International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

Research Article

Speed and Hardware Optimization of ORDP Algorithm Based Kalman Filter for 2D Object Tracking

Ashwani Kumar[Á*] and Atiika Aggarwal[Ḃ]

[Á]Department of Electronics & Communication Engineering, Indian Institute of Information Technology, Allahabad, India
[B]Department of Electronics & Communication Engineering, Meerut Institute of Technology, Meerut, India

Accepted 10 May 2014, Available online 01 June2014, **Vol.4, No.3 (June 2014)**

Abstract

Tracking of a dynamic object is a challenging task and it becomes more tedious when multiple dynamic objects are present in the targeted zone. The main task in object tracking is to filter the movement information from undesired dynamic objects because this information is considered as noise. To overcome this problem, the implementation of kalman filter is presented which is used to track the desired dynamic object and to filter the noise in 2D object tracking by the estimation of past, present and future states of object. The estimation of current state depends on the variables i.e. time, velocity, covariance and noise mainly. The hardware implementation of kalman filter is done on FPGA (Virtex 5) for parallel and pipelined architectures for optimum recursive data processing (ORDP) algorithm, using Verilog HDL on Xilinx ISE simulator in the range of MHz clock frequency by keeping the focus on reducing the hardware area and increasing the speed of operation.

Keywords: *Kalman Filter, FPGA, Object Tracking, Prediction Model, Measurement Model, Verilog HDL*

1. Introduction

Kalman Filter follows an ORDP algorithm. Due to the presence of real time input there is a need of result optimization (F. A. Faruqi at el, 1980). Kalman filter is used to estimate the state of a linear system where state assumed to be distributed by a Gaussian. Kalman filter is derived from a principle which explains a property that specifies that product of two Gaussian distribution is another Gaussian distribution. Kalman Filter using state techniques as state space methods helps in simplifying the implementation of the filter in the discrete domain. As the inputs are not fixed so the location of object is shown in terms of probability. By predicting the object position from the previous information and verifying the existence of the object at the predicted position. Estimation is performed to reach to the real value by the help of sampling process which further get extended for the larger domain. Estimation Filter theory states that the state vector is estimated for a given time based upon all past measurements. It is an optimal algorithm because of its less computational requirements. There are two approaches to implement kalman filter either as hardware or software. There are two types of architecture can be possible for kalman filter and they are:

1) Loop Rolled Architecture

In loop architecture common hardware is using for comm-on logic as division, multiplication, etc which is reducing the hardware (F. A. Faruqi et al, 1980).

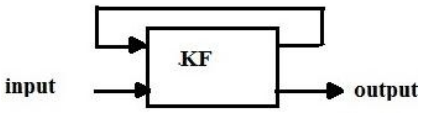

Figure 1 Loop rolled architecture

2) Loop Unrolled Architecture

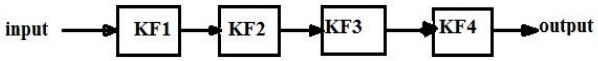

Figure 2 Loop unrolled architecture

In loop unrolling architecture the area get increased as number of blocks is increasing due to the use of separate hardware for different states. But with that the throughput is increasing as well with the speed (F. A. Faruqi et al, 1980).

The further sections describe the implementation of ORDP algorithm at both RTL level and hardware level.

2. Previous Work Analysis

When we analyse the previous works it is noticed that main concentration is done on the hardware area and the speed of the filter as in reference. Many hardware and software solutions have been proposed to achieve this objective. An Algebraic transformation method is

*Corresponding author: **Ashwani Kumar**

proposed to reduce the differential equation and to obtain explicit expression for the filter gains which results in a substantial reduction of the computer burden involved in estimating the targets states (Y. Bar-Shalom et al, 1993). After that a mapping methodology is proposed to delivering systolic and wave front array which allow the fastest pipelining rates (S. Y. Kung et al, 1991). Many more approaches were proposed but by seeing the era the major design issues arrive of optimizing area and power consumption and reduction of mean squared error. Then Kalman Filter is introduced which reduces the mean squared error. The overall objective is to estimate x (k).The difference between the estimate of $X^{\wedge}(k)$ and x (k) itself is termed the error; f (e(k))= $f(x(k)-X^{\wedge}(k))$ this function should be positive and increase monotonically (L. P. Maguire et al 1991). An error function which exhibits these characteristics is the squared error function and represent as f (e(k))= $(x(k)-X^{\wedge}(k))^2$. For the ability of the filter to predict different input data over a period of time a metric is the expected value of the error function. Therefore it represent as E [f(e(k))]. This result in the mean squared error as e(t)= $E[e^2(k)]$.

