High Speed Tracking System Using Single Chip Fpga

Assist. Prof. Dr. Dhafer R. Zaghar,
M. Sc. Fatimah S. Abdulsattar **Assist. Prof. Dr. Khamis A. Zidan**
Al-Mustansiriyah University / College of Engineering / Computer & Software Eng. Dept
Baghdad, Iraq
drz_raw@yahoo.com, khamis_Zidan@yahoo.com

Abstract
The main contribution of this paper to describes and implements the Castella tracking system (CTS) in high volume Field Programmable Gate Arrays (FPGA) devices, which presents the complete design of an adaptive two-state Kalman tracking filter that is suggested by Castella to track the maneuvering and nonmaneuvering targets using FPGA. The basic design for this system required a very high cost lie out of range of FPGA capacity. This paper will present a novel approach to reduce the cost of this system. The new method depends on the reduction of the width of data bus of the system without reduction the accuracy of the system. However the novel approach will reduce the cost to about 10% from the original cost to implement the system in a single chip FPGA. Finally, two simulation scenarios are also given to illustrate the efficiency of this adaptive filter comparing with the conventional Kalman filter.

Keywords: Adaptive tracking filter, two state Kalman filter, FPGA, Maneuvering target.

1- Introduction
In many researches such as [1-5], Kalman filter has been successfully applied to target tracking. However, the Kalman filter is computationally demanding if the input measurement rate is high and/or if the state dimension is large. Furthermore, noisy measurements may decrease Kalman filter tracking accuracy [6]. One way to possibly reduce the computational rate and sensitivity to noisy measurements and biases of the maneuvering targets is to use a simple type of target tracking filters (estimating position, and velocity) that modified its operation in order to maintain on maneuvering targets. One of these filters that compass this goal is the adaptive two state Kalman filter that suggested by Castella [7] to track the maneuvering targets for a low data rate Track While Scan (TWS) operation.

In other word, target tracking filters in real-time requires dedicated hardware to meet demanding time requirements. Modern FPGAs [8,9] include the resources that are needed to design such efficient filtering structures.

The paper is organized as follows. Section 2 provides a brief overview of Castella tracking system that present in this paper. Section 3 presents the complete design for the CTS by FPGA. Section 4 describes test scenarios and presents simulation results. Finally in section 5 it draws the conclusion.

2. Castella Tracking System (CTS)

The CTS consists of two-state Kalman filter, maneuvering detector, which continuously help in modify the maneuver noise spectral density q, where the measures of track quality are sensed, normalized to unity variance, and filtered in a single-pole filter. The output of the single pole filter, when it exceeds a threshold is used to vary the maneuver noise spectral density Q_k in Kalman filter in continuous manner. This has the effect of increasing the tracking filter gains and containing the bias developed by the tracker due to the maneuvering target.

The CTS under consideration (see ref. [7]) can be used independently for range and bearing coordinate. Therefore, it can be study the structure of this filter for a single coordinate, for example. The block diagram of the CTS for range coordinate shown in figure (1).

