Design and Implementations of a Mobile Target Tracking System Using FPGA

Dr. Laiyth M. Al-Rawi
Engineering College, University of Al-Mustansiriyah /Baghdad

Dr. Dhafer R. Zaghar
Engineering College, University of Al-Mustansiriyah /Baghdad
Dr. Ekhlas H. Karam
Engineering College, University of Al-Mustansiriyah /Baghdad

ABSTRACT

The design and implementations of a mobile target tracking system using field-programmable gate array (FPGA) are presented in this work. The idea of variable dimension (VD) filter which is used for tracking the nonmaneuver and maneuvering target is simplified and demonstrated by the FPGA implementations.

In general, the VD filter consist of two different Kalman filter dimensions and the fading memory detection scheme. In this tracking algorithm, the first Kalman filter is operates in its normal mode in the absence of any maneuvers, at same time, from the property of the innovation sequence and state estimates of this filter, the fading memory detector switch is used to determine that a maneuver is occurring, once a maneuver is detected the second augmented Kalman filter which uses a different state model is used to track the target in maneuvering motion course.

In this paper, the single Kalman filter is used to replace the second augmented filter of the VD algorithm, in this case when the maneuver is occur, the single filter is used in parallel with the first Kalman filter to track the target in maneuvering motion course without modifying the operation of the first Kalman filter. This step will simplified and reduce the calculation of the VD filter.The implementation for this system using FPGA will discuss in details, it will resulted to implement a low cost and mobile tracking system with high flexibility. Many of the general results presented in this paper are also useful for performance evaluation of this simplified variable dimension (SVD) filter algorithm as a compared with the VD filter algorithm.

Keywords : Target tracking, VD Kalman filter, FPGA implementation, Hardware design, Fading memory detection, Non-maneuvering and maneuvering.

<div dir="rtl">

تصميم وتنفيذ نظام تتبع للهدف المتحرك باستخدام الـFPGA

الخلاصة:

في هذا العمل تم عرض التصميم والتنفيذ لنظام تتبع الهدف المتحرك باستخدام بوابة مجموعة الميدان للبرمجة (FPGA). حيث ان فكرة البعد المتغير (VD) مرشح الذي يستخدم لتتبع لهدف

</div>

المناور وغير المناورة ا يتم تبسيطه وتعزيزة بواسطة تنفيذ ال FPGA. بشكل عام، مرشح VD يتكون من مرشحي كالمان مختلفين لأبعاد ونظام الكشف fading memory. في خوارزمية التتبع هذه ، مرشح كالمان ألاول يعمل في الوضع العادي في غياب أي مناورات، في الوقت نفسه، من خصائص التقديرات المتسلسله وتخمين الحالة لهذا المرشح، يتم استخدام الكاشف fading memory لتحديد أن مناورة حدثث، بمجرد الكشف عن مناورة فان مرشح كالمان المضاف الثاني والذي يستخدم نموذج مختلف يستعمل لتتبع الهدف خلال فترة المناورة.

في هذا البحث, تم استخدام كالمان فلتر المفرد ليحل محل المرشح الثاني المضاف في خوارزمية ال VD، في هذه الحالة عند حدوث المناورة ، يتم استخدام مرشح كالمان المفرد في نفس الوقت مع مرشح كالمان ألاول لتتبع الهدف خلال فترة المناورة دون تعديل في عمل مرشح كالمان الاول. وهذه الخطوة سوف تبسط وتقلل من حسابات مرشح VD. التنفيذ لهذا النظام باستخدام FPGA سوف يناقش بالتفاصيل، وسوف يؤدي إلى تنفيذ نظام تتبع للهدف المتحرك منخفض التكاليف و ذات مرونة عالية. العديد من النتائج العامة الواردة في هذا البحث هي أيضا مفيدة لتقييم أداء خوارزمية التصفية (SVD) هذه مقارنتا مع خوارزمية التصفية (VD).

INTRODUCTION

Target tracking is a significant problem in field of object recognition and tracking. This problem is more challenging due to involvement of compute intensive loops asserted in the process of recognition and tracking of object. In most target detection algorithms difficult thing is to optimize the algorithm in time and space simultaneously.

The non-maneuvering and maneuvering conditions are usually gives a good accuracy in the tracking systems. Both non-maneuvering and maneuvering conditions are usually existed with together in a radar tracking system [2]. The use of the nonmaneuver-base filters (NMFs) and maneuver-base filters (MFs) corresponding to the actual target motion is complicated by the lack of knowledge of both the maneuver magnitude and the maneuver onset and ending times [3]. Several approaches have been used to solve this problem [3-12] and one from the most using solves is the variable dimensions (VD) filter [1].

The VD filter is a particular type of innovations based adaptive filter, which consist of both the NMF and the MF; the intermittent use of either filter is determined by a decision mechanism [3]. If This filter applied with spherical coordinate instate of Cartesian coordinate, then the NMF is a two state Kalman filter which is operate in the normal condition (when the target is not under maneuver condition), at same time this filter provide the detector (the fading memory detector) with the innovation sequence which help to detect the existence of the maneuver, once a maneuver is detected the second augmented Kalman (three state) filter which uses a different state model is used to track the target in maneuvering motion course (see Fig.(2)). The VDF may provide rather good performance with efficient computation.

Since target tracking filters in real-time requires dedicated hardware to meet demanding time requirements. Modern field programmable gate arrays (FPGAs) include the resources that are needed to design such efficient filtering structures. This media will give a portable, high speed and low cost system.

In this article, the VDF is simplified by use a single Kalman filter and the modification to the two state Kalman filter to track the maneuvering target instead of the augmented three state filter then the hardware design implementation for the simplified VD filter (SVDF) by the field-programmable gate array (FPGA) is presented. This article will shows the practical steps to solve of the design problems

for this system under FPGA to satisfy a low cost and high speed portable tricking system.

This paper is organized as follows. The next section explains the simplified variable dimension filter, while section 3 give the reduce SVDF equations, section 4 that outlines the hardware design for the SVDF by FPGA. The system cost and delay calculations are presented in section 5 and the implementation of the system using FPGA presented in section 6. In section 7 some comparative simulation results are presented to highlight the performance for the VD and SVD filters and improvement obtained by using these approaches. The sections 8 and 9 are the discussions and conclusions for the results of this paper respectively.

The Simplified Variable Dimension (SVD) Filter

In this paper, the suggested SVD filter in Fig.(1) is consist of two Kalman estimator with fading memory detection scheme (FMD), the first one is two state Kalman filter used to estimate the position and the velocity when the target in nonmaneuveing course, from the property of the innovation sequence and state estimates of this filter, the FMD switch can be worked, while the second filter (the acceleration filter) which depended on the residual sequence of the first filter, is single Kalman filter and it is used in parallel of the first Kalman filter to estimate the acceleration and the new estimate to the position and velocity of the maneuvered targets without modifying the operation of the first Kalman filter.

The VD and SVD filter that used in this paper is applied with spherical coordinate instate of Cartesian coordinate. The following subsection gives the details of each part in Fig.(1). The operations details the simulation examples for the suggested tracking algorithms are given in the following subsections.

