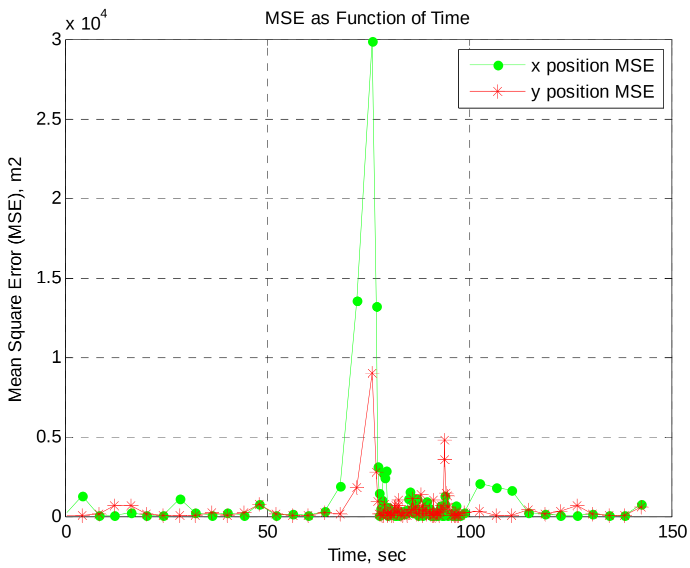

Figure 5-10: MSE in *x* and *y* coordinates versus Time, for one MC run

By implementing the Update Time Interval Determination Method-2 on Alpha-Beta filter for the control of the update time interval, the following remarks/comments are attained after the evaluation of the simulation results for tracking of Target Trajectory-1.

Remarks and Comments:

- After the maneuver begins, the measurement residuals (position residuals) and the estimation error increase, causing the update time interval to decrease.
- The reduction of the initial update time interval firstly occurs about a relatively long time after the maneuver begins which means that there is a considerable delay in the algorithm to decrease the update time interval from its maximum value.
- The update time interval is reduced to a minimum of 0.25 sec during the trajectory.
- When the decrease of the update time interval occurs in the trajectory, the tracking accuracy improves and the MSE in x and y coordinates decrease.
- When the decrease on the update time interval occurs in the trajectory, the decrease on the measurement residuals occurs which then causes the update time interval to increase again.
- After the maneuver is completed, the CV model of the filter becomes valid again, and the measurement residual and estimation error decreases. By the considerable decrease on the measurement residual, the maximum update time interval (4.00 sec) is attained. Then, the restored maximum update time interval is kept up to the completion of the trajectory.

5.3.3 Simulation Group 3 - Comparison of Update Time Interval Determination Method-1 and Method-2

The simulations results of Update Time Interval Determination Method-2 [5] in which time intervals may take continuous values is compared with the simulation results of Update Time Interval Determination Method-1 [2] in which the update time interval is chosen from a discrete set of values.

For this purpose, 100 MC simulations are performed for tracking of Target Trajectory-1 by using Update Time Interval Determination Method-1 (Discrete update time interval case) and Update Time Interval Determination Method-2 (Continuous update time interval case) on Alpha-Beta filter with CV Model in 2D Cartesian coordinates.

Results of MC Simulations:

The results of 100 MC simulations in terms of average update time interval, average number of updates and overall RMSE in position are given in the following table.

Table 5-2: Comparison of 100 MC Simulations Results for Update Time Interval Determination Methods

	Update Time Interval Determination Method	
	Method-1	**Method-2**
Average Update Time Interval (sec)	2.88	1.33
Average Number of Updates	50	108
Overall RMSE in Position (m)	47.27	29.63

By implementing the Update Time Interval Determination Method-1 and Update Time Interval Determination Method-2 on Alpha-Beta filter for tracking the Target Trajectory-1, the following remarks/comments are attained after the evaluation of the simulation results of both methods.

Remarks and Comments:

- There is a considerable improvement in the tracking performance when the method given in [5] named as Update Time Interval Determination Method-2 is adopted for choosing next update time interval, because overall estimation error in position is lower than the case where Update Time Interval Determination Method-1 is used.

- A great reduction in the average update time interval is obtained with the continuous update time interval case where Update Time Interval Determination Method-2 is used when compared with the discrete update time interval case where Update Time Interval Determination Method-1 is used.

- Due to the reduction on the average update time interval, the number of updates is higher in the continuous update time interval case when compared with the discrete case.

- The reason for the reduction in the average update time interval is that the update time intervals are not forced to any fixed value in the continuous update time interval case as opposed to the discrete update time interval case.

Note that, Update Time Interval Determination Method-1 and Update Time Interval Determination Method-2 will be investigated on Kalman filter and compared in detail in Section 5.4.

5.3.4 Simulation Group 4 - Comparison of Adaptive Time Interval and Fixed Time Interval Cases

The simulation results of adaptive time interval algorithm in which update time interval is controlled is compared with the simulation results of fixed time interval algorithm in which the update time interval is held constant by using Alpha-Beta filter. Update Time Interval Determination Method-1 is used as adaptive update time interval determination algorithm to simulate the case where the update time interval is controlled.

The important parameters of the simulation are given in the following table.

Table 5-3: Parameters of Simulation

Actual Measurement Noise	Range (m): Gaussian, $\sigma = 30$ Azimuth (rad): Gaussian, $\sigma = 0.003$
Radar Location	$x_r = 7000$ m, $y_r = 0$ m
Target Initial Location	$x = 0$ m, $y = 0$ m

In order to compare adaptive update time interval with fixed update time interval cases of 4.00 sec and 2.00 sec, "Position Residual as function of time" and "MSE as function of time" graphs are found at each cases for one MC run.

Case-1 of Simulation Group 4: Fixed Update Time Interval 4.00 sec case:

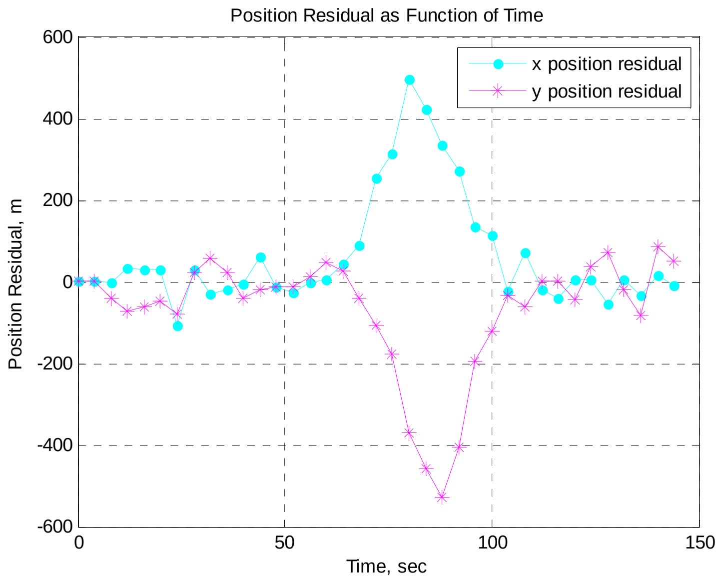

Figure 5-11: Position Residual in *x* and *y* coordinates versus Time, for one MC run

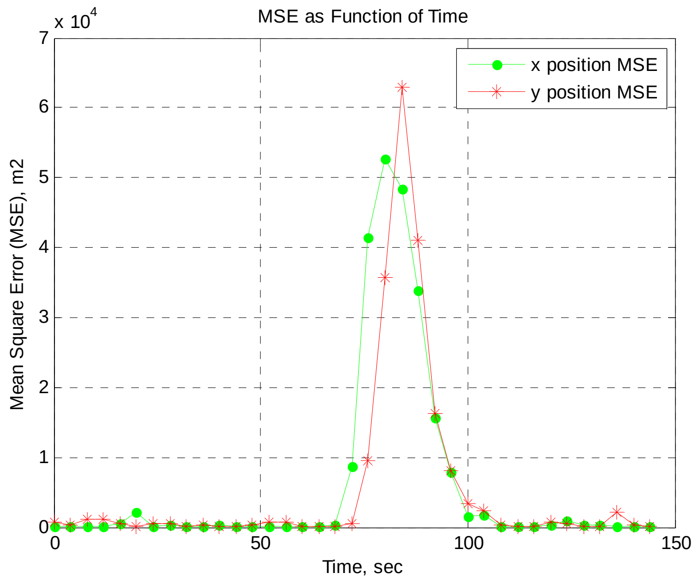

Figure 5-12: MSE in *x* and *y* coordinates versus Time, for one MC run

<u>Case-2 of Simulation Group 4: Fixed Update Time Interval 2.00 sec case:</u>

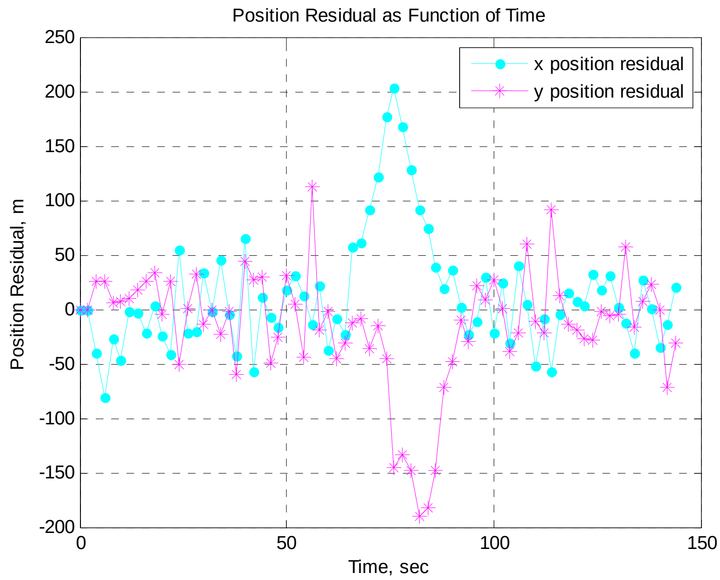

Figure 5-13: Position Residual in *x* and *y* coordinates versus Time, for one MC run

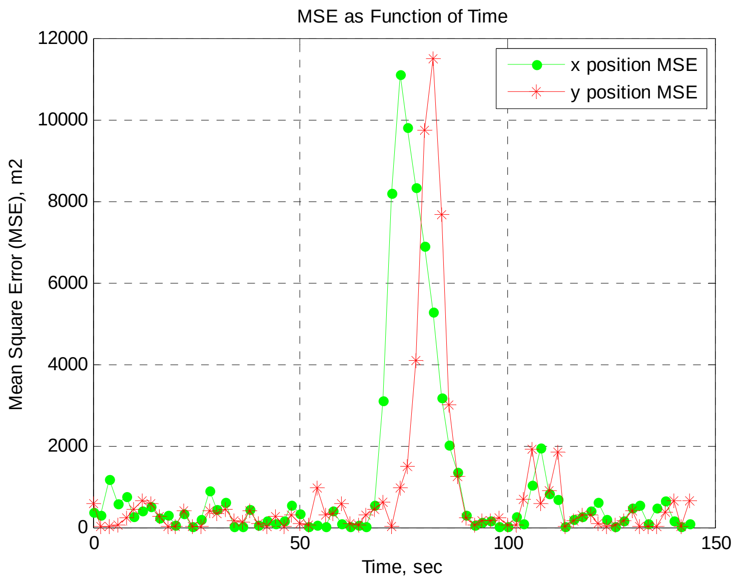

Figure 5-14: MSE in *x* and *y* coordinates versus Time, for one MC run

Case-3 of Simulation Group 4: Adaptive Update Time Interval case:

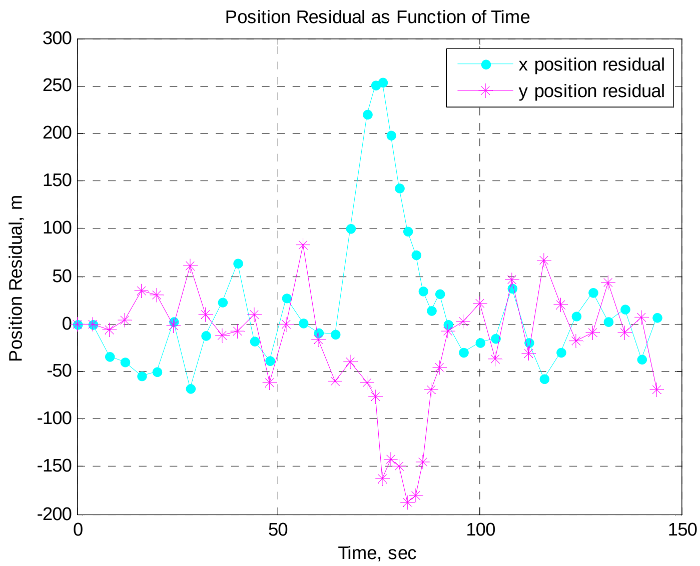

Figure 5-15: Position Residual in x and y coordinates versus Time, for one MC run

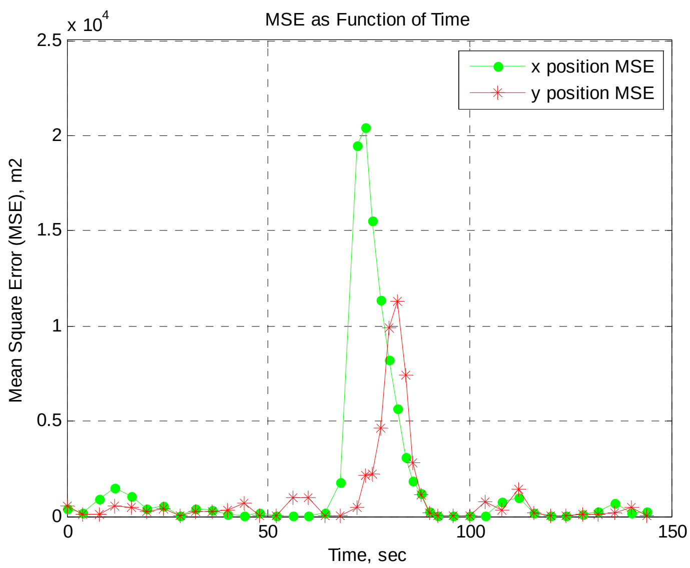

Figure 5-16: MSE in x and y coordinates versus Time, for one MC run

By implementing the Alpha-Beta filter for the cases update time interval is controlled and update time interval is fixed, the following remarks/comments are attained after the evaluation of the simulation results which are "Position Residual as function of time" and "MSE as function of time" graphs for adaptive update time interval, fixed update time interval cases of 4.00 sec and 2.00 sec for one MC run.

Remarks and Comments:

- During the constant-velocity segment of the trajectory, the measurement residuals increase as the update time interval becomes larger during the trajectory.

- During the maneuvering segment of the target trajectory, the measurement residual becomes larger as the update time interval increases. Moreover, the MSE gets higher values in the fixed time interval of 4.00 sec case than the MSE values obtained in the fixed time interval of 2.00 sec case. This shows that when the measurement residuals get higher values along the maneuvering segment of the target trajectory, the MSE also gets higher values in the case of longer update time interval compared with the shorter update time interval case.

- When MSE versus time graph is investigated for adaptive update time interval case, it is observed that the magnitude of the tracking error reflects the change in the update time interval as opposed to fixed update time interval cases. Because during maneuvering segment, in adaptive update time interval case the increase in MSE causes the update time intervals to decrease in order to keep the tracking accuracy relatively high.

Results of MC Simulations:

For the cases where the update time interval is controlled and is held constant in tracking with Alpha-Beta filter, 100 MC simulations are carried out. By comparing the number of updates and the estimation errors, the saving of phased array radar resources and the improvement on the tracking accuracy are investigated.

In order to compare adaptive and fixed time interval cases with regard to average number of updates, average update time interval and overall RMSE in position, 100 MC simulation results are summarized in Table 5-4 for adaptive time interval case and fixed time interval cases with different constant update time intervals. Note that, the overall RMSE in position is found by summing the RMSE values found in the whole trajectory and then dividing by the number of update data points. Tracking accuracy is represented by the overall RMSE in position.

If the energy needed for one update is considered equal, the use rate of the radar resource is the number of updates per unit time. In the present simulation, the average number of updates is considered as the amount of use of the radar resources [18].

Table 5-4: MC Simulations Results of Overall Trajectory for Adaptive and Fixed Update Time Interval Cases

	100 MC Simulation Results of Overall Trajectory (0-144 sec)		
Update time interval	**Average Number of Updates**	**Average Update Time Interval (sec)**	**Overall RMSE in Position (m)**
Adaptive	43	3.35	47.84
2.00 sec fixed	73	2.00	42.03
2.50 sec fixed	58	2.50	56.49
3.00 sec fixed	49	3.00	71.58
3.50 sec fixed	42	3.50	91.01
4.00 sec fixed	37	4.00	110.13

In order to observe the average RMSE of position versus time, the average RMSE of position is obtained by dividing the sum of the MSE in x and y coordinates computed at each track update data point by the number of updates carried out at the corresponding update times among overall MC runs and taking the square roots.

The results of 100 MC simulations in terms of average RMSE of position versus time is given in the following graphs for adaptive time interval, fixed time interval of 4.00 sec and fixed time interval of 2.00 sec cases.

<u>Adaptive Update Time Interval Case:</u>

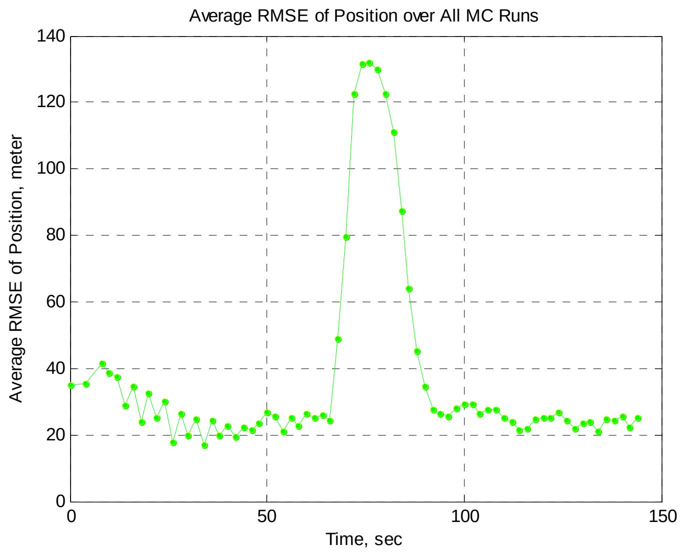

Figure 5-17: Average of the RMSE versus Time, over 100 MC runs for Adaptive Update Time Interval

Fixed Update Time Interval 4.00 sec Case:

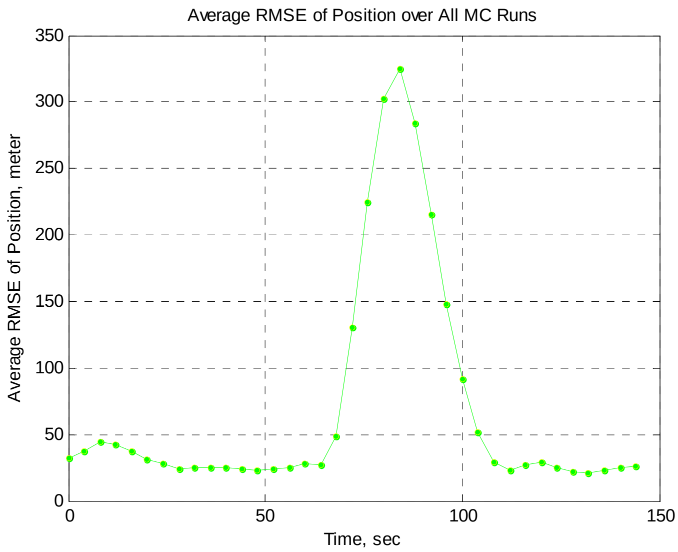

Figure 5-18: Average of the RMSE versus Time, over 100 MC runs for the Fixed Update Time Interval of 4.00 sec

Fixed Update Time Interval 2.00 sec Case:

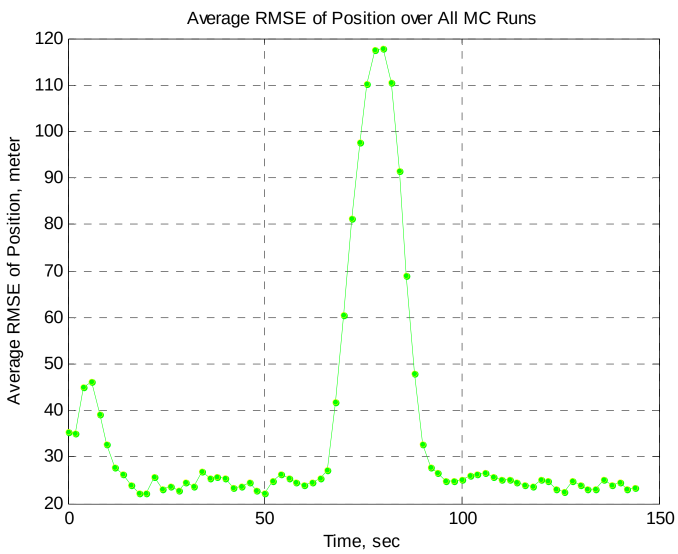

Figure 5-19: Average of the RMSE versus Time, over 100 MC runs for the Fixed Update Time Interval of 2.00 sec

The following remarks/comments are attained after the comparison and evaluation of 100 MC simulation results with regard to average number of updates, average update time interval, overall RMSE in position and the average RMSE of position versus time graphs for adaptive and fixed update time interval cases at different constant update time intervals.