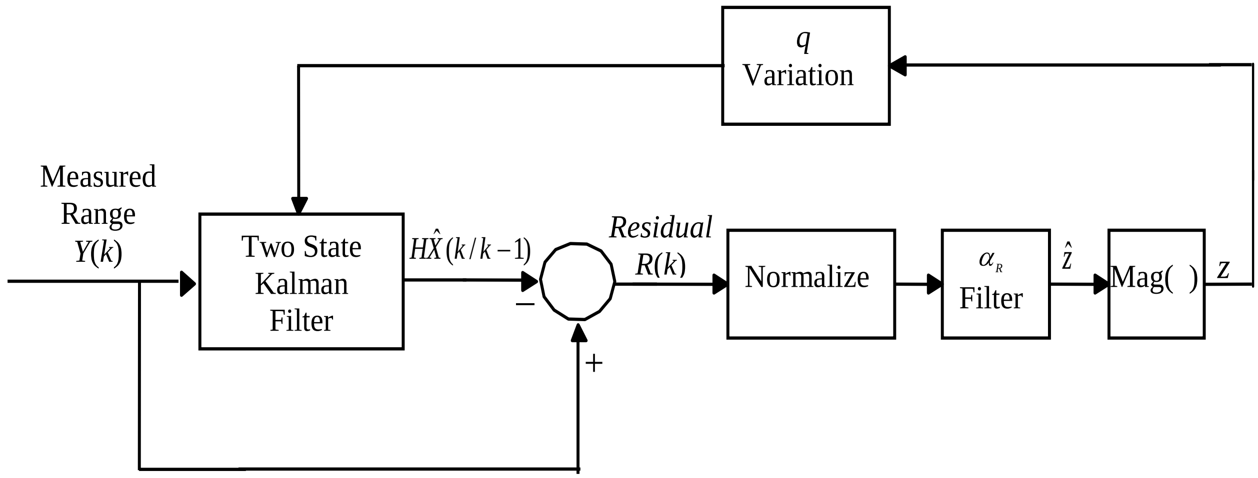

Figure (1): The block diagram of the adaptive CTS for range coordinate.

A briefly details about the target model, operation and the equations for each part of figure (1) are given as follows:

- **Target Model:** the target and the observation (measurement) dynamic model are represented by the following linear discrete-time model with sampling period T:

Target model: $X(k+1) = \Phi X(k) + W(k)$ … (1)

Observation model: $Y(k) = HX(k) + V(k)$ … (2)

Prior statistics:

$$E\{W(k)\} = E\{V(k)\} = 0$$
$$E\{W(k)W^T(j)\} = Q(k)\delta(k-j)$$
$$E\{V(k)V(j)\} = \sigma_r^2\delta(k-j)$$
$$E\{W(k)V(j)\} = 0 \quad \text{for all } j \text{ and } k$$

Matrices:

$$\Phi = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix},\ Q(k) = q^T \begin{bmatrix} T^2/3 & T/2 \\ T/2 & 1 \end{bmatrix},\ H = [1 \quad 0],\ X(k) = [r(k) \quad v(k)]\ ,\ q = \sigma_m^2 T$$

- **Kalman Filter Equations:** The recursive two state Kalman filter equations are:

Filter state prediction: $\hat{X}(k/k-1) = \Phi \hat{X}(k-1/k-1)$ (3)

Error covariance prediction: $P(k/k-1) = \Phi P(k-1/k-1)\Phi^T + Q(k)$ (4)

Filter gain: $K(k) = P(k/k-1)H^T[HP(k/k-1)H^T + \sigma_r^2]^{-1}$ (5)

Filter state update: $$\hat{X}(k/k) = \hat{X}(k/k-1) + K(k)[Y(k) - H\hat{X}(k/k-1)] \quad (6)$$

Error covariance update: $$P(k/k) = [I - K(k)H]P(k/k-1) \quad (7)$$

- **Maneuver Detection Scheme:** Maneuver is detected separately by monitoring the track residuals (i.e., measured minus predicted values) after appropriate normalization and filtering in a single-pole α_R filter, the equations for this step are
- :

Residuals: $$R(k) = Y(k) - H\hat{X}(k/k-1) \quad \ldots(8)$$

Normalization: $$z_n(k) = R(k)\left((P_{11}(k+1/k) + \sigma_r^2)^{½}\right)^{-1} \quad \ldots(9)$$

α_R Filter $$\hat{z}(k) = \alpha_R z_n(k) + (1-\alpha_R)\hat{z}(k-1) \quad \ldots(10)$$

Where $0 < \alpha_R < 1$

- ***q* Variation**: The magnitude of the output of the single-pole α_R filter *z* is used to adaptively vary the maneuver spectral density *q* in the Kalman filter model as shown in figure (2), the procedure for this step is as follows:

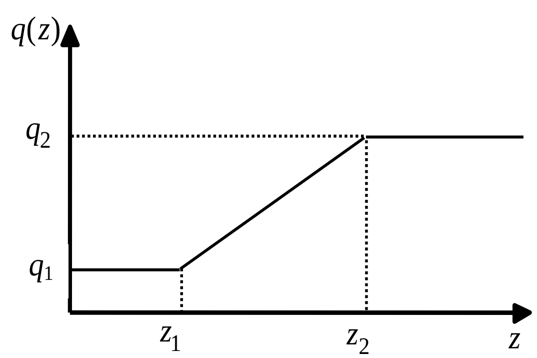

Figure (2): the relation between *z*(*k*) and the value of *q*.