3. Implementation

3.1 Functional Description of Designed Kalman Filter using (ORDP) algorithm for 2D Tracking

A basic top module block diagram of kalman filter is shown in figure 3. This is a looped rolled architecture of kalman filter which is used to implement the ORDP algorithm.

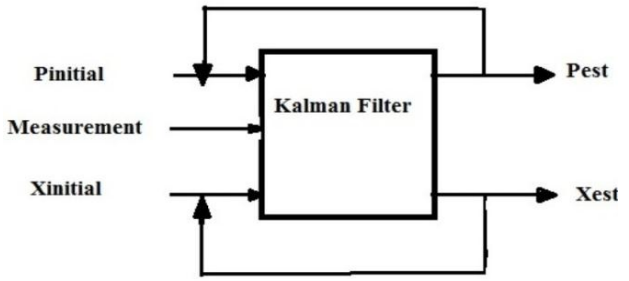

Figure 3 A basic diagram of kalman filter

Where, Pinitial is the predicted variance
Xinitial is the predicted state
Pest is the estimated variance
Xest is the estimated state
Kalman filter has two models as process model and measurement model. The whole procedure consist three steps and they are:

- Prediction
- Measurement
- Correction

Prediction is the state which is based on the previous state. Measurement is calculated by the help of measurement model.

Correction is estimated by the help of kalman gain, which got change with every sample.
The equation can be shown as:

$x(k+1) = A\,x(k) + B\,u(k) + w(k)$,
This equation is showing the prediction state for the time (k+1) where,
A is the state transition matrix,
B is the input transition matrix,
u (k) is the uncontrolled vector which is taken zero for the simplification,
w (k) is the process additive noise
$Y(k+1) = C\,x(k+1) + v(k+1)$,
This is the measurement equation where,
C is the observation matrix ,
v (k+1) is the measurement additive noise
$X^{\wedge}(k+1) = x(k+1) + K(k+1)[\,Y(k+1) - x(k+1)\,]$
This equation is showing the corrected estimated output.

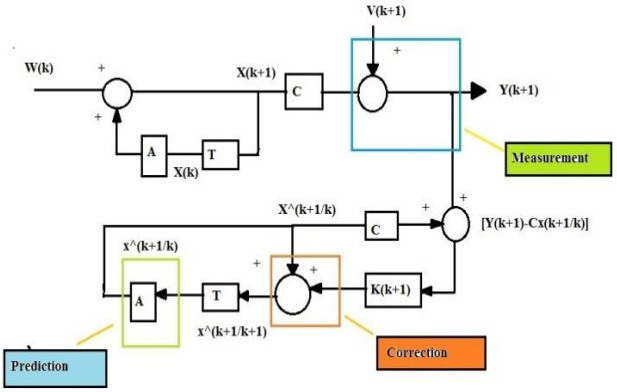

Figure 4 Block diagram showing three basic states of kalman filter

Kalman filter equation divided into two groups:

Time Update

1. Measurement Update

Time update equations can be represented as:
$X^{\wedge}(k/k-1) = A_k X^{\wedge}(k-1/k-1)$
$P(k/k-1) = A_k P(k-1/k-1) A_k^T + Q(k)$
Measurement equations can be represented as:
$X^{\wedge}(k/k) = X^{\wedge}(k/k-1) + K_k[Y(k) - C_k X^{\wedge}(k/k-1)]$
$K_k = P(k/k-1)\,C_k^T\,(C_k P(k/k-1) C_k^T + R(k))^{-1}$
$P(k/k) = (I - K_k C_k)\,P(k/k-1)$
Where, $X^{\wedge}(k/k-1)$ is predicted state
P(k/k-1) is predicted variance
$X^{\wedge}(k/k)$ and $X^{\wedge}(k-1/k-1)$ are updated state for (k-1) and k samples
P(k/k) and P(k-1/k-1) are updated variance for (k-1) and k samples
K_k is the kalman gain for state k
By assuming the process noise w(k) and measurement noise v(k) is uncorrelated and process noise is zero mean white noise having known covariance matrices.
$E[w(k), w(l)^T] = Q(k)$ if k=l;
= zero otherwise;
$E[v(k), v(l)^T] = R(k)$ if k=l;
= zero otherwise;
E [w (k), v (k)] = zero for all values of k and l
Where Q(k) is process covariance noise and R(k) is measurement covariance noise. As we the initial value of