Remarks and Comments:

- When the average numbers of updates are compared, the radar resources used in the case of the adaptive update time interval is almost the same as that for the fixed update time interval at 3.50 sec case.

- When the tracking accuracy with the adaptive update time interval is compared with the case of fixed interval at 3.50 sec, there is an improvement of about 48% in the overall trajectory with adaptive update time interval case.

- On the other hand, in order to obtain similar tracking accuracy with a fixed update time interval as in the adaptive update time interval case in the overall trajectory, the case of fixed interval at 2.00 sec seems to be necessary. Moreover, the number of updates is reduced by about 41% if adaptive update time interval is carried out instead of fixed interval case at 2.00 sec.

5.4 SIMULATION RESULTS WITH KALMAN FILTER

In this part of the simulations, the approach developed in literature for adaptive update rate algorithms on Alpha-Beta filter is implemented on Kalman filter. Several scenarios are created for the target trajectory. Then, the behavior of the adaptive update rate algorithms developed in literature for Alpha-Beta filter are investigated by using a single Kalman filter for tracking of a maneuvering target which can make 90° turn in its trajectory.

In order to analyze the application of adaptive time interval algorithms to target tracking with Kalman filter and to show clearly the performance of adaptive time interval algorithm, “Estimated Trajectory”, “Position Residuals as function of time”, “Track Update Time Interval as function of time” and “MSE as function of time” graphs are demonstrated for one MC run.

Besides, in order to evaluate adaptive update time interval case in terms of RMSE, average number of updates and average update time interval for several cases, 100 MC simulations are realized by changing simulation parameters such as Update Time Interval Determination Method, target trajectory, the duration of maneuver and measurement noise in each simulation.

5.4.1 Simulation Group 1 – Investigation of Update Time Interval Determination Method-1

Update Time Interval Determination Method-1 of [2] (which compares measurement residual with the standard deviation of measurement noise to decide the next update time interval) is used to control the update time interval by using KF with CV Model in 2D Cartesian coordinates throughout the simulations in this part.

5.4.1.1 Part A – Application to the Target Trajectory with a Single 90° Maneuver

To observe the effectiveness of Update Time Interval Determination Method-1 for a target trajectory which includes a maneuvering segment consisting of a single 90° maneuver, the scenario defined for Target Trajectory-1 is tracked by using KF with CV Model in 2D Cartesian coordinates.

The important parameters of the simulation are given in the following table.

Table 5-5: Parameters of Simulation

Update Time Interval (sec)	Adaptive
Update Time Interval Determination Method	Update Time Interval Determination Method-1
Initial Update Time Interval (sec)	4.00 sec
Actual Measurement Noise	Range (m): Gaussian, σ = 30 Azimuth (rad): Gaussian, σ =0.003
Measurement Noise Model in the Filter	Same as the covariance of actual measurement noise
Process Noise Model (Q matrix) in the Filter	$Q = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix}$
Radar Location	$x_r = 7000$ m, $y_r = 0$ m
Target Initial Location	$x = 0$ m, $y = 0$ m

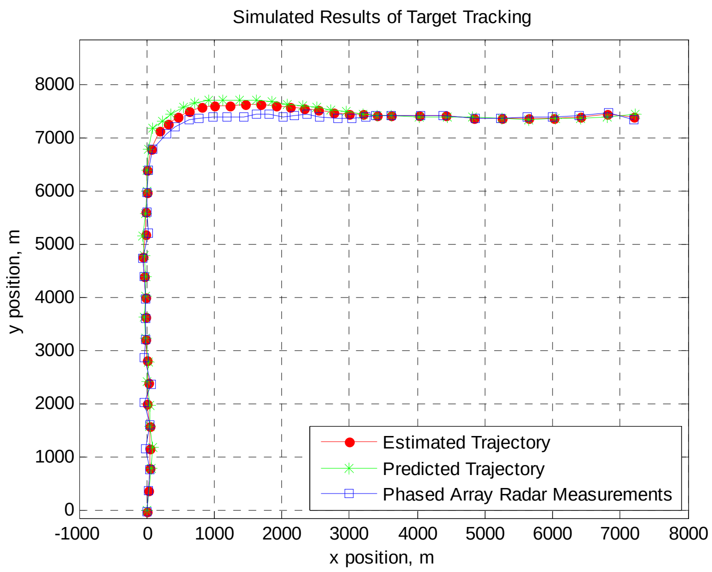

Figure 5-20: Predicted, Estimated Output of the Filter and Phased Array Radar Measurements in 2D, for one MC run

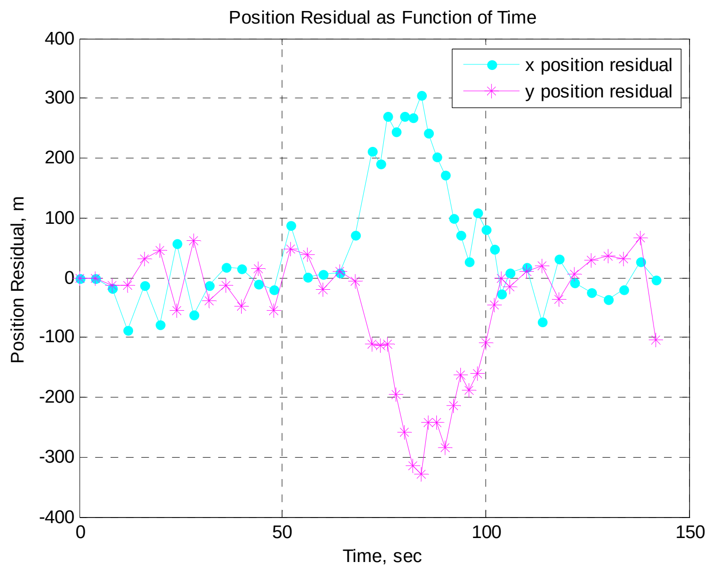

Figure 5-21: Position Residual in *x* and *y* coordinates versus Time, for one MC run

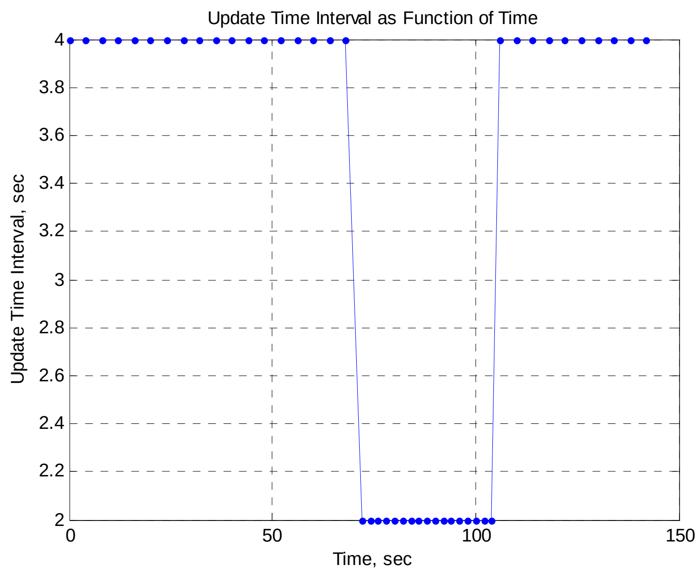

Figure 5-22: Track Update Time Interval versus Time, for one MC run

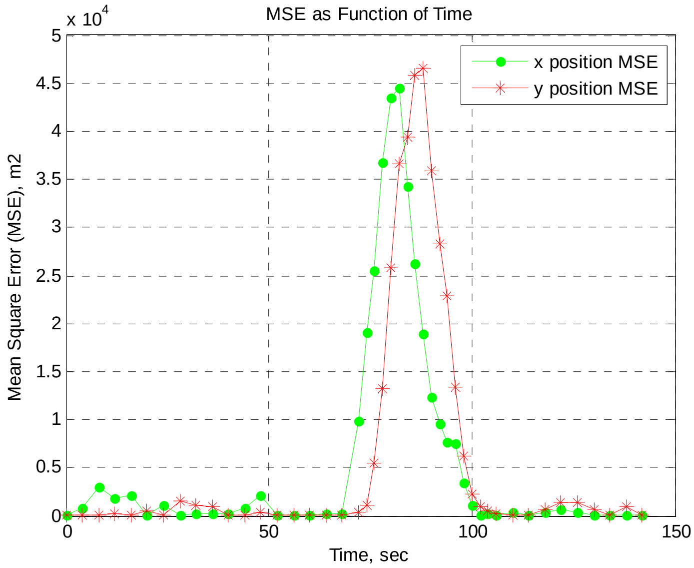

Figure 5-23: MSE in *x* and *y* coordinates versus Time, for one MC run

Results of MC Simulations:

The results of 100 MC simulations for the overall trajectory in terms of average update time interval and average number of updates are given in the following table.

Table 5-6: MC Simulations Results for Overall Trajectory

	Adaptive KF Result over 100 MC Runs
Average Update Time Interval (sec)	2.88
Average Number of Updates	50

The results of 100 MC simulations in terms of average update time interval and average number of updates versus time step are given in the following graphs. Each time step in the demonstrated graphs represents the stage of the trajectory with 8 sec of duration. For example, let the index of the stage be equal to 3 in the graph, then

this index represents the time duration between 16 sec - 24 sec time instants of the trajectory.

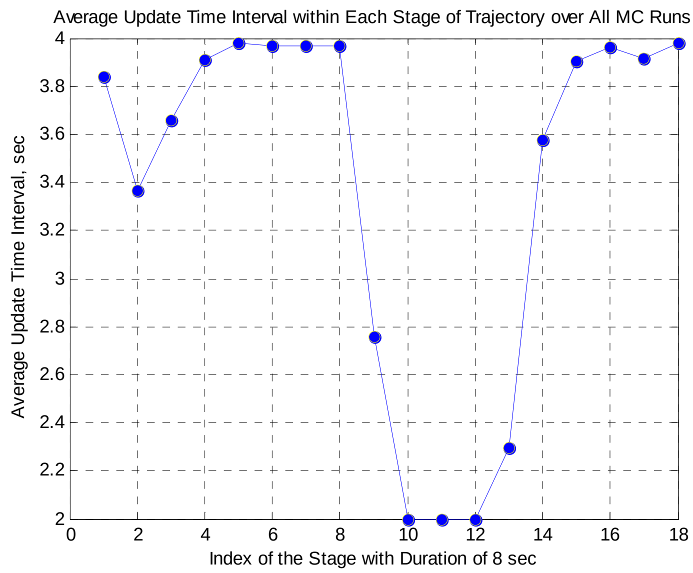

Figure 5-24: Average Update Time Interval versus Index of the Trajectory Stage, over 100 MC runs

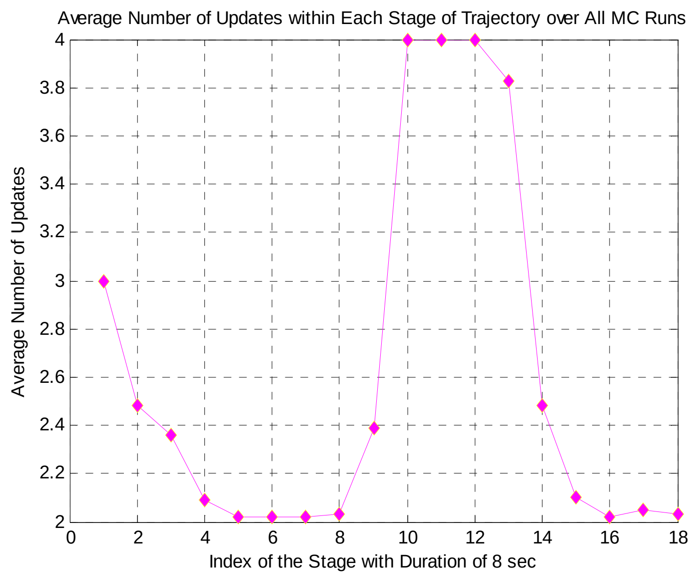

Figure 5-25: Average Number of Updates versus Index of the Trajectory Stage, over 100 MC runs

Moreover, the average RMSE of position is obtained by dividing the sum of the MSE in x and y coordinates computed at each track update data point by the number of updates carried out at the corresponding update times among overall MC runs and taking the square roots to get RMSE. It is believed that RMSE versus time graph obtained in this way will give an indication of the time dependence of RMSE for adaptive update rate case.

The results of 100 MC simulations in terms of RMSE versus time is given in the below graph.

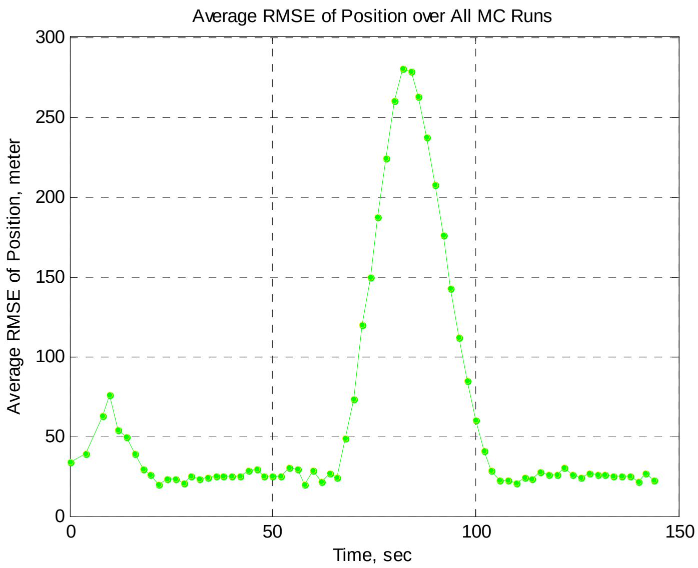

Figure 5-26: Average of the RMSE versus Time, over 100 MC runs

By implementing the Update Time Interval Determination Method-1 on KF for the control of the update time interval, the following remarks/comments are attained after the evaluation of the simulation results for tracking of Target Trajectory-1.

Remarks and Comments:

- After the maneuver begins, the measurement residuals (position residuals) and the estimation error increase, causing the update time interval to decrease.
- The reduction of the initial update time interval firstly occurs about 8 sec after the maneuver begins.
- The update time interval is reduced to a minimum of 2.00 sec during the trajectory.
- When the decrease of the update time interval occurs in the maneuvering segment, the tracking accuracy improves and the MSE in x and y coordinates decrease.
- When the decrease of the update time interval occurs in the maneuvering segment, the decrease on the measurement residuals occurs which causes the update time interval to increase again.
- While the target is maneuvering, the measurement residual has relatively high value which prevents the update time interval from increasing to the maximum value of 4.00 sec.
- On termination of the maneuver, the CV model of the filter becomes valid again, and the measurement residual and estimation error decreases.
- After the maneuver is completed, by the considerable decrease on the measurement residual, the maximum update time interval (4.00 sec) is attained within 24-32 sec. Then, the restored maximum update time interval is kept up to the completion of the trajectory.

5.4.1.2 Part B - Application to the Target Trajectory with 2 (two) 90° Maneuvers

To observe the effectiveness of Update Time Interval Determination Method-1 for a target trajectory which includes two maneuvering segments consisting of 90° maneuvers, the scenario defined for Target Trajectory-2 is tracked by using KF with CV Model in 2D Cartesian coordinates.

The parameter values of this simulation are same as the parameter values used in Part A of Simulation Group 1 given in Table 5-1.

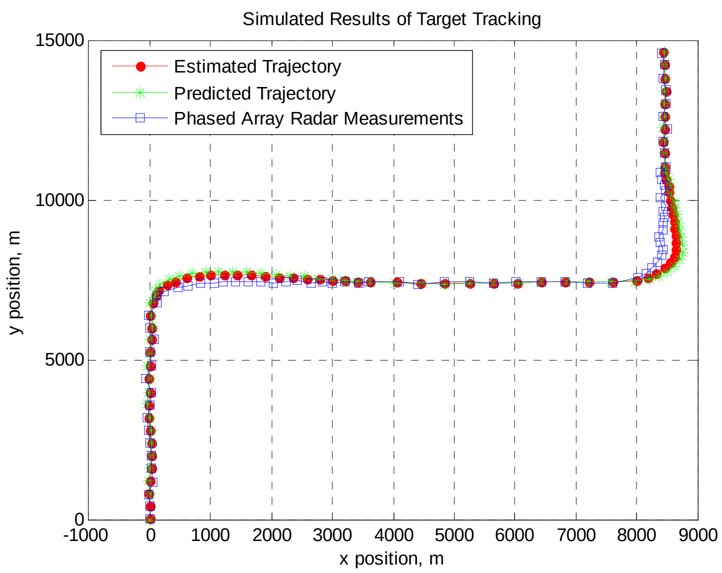

Figure 5-27: Predicted, Estimated Output of the Filter and Phased Array Radar Measurements in 2D, for one MC run

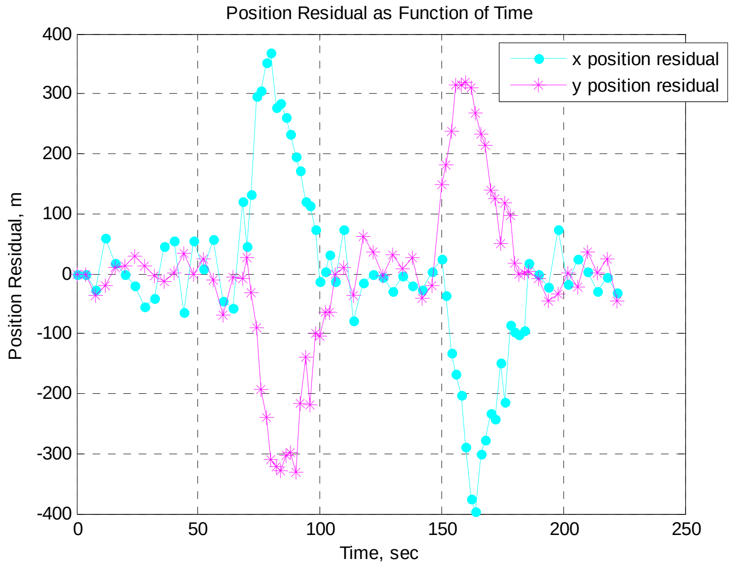

Figure 5-28: Position Residual in *x* and *y* coordinates versus Time, for one MC run

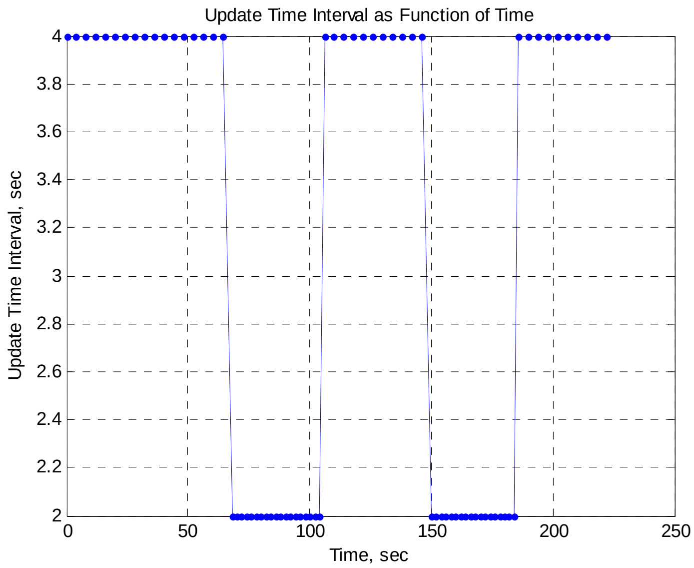

Figure 5-29: Track Update Time Interval versus Time, for one MC run

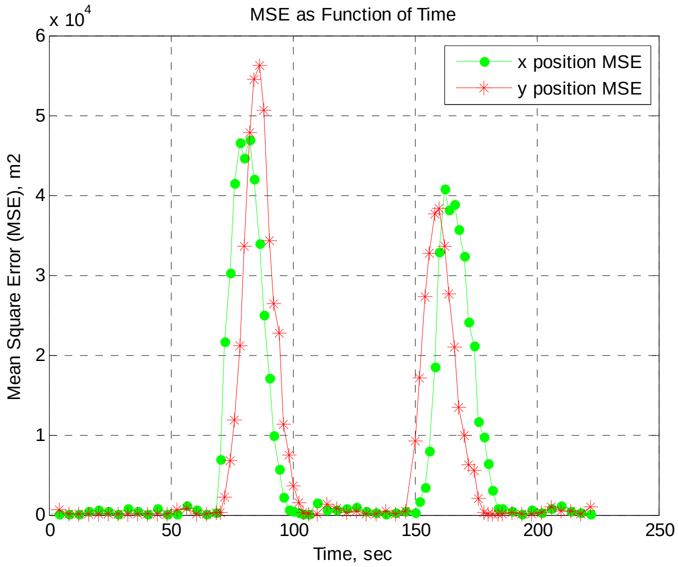

Figure 5-30: MSE in *x* and *y* coordinates versus Time, for one MC run

Results of MC Simulations:

The results of 100 MC simulations for the overall trajectory in terms of average update time interval and average number of updates are given in the following table.