1) When $z \le z_1$, where z_1 is selected as the 95 percent value for a nonmaneuvering target, $q = q_1$. The value of q_1 is selected to achieve the tracking accuracy required for a nonmaneuvering target.
2) When $z = z_2$, where z_2 is selected as the $99\frac{2}{3}$ percent value for a nonmaneuvering target, $q = q_2$. Thus when $z \ge z_2$, there is a high probability that a maneuver is in progress.
3) Between z_1 and z_2 the value of *q* is determined from the straight line established by two points (z_1, q_1) and points (z_2, q_2).
4) When $z > z_2$, use $q(z) = q_2$ since it is not necessary to increase the filter gains any more than that required containing the largest maneuver anticipated.

More details about the selection of q_1, q_2, z_1, and z_2 can be found in [7,10,11].

3. The Proposed Design of the CTS by FPGA

In order to present the complete design for the CTS, first it is re-write the two state Kalman filter equations as three equation instate of five equations after redress each of matrices *H*, *q*, equation (3) in equation (6), and equation (4) in equation (7) in addition to refer to each of the present and the previous error covariance, filter gain, and filter state element as:

$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}, K(k) = \begin{bmatrix} K_1 \\ K_2 \end{bmatrix}, \hat{X} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ with these equations, the new equations are:

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

Filter state: $$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 + Tx_2 + K_1 * (Y - (x_1 + Tx_2)) \\ x_2 + K_2 * (Y - x_1 + Tx_2) \end{bmatrix} \tag{12}$$

Error covariance: $$\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} (1+K_1)*\{(q*q_{11}) + [P_{11} + TP_{21} + T(P_{12} + TP_{22})]\} & (1+K_1)*\{(q*q_{12}) + [P_{12} + TP_{22}]\} \\ K_1 * \{(q*q_{11}) + (q*q_{12}) + [P_{11} + TP_{21} + T(P_{12} + TP_{22})]\} + (q*q_{11}) + (P_{21} + TP_{12}) & K_1 * \{(q*q_{12}) + (P_{12} + P_{22})\} \ (q*q_{22}) + P_{12} \end{bmatrix} \cdots \tag{13}$$

The following subsection explained in details the complete design for the CTS:

3.1 Principles Implementation for the CTS

The main challenge in the implementation of this system is the wide range of the data whereas the values of Kalman gain *K* fill in the range (0, 1) while the values of state $\hat{X}$ is an integer fill in the range (0, 50000), that is mean the system required to 24-bit data bus at least 8-bit for fractions and 16-bit for integers. One solution to this problem is the de-normalized to the values of gain *K* by multiply them by $2^8 = 256$to become integer, and then divided all the equations that have gain *K* by 256. This case will reduce the data bus to 16-bit only. Also that's mean the equation of f5 will become $f_5 = 256/(P_{11} + \sigma_r^2)$. This step will reduce the cost of the system to about 25% from the basic calculated cost.

Figure (3) shows the block diagram of the overall system that has 7 main units U0 process the initial values problem while the units U1, U2 and U3 implement the Kalman filter function *Ks*, $\hat{X}s$ and *Ps* respectively. The units U4 and U5 cover the other function in the tracking system in figure (1), finally the control unit is U6 that controlled on the data stream in the system.

Figure (3) draw as separated units because the high complexity in connections between the units, such example all the outputs values $\hat{X}$, *q*, *P*, and *K* must go to the run values in U0 while all the inputs values x, q, p, and k must tack from the final values in U0. However the system in figure (1) will implement in FPGA hardware in the approach of object oriented whereas the details of U1 are a sub-blocks U11 for first sub-unit of the unit U1, so on for all the details of the system as shown in the next section.