both mean and covariance matrix are unknown so we are assuming the initial value of state as

$X^{(0/0)} = E\{x(0)\}$ and

$P(0/0) = E[\{x(0) - X^{(0)}\}\{x(0) - X^{(0)}\}^T]$

$E[\|x(k+1) - X^{(k+1)}\|^2] = E[\{x(k+1)-X^{(k+1)}\}^* \{x(k+1)-X^{(k+1)}\}^T]$

A. Derivation of implemented ORDP algoritm

The estimation of state $X^{(k+1)}$ based on the observations up to time k, z1,z2…,zk, namely is considered (M. Munu et al, 1992).

$X^{(k+1)} /Zk$.

$X^{(k+1/k)} = E[x(k+1)/z1,\ldots z k] = E[x(k+1)/Zk]$

$X^{(k+1/k)} = E[x(k+1)/Z k]$

$= E[Ax(k)+Bu(k)+w(k)/Zk]$

$= AE[x(k)/Zk] + Bu(k) +E[w(k)/Zk]$

$X^{(k+1/k)} = A X^{(k/k)} + Bu(k)$

$P(k+1/k) = E[\{x(k+1) - X^{(k+1/k)}\}\{x(k+1) - X^{(k+1/k)}\}^T/Zk]$

$= E[\{x(k) - A X^{(k/k)}\}\{x(k) - A^T X^{(k/k)^T}\}]$

$= AP(k/k) A^T + Q(k)$

$X^{(k+1/k+1)} = K'_{k+1} X^{(k+1/k)} + K_{k+1} Y(k+1)$

Where K'_{k+1} and K_{k+1} are weighting or gainmatrices

$E[X^{(k+1/k+1)}] = E[K'_{k+1} X^{(k+1/k)} + K_{k+1} Y(k+1)]$

$= E[K'_{k+1} X^{(k+1/k)} + K_{k+1} C(k+1) x(k+1) + K_{k+1} v(k+1)]$

$= K'_{k+1} E[X^{(k+1/k)}] + K_{k+1} C(k+1)^* E[x(k+1)] + Kk+1 E[v(k+1)]$

$E[X^{(k+1/k)}] = E[A X^{(k/k)} + Bu(k)]$

$= A E[X^{(k/k)}] + B u(k)$

$= E[x(k+1)]$

$E[X^{(k+1)}] = E[K'_{k+1} + K_{k+1} C] E[x(k+1)]$

$K'_{k+1} + K_{k+1} C = I$

Or $K'_{k+1} = I - K_{k+1} C$

$X^{(k+1/k+1)} = (I - K_{k+1}C) X^{(k+1/k)} + K_{k+1} Y(k+1)$

$= X^{(k+1/k)} + K_{k+1}[Y(k+1) - C X^{(k+1/k)}]$

$P(k+1/k+1) = E[\{x(k+1) - X^{(k+1/k+1)}\}\{x(k+1) - X^{(k+1/k+1)}\}^T/Zk]$

$= (I - K_{k+1}C) E[\{x(k+1) - X^{(k+1/k)}\}\{x(k+1) - X^{(k+1/k)}\}^T] (I - K_{k+1}C)^T + K_{k+1} E[v(k+1)v(k+1)^T] K_{k+1}^T + 2(I - K_{k+1}C) E[\{x(k+1)-X^{(k+1/k)}\} v(k+1)^T] K_{k+1}^T$