Non-Maneuvering (Two state) Filter

The target state is defined in the measurement vector (such as range, bearing and elevation in radar system) direction. Then, the tracking filter may work separately in each direction approximately [5]. One single direction (like range) operation is described in the following.
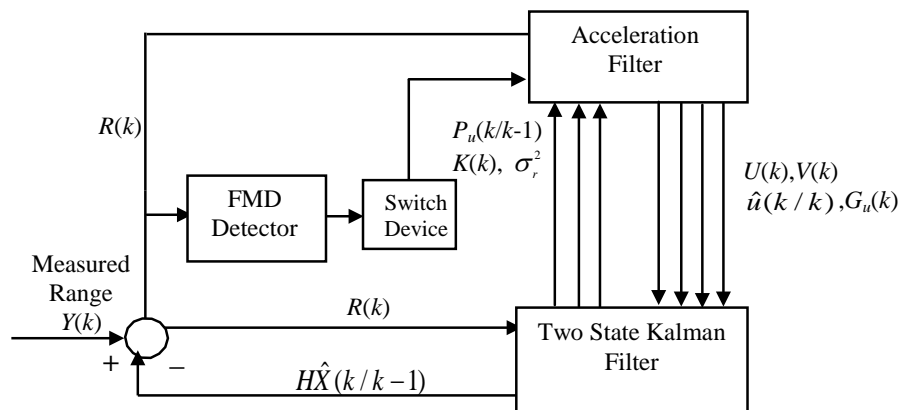

Figure.(1):the block diagram for the SVD filter.

When the target is moving in non-manewering course, the target motion and the measurement can be modeled as a constant velocity model by a state with two

dimensional vector $(X = [r \ v]^T)$ with some process noise that models slight changes in the velocity, the target process model discredited over time interval of length T is [6].

$$X(k+1) = \Phi X(k) + GW(k) \qquad \qquad ...(1)$$
$$Y(k) = HX(k) + V(k) \qquad \qquad ...(2)$$

$$\Phi = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} T/2 \\ 1 \end{bmatrix}, \quad H = [1 \ 0]$$

Where

Φ :   transition matrix.

X(k):   the state vector consisting of the radial range and range rate components denoted by r(k) and v(k) respectively.

Y(k):   measurement data, the observation (the range measurement) from the radar system.

H   :   the observation matrix.

W(k):   random acceleration process.

V(k):   the measurement noise.

W(k) is modeled as a white Gaussian noise with zero-mean and covariance matrix Q(k) defined as:

$$E\{W(k)W^T(j)\} = Q(k)\delta(k-j) \qquad \qquad ...(3)$$

Where $\delta(.)$ is the Kroneker-delta function.

The covariance matrix Q (k) is defined by [13],

$$Q(k) = q^T \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} = \sigma_m^2 T \begin{bmatrix} T^2/3 & T/2 \\ T/2 & 1 \end{bmatrix} \qquad \qquad ...(4)$$

where q is the spectral density of the continuous white noise change in acceleration process and σ_m^2 is the variance of the change in acceleration noise.

The measurement noise $V(k)$ is modeled as an independent white Gaussian process with zero-mean and variance (the error of the measured range) σ_r^2 defined as;

$$E\{V(k)V^T(k)\} = \sigma_r^2 \delta(k-j) \qquad \qquad ...(5)$$

The noise process W(k) & $V(k)$ are uncorrected (i.e. $E\{W(k)V(j)\} = 0$ for all j and k) .

The two state Kalman filter equations can be described by the following state space equations:

Filter state prediction:

$$\hat{X}(k/k-1) = \Phi\hat{X}(k-1/k-1) \qquad \qquad ...(6)$$

Error covariance prediction:

$$P(k/k-1) = \Phi P(k-1/k-1)\Phi^T + Q(k) \qquad \qquad ...(7)$$

Filter gain:

$$K(k) = P(k/k-1)H^T [HP(k/k-1)H^T + \sigma_r^2]^{-1} \qquad \qquad ...(8)$$

Filter state update:

$$\hat{X}(k/k) = \hat{X}(k/k-1) + K(k)[Y(k) - HX(k/k-1)] \qquad \dots(9)$$

Error covariance update:

$$P(k/k) = [I - K(k)H]P(k/k-1) \qquad \dots(10)$$

The two state Kalman is initialized as

$$\hat{r}(0/0) = y(0)$$

and

$$\hat{v}(0/0) = [y(1) - y(0)]/T$$

where y(0) and y(1) are, respectively, the first and second received sensor measurements.

While initial estimation error covariance for this coordinate is then:

$$p(0/0) = \begin{bmatrix} \sigma_r^2 & \sigma_r^2/T \\ \sigma_r^2/T & 2\sigma_r^2/T^2 \end{bmatrix}$$

2.2. Fading Memory Detection (FMD) Scheme

A simple fading memory average of the innovations from the two state Kalman filter is computed as follows [1]:

$$L(k) = \alpha_F L(k-1) + d(k) \qquad \dots (11)$$

with $0 < \alpha_F < 1$ and

$$d(k) = R^T(k)N^{-1}(k)R(k) \qquad \dots (12)$$

R(k) is the "innovation process" of Yk),

$$R(k) = Y(k) - H\hat{X}(k/k-1) \qquad \dots (13)$$

which is zero mean and variance given by;

$$N(k) = HP(k/k-1)H^T + \sigma_r^2 \qquad \dots (14)$$

Since d(k) is under the Gaussian assumption chi-squared distribution with n_y (dimension of measurement) degrees of freedom. The effective window length (M) of the fading memory average over which the presence of a maneuver is [1];

$$M = \frac{1}{1 - \alpha_F} \qquad \dots (15)$$

Accept the hypothesis that a maneuver is taking place if L(k) exceeds a certain threshold [1].

Here in this paper two threshold are used, λ_1 is the first oneand λ_2 is the second threshold.

2.3. The Second (Acceleration) Kalman Filter

When a maneuver occurs, an acceleration item Bu is applied in Eq.(1) such that the target process model in Eq.(1) Becomes as below [5,6]:

$$X(k/k) = \Phi X(k) + GW(k) + Bu(k) \qquad \dots (16)$$

Where B is acceleration transition vector defined by[6],

$$B = [T^2/2 \quad T]^T \qquad \dots(17)$$

u(k) is the unknown acceleration term, this term can be deemed as either a random signal or a deterministic signal. Here we consider it as a random signal that influences the system dynamics. The dynamics governing this random maneuver term is [14].

$$u(k) = u(k-1) + G_u W_u(k) \qquad \qquad \dots (18)$$

The $W_u(k)$ is uncorrelated white Gaussian sequence with zero mean and variance given by [6]:

$$E\{W_u(k)W_u^T(j)\} = Q_u(k) \qquad \qquad \dots (19)$$

The state measurement model which define in Eq.(2) is become,

$$Y(k) = HX(k) + Cu(k) + V(k) \qquad \qquad \dots (20)$$

With $C = T^2 \big/ 2$, while H & V(k) is define previously.