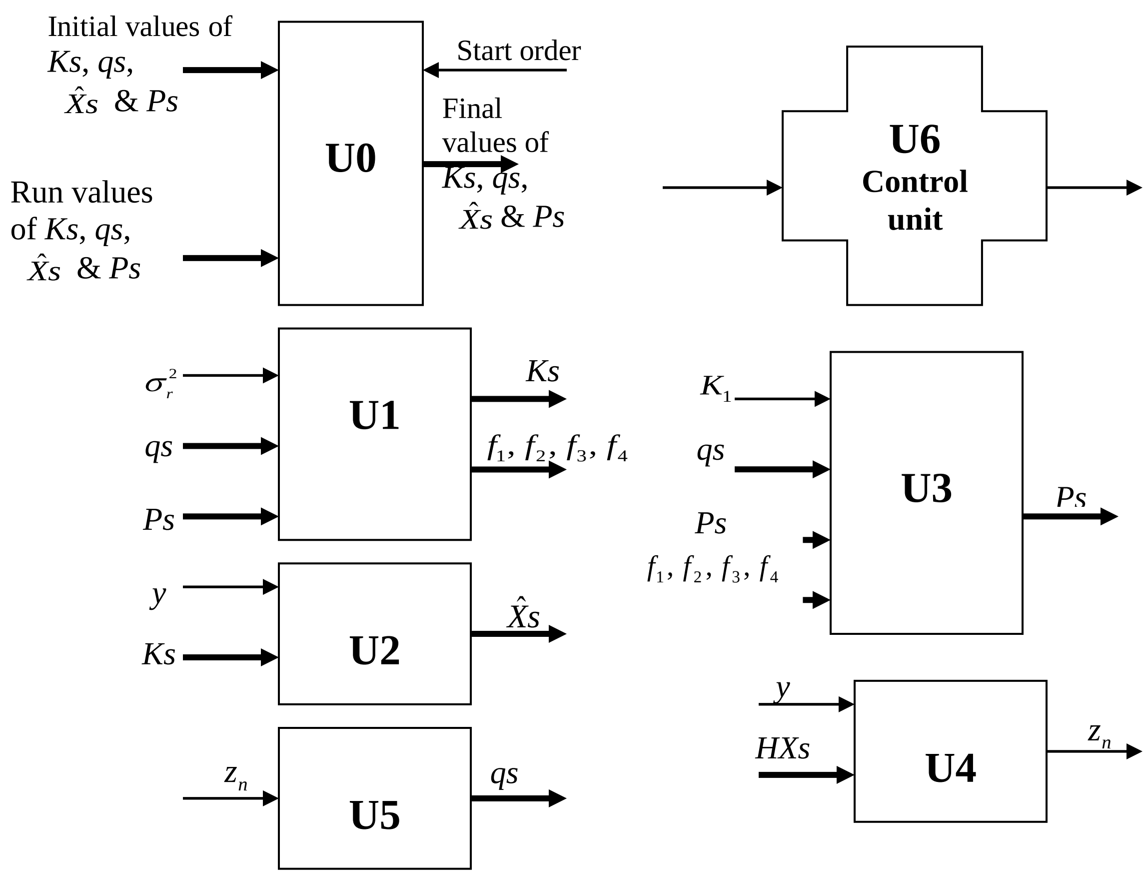

Figure (3): The block diagrams for implemented the two state Kalman filter.

The main complexity problem is the wide data bus (16-bits) that gives a high cost in multiplication and LUTs. Therefore the details of implementation will try to reduce the data bus width, so it will use a variable data bus between 6-to-16 bits depending on the requirements of each sub-unit. Generally, it will use 6-bits for *qs*, 8-bit for *Ks*, and other is 16-bit except in the sub-unit U15 as shown later. However, this approach will increase the complexity of the design but it will reduce the 16-bit system cost to about 40%, i.e. The new design will required to about 10% from the original (24-bit system) cost.