And with

$E[v(k+1) v(k+1)^T] = R(k)$

$E[\{x(k+1) - X^{(k+1/k)}\}\{x(k+1) - X^{(k+1/k)}\}^T]$

$= P(k+1/k) E[\{x(k+1) - X^{(k+1/k)}\} v(k+1)^T] = 0$

We get

$P(k+1/k+1) = (I - K_{k+1}C) P(k+1/k) (I - K_{k+1}C)^T + K_{k+1} Q(k+1) K_{k+1}^T$

$X(k)= [X1(k), X2(k), X3(k), X4(k)]^T$

$Y(k)= [Y1(k), Y2(k)]^T$

$W(k) = [0, U1(k), 0, U2(k)]^T$

$V(k) = [V1(k), V2(k)]^T$

$Pl(k/k-1) = A P(k-1/k-1) A^T + Q(k-1)$

$X^{l}(k/k-1) = \mathbf{A} X^{(k-1/k-1)}$

$X^{(k)} = C X^{l}(k/k-1)$

$G(k) = Pl(k/k-1) C^T [CP1(k/k-1) C^T + R(k)]^{-1}$

$X^{(k/k)} = X1(k/k-1) + G(k)[Y(k) - X^{(k)}]$

$P(k/k) = Pl(k/k-1) - G(k) C Pl(k/k-1)$

Where

$$G(k) = \begin{vmatrix} G11 & 0 \\ G21 & 0 \\ 0 & G32 \\ 0 & G42 \end{vmatrix},$$

$$R(k) = \begin{vmatrix} \sigma\rho 2(k) & 0 \\ 0 & \sigma\theta 2(k) \end{vmatrix}$$

$$P(k/k) = \begin{vmatrix} P11 & P12 & 0 & 0 \\ P21 & P22 & 0 & 0 \\ 0 & 0 & P33 & P34 \\ 0 & 0 & P43 & P44 \end{vmatrix}$$

$$P1(k/k-1) = \begin{vmatrix} P111 & P112 & 0 & 0 \\ P121 & P122 & 0 & 0 \\ 0 & 0 & P133 & P134 \\ 0 & 0 & P143 & P144 \end{vmatrix}$$

$$Q(k) = \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & \sigma 12(k) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma 22(k) \end{vmatrix}$$

$X^{1}(k/k-1) = [\, X1_1 \;\; X1_2 \;\; X1_3 \;\; X1_4 \,]$,

$X^{(k/k)} = [\, X1 \;\; X2 \;\; X3 \;\; X4 \,]$,

$Y(k) = [\, Y1, Y2 \,]$

Pl (k/k-1) is the priori error covariance estimate, **X**1(k/k-1) is the priori state estimate, Y(k) is the output estimate, G (k) is the Kalman gain, X (k/k) is the posterior state estimate, and

P (k/k) is the posterior error covariance estimate. $Q(k) = E[W(k)W^T(k)]$ is the system noise covariance matrix and $R(k) = E[V(k)V^T(k)]$ is the measurement noise covariance matrix $\sigma_1^2 = E[U_1^2(k)]$ and $\sigma_2^2(k) = E[U_2^2(k)]$ are the variances of T multiplied by the radial and angular acceleration respectively and $\sigma_\rho^2(k) = E[V_1^2(k)]$ and $\sigma_\theta^2(k) = E[V_2^2(k)]$ are the variances of T multiplied by the radial and angular measurement noise respectively. The tracking systems under consideration utilize sensors that provide measurements of range and bearing. Vehicle modelling is related to process model which includes two variables range and bearing. Present model is designed to track an object moving with constant speed, hence there should be four variables range, rate of change of range, bearing and rate of change of bearing. Sensor modelling is related to measurement model which includes only two variable range and bearing.

B. Architecture Hardware Design Approaches of Kalman Filter

In top module there are two measurement signals Y1 and Y2 at the input having 32 bits each with a clock signal clk as shown in figure 5. At the output of kalman filter there are four signals which represent the updated states X1, X2, X3, X4.