Table 5-7: MC Simulations Results for Overall Trajectory

	Adaptive KF Result over 100 MC Runs
Average Update Time Interval (sec)	2.77
Average Number of Updates	81

The results of 100 MC simulations in terms of average update time interval and average number of updates versus time step are given in the following graphs. Each time step in the demonstrated graphs represents the stage of the trajectory with 8 sec of duration. For example, let the index of the stage be equal to 3 in the graph, then

this index represents the time duration between 16 sec - 24 sec time instants of the trajectory.

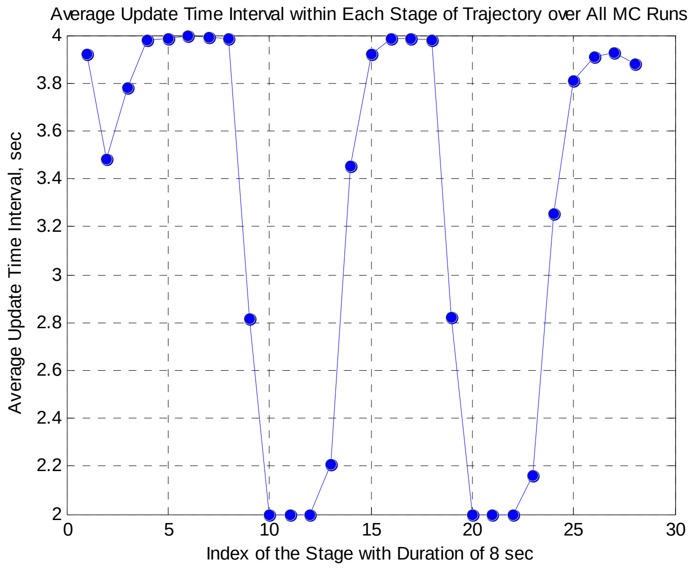

Figure 5-31: Average Update Time Interval versus Index of the Trajectory Stage, over 100 MC runs

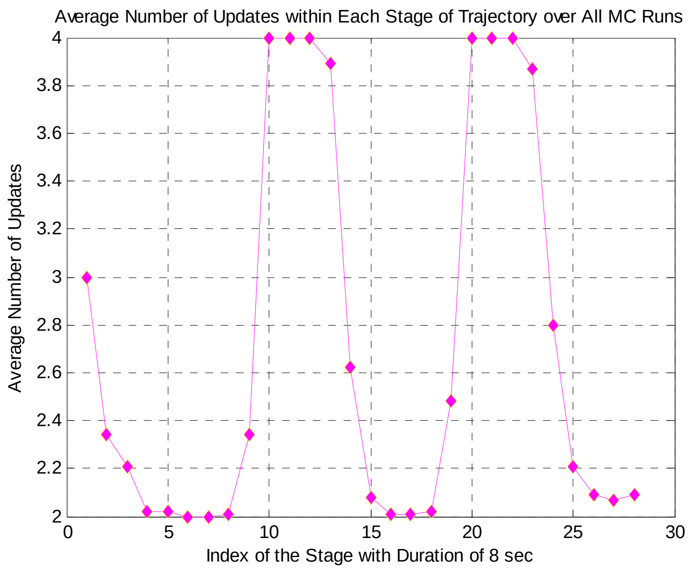

Figure 5-32: Average Number of Updates versus Index of the Trajectory Stage, over 100 MC runs

Moreover, the average RMSE of position is obtained by dividing the sum of the MSE in x and y coordinates computed at each track update data point by the number of updates carried out at the corresponding update times among overall MC runs and taking the square roots. It is believed that RMSE versus time graph obtained in this way will give an indication of the time dependence of RMSE for adaptive update rate case.

The results of 100 MC simulations in terms of RMSE versus time is given in the below graph.

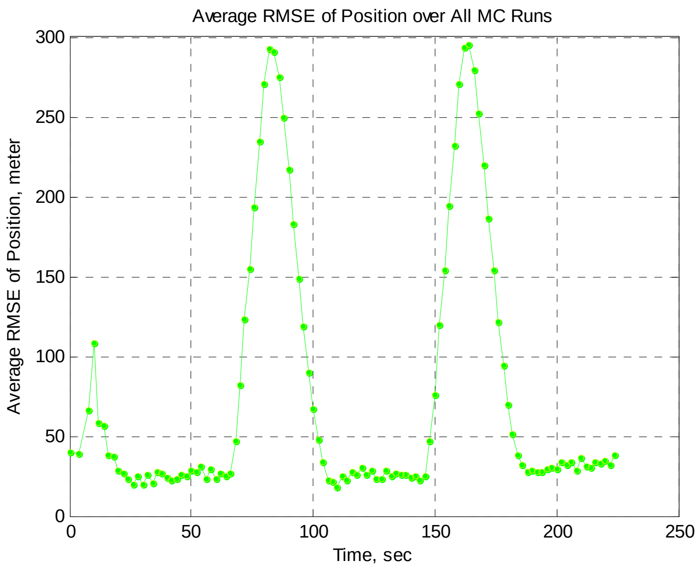

Figure 5-33: Average of the RMSE versus Time, over 100 MC runs

By implementing the Update Time Interval Determination Method-1 on KF for the control of the update time interval, the following remarks/comments are attained after the evaluation of the simulation results for tracking of Target Trajectory-2.

Remarks and Comments:

- In both maneuvering segments, after the maneuver begins, the measurement residuals and estimation error increase, causing the update time interval to decrease.

- In both maneuvering segments, reduction of the initial update time interval firstly occurs about 8 sec after the maneuver begins.

- In both maneuvering segments, the update time interval is reduced to a minimum of 2.00 sec during the trajectory.

- When the decrease of the update time interval occurs in each maneuvering segment, the tracking accuracy improves and the MSE in x and y coordinates decrease.

- When the decrease of the update time interval occurs in each maneuvering segment, the decrease on the measurement residuals occurs which causes the update time interval to increase again.

- In both maneuvering segments, while the target is maneuvering, measurement residuals have relatively high values which prevent the update time interval from increasing to the maximum value of 4.00 sec.

- In both maneuvering segments, on termination of the maneuver, the CV model of the filter becomes valid again, and the measurement residual and estimation error decrease.

- In both maneuvering segments, after the maneuver is completed, by the considerable decrease on the measurement residual, the maximum update time interval (4.00 sec) is attained within 32-40 sec.

General Conclusions for Part A and Part B of Simulation Group 1:

- In the maneuvering segments of each trajectory, good tracking performance can only be achieved by a drastic reduction in the update time interval. This is due to the fact that the implemented KF which is based on CV model is strictly valid when applied to the tracking of a non-maneuvering target which has constant velocity.

- During the non-maneuvering segments of the trajectory, the estimation errors are fairly low when compared with the errors obtained in the maneuvering segments, because CV model of the filter is valid in non-maneuvering segments.

- During constant-velocity motion of the target in non-maneuvering segments, the update time interval is made larger so that the phased array radar resource is conserved.

- During the maneuvering segments of the target, the update time interval is made smaller so that the degradation of tracking is prevented.

- The update time interval must be reduced rapidly in the maneuvering segments by an appropriate amount in order to keep the tracking of the target.

- The longest predefined update time interval is automatically restored after the maneuver so that more time is made available to the phased array radar for other radar tasks.

5.4.1.3 Part C - Application to the Target Trajectory with Constant Acceleration Motion

The aim of this simulation is to demonstrate the behavior of the adaptive time interval algorithm for the case of tracking of a target with constant acceleration motion along a straight-line by using the CV model KF. The scenario defined below for Target Trajectory-3 is tracked in 2D Cartesian coordinates.

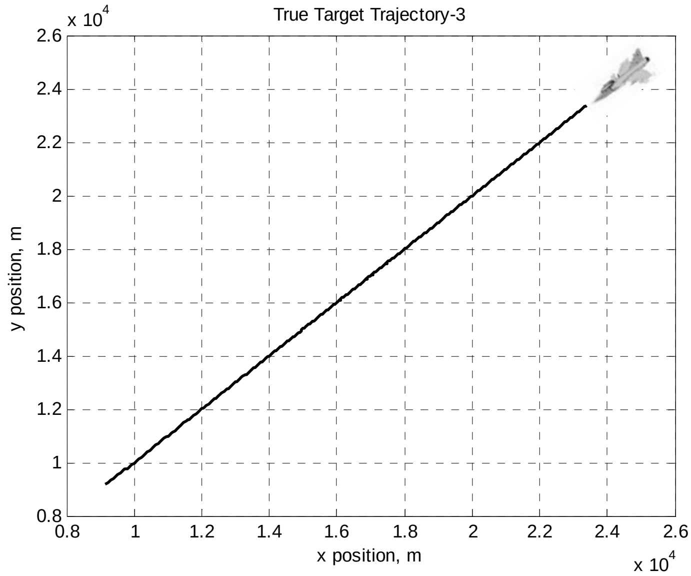

Figure 5-34: True Target Trajectory-3 in 2D

<u>Details of True Target Trajectory-3:</u>

Trajectory Duration: 144 sec

Target Initial Speed: 100 m/sec

Target Final Speed: 300 m/sec

Trajectory is composed of 2 segments:

Segment 1: A Non-maneuvering segment, constant velocity motion along straight line, lasting 64 sec, from 0 to 64 sec

Segment 2: A Constant acceleration segment, Motion with constant acceleration along a straight-line, lasting 80 sec, from 64 to 144 sec

The parameter values of this simulation are same as the parameter values used in Part A of Simulation Group 1, except the radar location and the target initial location.

The important parameters of the simulation are given in the following table.

Table 5-8: Parameters of Simulation

Update Time Interval (sec)	Adaptive
Update Time Interval Determination Method	Update Time Interval Determination Method-1
Initial Update Time Interval (sec)	4.00 sec
Actual Measurement Noise	Range (m): Gaussian, $\sigma = 30$ Azimuth (rad): Gaussian, $\sigma = 0.003$
Measurement Noise Model in the Filter	Same as the covariances of actual measurement noise
Process Noise Model (Q matrix) in the Filter	$Q = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix}$
Radar Location	$x_r = 0$ m, $y_r = 0$ m
Target Initial Location	x=25000 m, y=25000 m

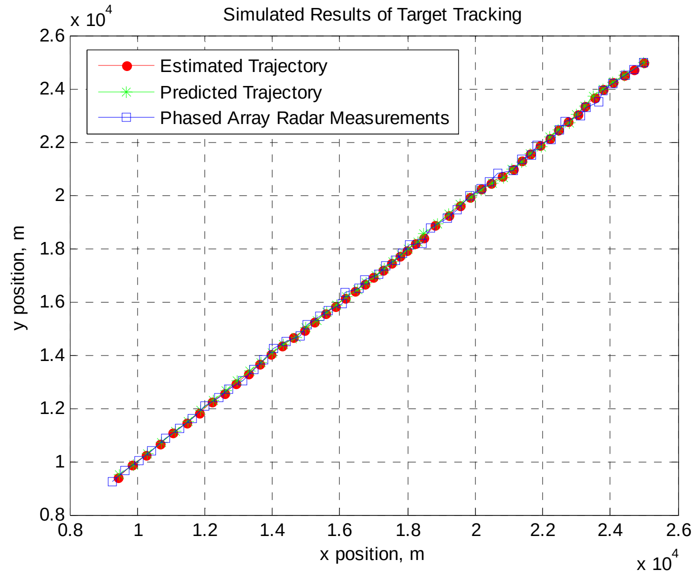

Figure 5-35: Predicted, Estimated Output of the Filter and Phased Array Radar Measurements in 2D, for one MC run

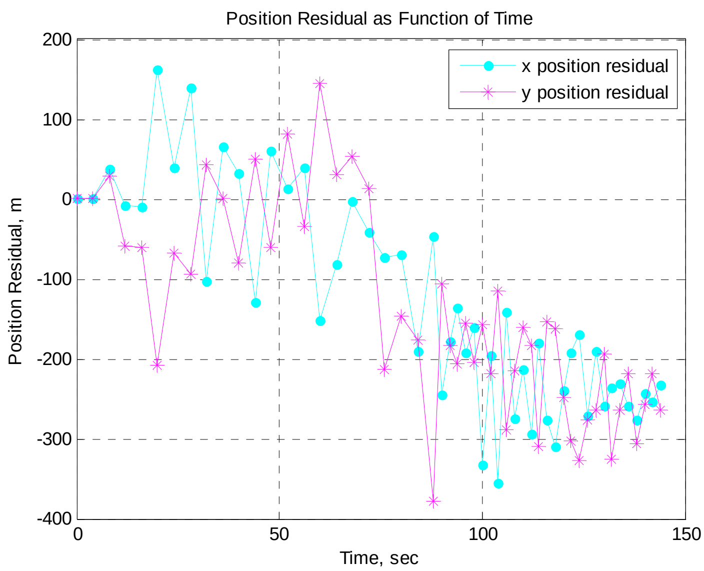

Figure 5-36: Position Residual in *x* and *y* coordinates versus Time, for one MC run

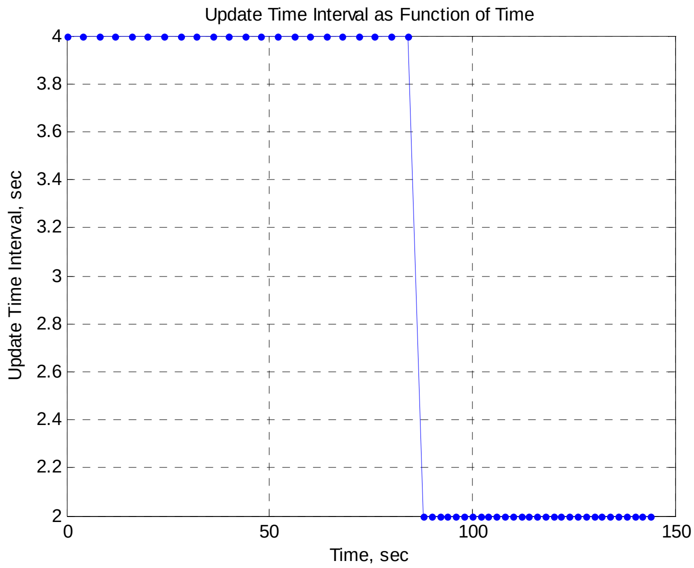

Figure 5-37: Track Update Time Interval versus Time, for one MC run

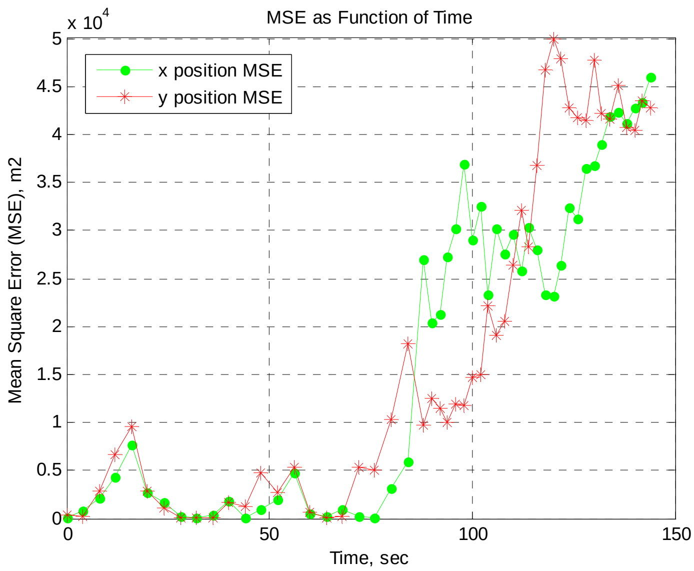

Figure 5-38: MSE in *x* and *y* coordinates versus Time, for one MC run

By implementing the Update Time Interval Determination Method-1 on CV model KF for tracking of the target with constant acceleration, the following remarks/comments are attained after the evaluation of the simulation results.

Remarks and Comments:

- After the constant acceleration motion begins, the measurement residuals and estimation error increase, causing the update time interval to decrease.
- The reduction of the initial update time interval firstly occurs about 10-20 sec after the constant acceleration motion begins.
- The update time interval is reduced to a minimum of 2.00 sec during the trajectory.
- When the decrease of the update time interval occurs in the constant acceleration segment, the tracking accuracy does not improve and the MSE in x and y coordinates do not decrease.
- In the constant acceleration segment, the measurement residuals in x and y coordinates are fairly high and even the decrease in the update time interval can not cause the residuals to decrease again.
- When the decrease of the update time interval occurs in the constant acceleration segment, the measurement residuals have relatively high values which prevent the update time interval to increase again and the update time interval remains at 2.00 sec during the constant acceleration segment.
- While the target is moving with constant velocity in non-maneuvering segment, the measurement residual has relatively low value which keeps the update time interval at the maximum value of 4.00 sec.
- By termination of the constant velocity segment, the CV model of the filter does not match with the target's actual motion again in rest of the trajectory,

measurement residuals and the MSE in x and y coordinates have relatively high values and do not decrease.

- Since the CV model filter predicts the target's position as if the target goes with a constant velocity and the target actually goes with constant acceleration during the constant acceleration segment, the measurement residuals in x and y coordinates have always same sign (which are found to be negative for the defined scenario) in constant acceleration segment. In other words, the predicted position is smaller due to the CV model of the filter than the actual measured position and since the measurement residual represents the difference between the actual measured position and the predicted position, the position residuals in both x and y coordinates remain at negative values during the constant acceleration segment.

General Conclusions for Part C of Simulation Group 1:

- In the constant acceleration segment of the trajectory, good tracking performance can not be achieved although the reduction in the update time interval occurs. This is due to the fact that the implemented filter, which is based on CV model, is strictly valid when applied to the tracking of a non-maneuvering target which has constant velocity.

- During the non-maneuvering segment of the trajectory where the target moves with constant velocity, the estimation errors are fairly low when compared with the errors obtained in the constant acceleration segment, because CV model of the filter is valid in the non-maneuvering segment.

- During constant-velocity segment of the target trajectory, the update time interval is kept at its longest predefined value so that the phased array radar resource is conserved.

- During the constant acceleration segment of the target, although the update time interval is made smaller, the measurement residuals remain at relatively high values. Consequently MSE in x and y coordinates have relatively high values and the degradation of tracking can not be prevented.

5.4.1.4 Part D - Application to the Target Trajectory with Constant Velocity Motion

The aim of this simulation is to observe whether or not the decrease in the update time interval occurs only when the tracking filter detects a maneuver in the target motion. For this purpose, a scenario for a constant-velocity target is created and the tracking is performed by using Update Time Interval Determination Method-1 on KF with CV Model in 2D Cartesian coordinates for the True Target Trajectory-4 which is given in Figure 5-39.

The parameter values of this simulation are same as the parameter values used in Part C of Simulation Group 1 given in Table 5-8.