3.2 Details Implementation for the Proposed CTS

The implementations details of the system units will present as separated units as following:

3.2.1 Implementation of U0

The unit U0 is the initial values unit process, it process the problem of the initial values of the system, where it has a set of registers to store the initial vales of

$x_1, x_2, q, q_{11}, q_{12}, q_{21}, q_{22}, P_{11}, P_{12}, P_{21}, P_{22}, K_1, and\ K_2$

with set of selections controlled by the start order that resaved from the control unit, also it has a storage register to the value of σ_r^2. This unit required in FPGA to 224-bit memory storage with 208 cells, also it will add 1 cell delay for all the system.

3.2.2 Implementation of U1

Figure (4) shows the details of the unit U1 that represent the following set of equation as example to implement the units of the system in FPGA. The main problem in this unit is the subunit U15 in equation (14e) that has two problems first it required a very high cost and the output is a fraction value. The spatial study for this equation shows the practical values of this output function fill in the range (0, 2) with minimum resolution 0.017 that can be represented in 8-bits LUT to reduce the cost. The fraction values in its output will process in the spatial multiplier in the subunits U16A and U16B as shown in the following:

$$f_1 = P_{12} + P_{22} \quad \ldots\ldots(14a)$$

$$f_2 = P_{12} + P_{21} + f_1 \quad \ldots..(14b)$$

$$f_3 = q * q_{11} \quad (14c)$$

$$f_4 = q * q_{21} \quad (14d)$$

$$f_1 = 256/(P_{11} + \sigma_r^2) \quad (14e)$$

$$256 * K_1 = f_5 * (f_3 + f_2) \quad .(14f)$$

$$256 * K_2 = f_5 * (f_4 + f_1) \quad (14g)$$

The detail of implementation is summarized in table (1).

Table (1): The detail of implementation of U1 unit.

Unit	Requirements	Cost	Delay/cell delay
U11	1 adder/16-bit	40	8
U12	2 adders/16-bit	80	8
U13	6x6 multiplier	68	8
U14	6x6 multiplier	68	8
U15	1 adder/16-bit+ 8x8 LUT	400	9
U16A&B	2 adders/16-bit+2(8x12) multipliers	816	28
U1	------	1472	44

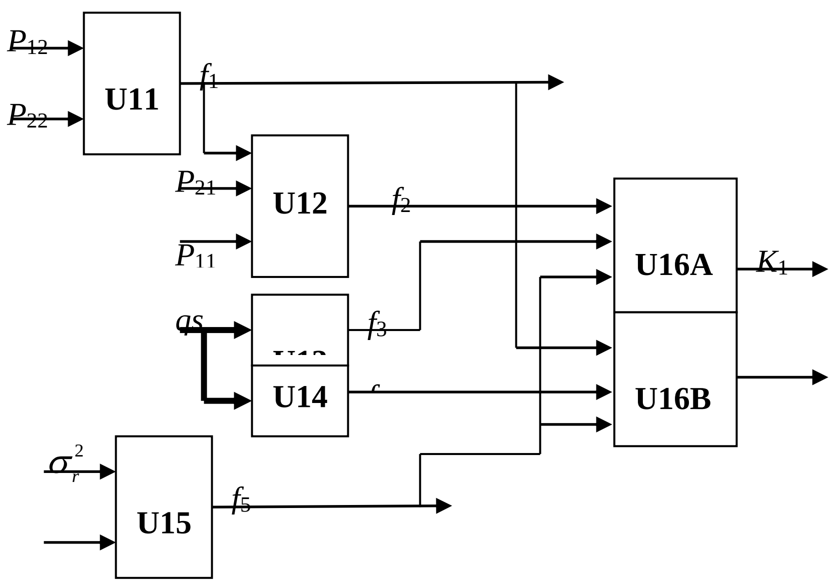

Figure (4): the details of U1 unit.

This unit required to 6 adders/16-bits and two 8x16 bits multipliers with total cost is 1500 cells with general maximum delay is 44 cell delay.

3.2.3 Implementation of U2

This unit works serially with U1 to satisfy the following equations:

$$f_6 = x_1 + x_2$$
$$f_7 = x_1 - x_2$$
$$x_1 = f_6 + 256K_1(Y - f_6)/256 \qquad .(15)$$
$$x_2 = x_2 + 256K_2(Y - f_7)/256$$

This unit required to 6 adders/16-bits and two 8x16 bits multipliers with total cost is 650 cells with general maximum delay is 42 cell delay.