The Equations for the acceleration filter steps are [6]:
Filter acceleration estimate:

$$\hat{u}(k/k) = \hat{u}(k/k-1) + k_u[R(k) - S(k)\hat{u}(k/k-1)] \qquad \qquad \dots (21)$$

Acceleration error variance prediction:

$$P_u(k/k-1) = P_u(k-1/k-1) + GQ_u(k)G^T . \qquad \qquad \dots (22)$$

Acceleration filter gain:

$$K_u(k) = P_u(k/k-1)S(k)^T[S(k)P_u(k/k-1)S(k)^T + HP(k/k-1)H^T + \sigma_r^2]^{-1} \qquad \dots (23)$$

Acceleration error variance update:

$$P_u(k/k-1) = [1 - K_u(k)S(k)]P_u(k/k-1) \qquad \qquad \dots (24)$$

Sensitivity matrices:

$$S(k) = HU(k) + C \qquad \qquad \dots (25)$$

$$U(k) = \Phi V(k-1) + B \qquad \qquad \dots (26)$$

$$V(k) = [1 - K(k)H]U(k) - K(k)C \qquad \qquad \dots (27)$$

The acceleration Kalman filter is initialized as

$$\hat{u}(0/0) = 0$$

and

$$P_u(0/0) = [.01]$$

While initial values for the sensitivity matrices U(k) and V(k) are assumed zero.

2.4. The Relation Between the First and Second Filters

The estimate of u is used to compensate the output of the two state Kalman filter by the following equation [6, 15];

$$\hat{X}_n(k/k) = \hat{X}(k/k) + V(k)\hat{u}(k/k) \qquad \qquad \dots (28)$$

$$P_n(k/k) = P(k/k) + V(k)P_u(k/k)V^T(k) \qquad \qquad \dots (29)$$

3. Reduce the Equations of the SVD Filter

The first step in reduce the equations in each part of the SVD filter is to refer to each of the present and the previous error covariance, filter gain, filter state element, the sensitivity matrices, the new filter state element, filter acceleration as

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}, \ K(k) = \begin{bmatrix} K_1 \\ K_2 \end{bmatrix}, \ \hat{X} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}, \ U = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}, \ V = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \text{ and } \ \hat{u} = u \ \text{ with}$$

these equations. Hence, the first two state Kalman filter equations in the SVD can be reduced to only three equations by substitute the predicated state Eq.(6) in Eq.(9) and the predicted error covariance Eq.(7) in Eq.(10) after redress each of element of $Q(k)$ in Eq.(7) and the element of H matrix in Eq.(8), Eq.(9), and Eq.(10), therefore the two state Kalman equations becomes as below;

Filter gain:
$$\begin{bmatrix} K_1 \\ K_2 \end{bmatrix} = \frac{1}{P_{11} + \sigma_r^2} \begin{bmatrix} (q*q_{11}) + [P_{11} + TP_{21} + T(P_{12} + TP_{22})] \\ (q*q_{21}) + [P_{21} + TP_{22}] \end{bmatrix} \quad \dots(30)$$

Filter state:
$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} X_1 + TX_2 + K_1*(Y - (X_1 + TX_2)) \\ X_2 + K_2*(Y - (X_1 + TX_2)) \end{bmatrix} \quad \dots(31)$$

Error covariance:
$$\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} (1-K_1)*\{(q*q_{11}) + & (1-K_1)*\{(q*q_{12}) + \\ [P_{11} + TP_{21} + T(P_{12} + TP_{22})]\} & [P_{12} + TP_{22}]\} \\ \\ -K_2*\{(q*q_{11}) + & \\ [P_{11} + TP_{21} + T(P_{12} + TP_{22})]\} & -K_2*\{(q*q_{12}) + (P_{12} + TP_{22})\} \\ + (q*q_{12}) + (P_{21} + TP_{22}) & + [(q*q_{22}) + P_{22}] \end{bmatrix} \dots(32)$$

In other word, the FMD equations can be summarized in the following single equation:
$$L = (\alpha_F + L) + \frac{(Y - X_1)*(Y - X_1)}{(P_{11} + \sigma_r^2)} \quad \dots(33)$$

The initial value for $L(k)=0$, while the second Kalman filter and the new estimation state equations can be reduced to the following equations after remove the C from Eq.(25) and the term($C*K(k)$) from Eq.(27) in addition to neglected the Eq.(29) from these calculations:

Filter acceleration estimate:
$$u = u + k_u[R - (U_1*u)] \quad \dots(34)$$
Acceleration filter gain:
$$K_u(k) = \frac{(P_u*U_1)}{[(U_1*P_u*U_1) + (P_{11} + \sigma_r^2)]} \quad \dots(35)$$
Acceleration error variance:
$$P_u = [1 - (K_u*U_1)]*(P_u + Q_u) \quad \dots(36)$$
$$Q_u = cons\tan t \ value$$
Sensitivity matrices:

$$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} V_1 + (T * V_2) + (0.5 * T^2) \\ V_2 + T \end{bmatrix} \quad ...(37)$$

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} (1 - K_1) * U_1 \\ (-K_2 * U_1) + U_2 \end{bmatrix} \quad ...(38)$$

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} X_1 + (V_1 * u) \\ X_2 + (V_2 * u) \end{bmatrix} \quad ...(39)$$

4. The Design of variable dimension (VD) filter using FPGA

4.1 Problem Definition

      The goal of this work is implement a portable tracking system with low price. The first we will think to use FPGA devices because it is a portable and low price.
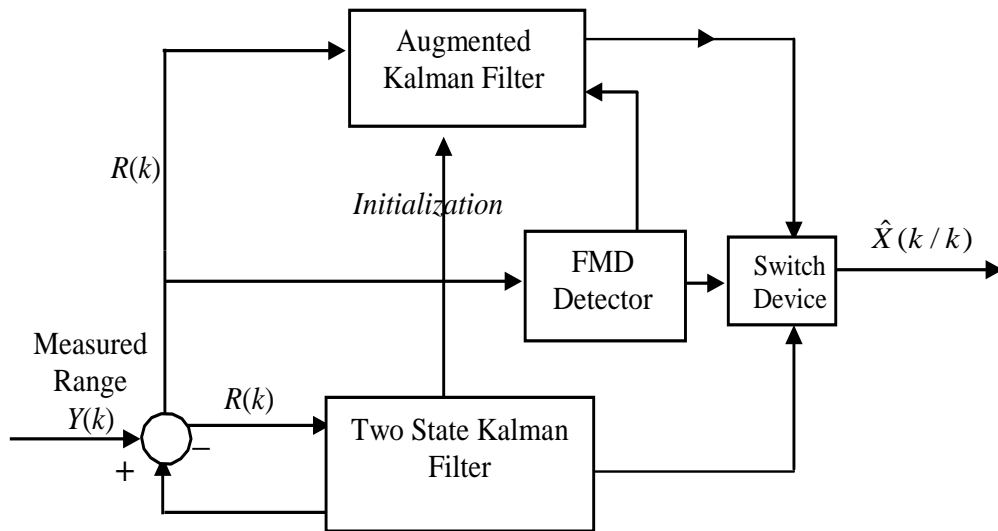

Figure.(2):the block diagram for the variable dimension (VD) filter.