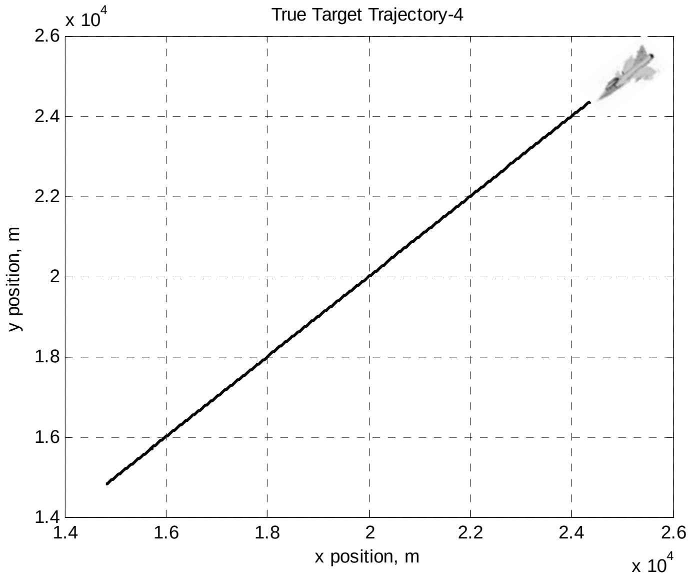

Figure 5-39: True Target Trajectory-4 in 2D

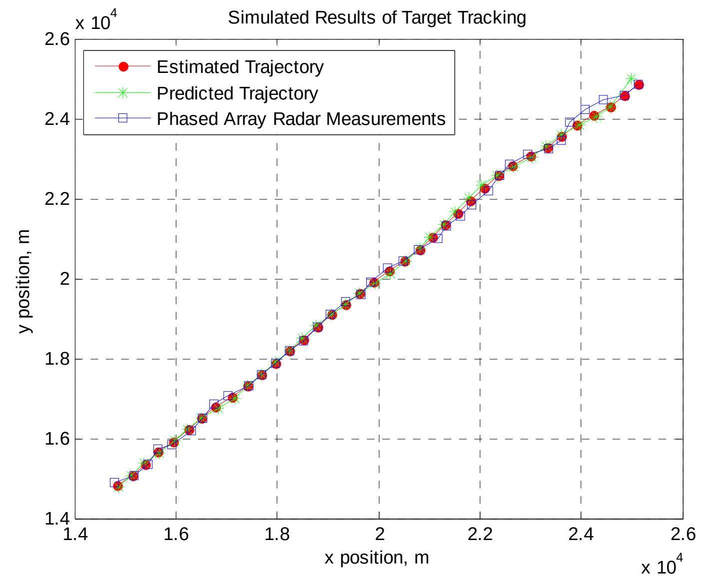

Figure 5-40: Predicted, Estimated Output of the Filter and Phased Array Radar Measurements in 2D, for one MC run

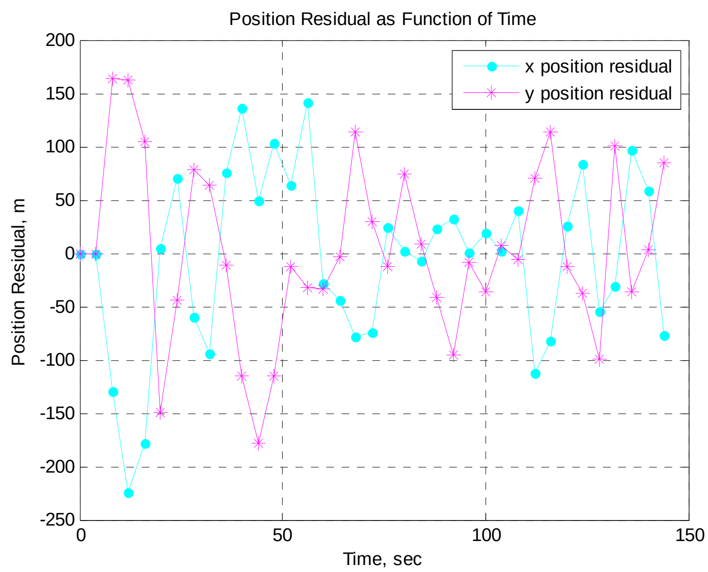

Figure 5-41: Position Residual in *x* and *y* coordinates versus Time, for one MC run

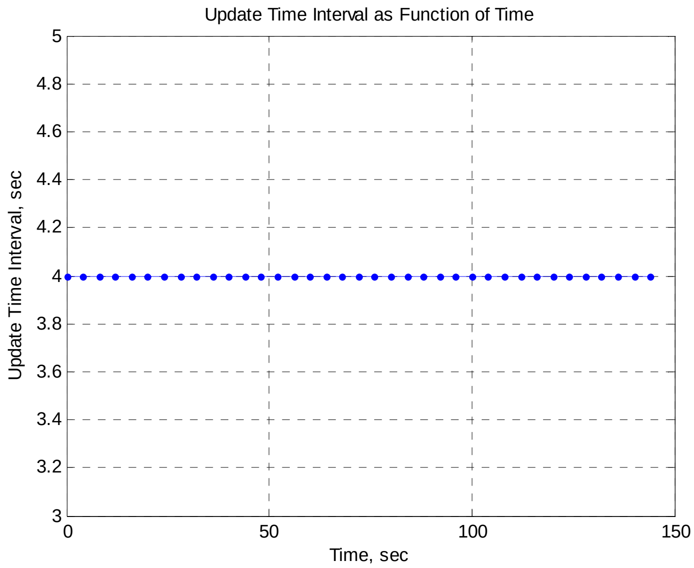

Figure 5-42: Track Update Time Interval versus Time, for one MC run

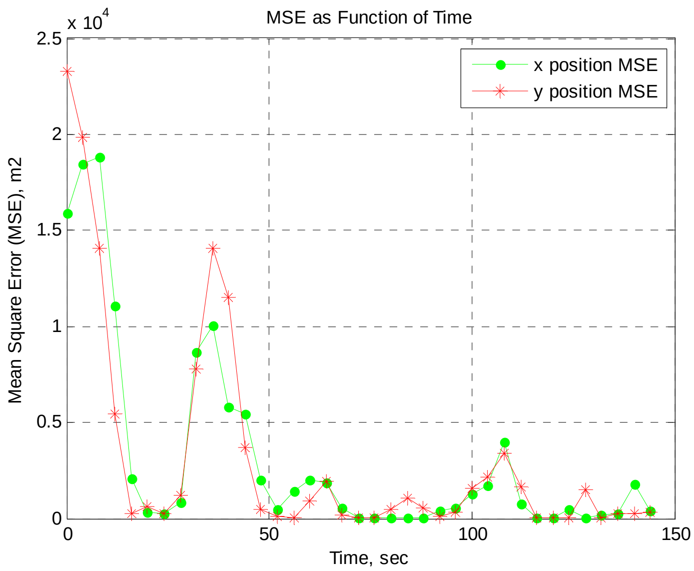

Figure 5-43: MSE in *x* and *y* coordinates versus Time, for one MC run

By implementing the Update Time Interval Determination Method-1 on KF for the control of the update time interval, the following remarks/comments are attained after the evaluation of the simulation results for tracking of a target with constant velocity.

Remarks and Comments:

- Since the CV model of the filter matches with the target's actual motion during the whole trajectory, the MSE and measurement residuals in *x* and *y* coordinates remain nearly constant at relatively low values. This is due to the fact that the implemented KF which is based on CV model is strictly valid when applied to the tracking of a target which moves with constant velocity.

- The random nature of the measurement noise can cause an increase in residuals. Due to the increase in residuals, an increase on the MSE may occur at some time instants as shown in Figure 5-43.

- Due to the fact that measurement residuals are nearly constant at a relatively low value during the whole trajectory, the update time interval does not decrease. The initial longest update time interval is kept during the trajectory.

General Conclusions for Part D of Simulation Group 1:

- In order to keep the tracking of a constant-velocity target by the phased array radar, there is no need for the update time interval to be reduced. The longest predefined update time interval is kept during the trajectory so that more time is made available to the phased array radar for other tasks.

- During the constant-velocity motion of the target, the estimation errors are fairly low compared with the errors obtained in the maneuvering segments of previous simulations in Part A and Part B of Simulation Group 1.

Because CV model of the tracking filter is valid for tracking of the target which moves with constant velocity.

5.4.1.5 Part E - Effect of the Measurement Noise

The effect of measurement noise is investigated by decreasing the covariances of the measurement noise. Update Time Interval Determination Method-1 is applied to track Target Trajectory-2 for the case in which the reduction in the measurement noise terms is exaggerated in order to demonstrate the performance difference compared with the case in which measurement noise terms are more realistic as in Part B. The parameter values of this simulation are same as the parameter values used in Part A of Simulation Group 1, except the covariances of measurement noise. The important parameters of the simulation are given in the following table.

Table 5-9: Parameters of Simulation

Update Time Interval (sec)	Adaptive
Update Time Interval Determination Method	Update Time Interval Determination Method-1
Initial Update Time Interval (sec)	4.00 sec
Actual Measurement Noise	Range (m): Gaussian, $\sigma = 2$ Azimuth (rad): Gaussian, $\sigma = 0.0001$
Measurement Noise Model in the Filter	Same as the covariances of actual measurement noise
Process Noise Model (Q matrix) in the Filter	$Q = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix}$
Radar Location	$x_r = 7000$ m, $y_r = 0$ m
Target Initial Location	$x = 0$ m, $y = 0$ m

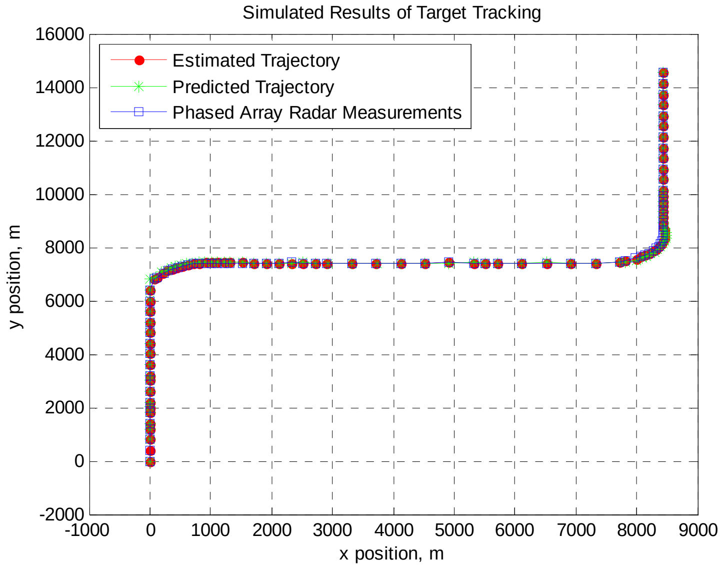

Figure 5-44: Predicted, Estimated Output of the Filter and Phased Array Radar Measurements in 2D, for one MC run

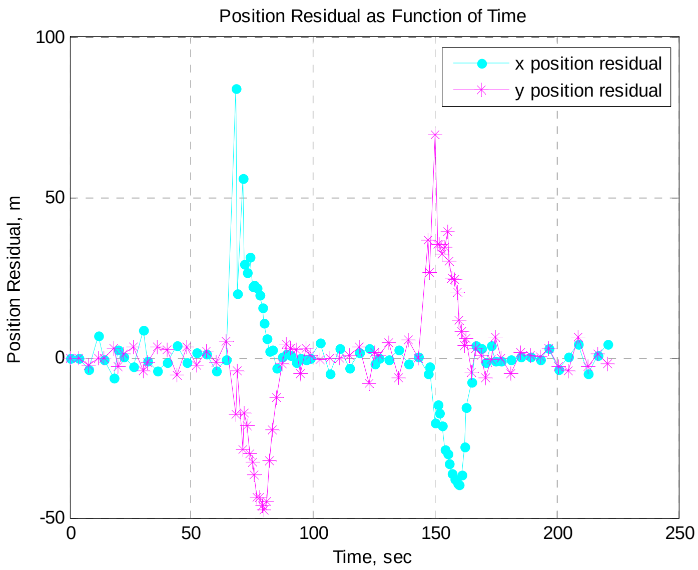

Figure 5-45: Position Residual in *x* and *y* coordinates versus Time, for one MC run

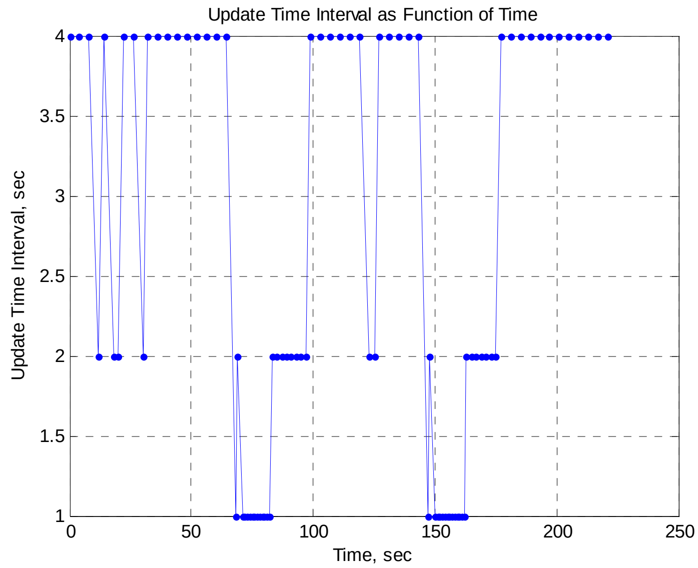

Figure 5-46: Track Update Time Interval versus Time, for one MC run

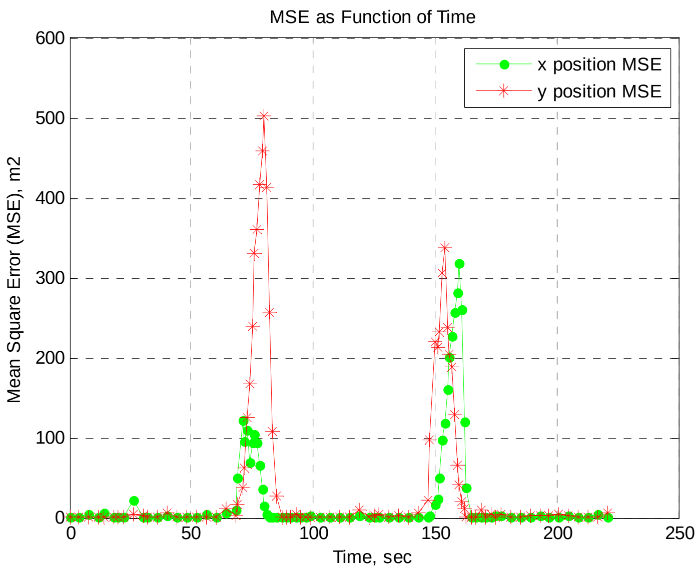

Figure 5-47: MSE in *x* and *y* coordinates versus Time, for one MC run

By implementing the Update Time Interval Determination Method-1 on KF for the control of the update time interval in the case of low measurement noise, the following remarks/comments are attained after the comparison of simulation results with the simulation results for tracking of the target where measurement noise terms are more realistic as in Part B.

Remarks and Comments:

- In maneuvering segments, the update time interval is reduced to a minimum of 1.00 sec for the low measurement noise case, while the update time interval is reduced to a minimum of 2.00 sec for the relatively high measurement noise case. Consequently, the increase in the number of updates and decrease in the average update time interval occurs during the whole trajectory in the low measurement noise case compared with relatively high measurement noise case.

- The reason for the shorter update time interval in the case of low measurement noise is that when the measurement noise is low, the tracking filter relies on the radar measurements. In maneuvering segments, since the target motion does not match with the filter's CV motion model, the tracking filter takes measurements more frequently in order to track the target accurately.

- The position residuals are smaller in the low measurement noise case in both maneuvering and non-maneuvering segments. This is due to the fact that the difference between the predicted position and the actual measured position is smaller.

5.4.1.6 Part F - Effect of the Process Noise in the Tracking Filter

The effect of low process noise used in the state model of the filter is investigated in the performance of target tracking for adaptive update time interval case.

The process noise assumption in the tracking filter is reduced to simulate a worse case in terms of model mismatch of the filter's motion model with the target's actual motion in the maneuvering segment. Update Time Interval Determination Method-1 is used for the tracking of Target Trajectory-1 and the comparison is made with the simulation results of Part A.

The parameter values of this simulation are same as the parameter values used in Part A of Simulation Group 1 given in Table 5-1, except process noise model,

$Q = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$ is taken.

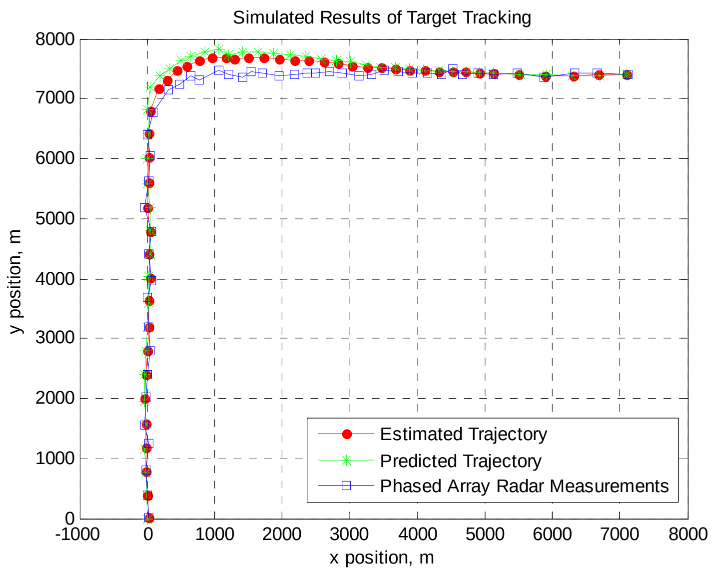

Figure 5-48: Predicted, Estimated Output of the Filter and Phased Array Radar Measurements in 2D, for one MC run

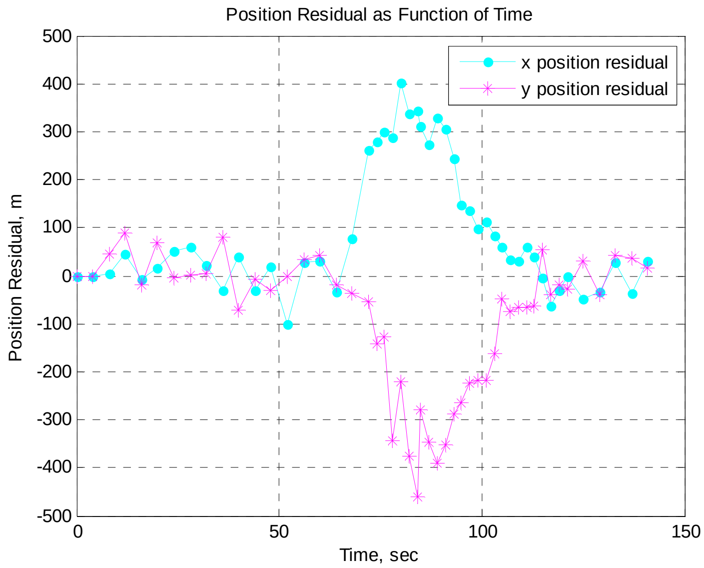

Figure 5-49: Position Residual in *x* and *y* coordinates versus Time, for one MC run

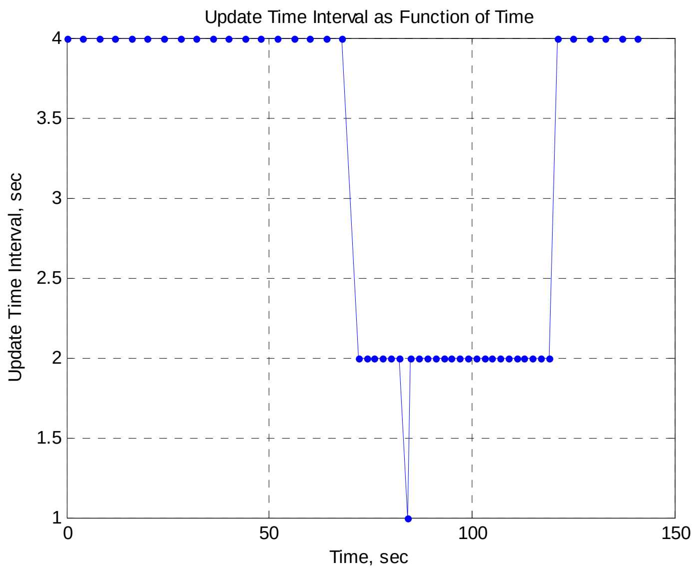

Figure 5-50: Track Update Time Interval versus Time, for one MC run

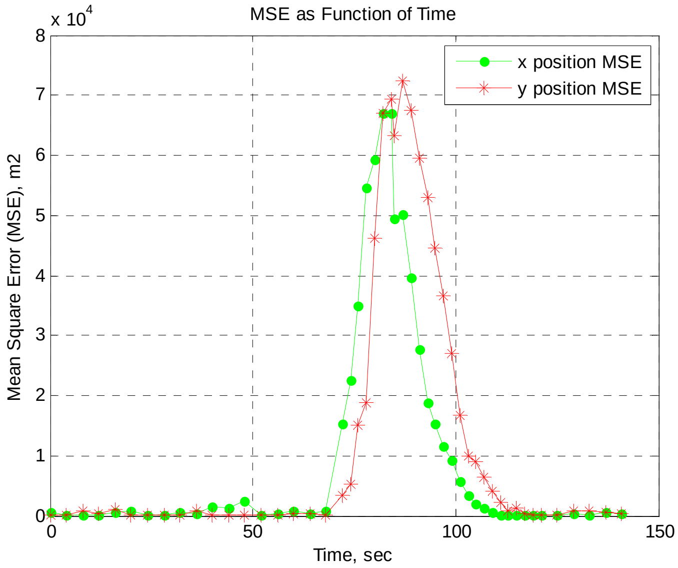

Figure 5-51: MSE in *x* and *y* coordinates versus Time, for one MC run

By implementing the Update Time Interval Determination Method-1 on KF for the control of the update time interval for the case of relatively low process noise in the filter, the following remarks/comments are attained after the comparison of simulation results with the simulation results for tracking of the target where the process noise is higher as in Part A.