3.2.4 Implementation of U3

This unit works serially with U1 and parallel with U2 to satisfy the following equations:

$$P_{11} = (256 + 256K_1)(f_3 + f_2)/256$$
$$P_{12} = (256 + 256K_1)(f_4 + f_1)/256$$
$$P_{21} = 256K_1(f_3 + f_4 + f_2)/(256 + f_3 + P_{12} + P_{21})$$

$$f_8 = q * q_{12} \qquad ….(16)$$
$$f_9 = q * q_{22}$$
$$P_{22} = 256K_1(f_1 + f_8)/(256 + P_{12} + f_9)$$

This unit required to 10 adders/16-bits, two 6x6 bits multipliers and four 8x16 bits multipliers with total cost is 1800 cells with general maximum delay is 32 cells delay.

3.2.5 Implementation of U4

This unit is the feedforward process of the Kalman filter output; it works serially with U2 to satisfy the equations (8), (9) and (10). This unit has a two 16-bit subtractor to implement equation (8), with four 6x16 multiplier to implement equation (9), while the equation (10) will required to two 8x8 fixed multiplier with 8-bit adders.

This unit required totally to 800 cells with general maximum delay is 40 cell delays.

3.2.6 Implementation of U5

This unit is the feedback process works serially with U4 to satisfy the function in figure (2) that has five 8x8 bits LUTs, it required to 640 cells with general maximum delay is 1 cell delays.

3.2.7 Implementation of U6

U6 is the control unit that is the supervisor of the system, the requirements of this unit 100-to-200 cells depending on the design approach and jobs of this unit, also it works parallel with all units therefore it not add any delay to the system.

3.2.8 System Cost and Delay

The calculations of the total cost and total delay are divided to three stapes, first is the Kalman filter that use U0, U1, U2 and U3, second is the calculations of the feedforward path that has Kalman filter with U4, while the unit U5 will represent the feedback path. The third step is using the second step calculations to calculate the total cost and delay of the system as shown in table (2).

The total cost and speed for the tracking system using xc2v1000 is 5800 cells with total delay is maximum 200 nsec. That is mean the tracking system will implement in a single chip with speed over 4MSPS.

4. Numerical Simulation Results

This section performs two numerical maneuver target trajectories to evaluate the performance of the CTS compared to the two state conventional Kalman filter (CKF). The track filters parameters for the two trajectories are:

-Sampling period *T*=1 *sec*.

-The standard deviation of the observation additive white Gaussian noise σ_r = 100 *m*.

-The anticipated standard deviation of the plant noise disturbance σ_m=5/sec^2.

-The constant target radial velocity *V*: 200 *m*/*sec*.

Table (2): The detail of implementation of tracking system.

Unit	Cost	Delay/cell delay	Notes
U0	208	1	
U1	1500	44	Serial with U0
U2	650	42	Serial with U1
U3	1800	43	Parallel with U2
Kalman filter	4158	88	
U4	800	40	Serial with U2
Forward bath	4958	128	
Feedback U5	640	1	
Control U6	200		
Total system	5800	130	

The value 100 is selected for σ_r to examine the filter performance in worst condition. Note that the design parameters for the CTS are taken from [11], these parameters are:

- a single-pole α_R=0.5.
- b_r =4.8 for 60 *m/sec*^{2} maneuver in the range coordinate.
- The value of q_1=2.16 and the value of q_2=200.
- The value of z_1 = 1.132 and the value of z_2=1.693.