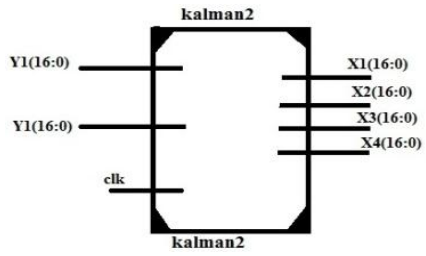

Figure 5 Top module of kalman filter

A. Parallel Architectural Design of Kalman Filter

In parallel structure as per the equation, the hardware circuit of kalman filter consists of four major parts (Z. Mehri et al, 2006; C. R. Lee et al, 1997). First part of kalman filter designed to calculate P_{11}, P_{l2}, P_{21}, P_{22} and G_{11}, G_{22}. Second part of the filter designed to calculate P_{33}, P_{34}, P_{43}, P_{44}, and G_{32}, G_{42}. Third part of the filter designed to calculate $X\hat{}l_1$, $X\hat{}l_2$ and Y_1. Fourth part of the filter designed to calculate $X\hat{}1_3$, $X\hat{}1_4$, and, Y_2.

Top module of parallel kalman filter is shown in figure 6 which has showing two measurement signals Y1 and Y2 at the input of the kalman filter as shown in figure 6. At the output of kalman filter updated states like X1= Range, X2=Rate of change of range, X3=Bearing angle, X4= Rate of change of bearing angle.

i. Variance 1

This module has four signals P11, P12, P21, P22 at the output as shown in figure 7, where P11 is the variance between actual range and the estimated range. P12 is the variance between the actual range and the estimated velocity.

P21 is the variance between actual velocity and the estimated range. P22 is the variance between the actual velocity and the estimated velocity. This variance is the estimated variance, which feed backed at the input for the next clock. Both variance blocks have adder, multiplier, constant multiplier or shifter, sub tractor, and divider. 6 adder, 3 constant multiplier, 4 sub tractor, 2 divider and 4 multiplier have been used. There is only one input (clock). Initial values are stored in resistor, hence don't need to apply input for first clock hence input wire has been reduced but register area increases.

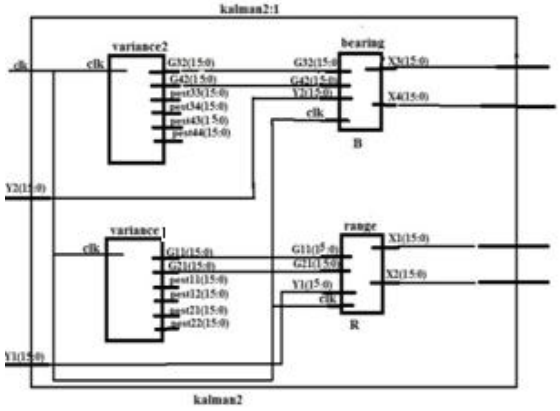

Figure 6 Top module of parallel kalman filter with sub-modules

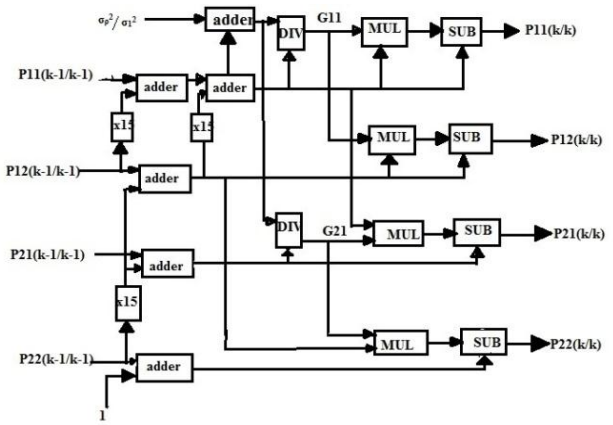

Figure 7 Internal architecture design of sub-module variance 1

ii. Variance 2

This module also has four signals P33, P34, P43, P44 at the output as shown in figure 8, where P33 is the variance between actual bearing angle and the estimated bearing angle. P34 is the variance between the actual bearing angle and the estimated bearing rate of change. P43 is the variance between actual bearing rate of change and the estimated bearing. P44 is the variance between the actual bearing rate of change and the estimated bearing rate of change. 6 adders, 3 constant multipliers, 4 sub-tractors, 2 dividers and 4 multipliers have been used.