The deep analysis for the tracking system in Fig.(2) show that this system required to about 35,532 cells as in table (1). Let an approximation cost for the basic elements [] in this system as 16-bits are adder, multiplier, divider and other elements (such as comparator, MUX).

Table (1a): The requirements of the system in Fig.(2).

Component	Adder	Multiplier	Divider	Other elements
2D filter	33	14	1	20
3D filter	65	27	1	35
Other system components	3	2	0	67
Total system	101	43	2	122

Table (1b): The approximation cost of the component in FPGA cells.

Component cost	Adder	Multiplier	Divider	Other elements
Cost/ cells	40	600	1000	40

Table (1c): The approximation cost of the system in FPGA in cells.

Component cost	Adders cost	Multipliers cost	Dividers cost	Other elements cost	Total cost	Rational cost
2D filter	132	8400	1000	800	10332	29%
3D filter	2600	16200	1000	1400	21200	60%
Other system components	120	1200	0	2680	4000	11%
Total system	4040	25800	2000	4880	35532	100%

The basic analysis for this system shows that. The system is huge and slow while the FPGA chips are fast and limited. That is like move a mountain from rocks to 1000 meter using small fly.

There is no direct solution for this problem, but it required to a novel and complex solution. The proposed solution has three levels:
1- The mathematics and physical level. This level will reduce the mathematics equations for the system with save its physical specifications.
2- The simulation level. The main target for this level is reducing the databus for minimum as possible with acceptable additional error.
3- The implementation level. In this level we will select low cost digital components to reduce the total cost of the system.

The Mathematics and Physical Level
This section discusses how the total system complexity can be reducing with same output result by the simplified variable dimension (SVD) filter that shown in Fig. (1). This solution will eliminate the 3D filter that has the great part from the cost, so it in result will reduce the total cost as in table 2.

Table (2): The approximation cost of the system in figure (1).

Component	Adder	Multiplier	Divider	Other elements	FPGA cost/ cells	Rational cost
2D filter	33	14	1	20	10332	49%
Acceleration filter	7	6	0	13	4400	21%
FMD detector	3	2	1	0	2320	11%
Other system components	3	2	0	67	4000	19%
Total system	46	24	2	100	21052	100%

The new cost for this level is less than 60% from the cost of the system in table (1c). That is mean this level reduce the cost of this system without any reduction in its efficiency or in its accuracy.

The Simulation Level

This level fox to reduce the databus for the system but the reduction for databus will increase the error in the output and in result reduce the accuracy of the output results. The additional error in this level must remain less than the effective values.

This level depends on the following procedure:

1- Calculate the values of the variables and constants in the system and detect the output sensitivity for these values.
2- Give initial databus for each value.
3- Simulate the output for new databuses.
4- Use try and error to find acceptable results with minimum databus width.
5- The databuses will limited to use three values 8, 12 and 16 bits.

The result for this level are summarize in table (3).

Table (3): The results of the simulation level with the maximum error.

Component	Variable	Range	Databus/ bits	Maximum error
system	X_1	0 to $5*10^6$ meter	16	38
	X_2	0 to 10^4 m/sec	16	38
	P	0 to 10^4	12	1.22
	K	0 to 1	8	0.002
	T	Scalar value 1 sec.	--	
	σ_r	Scalar value like 100 meter	8	0.2
	σ_{m1}	Scalar value like 5 m/ sec^2	8	0.2
	σ_{m2}	Scalar value like 20 m/ sec^2	8	0.2
2D filter	L	0 to 7,000	12	9
	N	0 to $7*10^4$ m	12	9
	R	0 to 6,000 m	12	9
	d	0 to $2*10^4$	12	9
	λ_1	Scalar value like7.5.	8	0.2
	λ_2	Scalar value like 11.04	8	0.2
	α_F	Scalar value can be from 0 to 1	8	0.2
FMD	V_1	0 to 21	12	0.014
	V_2	0 to 7	12	0.014
	U_1	0 to 28	12	0.014
	U_2	0 to 8	12	0.014
	P_u	0 to 110	12	0.014
	K_u	0 to 0.05	8	0.0001
second filter	u	0 to 60 m/sec^2	8	0.1

The Implementation Level
        This level work to reduce the cost of each element in the system separately to reduce the total cost. This level will depend on the low processing speed of the system (1 sec) and the high speed of the FPGA (few nsec) to implement a low cost and low speed components and circuits in view of FPGA but this components are suitable speeds for the system. This level will discuss a four main component called basic component as in sub-section 5.3.

The System Cost and Delay Calculations
The Cost Calculations of the Components
        The calculation of the cost is simple and the hand calculations are similar to the result of the hardware implementation in ISE 4.1i with error is +2% in maximum. The additional error comes from the long paths in the routing implementation that needs to connection points. []
The cost of the small circuit is very simple as in []. While the total cost of the complex system is the sum of the costs of the components in it.
The Delay Calculations of the Components
        The calculations of the delays of the components are very complex. The hand calculations like the result of the hardware implementation in ISE 4.1i with error reach to about ±20%. The high error comes from the paths in the long routing implementation.
        The delay of the small circuit is simple calculations as in []. While the delay of the complex system require to draw the circuit and estimate the longest path, then sum the maximum delays of the cascade connection components to find the maximum delay (longest asynchronous path) in the system. The total delay will calculate as in Eq.(40).

$$total\ delay = \ L * t \pm Et \qquad\qquad\qquad … (40)$$

When L :is the longest path measured in cell delay
        t :the cell delay, it is approximate value as example cell delay for XC2V1000 is 1.5 nsec.
        Et: is the additional time delay add because the long paths in the system. This value is not fixed and increase with complexity of the system.

The Cost and Delay Calculations of the Basic Components
The basic components that will use in this system are:

1- Adder: The basic used adder is a 4-bit ripple with carry lookahead as in Fig.(3) that required to 10c with 4d for the output and 2d for the last carry. The 8-bit adder implements using two units from the 4-bit adder, the 8-bit adder requires to 18c and 6d. The same way will use for the 12-bit adder that require to 28c and 8d, and the 16-bit adder that require to 38c and 10d.
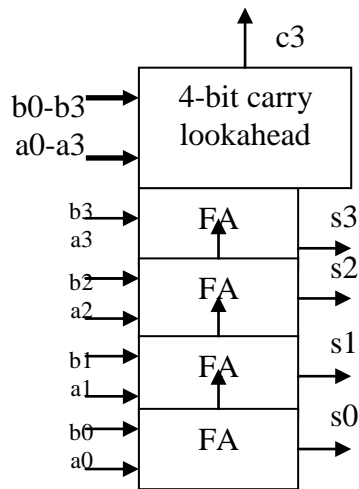
Figure.(3): The details of the implementation for the 4-bit adder.