<u>Remarks and Comments:</u>

- Since process noise model is much tighter than it should be to overcome the maneuver, the MSE during the whole trajectory is increased. Also, further reduction of the process noise covariance may result in divergence and the filter may fail to track the target.

- The position residual gets higher values in the case of low process noise as compared to the case of high process noise. This is due to the fact when the process noise covariance is lower, then the tracking filter trusts the motion model of the filter. However, the target does not make constant velocity motion during maneuvering segments. This causes the position residual to get higher values.

- In the case of low process noise, it takes relatively long time for the position residual to decrease after completion of the maneuver, which prevents the update time interval to attain its maximum value. The maximum value of the update time interval is attained in relatively shorter time for the case of higher process noise.

5.4.1.7 Part G - Effect of the Duration of Maneuvering Segment

The effect of the duration of the maneuvering segment of the trajectory is investigated by increasing the duration of the maneuvering segment of Target Trajectory-1. For this purpose, a new trajectory which is named as Target Trajectory-5 is created. Update Time Interval Determination Method-1 is applied to track Target Trajectory-5. The performance difference is compared with the case in which duration of the maneuvering segment is shorter as in Part A when Target Trajectory-1 is tracked.

The parameter values of this simulation are same as the parameter values used in Part A of Simulation Group 1 except measurement noise. The covariances of the measurement noise are reduced in order to observe easily the performance difference between long duration of the maneuvering segment case and short duration of the maneuvering segment case.

The important parameters of the simulation are given in the following table.

Table 5-10: Parameters of Simulation

Update Time Interval (sec)	Adaptive
Update Time Interval Determination Method	Update Time Interval Determination Method-1
Initial Update Time Interval (sec)	4.00 sec
Actual Measurement Noise	Range (m): Gaussian, $\sigma = 4$ Azimuth (rad): Gaussian, $\sigma = 0.0004$
Measurement Noise Model in the Filter	Same as the covariances of actual measurement noise
Process Noise Model (Q matrix) in the Filter	$Q = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix}$
Radar Location	$x_r = 7000$ m, $y_r = 0$ m
Target Initial Location	$x = 0$ m, $y = 0$ m

<u>Case-1 of Part G: Short Duration Of Maneuvering Segment Case:</u>

True Target Trajectory-1 is tracked.

Figure 5-52: Predicted, Estimated Output of the Filter and Phased Array Radar Measurements in 2D, for one MC run

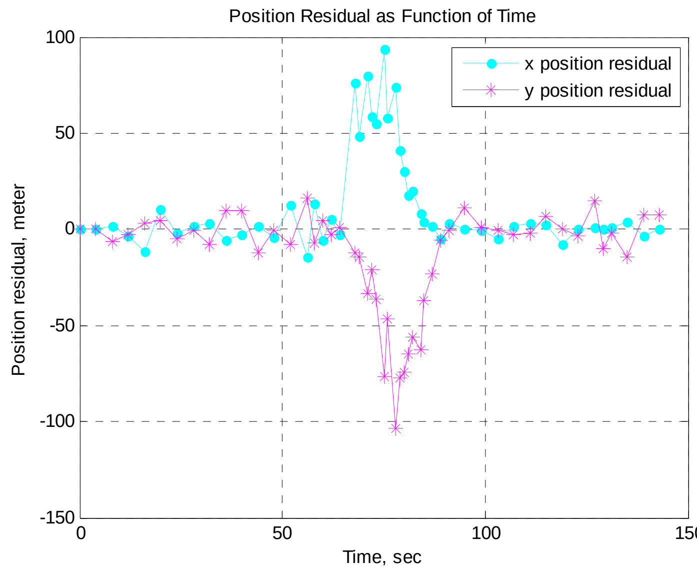

Figure 5-53: Position Residual in *x* and *y* coordinates versus Time, for one MC run

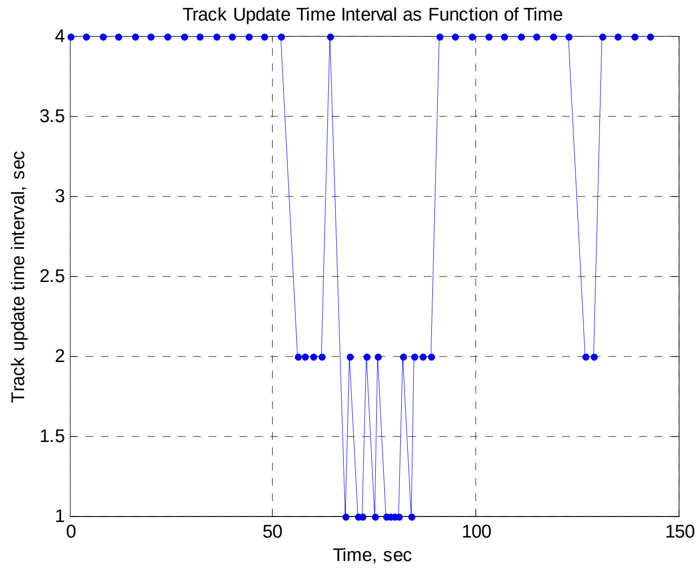

Figure 5-54: Track Update Time Interval versus Time, for one MC run

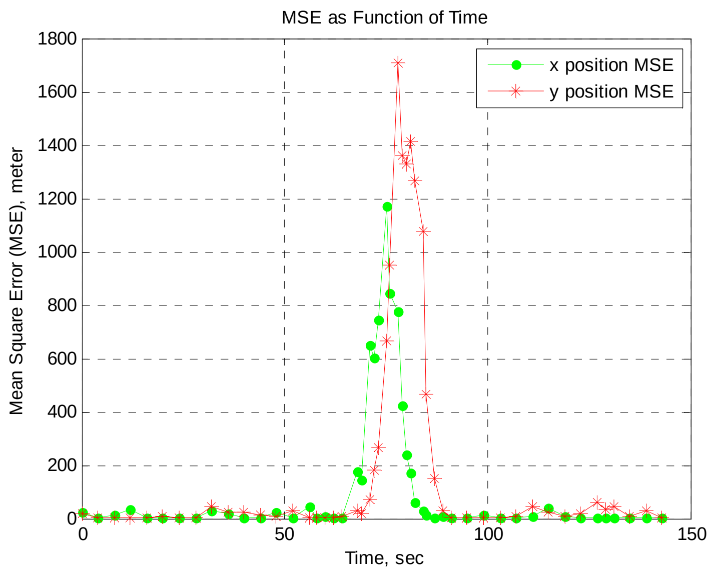

Figure 5-55: MSE in *x* and *y* coordinates versus Time, for one MC run

Case-2 of Part G: Long Duration Of Maneuvering Segment Case:

True Target Trajectory-5 which is given in detail below is tracked:

Trajectory Duration: 144sec

Target Speed: 100 m/sec (constant during trajectory)

Trajectory is composed of 3 segments:

Segment 1: A Non-maneuvering segment, constant velocity motion along y axis, lasting 48 sec, from 0 to 48 sec

Segment 2: A Maneuvering segment, 90° maneuver with constant speed, lasting 48 sec, from 48 to 96 sec

Segment 3: A Non-maneuvering segment, constant velocity motion along *x* axis, lasting 48 sec, from 96 to 144 sec

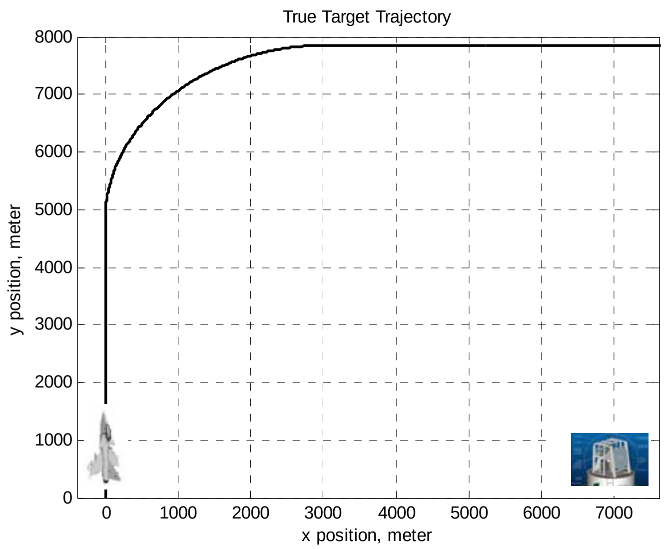

Figure 5-56: True Target Trajectory-5 with Long Duration Maneuver in 2D

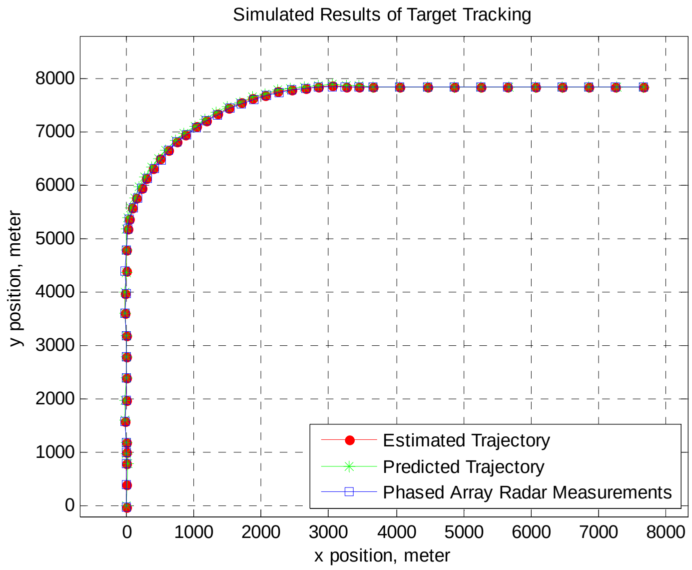

Figure 5-57: Predicted, Estimated Output of the Filter and Phased Array Radar Measurements in 2D, for one MC run

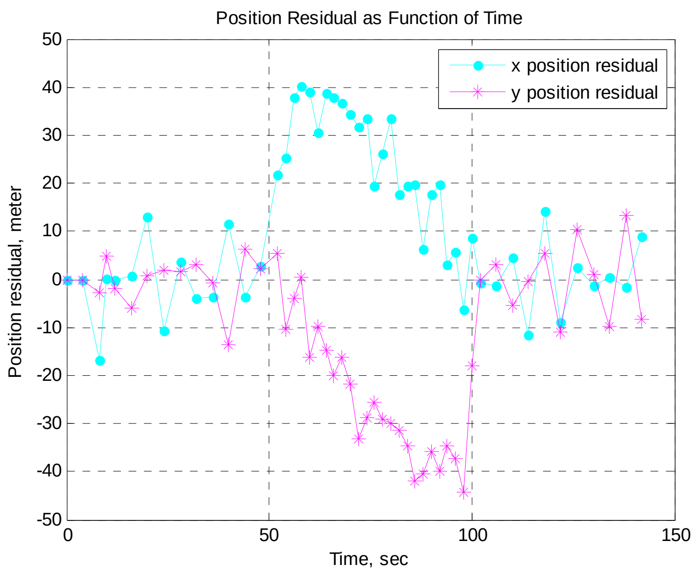

Figure 5-58: Position Residual in *x* and *y* coordinates versus Time, for one MC run

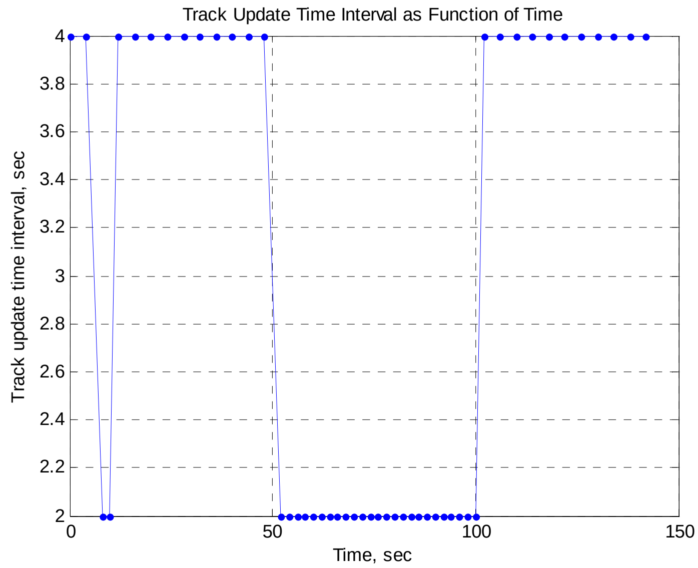

Figure 5-59: Track Update Time Interval versus Time, for one MC run

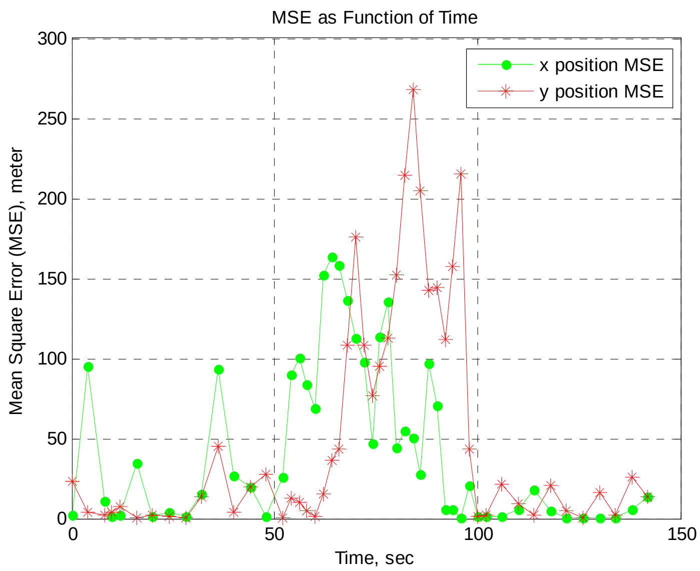

Figure 5-60: MSE in *x* and *y* coordinates versus Time, for one MC run

By implementing the Update Time Interval Determination Method-1 on KF for the control of the update time interval for the case of relatively long duration of the maneuvering segment, the following remarks/comments are attained after the comparison of simulation results with the simulation results for tracking of the target where the maneuvering segment is shorter as in Part A.

Remarks and Comments:

- As the duration of the maneuvering segment decreases, the update time intervals attained in the maneuvering segment decrease.

- In the maneuvering segment, the update time interval is reduced to a minimum of 1.00 sec for the short duration of maneuvering segment case, while the update time interval is reduced to a minimum of 2.00 sec for the relatively long duration of maneuvering segment case. This observation shows that in the short duration of maneuvering segment case, since the sharpness of 90° turn is higher compared with relatively long duration of maneuvering segment case, the tracking filter decreases the update time interval values and takes the measurements more frequently in order to keep the track of the target.

- In the maneuvering segment, the position residual is reached to higher values in the short duration of maneuvering segment case compared with the long duration of maneuvering segment case. This observation shows that in the short duration of maneuvering segment case, since the sharpness of 90° turn is higher compared with high duration of maneuvering segment case, the position residual gets higher values which causes the update time interval to get lower values compared with the high duration of maneuvering segment case in order to keep the track of the target.

5.4.2 Simulation Group 2 – Investigation of Update Time Interval Determination Method-3 (Newly proposed Method)

The effect of the number of discrete levels in the set from which the next update time interval is selected is investigated by defining a different set of discrete values for the update time interval than the set used in Update Time Interval Determination Method-1. Target Trajectory-1 is tracked for the case in which the number of discrete levels in the set of update time intervals is increased in order to demonstrate the performance difference compared with the case in which the discrete levels in the set of update time intervals are less as in Part A.

5.4.2.1 Overview of Newly Proposed Update Time Interval Determination Method-3

In this proposed method, the next update time interval $T(k)$ is selected by comparing the calculated measurement residual e_k with the standard deviation of the measurement noise σ_n at each data point as in the method of [2].

In this new method, $T(k)$ is chosen from the discrete set of values, depending on the maximum of the magnitudes of the measurement residuals on *x* and *y* directions as in the method of [2].

In the method performed by Cohen [2], the relation between the update time interval and the position residual is defined as below:

$$T(k) = \frac{4}{2^p}, \text{ if } |e_k| > 2^{2p} \times \sigma_n \quad \text{where, p=1, …, 4} \tag{5-3}$$

where,

$T(k)$: the update time interval for the next track update

σ_n: the standard deviation of measurement noise

e_k : the measurement residual at the data point t_k

In this newly proposed method, in order to investigate the effect of the number of discrete levels in the set from which the next update time interval $T(k)$ is selected, a different set of discrete values are defined according to the following rule defined as below:

$$T(k) = \frac{4}{2^{p/2}}, \text{ if } |e_k| > 2^p \times \sigma_n \quad \text{where, p=1, …, 8} \tag{5-4}$$

In the defined set of the update time intervals, the upper limit of the update time interval $T(k)$ is selected as 4.00 sec and the lower limit of $T(k)$ is selected as 0.25 sec as in the method of [2].

According to Equation (5-4), the set of the update time intervals that can be used are 0.25, 0.35, 0.50, 0.71, 1.00, 1.41, 2.00, 2.83 and 4.00 sec. On the other hand, in the method of [2], the set of update time intervals that can be used are 0.25, 0.50, 1.00, 2.00 and 4.00 sec.

Thus, a different number of discrete levels is used in this newly proposed method where 9 levels are used for the selection of the next update time interval as opposed to Cohen's method [2] where 5 discrete levels are used for the selection of the next update time interval.

5.4.2.2 Implementation of Simulation Group 2

The simulation results of adaptive update rate case where Update Time Interval Determination Method-1 is used as adaptive time interval algorithm is compared with the case where the newly proposed method is used as the adaptive time interval algorithm for tracking of Target Trajectory-1.

The parameter values of this simulation are same as the parameter values used in Part A of Simulation Group 1 given in Table 5-1 except the Update Time Interval Determination Method.

In order to analyze the application of the newly proposed method as the adaptive time interval algorithm and to show clearly the performance of the newly proposed method, “Estimated Trajectory”, “Position Residuals as function of time”, “Track Update Time Interval as function of time”, “MSE as function of time” graphs are demonstrated.