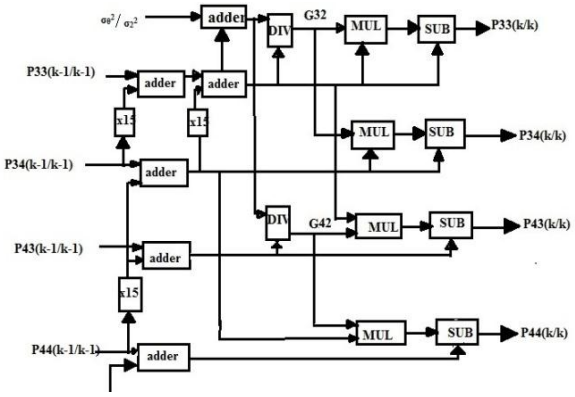

Figure 8 Internal architecture design of submodule variance 2

iii. Range

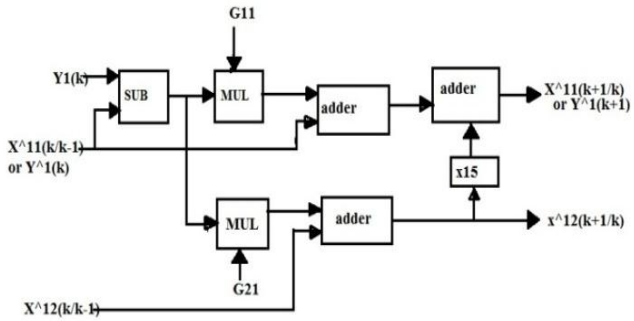

Figure 9 Internal architectural design of submodule of range

This has two signals X1 and X2 at the output as shown in figure 9, where X1 is the range of the object for particular sample and X2 is the velocity for that sample. 3 adders, 1 constant multiplier, 1 sub tractor and 2 multipliers have been used and verified. There are four input signal 3 of them having 32 bit fourth one is clk. Initial values present state are stored in resistor so we don't need to apply input for first clock.

iv. Bearing

This module has two signals X3 and X4 at the output as shown in figure 10, where X3 is the Bearing angle of the object for particular sample and X4 is the bearing rate of change for that sample. 3 adders, 1 constant multiplier, 1 sub tractor and 2 multipliers.

B. Fully Pipelined Architectural Design of Kalman Filter

Fully hardware architecture has been developed which result in big area overhead. This overhead area is not suitable for today's area constraints problem such as sensor nodes in a sensor network. So after breaking the arrays into their scalar forms fully pipelined hardware architecture has been developed for the radar tracking.

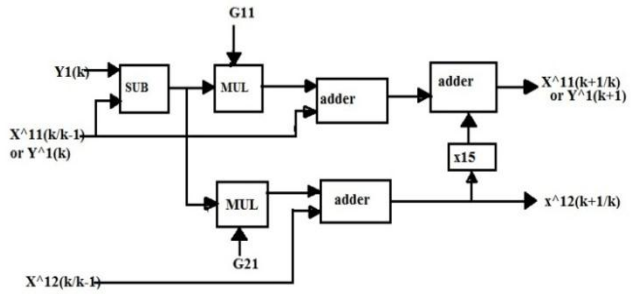

Figure 10 Internal architectural design of submodule bearing

Kalman filter having time division multiplexing of blocks has been used to decrease the silicon area. The proposed architecture divided the fully hardware structure in four parts. There are two pairs of common structured blocks which have been converted into two blocks using time division multiplexing. Hardware has been reduced to half.

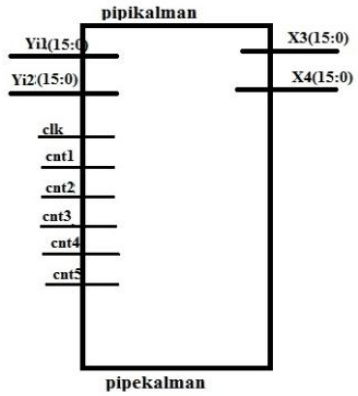

Figure 11 Top module of fully-pipelined kalman filter

Now kalman block contains 6 multipliers, 2 dividers, 9 adders, 5 sub tractors, 3 constant multiplier one control unit, some registers and multiplexers for pipelining and control. Because of this external circuit hardware doesn't reduce to exactly half, but lot of area has been saved (D. P. Atherton et al, 1994).