2- Multiplier: The basic used multiplier is a parallel shift and adds 8x8-bits multiplier (8-bit mult. for simplicity) that required to 200c and 22d. The 12-bit mult. require to 466c and 38d, it implements using same techniques for 8-bit mult.. The same way will use for the 16-bit mult. require to 860c and 57d.

3- Divider: The divider will implement directly as a LUT that require to $2^{(n-4)}$c and 1d.

4- Other components: the other components are n-bit complement that required to n cell with 1d, n-bit switch that required to n cell with 1d and n-bit registers that required to n cell with 0d.

6. The Implementation Details

The details of the design of the system will repeat the system in Fig.(1) in hierarchical form as in Fig.(4).
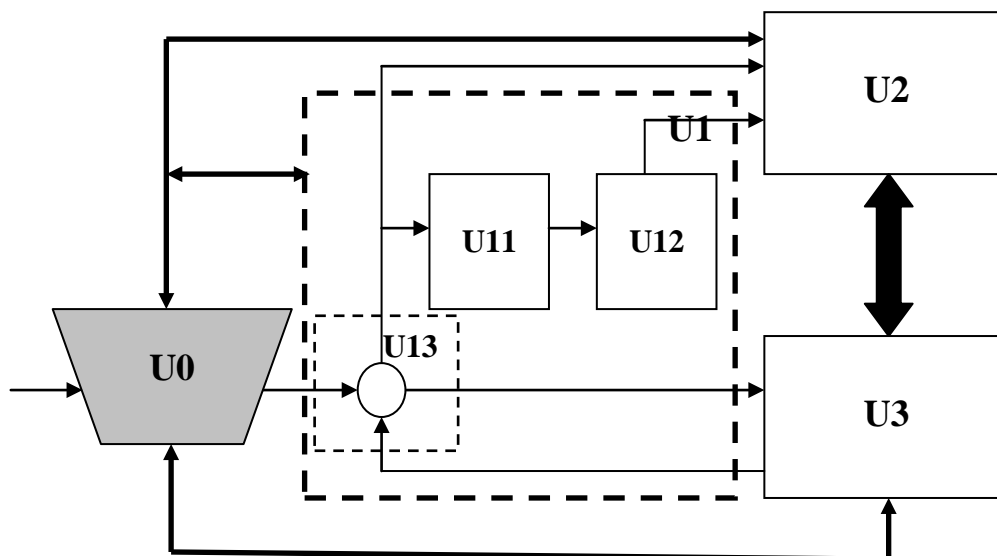

**Figure. (4): The blocks of the implementation for the system in
Fig.(1).**

This figure has 4 unit assigned as U0-to-U3. The units U1-to-U3 are like the blocks units in Fig.(1), while the unit U0 is the control and initial values and it service and control for all other units in the system.

6.1. The Cost and the Delay of the Unit U1
        This unit is the service unit for the system; it has three sub-units connected as in Fig.(3) with the following details:

1- U11 is the FMD block shown in Fig.(2). The design details show it has six sub-units as in table (4) and connected as in Fig.(5). This sub-unit required to 4914 cells with total delay is 132 cell delays.
2- U12 is 4 parallel groups of switches each one has 12-bits that required to 12 cells and 1 d. in result this sub-unit require to 48 cell and 1d delay.
3- U13 is a 16-bit adder with total cost is 38 cells and 10d delay.
        The unit U1 has a total cost is 5012 cells and over all delay is 143 cell delays.

Table (4): The simplified equations, the costs in cells and the delays in cell delay of the sub-unit U11.

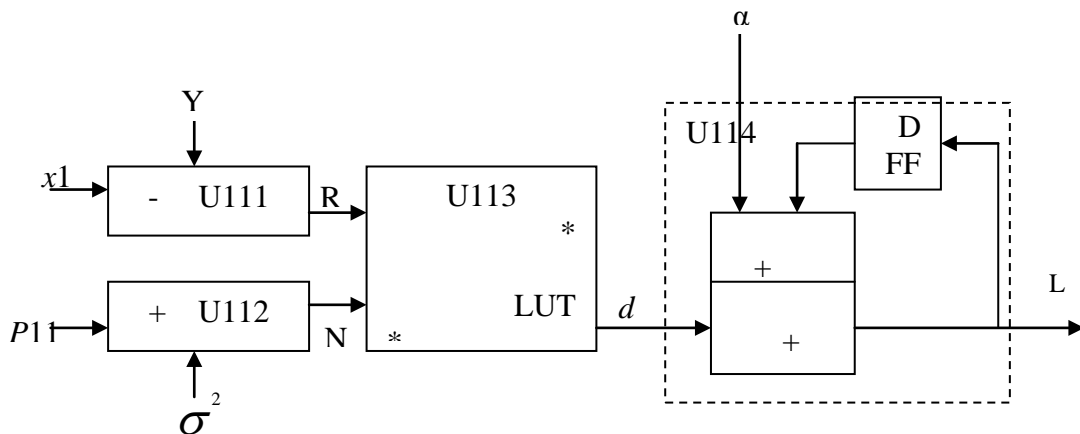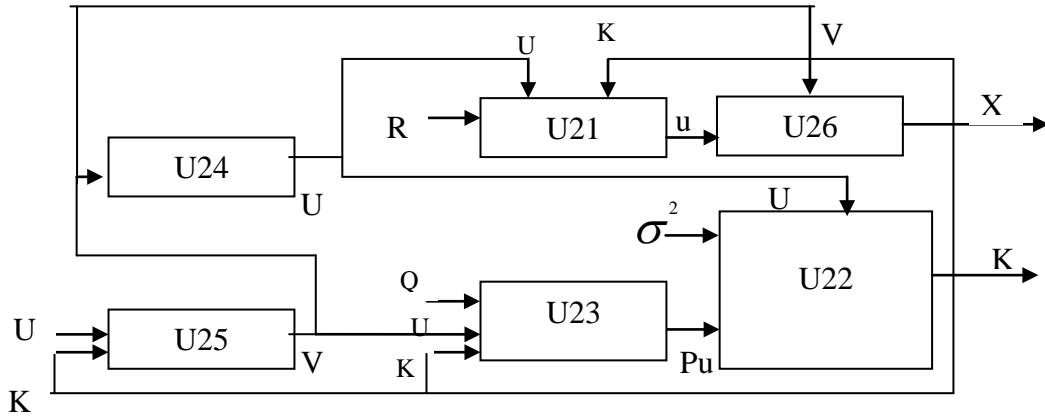| Sub-unit	Equation	Components	Total cost	Total delay
U111	$R = Y - X_1$	16-bit adder	38	10
U112	$N = P_{11} + \sigma_r^2$	12-bit adder	28	8
U113	$d = R * R * 1/N$	2x16-bit mult., 12-bit divider	860+3072+860 =4792	57+57 =114
U114	$L(k) = \alpha_F + L(k-1) + d$	2x12-bit adder, 12-bit registers	28+28+12 =68	8
U11	FMD $$L = (\alpha_F + L) + \frac{(Y - X_1) * (Y - X_1)}{(P_{11} + \sigma_r^2)}$$	-----	4926	132

Figure.(5): The details connections of the unit U11.