The two states Kalman is initialized as

$$\hat{r}(1/1) = y(1) \quad \text{, and}$$

$$\hat{v}(0/0) = [y(1) - y(0)]/T \qquad \text{…..(17)}$$

$$\hat{X}(1/1) = \left|\hat{r}(1/1) \quad \hat{\dot{r}}(1/1)\right|$$

Where y(0) and y(1) are, respectively, the first and second received sensor measurements. The initial error covariance matrix for this coordinate is then:

$$P(1/1) = \begin{bmatrix} \sigma_r^2 & \sigma_r^2/T \\ \sigma_r^2/T & 2\sigma_r^2/T^2 \end{bmatrix} \qquad \text{…..(18)}$$

In the first target trajectories, it assumes that target is on a constant course and velocity until time t=85 sec, when it maneuvers a slow 90° turns with acceleration input 35 m/sec^2. It completes a turn at t=110 sec, remaining course is constant velocity.

To test the filters performance, a Monte Carlo simulation of 50 runs has been carried out for each filter, the roots mean square (rms) values of the range and velocity estimation errors are plotted in figure (5).

In the second maneuver trajectories, it assume that the target moves in a plane on constant course with constant velocity until time t=100sec, when it maneuvers a 90° turn with acceleration 10 m/sec^2 and it still with the end of this scenarios.

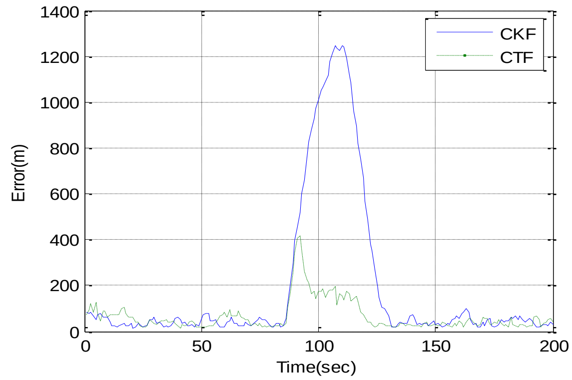

Figure (5a): The rms error of position.

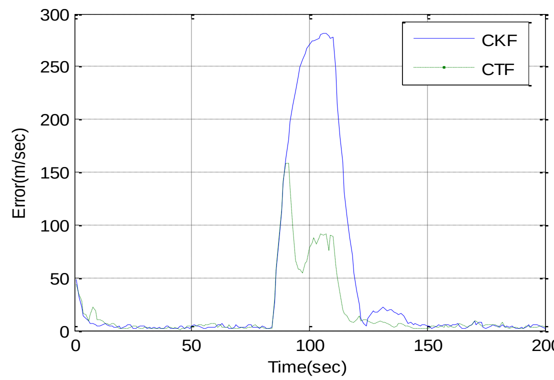

Figure (5b): The rms error of velocity.

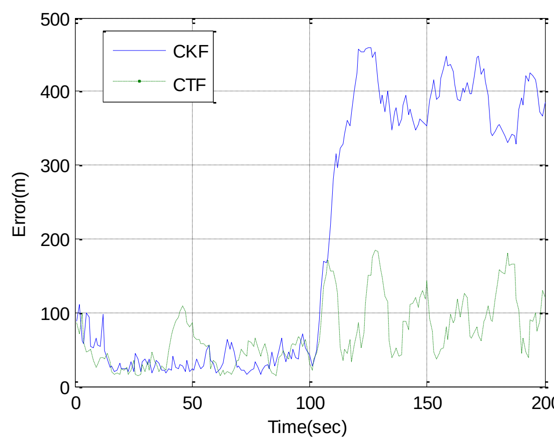

Figure (6a): The rms error of position.

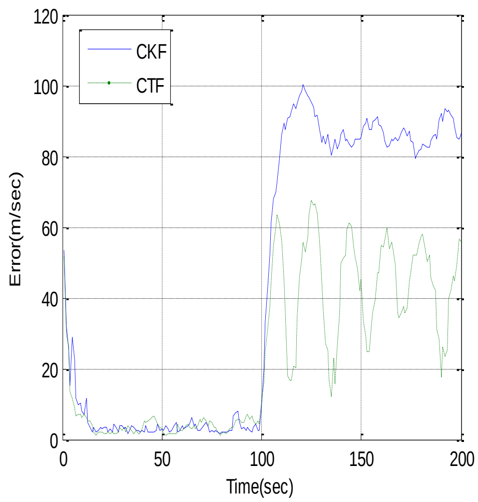

Figure (6b): The rms error of velocity.