Top module of pipelined kalman filter as shown in figure 11, it used two measurement signal at the input having 32 bits each with clock (clk) and others 5 control signal cnt1, cnt2, cnt3, cnt4, cnt5. At the output of filter we get two signals (X1 and X2) for odd clock and (X3 and X4) for even clock.

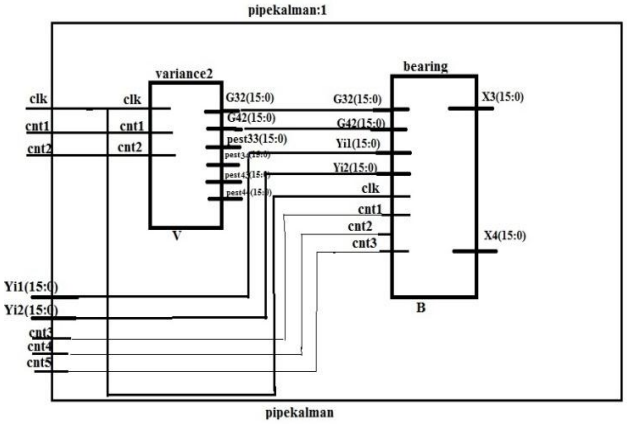

Figure 12 Top module of fully-pipelined kalman filter with sub-modules

i. Variance 1/ Variance 2

This block as shown in figure 13 uses four signals P11, P12, P21, P22 at the output for odd clock and P33, P34, P43, P44 at the output for even clock as shown in figure 13. Here time-multiplexing has been used to implement the pipelined structure to reduce the hardware. Here are two control signal cnt1 and cnt2 at the input with clock clk to select which group of output should feed backed to input.

ii. Range/ Bearing

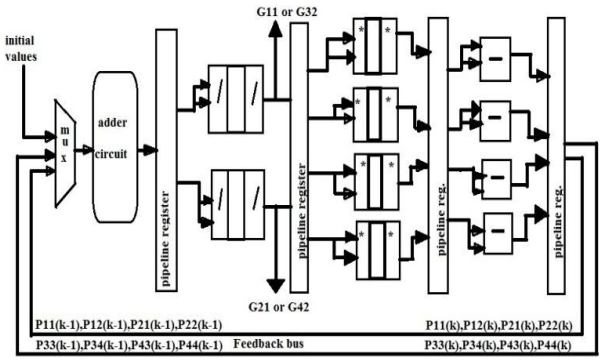

Figure 13 Internal architectural design of Variance1/ Variance2

This block as shown in figure 14 consist two signals X1 and X2 at the output for odd clock and X3and X4 for even

clock as shown in figure 13. Here are three control signal cnt1, cnt2 and cnt3 at the input with clock clk and others inputs as measured input (Y1, Y2/Y3,Y4) at odd/even clock respectively and kalman gain (G11, G21/G32, G42) at odd/even clock respectively.

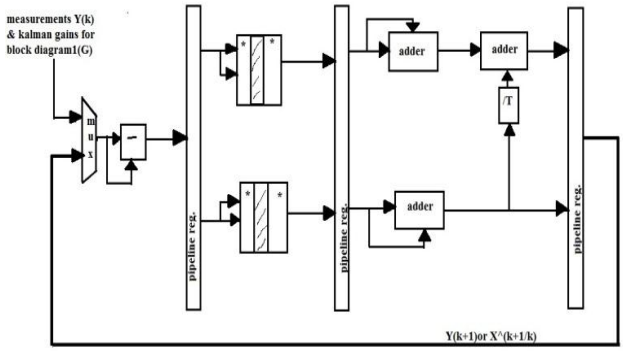

Figure 14 Internal architectural design of Range/ Bearing

Adder Block

This block contains 6 adders and 3 constant multipliers as shown in following. Ripple carry adder has been used instead of any other fast adder to implement the addition/subtraction operation as shown in figure 15.