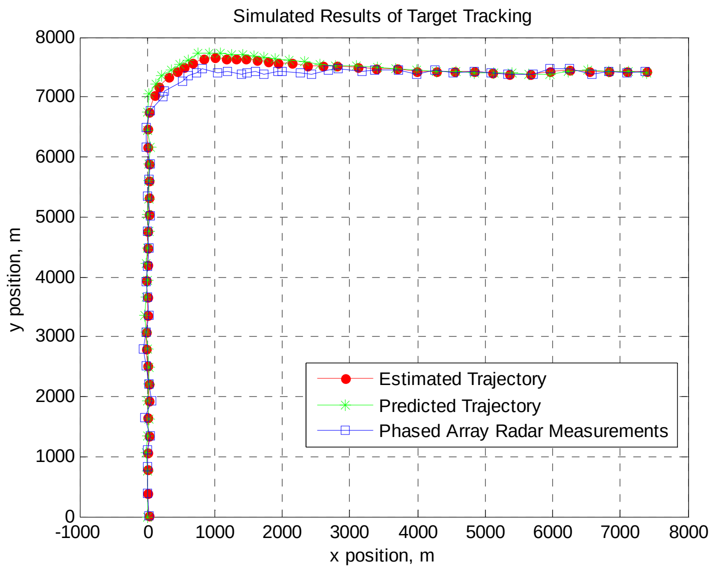

Figure 5-61: Predicted, Estimated Output of the Filter and Phased Array Radar Measurements in 2D, for one MC run

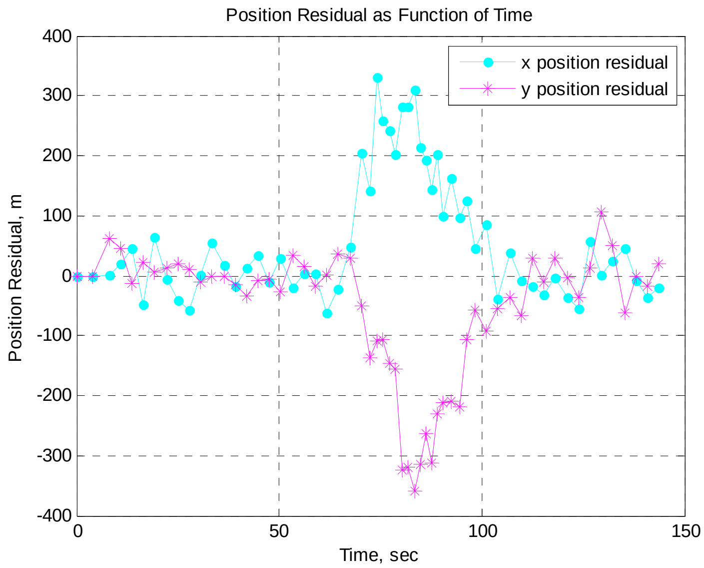

Figure 5-62: Position Residual in *x* and *y* coordinates versus Time, for one MC run

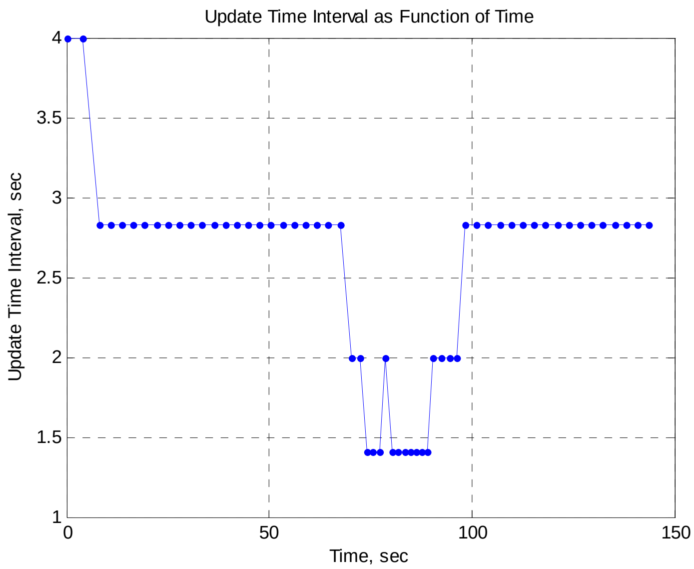

Figure 5-63: Track Update Time Interval versus Time, for one MC run

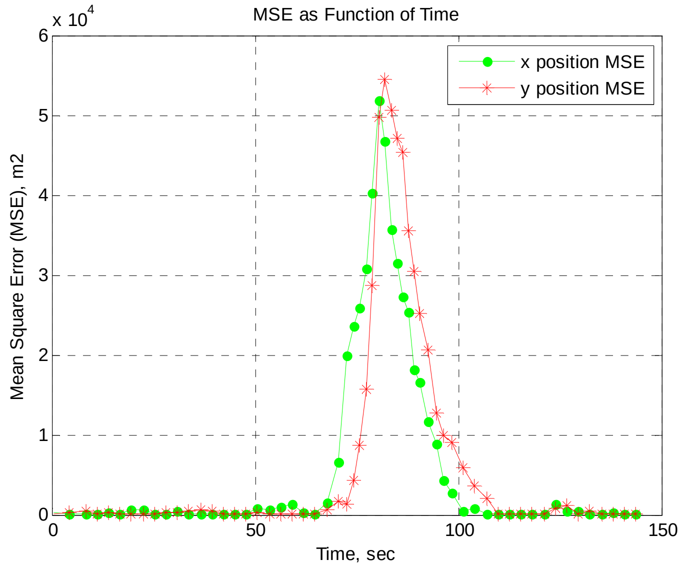

Figure 5-64: MSE in *x* and *y* coordinates versus Time, for one MC run

<u>Results of MC Simulations:</u>

The results of 10 MC simulations in terms of average update time interval versus time step are given in the following graphs for Update Time Interval Determination Method-1 and Update Time Interval Determination Method-3. Each time step in the demonstrated graphs represents the stage of the trajectory with 8 sec of duration.

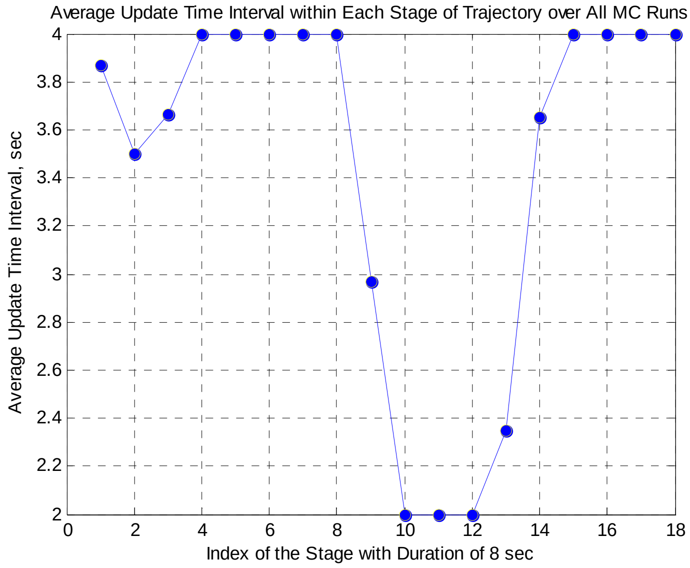

Figure 5-65: Average Update Time Interval versus Index of the Trajectory Stage with Method-1, over 10 MC runs

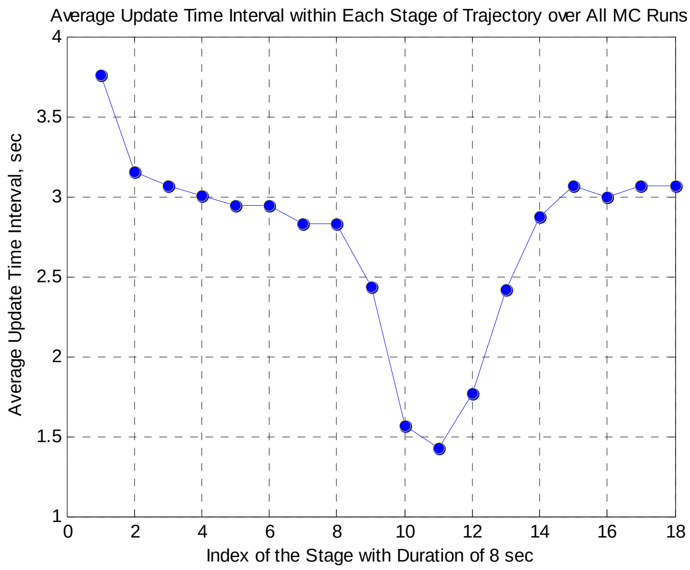

Figure 5-66: Average Update Time Interval versus Index of the Trajectory Stage with Method-3, over 10 MC runs

Furthermore, average RMSE versus time graphs are given for Update Time Interval Determination Method-1 and Update Time Interval Determination Method-3 after realizations of 10 MC simulations.

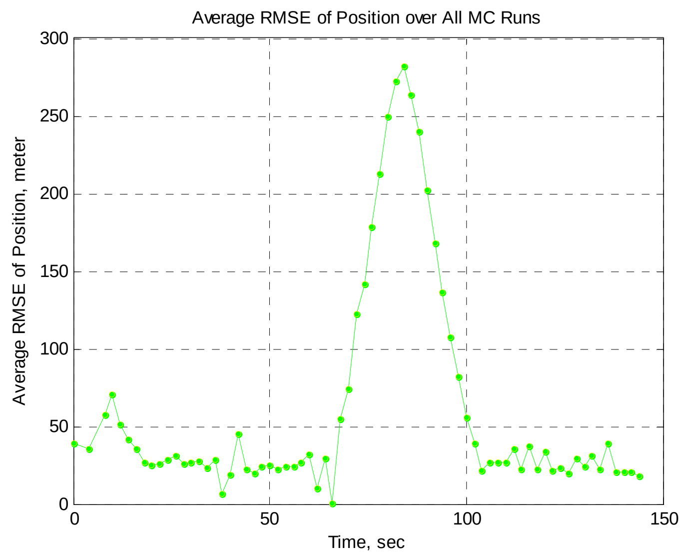

Figure 5-67: Average of the RMSE versus Time with Method-1, over 10 MC runs

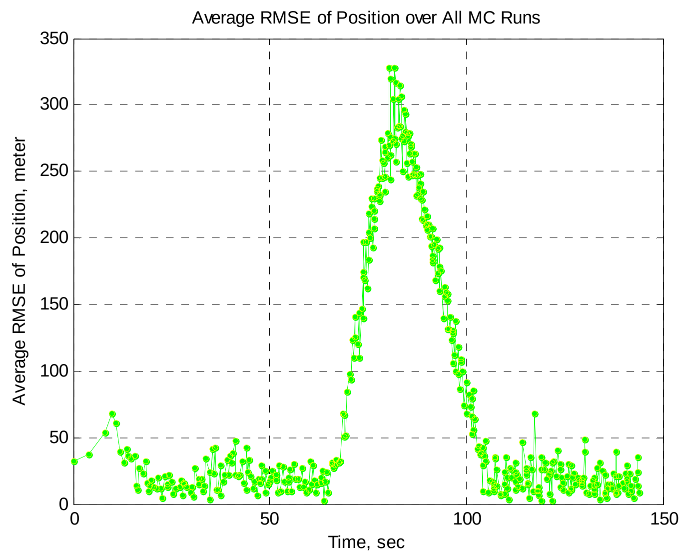

Figure 5-68: Average of the RMSE versus Time with Method-3, over 10 MC runs

Comparison of Update Time Interval Determination Method-1 (5 discrete level case) and Method-3 (9 discrete level case) is done in terms of overall RMSE in position, average number of updates and average update time interval after realization of 10 MC simulations. The results are summarized in the table given below.

Table 5-11: Comparison of 10 MC Simulations Results for Update Time Interval Determination Methods

	Update Time Interval Determination Method	
	Method-1 (5 discrete levels)	**Method-3 (9 discrete levels)**
Average Update Time Interval (sec)	3.13	2.48
Average Number of Updates	46	58
Overall RMSE in Position (m)	93.61	128.99

By implementing the proposed Update Time Interval Determination Method-3 on KF for the control of the update time interval, the following remarks/comments are attained after the comparison of the simulation results for tracking of Target Trajectory-1 for the cases where Update Time Interval Determination Method-1 and newly proposed method (Update Time Interval Determination Method-3) are used.

<u>Remarks and Comments:</u>

- In the method proposed by Cohen in [2], the update time interval is kept at the maximum value of 4.00 sec, until the maneuver begins. However, the update time interval is mainly kept at a smaller value of 2.83 until the maneuver begins with this newly proposed method. Consequently, in the first non-maneuvering segment, the number of updates is higher with this newly proposed method than the case where the method given in [2] is used.

- When the newly proposed method is used, the update time interval is reduced to a minimum of 1.41 sec whereas when Cohen's method [2] is used, the update time interval is reduced to a minimum of 2.00 sec. Consequently, the number of updates is higher with this newly proposed method than the case where the method given in [2] is used.

- By the decrease of the update time interval in the maneuvering segment, the decrease on the measurement residual occurs which then causes the update time interval to increase again to 2.83 sec. On the other hand, when Cohen's method [2] is used, by the decrease on the measurement residual the update time interval is increased again to 4.00 sec which is the maximum allowable update time interval.

- As the number of discrete levels in the set of time intervals increase, the number of updates increases. Thus, the radar resources used in the case of the newly proposed method is more than the case where Cohen's method [2] is used.

5.4.3 Simulation Group 3 - Comparison of Adaptive Time Interval and Fixed Time Interval Cases

The simulation results of adaptive time interval algorithm in which update time interval is controlled is compared with the simulation results of fixed time interval algorithm in which the update time interval is held constant. Update Time Interval Determination Method-1 is used as adaptive update time interval determination algorithm to simulate the case where the update time interval is controlled.

The parameter values of this simulation are similar to the parameter values used in Part A of Simulation Group 1.

The important parameters of the simulation are given in the following table.

Table 5-12: Parameters of Simulation

Actual Measurement Noise	Range (m): Gaussian, $\sigma = 30$ Azimuth (rad): Gaussian, $\sigma = 0.003$
Measurement Noise Model in the Filter	Same as the covariances of actual measurement noise
Process Noise Model (Q matrix) in the Filter	$Q = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix}$
Radar Location	$x_r = 7000$ m, $y_r = 0$ m
Target Initial Location	$x = 0$ m, $y = 0$ m

In order to compare adaptive update time interval case with fixed update time interval cases of 4.00 sec and 2.00 sec, "Position Residual as function of time" and "MSE as function of time" graphs are demonstrated at each cases for one MC run.

<u>Case-1 of Simulation Group 3: Fixed Update Time Interval 4.00 sec case</u>

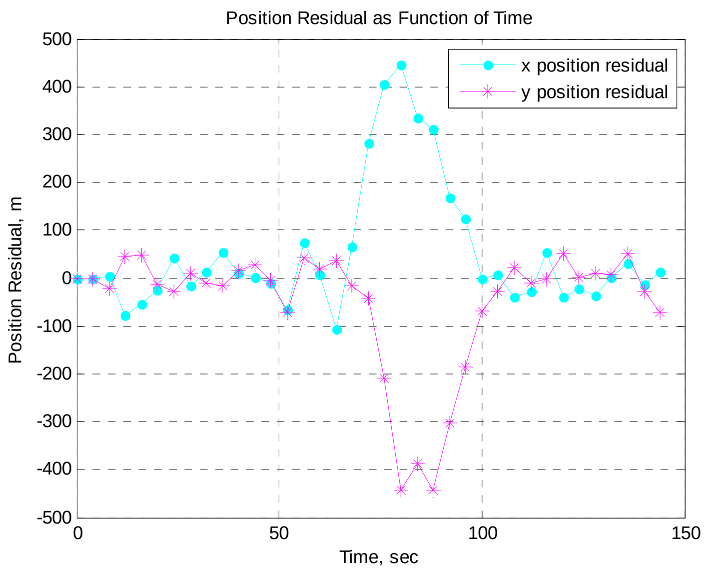

Figure 5-69: Position Residual in x and y coordinates versus Time, for one MC run

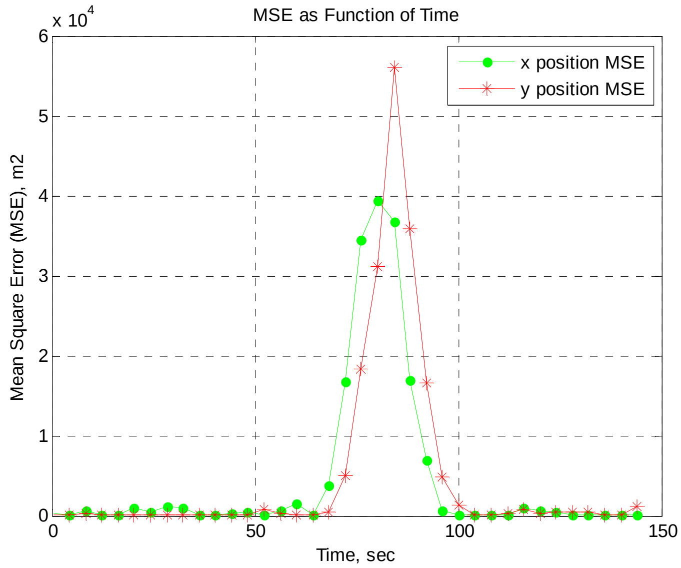

Figure 5-70: Track Update Time Interval versus Time, for one MC run

<u>Case-2 of Simulation Group 3: Fixed Update Time Interval 2.00 sec case</u>

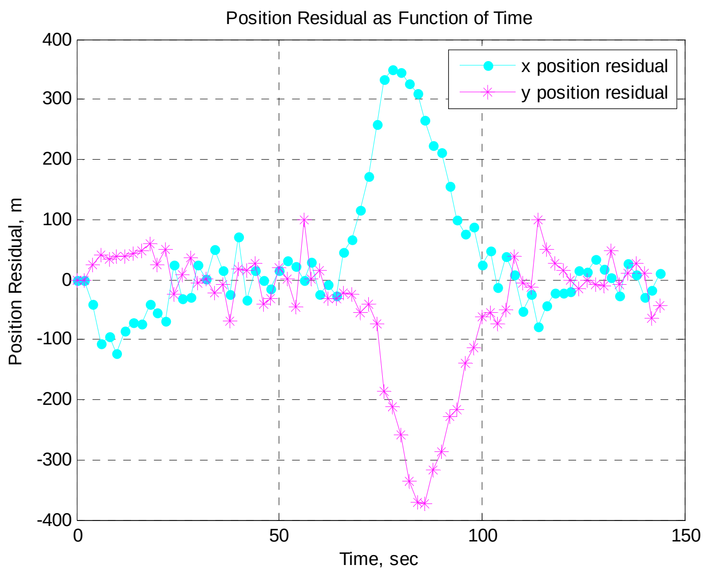

Figure 5-71: Position Residual in *x* and *y* coordinates versus Time, for one MC run

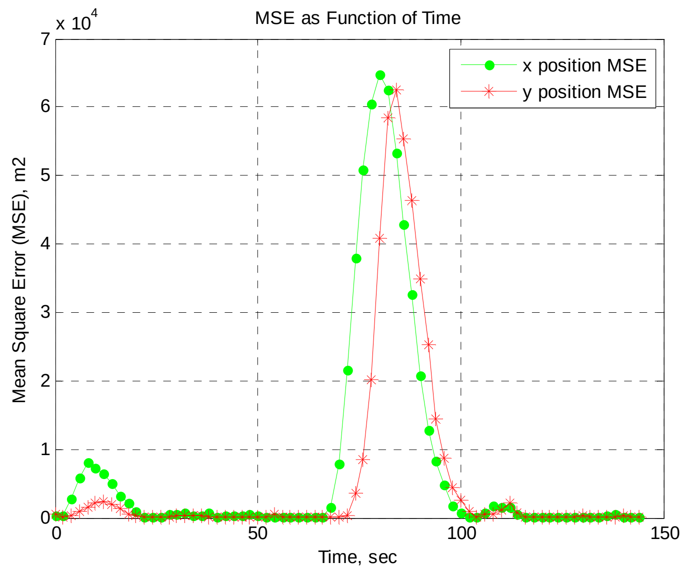

Figure 5-72: Track Update Time Interval versus Time, for one MC run

<u>Case-3 of Simulation Group 3: Adaptive Update Time Interval case</u>

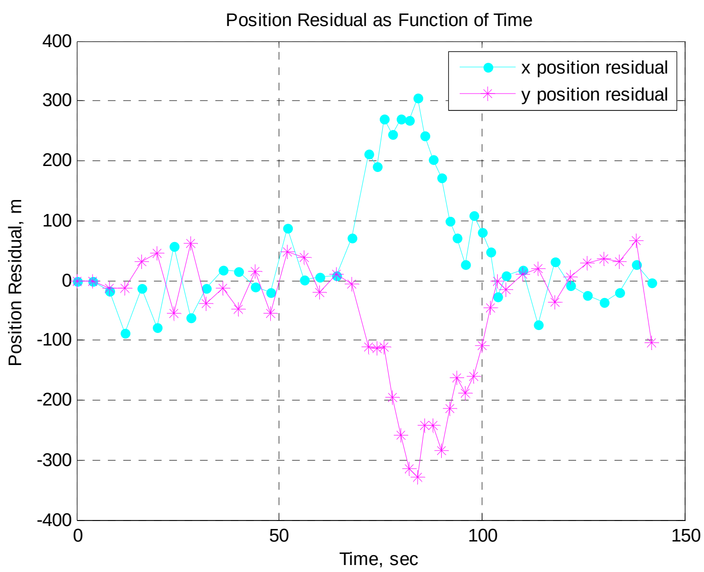

Figure 5-73: Position Residual in *x* and *y* coordinates versus Time, for one MC run

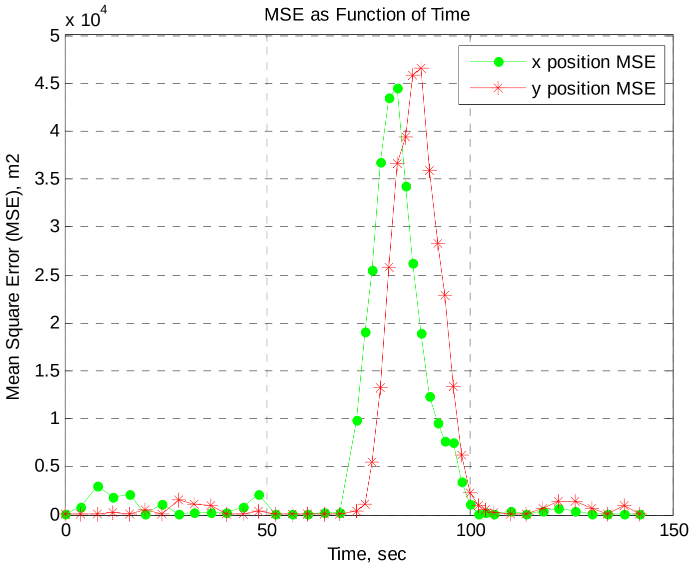

Figure 5-74: MSE in *x* and *y* coordinates versus Time, for one MC run

By implementing the KF for the cases update time interval is controlled and update time interval is fixed, the following remarks/comments are attained after the evaluation of the simulation results which are "Position Residual as function of time" and "MSE as function of time" graphs for adaptive update time interval, fixed update time interval cases of 4.00 sec and 2.00 sec for one MC run.