The rms range and velocity errors of the second scenarios for the CTF and CKF are shown in figure (6). It can be seen from the simulation results that the two filters appear to be equally effective in the constant course of the target trajectories. During the maneuvering period, the CTS provide a lower rms error than the tracking CKF.

5. Conclusions

This paper presented the complete design and implements of a novel approach to reduce the cost of adaptive Kalman tracking filter by Castella to track the maneuvering and nonmaneuvering targets for a low rate TWS using a single chip FPGA.

This new method depends on the reduction of the data bus width of the system without reduction the accuracy of the system (reduces the cost about 10% from the original cost), when implemented the system in a single chip FPGA. *Also the* CTF is simulated under different flight environments

References

[1] R. A. Singer. Estimating optimal tracking filter performance for manned maneuvering targets. In IEEE Trans. Aerosp. and Electron. Syst., volume AES-5, Nov. 1970.

[2]Y. T. Chan, A. G. C. Hu, and J. B. Plant, “A Kalman filter based tracking scheme with input estimation,” IEEE Trans. Aerosp. Electron. Syst., vol. 15, pp. 237–244, 1979.

[3] Y. Bar-Shalom and K. Birmiwal, “Variable dimension filter for maneuvering target tracking,” IEEE Trans. Aerosp. Electron. Syst., vol. 18, pp. 621–629, 1982.

[4] P. L. Bogler, “Tracking a maneuvering target using input estimation,” IEEE fians. Aerosp. Electron. Syst., VOI. 23, pp. 298-310, 1987.

[5] Y. Bar-Shalom, K. C. Chang, and H. A. P. Blom, “Tracking a maneuvering target using input estimation versus the interacting multiple model algorithm,” IEEE Trans. Aerosp. Electron. Syst., vol. 25, pp. 296-300, 1989.

[6] P. De Leon, W. Kober, K. Krumvieda, and J. Thomas,”2 Subband Kalman Filtering with Applications to Target Tracking”, New Mexico State University, Klipsch School of Electrical Engineering, Las Cruces, New Mexico 88003-8001, USA.

.[7] 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.

[8] John F. Wakerly, “Digital Design Princilies and Practices”, Prentice Hall, 2001.

[9]Xilinx: Preliminary Product Specification – Virtex 2.5 V Field Programmable Gate Arrays, http:// www.xilinx.com/products 9/5-00.

[10] R. N. Razzaq:”Dynamic Target Tracking Using Composite Hypothesis Testing“; M. Sc. Thesis, College of Engineering, Al_Nahreen University, Dec., 1998.

[11] E. Hameed: ”Modified Adaptive Kalman Filter for Tracking Maneuvering Targets”; M.Sc. Thesis, Computer and Control Engineering Departmemt, University of Technology, May 2001.

نظام تتبع عالي السرعة باستخدام رقاقة FGPA

ظافر رافع زغير فاطمة عبد الستار خميس زيدان
الجامعة المستنصرية – كلية الهندسة – قسم هندسة الحاسوب

الخلاصة:

الاسهام الرئيسي لهذا البحث هو وصف وبناء نظام تتبع CTS باستخدام FPGA. لذا هذا البحث يقدم تصميم متكامل بواسطة Adaptive two-state Kalman tracking filter لتتبع الاهداف المناوره وغير المناوره باستخدام FPGA. وحيث ان تصميم النظام الاعتيادي يتطلب كلفة عالية في FPGA, لذا في هذا البحث تم تقديم طريقة مبتكرة لتقليل كلفة النظام المستخدم. هذه الطريقة الجديدة تعتمد على تقليل نطاق مجرى البيانات للنظام دون تقليل دقة النظام. لهذا فان النظام الجديد يعمل على تقليل الكلفة بحدود ١٠% من كلفة النظام الاصلي باستخدام شريحة واحدة FPGA. وكذلك تم عرض مخططين لمحاكاة النظام ولتوضيح كفاءة استخدام adaptive filter مقارنة مع conventional Kalman filter.