The Cost and the Delay of the Unit U2
This unit is the acceleration filter in the system; it has three sub-units as follows (Fig. (6) Shows the details connections of the unit U2):

Figur.(6): The details connections of the unit U2.

Table (5): The simplified equations, the costs in cells and the delays in cell delay of the unit U2.

Sub-unit	Equation	Components	Total cost	Total delay
U21	$u = u + k_u[R - (U_1 * u)]$	12-bit mult., 8-bit mult., 2x12-bit adder, 12-bit registers	466+200+2*28+12=734	38+8+22 =68
U22	$T = P_u * U_1$ $K_u = \dfrac{T}{[(U_1 * T) + (P_{11} + \sigma_r^2)]}$	2x12-bit adder, 2x12-bit mult., 12-bit divider	2*28+2*466+256=1244	38+38+8+1=85
U23	$P_u = [1 - (K_u * U_1)] * (P_u +$	2x12-bit mult., 12-bit adder, 12-bit complement	2*466+2*28+12=1000	38+38 = 76
U24	$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} V_1 + V_2 + 0.5 \\ V_2 + 1 \end{bmatrix}$	3x12-bit adder	3*28=84	28
U25	$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} U_1 * (1 - K_1) \\ U_2 - K_2 * U_1 \end{bmatrix}$	2x12-bit mult, 12-bit adder, 12-bit complement	2*466+2*28+12=1000	38
U26	$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 + (V_1 * u) \\ x_2 + (V_2 * u) \end{bmatrix}$	2x12-bit mult., 2x16-bit adder	2*466+2*28=988	38+38 =76
U2	Acceleration filter	-----	5050	199

 The unit U2 has a total cost is 5050 cells and over all delay is 199 cell delays.
The Cost and the Delay of the Unit U3.
        This unit is the two states filter in the system; it has seven sub-units as follows (Fig. (7) Shows the details connections of the unit U3):
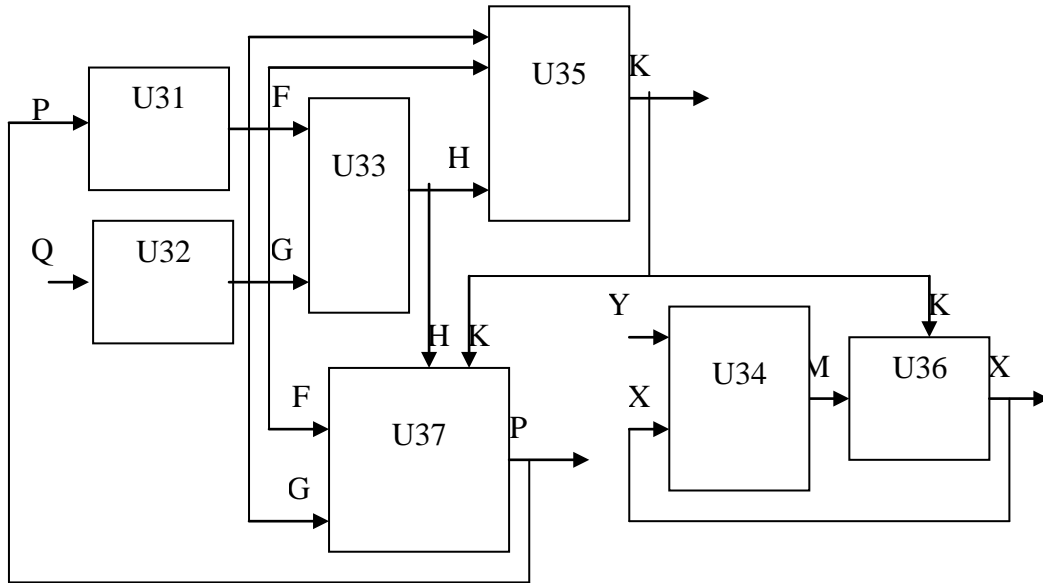

Figure.(7): The details connections of the unit U3.

Table (6): The simplified equations, the costs in cells and the delays in cell delay of the unit U3.

Sub-unit		Equation	Components	Total cost	Total delay
U31	U311	$f1 = P_{11} + P_{21} + P_{12} + P_{22}$	3x12-bit adder	3*28=84	8+8=16
	U312	$f2 = P_{21} + P_{22}$	12-bit adder	28	8
	U313	$f3 = P_{12} + P_{22}$	12-bit adder	28	8
U32	U321	$g1 = q * q_{11}$	12-bit mult.	466	38
	U322	$g2 = q * q_{12}$	12-bit mult.	466	38
	U323	$g3 = q * q_{21}$	12-bit mult.	466	38
	U324	$g4 = q * q_{22}$	12-bit mult.	466	38
U33	U331	$h1 = f1 + g1$	12-bit adder	28	8+38^=46
	U332	$h2 = f3 + g2$	12-bit adder	28	8+38^=46
U34	U341	$m1 = x_1 + x_2$	16-bit adder	28	8
	U342	$m2 = Y - m1$	16-bit adder	28	8+8^=16

Continuation Table(6)

U35	$\begin{bmatrix} K_1 \\ K_2 \end{bmatrix} = \dfrac{1}{P_{11} + \sigma_r^2} \begin{bmatrix} h1 \\ g3 + f2 \end{bmatrix}$	2x12-bit mult., 12-bit divider, 2x12-bit adder	2*466+256 +2*28 =1244	38+46^=84
U36	$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} m1 + K_1 * m2 \\ x_2 + K_2 * m2 \end{bmatrix}$	2x16-bit mult., 2x16-bit adder	2*860+2*38 = 1796	38+16^+8 = 62
U37	$\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} (1-K_1)*h1 & (1-K_1)*h2 \\ -K_2*h1 & -K_2*h2 \\ +g2+f2 & +g4+P_{22} \end{bmatrix}$	4x12-bit mult., 4x12-bit adder	4*466+4*28 =1976	46^+38+8 = 92
U3	Two states filter	-----	7598	146

^ Additional delay comes from the previous unit(s)

Note: The variables in table (6) is grouped as capital later (example F=f1, f2 and f3).
The unit U2 has a total cost is 7598 cells and over all delay is 146 cell delays.
The Cost and the Delay of the Unit U0

This unit is the control and service filter in the system; it has three sub-units as follows:

1- U01- is the constant memory registers. This sub-unit has seven sub-units, the first six sub-units are an 8-bit registers, while the U017 is the interface circuit for the constants. U017 receive the values of the constants serially then sent them to the suitable register. This sub-unit receives the control instruction from the control unit U03. The details of the cost and delay for U01 are in table (7).

Table (7): The simplified equations, the costs in cells and the delays in cell delay of the sub-unit U01.