Remarks and Comments:

- During the constant-velocity segment of the trajectory, the measurement residual gets higher values in the fixed time interval of 4.00 sec case than the measurement residual values obtained in the fixed time interval of 2.00 sec case. Moreover, in the case of adaptive update time interval, the update time interval is mainly found to be 4.00 sec which causes the measurement residual to have the values approximately on the same order of magnitude with the fixed time interval of 4.00 sec case.

- During the maneuvering segment of the trajectory, measurement residuals get higher values in the fixed time interval of 4.00 sec case compared with the fixed time interval of 2.00 sec case. On the other hand, the MSE gets lower values in the fixed time interval of 4.00 sec case than the MSE values obtained in the fixed time interval of 2.00 sec case. This shows that although the measurement residuals get higher values along the maneuvering segment of the target trajectory, the MSE gets lower value in the case of longer update time interval compared with the shorter update time interval case.

- Along the maneuvering segment of the trajectory, when the update time interval is relatively shorter, the measurement residual gets relatively low values compared with the longer update time interval case as expected. Because, when the update time interval is small, although the target's motion does not match with the filter's CV motion model, the difference between the predicted position and the measured position gets relatively small value. Then, the calculated small measurement residual causes the Kalman filter to estimate the target's position closer to the filter's predicted position. Another reason for the estimated position be closer to the predicted position is that the assumed process noise model in the implemented Kalman filter is relatively low. In addition to this, the measurement noise is not so low to be able to force the estimated position be closer to the measured position rather than closer to the predicted position. In such a case, the filter's estimated position is found to be closer to the predicted position despite the fact that the target does not actually move at constant velocity, but it makes maneuver. Accordingly, the difference between the target's true position and the estimated position has relatively higher values in the case of shorter update time interval which causes the MSE values to have higher values.

- Along the maneuvering segment of the trajectory, when the update time interval is relatively longer, the measurement residual gets relatively high values compared with the shorter update time interval case as expected. Because, when the update time interval is high, the difference between the predicted position and the measured position gets relatively high value since the target's motion does not match with the filter's CV motion model. Then, the calculated high measurement residual causes the Kalman filter to estimate the target's position closer to the radar's measurement. Another reason for the estimated position be closer to the radar's measurement is that the measurement noise in the simulation is not so high to be able to force the estimated position be closer to the predicted position rather than closer to the measured position. In such a case, the filter's estimated position is found to be closer to the measured position due to the fact that the target does not move at constant velocity. Accordingly, the difference between the target's true position and the estimated position has lower values in the case of relatively longer update time interval which causes the MSE values to have lower values.

- When the assumed process noise model in the implemented Kalman filter is increased and assumed to be relatively high, the MSE can get higher values in the longer update time interval case than the MSE values obtained in the relatively short update time interval case. But in the high process noise case, the obtained measurement residual values are approximately on the same order of magnitude throughout the whole trajectory. Since the aim in this simulation is to compare the adaptive update time interval case with the fixed update time interval cases, to assume high process noise in the filter causes the filter can not detect the target's maneuver. The main reason for this is that the update time interval determination methods used in this study are based on the measurement residual approach, which uses the fact that the increase on the measurement residual means the target makes a maneuver.

- When the MSE versus time graph is investigated for adaptive update time interval case, it is observed that the magnitude of the tracking error reflects the change in the update time interval as opposed to fixed update time interval cases. Because during maneuvering segment, in adaptive update

Results of MC Simulations:

For the cases where the update time interval is controlled and is held constant in tracking with KF, 100 MC simulations are carried out. By comparing the number of updates and the estimation errors, the saving of phased array radar resources and the improvement on the tracking accuracy are investigated.

In order to compare adaptive and fixed time interval cases with regard to average number of updates, average update time interval and overall RMSE in position, 100 MC simulation results are summarized in Table 5-13 for adaptive time interval case and fixed time interval cases with different constant update time intervals. Note that, the overall RMSE in position is found by summing the RMSE values found in the whole trajectory and then dividing by the number of update data points. Tracking accuracy is represented by the overall RMSE in position.

Table 5-13: MC Simulations Results of Overall Trajectory for Adaptive and Fixed Update Time Interval Cases

	100 MC Simulation Results of Overall Trajectory (0-144 sec)		
Update time interval	**Average Number of Updates**	**Average Update Time Interval (sec)**	**Overall RMSE in Position (m)**
Adaptive	45	3.20	95.45
2.00 sec fixed	73	2.00	112.37
2.50 sec fixed	58	2.50	106.20
3.00 sec fixed	49	3.00	102.04
3.50 sec fixed	42	3.50	97.27
4.00 sec fixed	37	4.00	92.47

In order to observe the average RMSE of position versus time, the average RMSE of position is obtained by dividing the sum of the MSE in x and y coordinates computed at each track update data point by the number of updates carried out at the corresponding update times among overall MC runs and taking the square roots.

The results of 100 MC simulations in terms of average RMSE of position versus time is given in the following graphs for adaptive time interval, fixed time interval of 4.00 sec and fixed time interval of 2.00 sec cases.

Adaptive Update Time Interval Case:

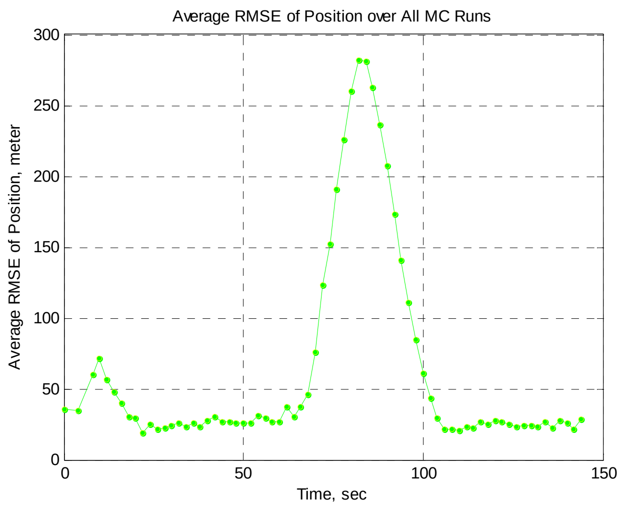

Figure 5-75: Average of the RMSE versus Time, over 100 MC runs for Adaptive Update Time Interval

Fixed Update Time Interval 4.00 sec Case:

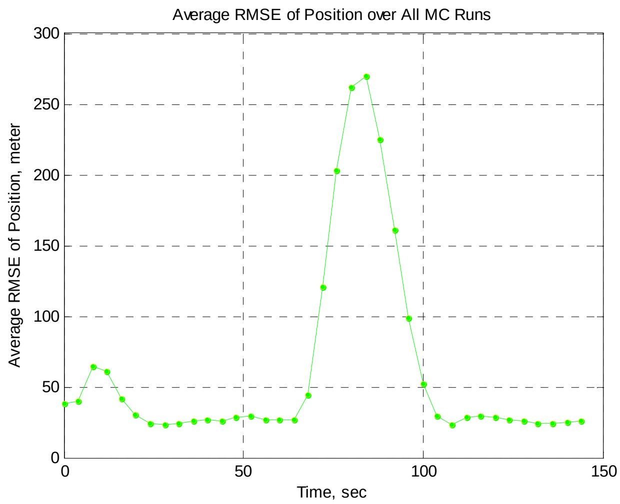

Figure 5-76: Average of the RMSE versus Time, over 100 MC runs for the Fixed Update Time Interval of 4.00 sec

Fixed Update Time Interval 2.00 sec Case:

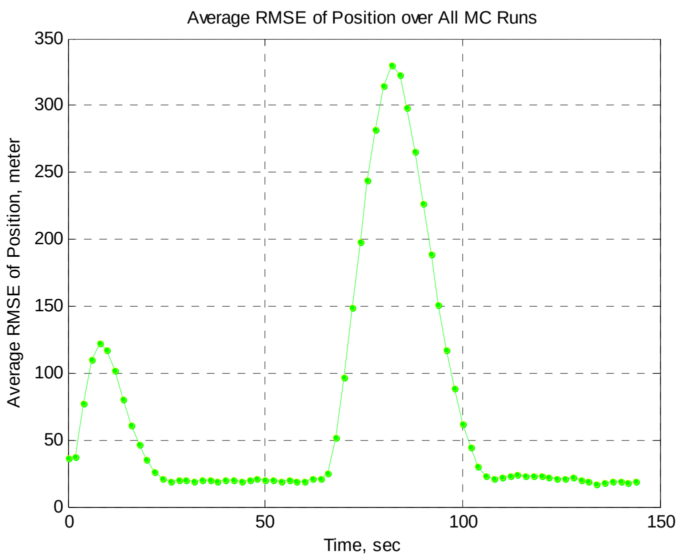

Figure 5-77: Average of the RMSE versus Time, over 100 MC runs for the Fixed Update Time Interval of 2.00 sec

The following remarks/comments are attained after the comparison and evaluation of 100 MC simulation results with regard to average number of updates, average update time interval, overall RMSE in position and the average RMSE of position versus time graphs for adaptive and fixed update time interval cases at different constant update time intervals.

Remarks and Comments:

- When the average numbers of updates are compared, the radar resources used in the case of the adaptive update time interval is almost the same as that for the fixed update time interval at 3.50 sec case.

- When the tracking accuracy in the case of adaptive update time interval is compared with the case of fixed interval at 3.50 sec, it seems that there is not a considerable improvement in the average RMSE with adaptive update time interval case as opposed to the situation in the Alpha-Beta filter where there is a relatively high improvement on the tracking accuracy when adaptive update time interval is adapted. It can be concluded that when the Alpha-Beta filter is used, the gain of the filter is a constant value which can cause the average RMSE values to get higher values when the update time interval is longer in the maneuvering segment. On the other hand when the Kalman filter is used, the average RMSE is affected from both the level of the measurement noise and the magnitude of the process noise in the filter.

- As stated in previous Remarks/Comments part, in the simulations of this thesis, the process noise in the filter is assumed so that it can represent target maneuvers in KF with CV Model. Besides, the process noise in the filter is assumed to be relatively low so that in the case of adaptive update time interval, the update time interval determination methods which is based on the measurement residual approach, can detect the maneuver depending on the increase in the measurement residual. For the relatively high process noise case, the obtained measurement residual values are approximately on the same order of magnitude along the whole trajectory and the filter can not detect the maneuver.

- During the constant-velocity segment of the trajectory, the average RMSE get higher values in the fixed time interval of 4.00 sec case than the average RMSE values obtained in the fixed time interval of 2.00 sec case. Moreover, in the case of adaptive update time interval, the update time interval is mainly found to be 4.00 sec which causes the average RMSE to have the values approximately on the same order of magnitude with the fixed update time interval of 4.00 sec case.

- During the maneuvering segment of the trajectory, the average RMSE gets lower values in the fixed time interval of 4.00 sec case than the average RMSE values obtained in the fixed time interval of 2.00 sec case. This shows that the average RMSE get lower values in the case of longer update time interval compared with the shorter update time interval case, although the measurement residuals get higher values in the fixed time interval of 4.00 sec case compared with the fixed time interval of 2.00 sec case as stated previously.

- As stated in previous Remarks/Comments part, along the maneuvering segment of the trajectory, when the update time interval is relatively shorter, the calculated small measurement residual causes the Kalman filter to estimate the target's position closer to the filter's predicted position. The other reasons for the estimated position be closer to the predicted position are the relatively low process noise in the implemented Kalman filter and the relatively high measurement noise. When the measurement noise is not so low, it can not force the estimated position be closer to the measured position. In such a case, the filter's estimated position is found to be closer to the predicted position although the target makes maneuver. Accordingly, the difference between the target's true position and the estimated position has relatively higher values in the case of shorter update time interval which causes the average RMSE values to have higher values in the maneuvering segment.

- As stated in previous Remarks/Comments part, along the maneuvering segment of the trajectory, since target's motion does not match with the filter's CV motion model, when the update time interval is relatively longer, the calculated relatively high measurement residuals causes the Kalman filter to estimate the target's position closer to the radar's measurement. Moreover, since the measurement noise in the simulation is not so high, it can not force the estimated position be closer to the predicted position rather. In such a case, the filter's estimated position is found to be closer to the measured position due to the fact that the target does not move at constant velocity. Accordingly, the difference between the target's true position and the estimated position has lower values in the case of relatively longer update time interval which causes the average RMSE values to have lower values.

5.4.4 Simulation Group 4 - Investigation of Update Time Interval Determination Method-2

Update Time Interval Determination Method-2 proposed in [5] (which decides the value of the next update time interval after computing the next update time interval by the formula given in Equation (3-20) where the value of the next time interval is directly proportional with the previous update time interval and inversely proportional with the standard deviation of the measurement noise) is used to control the update time interval by using KF with CV Model in 2D Cartesian coordinates throughout the simulations in this part.

In the simulations, for the ease of computational effort, the value of the next update time interval which is calculated by the formula given in Equation (3-20) is rounded to the nearest value with the resolution of 0.05 sec.

To observe the effectiveness of Update Time Interval Determination Method-2 for a target trajectory which includes maneuvering segment consisting of 90° maneuver, the scenario defined for Target Trajectory-1 is tracked.

In order to analyze the application of adaptive time interval algorithm to target tracking by using Update Time Interval Determination Method-2, “Estimated Trajectory”, “Position Residuals as function of time”, “Track Update Time Interval as function of time”, “MSE as function of time” graphs are demonstrated.

The parameter values of this simulation are same as the parameter values used in Part A of Simulation Group 1 given in Table 5-1 except the Update Time Interval Determination Method.

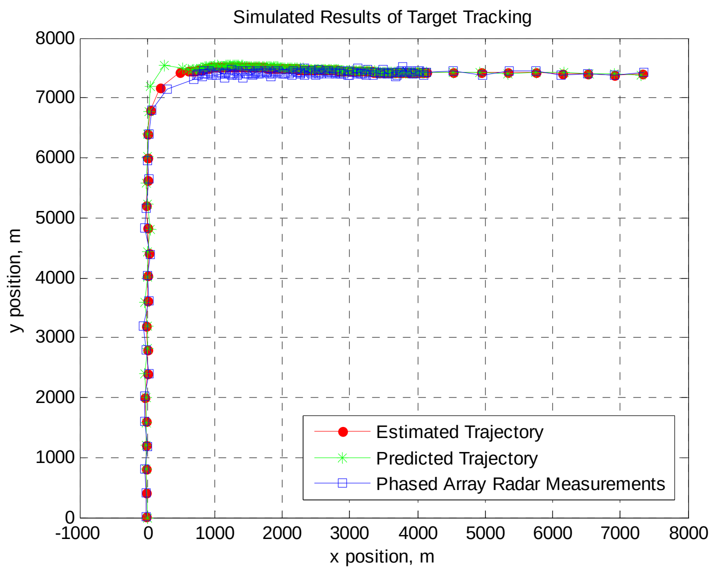

Figure 5-78: Predicted, Estimated Output of the Filter and Phased Array Radar Measurements in 2D, for one MC run

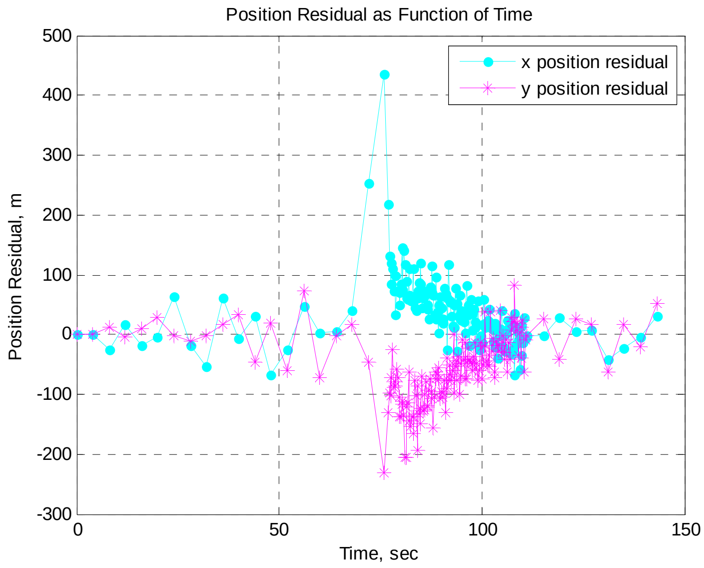

Figure 5-79: Position Residual in *x* and *y* coordinates versus Time, for one MC run

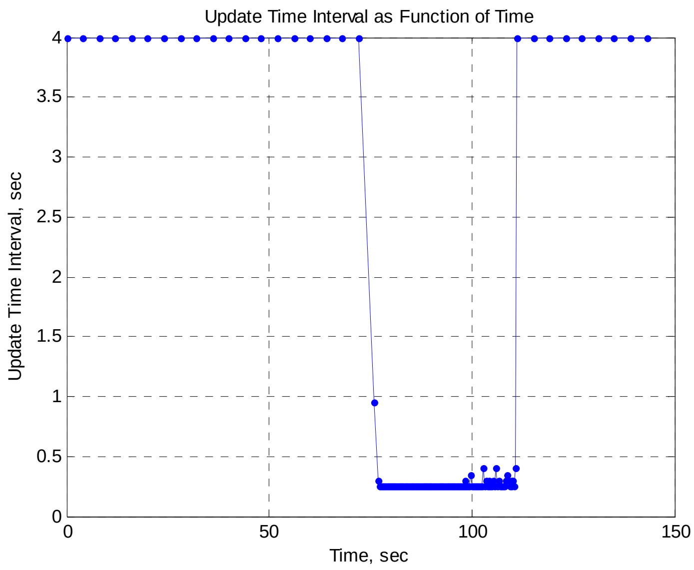

Figure 5-80: Track Update Time Interval versus Time, for one MC run

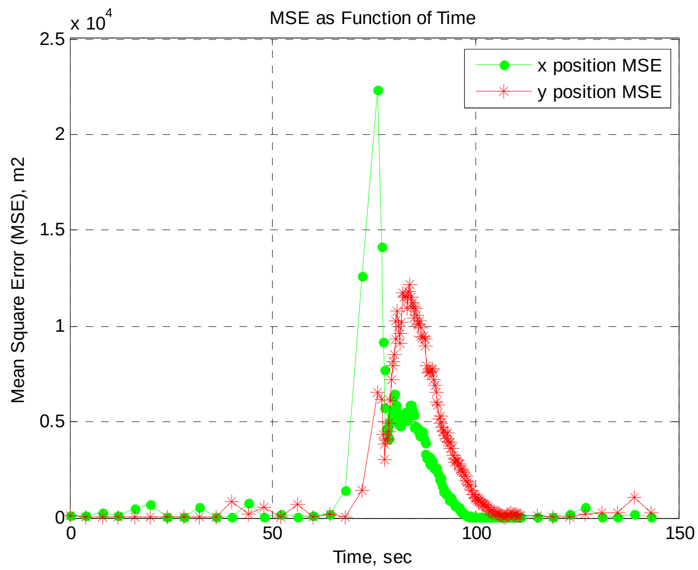

Figure 5-81: MSE in *x* and *y* coordinates versus Time, for one MC run

Results of MC Simulations:

The results of 100 MC simulations for the overall trajectory in terms of average update time interval and average number of updates are given in the following table.

Table 5-14: MC Simulations Results for Overall Trajectory

	Adaptive KF Result over 100 MC Runs
Average Update Time Interval (sec)	0.99
Average Number of Updates	146

The results of 100 MC simulations in terms of average update time interval and average number of updates versus time step are given in the following graphs. Each time step in the demonstrated graphs represents the stage of the trajectory with 8 sec of duration. For example, let the index of the stage be equal to 3 in the graph, then

this index represents the time duration between 16 sec - 24 sec time instants of the trajectory.