Sub-unit	Equation- constant	Components	Total cost	Total delay
U011	σ_r	8-bit register	8	0
U012	σ_{m1}	8-bit register	8	0
U013	σ_{m2}	8-bit register	8	0
U014	λ_1	8-bit register	8	0
U015	λ_2	8-bit register	8	0
U016	α_F	8-bit register	8	0
U017	Input interface for constants	8-bit register, 8-bit decoder	8+8=16	
U01		-----	54	0

2- U02- is the initial values part. This part has three sub-units U021-U023 that generate the initial values from the equations in table (8). Each one from these sub-units has MUXs that has two groups of inputs (each one is 8 or 12 bits) with one output as in Fig.(8). The first input is the initial value while the other is the value of the system after starting. The sub-unit work as switching with a starting order input that received from the control unit U03. The details of the cost and delay for U02 are in table (8).

3- U03- is the control part. This part is very simple because the system in general is an auto data stream. As example the data pass from U2 to U3 automatically and no need to any control signal. The sub-unit U03 design to controlled on:

i- The input of the initial and constant values in U01 and U02.

ii- The reset signal for the system.

iii- The switch signal for U12.

iv- Work as interface with the system as optional job. This part will neglect in this paper because it depend on the properties of the total system.

Hence, the delay of U0 will neglect because it delayed the system at start point then it withdraw for the next time of work. The cost of the unit U0 is about 300 cells.

Table (8): The simplified equations, the costs in cells and the delays in cell delay of the sub-unit U02.

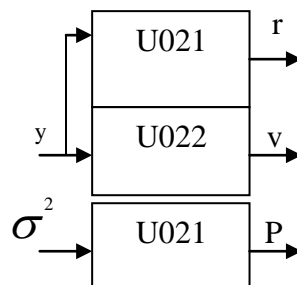| Sub-unit	Equation	Components	Total cost	Total delay
U021	$r(0/0) = y(0)$	12-bit register, 12-bit MUX	12+12=24	0
U022	$v(0/0) = [y(1) - y(0)]$	12-bit adder, 12-bit register, 12-bit MUX	28+12+12 =52	8
U023	$p(0/0) = \sigma_r^2 \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$	8-bit register, 8-bit MUX	8+8=16	1
U02		-----	92	0

Figure.(8): The block diagram of the sub-unit U02.

The Total Cost and the Delay of the System

The cost of the system is the direct sum of the costs of the four units U0-U3 as in table (9). The practical costs in ISE 4.1i has some different in the calculation of the costs because some small internally details not discuss in this work but it put in the design programs. Examples of these details are the neglected bitsrein the last adder in multiplier, neglected LSB in the adder, merge the n-bit register with the n-bit MUX in U02 and the neglect the details of U03.

Table (9): The final costs in cells and the delays in cell delay of the total system and its units.

Unit	Function	Calculated Cost/ cell	Cost*/ cell	Delay/ cell delay	Delay*/ nsec
U0	Control and inertial	300	NA	1	NA
U1	FDM and switching	5012	4942	143	219
U2	Acceleration filter	5050	4868	199	266
U3	Two states filter	7598	7476	146	198
System	Target tracking	17960	17286	343	457

*Under software implementation in ISE 4.1i

Simulation and Comparison

The performance of the SVD filter is tested and compared with VD filter using the two maneuver scenarios that shown in Fig.(9)[16]. with parameters for the two scenarios and the first two state Kalman filter in SVD and VD filters are given by the following:

- Sampling interval T:1 sec.
- The standard deviation for the measurement noise σ_r: 100 m
- The standard deviation of the target noise disturbance σ_{m1}: 5m/sec^2 and σ_{m2}: 20m/sec^2.
- The constant target radial velocity v: 250 m/sec.
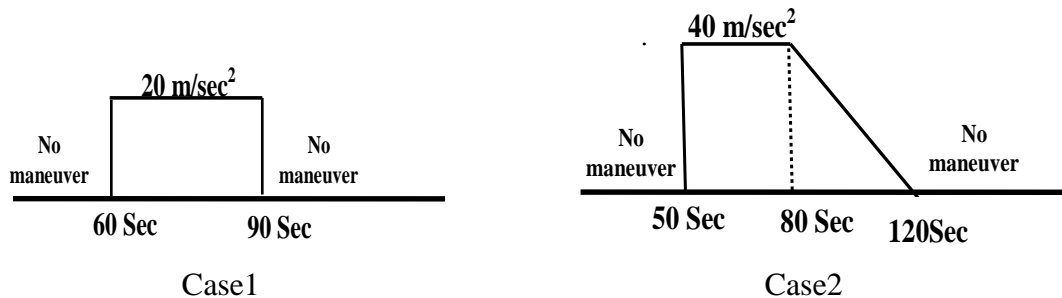- The initial value for the target range r(0): 1000 m
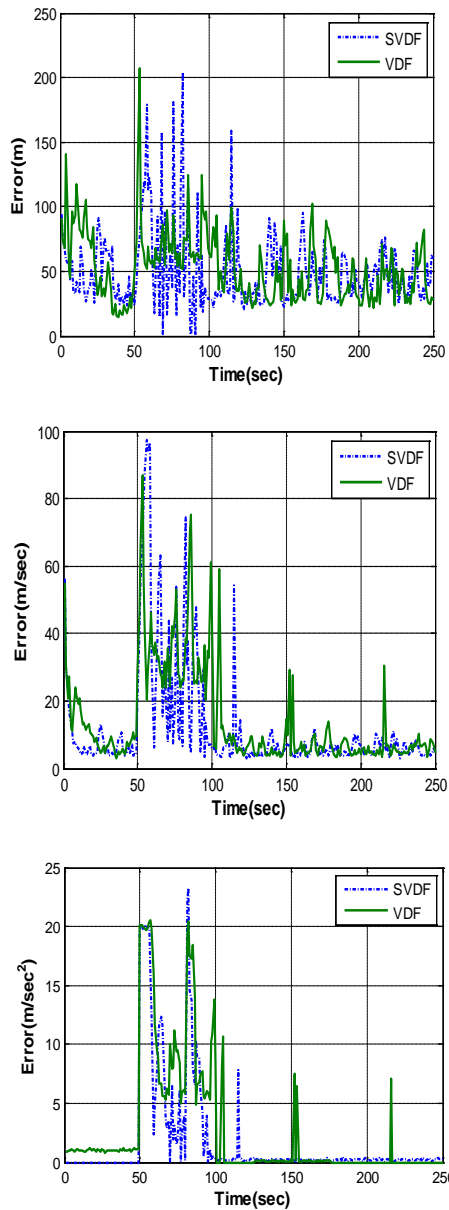

Figure.(9):Scenarios of maneuver level.

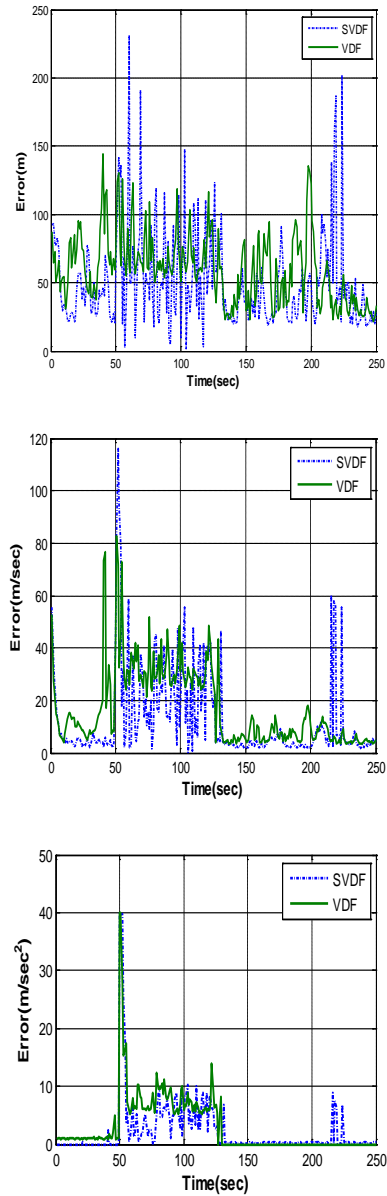**Figure.(10): RMS error for both filters with trajectory 1**

Figure.(11): RMS error for both filters with trajectory 2

A Monte Carlo simulation of 50 runs was obtained for each algorithm and the roots mean square (rms) values of the estimation errors were computed and plotted for range (m), velocity(m/sec.), and acceleration (m/sec^2) errors for both filters are shown in Fig.(10) &Fig.(11). It can be seen from these simulation results that SVDF not only yields improved performance during the maneuvering period, but also provides for better estimation during the non maneuvering period that mean the both filters give nearly similar results.

Discussions
  1- The complex problems are required to complex solutions or multiple methods and sources to reach to good results.
  2- The mathematics and physical solution for the complex problem is the most powerful methods, because it gives simple systems without additional problems. This paper shows a good example, it shows how this solution reduces the cost to about 60%, and it also reduces the total time for the system.
  3- The combination of level 2 and level 3 is reduce the cost by 10% from the original cost or by 17% from the cost of level 1, because the reduction in level 1 generate a simple system. Hence, the simulation and implementation levels can be reduce the costs and the times delay for the more complex systems.
  4- The result of the cost in the implementation using ISE 4.1i software is very similar to the result of level 3 because the clear image of the design.
  5- The total time of processing is less than 0.5 μsec, this time is very short in comparation with the sampling time (1 sec).
  6- We can return to level 3 to reduce the total cost but this work is not effective because the final total cost is low and available in the low cost FPGA chips and the redesign in serial form will increase the complexity and not reduce the cost for all circuits.

CONCLUSIONS
    The VD filter that used for tracking the maneuvering target is simplified in this paper by replace the second three state Kalman filter of it by only single acceleration Kalman filter and the modified the two state Kalman filter is initiated to perform the new estimation in conjunction with the detection scheme to track the maneuvering target. The complete designed and implemented for both filters VD and SVD by using the FPGA technique are explained in this paper to show the low cost and small delay calculations for SVD filter FPGA design as acompare with VD filter FPGA design. Also some comparative simulation results are presented in this paper, to highlight the performance for the VD and SVD filters and improvement obtained by using these approaches.

REFERENCES
[1] Y. Bar-Shalom and K. Birmiwal:" Variable Dimension Filter for Maneuvering Target Tracking", IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-18, No. 5, pp.621-629, Sep. 1982.
[2] Ming-Liang Li1, Yu-Kuei Chiu1, Yi-Nung Chung2 and Chao-Hsing Hsu3:" A Dynamic Controlling Scheme For A Tracking System"; ICIC Express Letters, ICIC International °c 2009 ISSN 1881-803X, Vo.3, No. 2, pp. 219-223, June 2009.
[3] James R. Cloutier, Ching-Fang Lin, Chun Yang:" Enhanced Variable Dimension Filter for Maneuvering Target Traking", IEEE Transactions on Aerospace and Electronic Systems, Vol. 29, No. 3, pp.786-796, July. 1993.

[4] Y. Barshalom and K.C. Chang:" Tracking a Maneuvering Target Using Input Estimation Versus the Multiple Model Algorithm"; IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-25, No. 2, pp.296-300, Nov. 1989.

[5] Jiin-An Guu Che-Ho Wel;" Maneuvering Target Tracking Using IMM Method at High Measurement Frequency"; IEEE Transactions on Aerospace and Electronic Systems, Vol. 27, No. 3, pp.514-519, May 1991.

[6] W. D. Blair:" Fixed-Gain Two-Stage Estimators for Tracking Maneuvering Targets", IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-29, No. 3, pp.1004-1014, July 1993.

[7] Bum-Jik Lee, Young-Hoon Joo, and Jin Bae Park:" An Intelligent Tracking Method   for a   Maneuvering Target "; International Jornal of control, Automation, and system Vol.1, No.1, pp93-100, March 2003.

[8] Zhongzhi LI†, Xuegang WANG:" Reverse Prediction Adaptive Kalman Filtering Algorithm for Maneuvering Target Tracking"; Journal of Computational Information Systems Vol.6, No.10, pp. 3257-3265,October, 2010.

[9] Dahmani Mohammed _, KecheMokhtar,OuamriAbdelaziz,MecheAbdelkrim:" A new   IMM   algorithm using fixed coefficients filters (fastIMM)";Int. J. Electron. Commun. (AEU¨), Algeria 64, pp. 1123–1127, 2010.

[10] Hodjat Rahmati, Hamid Khaloozadeh, Moosa Ayati:" Maneuvering Target Tracking Method Based on Unknown But Bounded Uncertainties" ; International Federation of Automatic Control (18th IFAC), Italy , pp. 4290-4295, 2011.

[11] Li-ping Song and Hong-bing Ji:" Interacting Fading Memory Modified Input Estimation for Maneuvering Target Tracking"; International Journal of the Physical Sciences Vol. 6(25), pp. 6110-6117, 23 October, 2011.

[12] Zheng Tang, Chao Sun, and Zongwei Liu:" The Tracking Algorithm for Maneuvering Target Based on Adaptive Kalman Filter"; The International Arab Journal of Information Technology, Vol. 10, No. 5, pp. 453-459, September 2013.

[13] F. R. Castella:" An Adaptive Two-Dimensional Kalman Tracking Filter"; IEEE Transactions on Aerospace and Electronic Systems, Vol.AES-16, No.6, pp.822-829, Nov.1980.

[14] E. H. Karam:" Modified Adaptive Kalman Filter For Tracking Targets "; M.Sc. Thesis, Computer and Control Engineering Department, University of Technology, 2001.

[15]A. Osama:"State Space Estimation Using Kalman Filter for Tracking Manreuvering Targets"; M.Sc. Thesis, College of Engineering, Al-Naherren University, 1993.

[16] Ick Ho Whang, Jang Gyu Lee, Tae Kyung Sung:" A Modified Input Estimation Technique Using Pseudoresiduals"; IEEE Transactions on Aerospace and Electronic Systems, Vol.30, No.2, pp.591-598, Apr.1994.