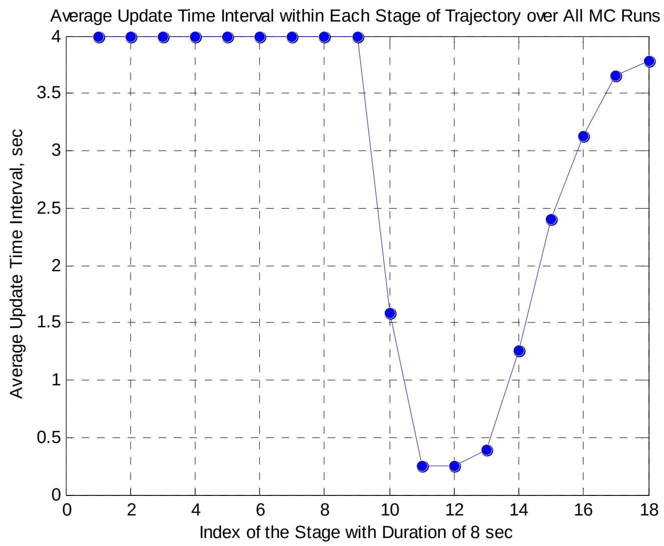

Figure 5-82: Average Update Time Interval versus Index of the Trajectory Stage, over 100 MC runs

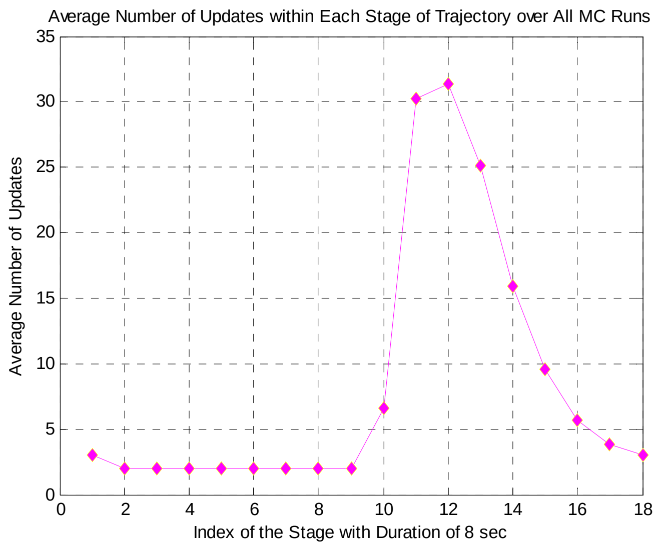

Figure 5-83: Average Number of Updates versus Index of the Trajectory Stage, over 100 MC runs

Moreover, the average RMSE of position is obtained by dividing the sum of the MSE in x and *y* coordinates computed at each track update data point by the number of updates carried out at the corresponding update times among overall MC runs and taking the square roots. It is believed that RMSE versus time graph obtained by this way will give an indication of the time dependence of RMSE for adaptive update rate case.

The results of 100 MC simulations in terms of RMSE versus time is given in the below graph.

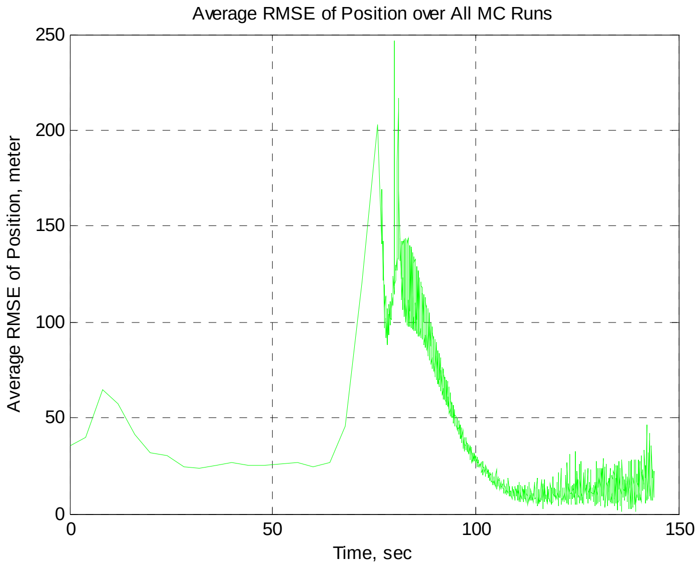

Figure 5-84: Average of the RMSE versus Time, over 100 MC runs

By implementing the Update Time Interval Determination Method-2 on KF for the control of the update time interval, the following remarks/comments are attained after the evaluation of the simulation results for tracking of Target Trajectory-1.

<u>Remarks and Comments:</u>

- After the maneuver begins, the measurement residuals (position residuals) and the estimation error increase, causing the update time interval to decrease.

- The reduction of the initial update time interval firstly occurs about 16 sec after the maneuver begins which means that there is a considerable delay in the algorithm to decrease the update time interval from its maximum value.

- The update time interval is reduced to a minimum of 0.25 sec during the trajectory.

- When the decrease of the update time interval occurs in the trajectory, the tracking accuracy improves and the MSE in x and y coordinates decrease.

- When the decrease of the update time interval occurs in the trajectory, the decrease on the measurement residuals occurs which then causes the update time interval to increase again.

- After the maneuver is completed, the maximum update time interval is attained within 40-48 sec. Then, the restored maximum update time interval is kept up to the completion of the trajectory.

5.4.5 Simulation Group 5 - Comparison of Update Time Interval Determination Method-1 and Method-2

The simulations results of Update Time Interval Determination Method-2 [5] in which time intervals may take continuous values is compared with the simulation results of Update Time Interval Determination Method-1 [2] in which the update time interval is chosen from a discrete set of values. For this purpose, Target Trajectory-1 is tracked by using these methods on KF with CV Model in 2D Cartesian coordinates.

Comparison of Update Time Interval Determination Method-1 (Discrete update time interval case) and Method-2 (Continuous update time interval case) is done in terms of average RMSE, average number of updates and average update time interval after realization of 100 MC simulations. The simulation results are summarized in Table 5-15 and Table 5-16.

Table 5-15: Comparison of MC Simulations Results for Average Number of Updates

	Average Number of Updates for the Overall Trajectory over 100 MC Runs
Discrete update time interval case	50
Continuous update time interval case	146

Table 5-16: Comparison of MC Simulations Results for Average Update Time Interval

	Average Update Time Interval for the Overall Trajectory over 100 MC Runs (sec)
Discrete update time interval case	2.88
Continuous update time interval case	0.99

Additionally, the overall RMSE in position is found by summing the RMSE values found in the whole trajectory at each update time instant and then dividing by the number of updates occurred in the whole trajectory. The results of 100 MC simulations in terms of overall RMSE is given in the following table for discrete update time interval and continuous update time interval cases.

Table 5-17: Comparison of MC Simulation Results for Overall RMSE

	Overall RMSE in Position (m)
Discrete update time interval case	95.45
Continuous update time interval case	56.51

By implementing the Update Time Interval Determination Method-1 and Update Time Interval Determination Method-2 on KF for tracking the Target Trajectory-1, the following remarks/comments are attained after the evaluation of the simulation results of both methods.

Remarks and Comments:

- There is a considerable improvement in the tracking performance when the method given in [5] named as Update Time Interval Determination Method-2 is adopted for choosing $T(k)$, because overall RMSE in position is lower than the case where Update Time Interval Determination Method-1 is used.

- A great reduction in the average update time interval is obtained with the continuous update time interval case where Update Time Interval Determination Method-2 is used when compared with the discrete update time interval case where Update Time Interval Determination Method-1 is used.

- Due to the reduction on the average update time interval, the number of updates is higher in the continuous update time interval case when compared with the discrete case.

- The reason for the reduction in the average update time interval is that the update time intervals are not forced to any fixed value in the continuous update time interval case as opposed to the discrete update time interval case.

- The update time intervals may have an almost infinite set of values within the set limits where maximum value is 4.00 sec and minimum value is 0.25 sec in the continuous update time interval case as opposed to the discrete update time interval case where the update time intervals may have discrete values within the predefined set of time interval values.

- In the continuous update time interval case, due to the direct usage of $T(k) = \frac{T(k-1)}{\sqrt{e_{0,k}}}$ equation (i.e. Equation (3-20)), the next update time interval is directly proportional to the previous update time interval.

- When average RMSE versus time graphs are compared for both cases, it is observed that the estimation errors for non-maneuvering segments are around the same order of magnitude.

- In the maneuvering segment, the estimation errors obtained are less with continuous case than discrete case, at a cost of a larger number of updates. This shows that the choice of Update Time Interval Determination Method-1 and Method-2 is a compromise between reducing the estimation errors and keeping the average update time interval at a significant value.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This chapter provides the summary and the concluding remarks of this study. Moreover, the possible future studies are given.

6.1 CONCLUSIONS

Phased array radar, provides the flexibility of using variable track update time interval during target tracking by means of its electronically beam steering property.

In this thesis, as an application to phased array radar, adaptive update time interval (adaptive control of update rate) algorithms are investigated on Kalman filter for the single target tracking problem where the target has 90° maneuvering segment during the trajectory.

Historically, the problem of adaptive update rate in target tracking has been examined by many researchers. Generally, the algorithms developed so far are based on measurement residual or predicted error covariance as the control parameter of the adaptive update rate algorithm to detect maneuver of the target.

The degree of adaptability of the target motion model in the implemented filter plays an important role for the filter to detect whether or not the target makes a maneuver. In Kalman filter, it is not possible to use predicted error covariance based adaptive update time interval algorithms since the degree of adaptability of the target motion to the filter's model is not linked to the Kalman filter's predicted error covariance matrix. Therefore, it is not possible to decide whether or not the target deviates from the motion model by observing the predicted error covariance matrix in Kalman filter. On the other hand, by using the measurement residual as the control parameter for adjusting the next update time interval, it is possible to decide whether or not the target deviates from the motion model by observing the measurement residual at each track update data point.

After the literature survey, it is seen that for adaptive update rate, there are several algorithms implemented on Alpha-Beta filter based on measurement residual approach. Moreover, there are many algorithms implemented on multiple model filters based on the trace of predicted error covariance approach. However, as far as we know, there is not a direct application of adaptive update rate algorithms on a single Kalman Filter.

In this study, the adaptive update rate algorithm approach, which is based on keeping the track update time interval by an amount that tends to maintain a constant measurement residual, developed in literature for Alpha-Beta filter is extended to the single Kalman filter with CV Model.

Firstly, the application of adaptive update time interval algorithm to Alpha-Beta filter is explained by giving the necessary theoretical bases. The adaptive update rate algorithms developed in literature for Alpha-Beta filter are introduced.

One of the most important contributions of this thesis is the application of adaptive update rate on Kalman filter. After presenting the application of adaptive update rate on Alpha-Beta filter, the bases for the application of adaptive update time interval algorithms to the Kalman filter equations is proposed and explained.

The two adaptive update time interval determination methods developed in literature for Alpha-Beta filter are used in this thesis throughout the simulations for the implementation of adaptive update rate on Kalman filter.

The first method, which is named as "Update Time Interval Determination Method-1" throughout this thesis, decides the value of the next update time interval from a discrete set by comparing the measurement residual at the current data point with the standard deviation of the measurement noise. Whereas, the second method, which is named as "Update Time Interval Determination Method-2" throughout this thesis, decides the value of the next update time interval after computing the next update time interval by the formula where the value of the next time interval is directly proportional with the previous update time interval and inversely proportional with the standard deviation of the measurement noise.

The main difference of these two update time interval determination methods is that while the first method uses the discrete predefined values for the next update time interval, the second method can use the direct continuous result of the formula for the next update time interval.

In this thesis, the simulations are mainly implemented by using Kalman filter since the aim of this study is to apply the approach developed in literature for adaptive update rate algorithms on Alpha-Beta filter to the Kalman filter. However, in order to demonstrate the application of adaptive update time interval algorithms developed for Alpha-Beta filter historically and to analyze the performance of adaptive update time interval algorithms on Alpha-Beta filter, firstly update time interval determination methods developed in literature are implemented on Alpha-Beta filter.

In order to investigate the properties of target tracking by using adaptive update rate, several scenarios are constructed for the target trajectory and the environment. Then, adaptive update time interval algorithms are applied to these scenarios with different simulation parameters.

Moreover, the implementation details of MC simulations for adaptive update rate case is introduced because the track updates may occur at different time instants for each MC run which accordingly causes a difficulty in applying MC simulations to variable update rate case.

In order to evaluate target tracking with adaptive update rate in terms of tracking accuracy and the resource allocation time, several simulations are realized for different scenarios by using Update Time Interval Determination Method-1 and Update Time Interval Determination Method-2 on Kalman filter. Also, the effect of the number of discrete levels in the set from which the next update time interval is selected in Update Time Interval Determination Method-1 is investigated by proposing a different set of discrete values for the next update time interval.

Besides, the adaptive update time interval case is compared with the fixed update time interval case for both the Alpha-Beta filter and the Kalman filter. For the Alpha-Beta filter case, it is observed that in order to obtain similar tracking accuracy with a fixed update time interval as in the adaptive update time interval case after the start of maneuver, the case of fixed update interval at a shorter update time interval than the average update time interval of adaptive case is necessary. On the other hand, it is observed that when the Kalman filter is used, the tracking accuracy is affected from both the level of the measurement noise and the magnitude of the process noise in the filter. Moreover, the process noise in the Kalman filter should be assumed to be relatively low so that in the case of adaptive update time interval, the update time interval determination methods which is based on the measurement residual approach, can detect the maneuver depending on the increase on the measurement residual throughout the trajectory.

The simulation results of Update Time Interval Determination Method-2 in which the update time intervals may take continuous values and the simulation results of Update Time Interval Determination Method-1 in which the update time interval is chosen from a discrete set of values are compared. After the evaluation of these simulation results, it is concluded that since the update time intervals are not forced to any fixed value in the continuous case as in the discrete case, the reduction in the average update time interval and consequently the increase in the number of updates occur in the continuous update time interval case. Furthermore, one of the noticeable results of the simulations is that there is a considerable improvement in the tracking performance when Update Time Interval Determination Method-2 is used as expected. The main reason for this is that in the continuous case, the next update time interval may take a continuous value between predefined maximum and minimum values, whereas in the discrete case the next update time interval is allowed to take the predefined discrete values.

To conclude, by using adaptive update rate on target tracking, the degradation of tracking can be prevented while allocating the time of the phased array radar economically for the tracking task. This is achieved by using longer update time intervals in the non-maneuvering segments to allocate radar resource economically and by using shorter update time intervals in the maneuvering segments not to lose the track.

6.2 FUTURE WORK

In this thesis, the main aim is to demonstrate the effectiveness of the use of adaptive update time interval in terms of maintaining tracking accuracy and resource allocation of radar. By considering the real-time applications, some of the assumptions which are used in the simulations of this thesis may not be realistic. However, it is believed that, this study can be easily extended to be used for real time applications by some additional effort.

In this thesis a single target is assumed on the environment, so one of the future topics of interest includes extension to a multi target environment, In addition to this, an environment with clutter case is another future topic of interest.

Besides, as a future study, verifications of adaptive update time interval algorithms should be performed with real trajectory and measurement data. Also, further modifications can be performed in the implemented adaptive update time interval algorithms. Using other filters such as EKF, IMM filter etc. to obtain performance figures in terms of tracking accuracy and economical radar resource allocation is also possible.

REFERENCES

[1] F. R. Castella, “An adaptive two-dimensional Kalman tracking filter”, IEEE Transactions on Aerospace and Electronic Systems, Page(s): 822-829, 1980

[2] S. A. Cohen, "Adaptive Variable Update Rate Algorithm for Tracking Targets with a Phased Array Radar", IEE Proceedings, Volume 133, No: 3, Page(s): 277-280, June 1986

[3] K. Nomoto, T. Kirimoto, and M. Kondo “An adaptive tracking filter controlling sampling frequency”, IEEE ICASSP, Page(s): 2571-2574, 1986

[4] L. A. Gardner and R. J. Mullen, “Constant gain tracker with variable frame time”, IEEE Transactions On Aerospace And Electronic Systems Volume 24, No: 4, Page(s): 322-326, July 1988

[5] D. J Wilkin, I. Harrison, and M. S Woolfson, “Target tracking algorithms for Phased array radar”, IEE PROCEEDINGS-F, Volume 138, No: 3, Page(s): 255-262, June 1991

[6] M. Munu, I. Harrison, D. Wilkin, and M. S. Woolfson, “Comparison of Adaptive Target-Tracking Algorithms for Phased-Array Radar”, IEE Proceedings-F, Volume 139, No: 5, Page(s): 336-342, October 1992

[7] P. R. Kumar, Pravin Varaiya, “Stochastic Systems: Estimation, Identification, Adaptive Control”, Prentice Hall, 1986.

[8] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, “Estimation with Applications to Tracking and Navigation”, John Wiley & Sons, NY, 2001

[9] K. V. Ramachandra, "Kalman Filtering Techniques for Radar Tracking", Marcel Dekker, New York, 2000

[10] Y. Bar-Shalom, T. E. Fortmann, “Tracking and Data Association”, Academic Press Inc., San Diego, California, 1988

[11] G. Welch, and G. Bishop, “An Introduction to the Kalman Filter”, Transaction of Department of Computer Science University of North Carolina, Chapel Hill, 2006

[12] Wikipedia, http://en.wikipedia.org/wiki/Kalman_Filter, November 15th 2009, last visited on December 2009

[13] Wikipedia, http://en.wikipedia.org/wiki/Alpha_beta_filter, November 12th 2009, last visited on December 2009

[14] L. Anderson, A. Bakshi, K. Elnahal, J. Kelly, D. Kim, V. Modi, A. Pantel, and J. Park, A. Schnayder, A. Sood, A. Weintraub, “Principles Of Radar Tracking”, Page(s): 6.1-6.20

[15] A. J. Orman, A. K. Shahani and A. R. Moore, “Modelling for the Control of A Complex Radar System”, Faculty of Mathematical Studies, University of Southampton, Volume 25, No: 3, Page(s): 239-249, 1998

[16] G. A. Watson, W.D. Blair, "Revisit Calculation and Waveform Control for a Multifunction Radar", Proceedings of the 32nd Conference on Decision and Control, Page(s): 456-461, December 1993

[17] P. W. Sarunic, “Adaptive Variable Update Rate Target Tracking for a Phased Array Radar”, IEEE International Radar Conference, Page(s): 317-322, 1995

[18] Akio Yanbe, Masayoshi Ito, Shingo Tsujimichi, and Yoshio Kosuge, “Target Tracking with Adaptive Sampling Intervals Using a Phased Array Radar”, Electronics and Communications in Japan, Part 1, Volume 85, No: 10, Page(s): 44-54, 2002

[19] Cheng Ting, He Zi-shu, Tang Ting, “An IMM-Based Adaptive-Update-Rate Target Tracking Algorithm for Phased-Array Radar”, Proceedings of 2007 International Symposium on Intelligent Signal Processing and Communication Systems, Page(s): 145-148, 2007

[20] H. Benoudnine, M. Keche, “Fast Adaptive Update Rate for Tracking a Maneuvering Target with a Phased Array Radar, Using IMM amd MRIMM Algorithms”, Journal of Applied Sciences, Page(s): 201-213, 2009

[21] H. Benoudnine, M. Keche, A. Ouamri and M. S. Woolfson, “Fast Adaptive Update Rate for Phased Array Radar Using IMM Target Tracking Algorithm”, IEEE International Symposium on Signal Processing and Information Technology, Page(s): 277-282, 2006

[22] X. R. Li, and V. P. Jilkov, “A Survey of Maneuvering Target Tracking: Dynamic Models”, Proceedings of SPIE Conference on Signal and Data Processing of Small Targets, Orlando, FL, USA, April 2000

[23] VPS Naidu and J. R. Raol, “Performance Evaluation of Interacting Multiple Model Kalman Filter”, IETE Journel of Education, Volume 49, No:3, Page(s): 95-108, 2008

[24] G. van Keuk, S. S. Blackman, “On Phased-Array Radar Tracking and Parameter Control” IEEE Transactions on Aerospace and Electronic Systems, Volume 29, No:1, Page(s): 186-194, 1993

[25] W. D. Blair, J.E. Gray, M.D. Boyd “Design Analysis for the Two-Stage Alpha, Beta, Gamma Estimator”, IEEE, 1991

[26] Chi-Man Liu Kuo-Guan Wu and Jer-Heh Sheu, “A Single Kalman Filtering Algorithm for Maneuvering Target Tracking”, IEEE, Page(s): 193-196, 1994

[27] D. Klamer, “Nonparametric Maneuver Detection In Kalman Filtering”, Naval Ocean Systems Center, Page(s): 544-548