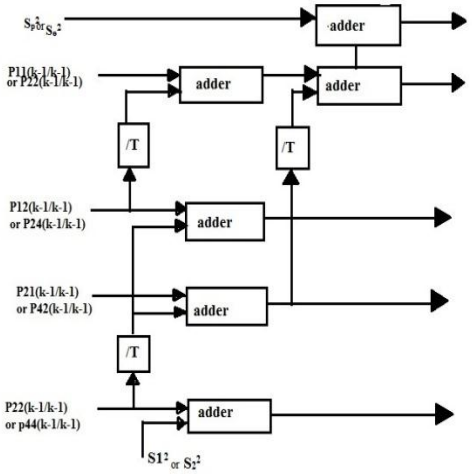

Figure 15 Adder block diagram

Sub-Tractor Block

This block contains four Sub tractors in parallel as shown in figure 16.

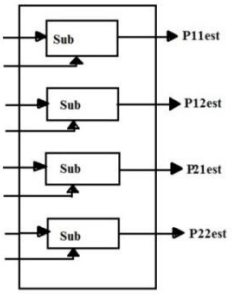

Figure 16 Sub-tractor block diagram

Multiplier Block

Multiplication has been implemented using non restoring shift and add algorithm as shown in figure 17.
This block contains four multipliers in parallel.

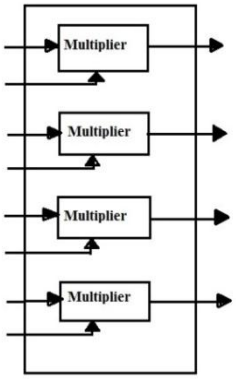

Figure 17 Multiplier block diagram

Divider Block

This block contains two dividers in parallel. A 32 bit non-restoring divider uses 15 sub tractor 16 comparator and 16 shifters. Subtraction operation occurred 15 times to got division process completed. Radix-4 non-restoring divider has been used instead of radix-2 to reduce the subtraction operation as shown in figure 18. There are two methods which could speed up the division:

- Fasten the each subtraction operation.
- Reduce the number of subtraction operation.

The top module with sub-modules of fully pipelined kalman filter is shown in figure 18.

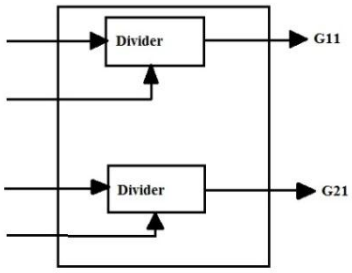

Figure 18 Divider Block diagram

4.Results and Analysis

We can analysis the results of kalman filter by different methods. In this paper we are analysing the result by Verilog HDL.

4.1 RTL Level Implementation of Parallel and Pipelined Architectures using Verilog HDL

Hardware for Kalman filter has been described in Verilog HDL.
There are two types of architecture proposed in this work
1) Parallel structure
2) Fully Pipelined structure

To minimize the hardware circuit in fixed data accuracy or to maximize data accuracy with a fixed number of bits, we normalized Pl_{11}, $P1_{12}$, $P1_{21}$, $P1_{22}$ dividing by $\sigma1^2$ in each term and normalized $P1_{33}$, $P1_{34}$, $P1_{43}$, $P1_{44}$ dividing by $\sigma2^2$. Followings are the equations reduced into scalar form.

Prediction

$P1_{l1} = P_{11} + TP_{21} + TP1_{12}$, $P1_{12} = P_{12} + TP_{22}$

$P1_{21} = P_{21} + TP_{22}$, $P1_{22} = P_{22} + \sigma_1^2$

$P1_{33} = P_{33} + TP_{43} + TP1_{34}$, $P1_{34} = P_{34} + TP_{44}$

$P1_{43} = P_{43} + TP_{44}$, $P1_{44} = P_{44} + \sigma_2^2$

$X1_1 = X_1 + TX_2$, $X1_2 = X_2$, $X1_3 = X_3 + TX_4$, $X1_4 = X_4$

$Y_1 = X_1$, $Y_2 = X_3$

Updating

$G_{11} = P1_{11} / (P1_{11} + \sigma_\rho^2)$, $G_{21} = P1_{11} / (P1_{12} + \sigma_\rho^2)$

$G_{32} = P1_{33} / (P1_{33} + \sigma_\theta^2)$, $G_{42} = P1_{43}/(P1_{33} + \sigma_\theta^2)$

$P_{11} = P1_{11} - P1_{l1} G_{11}$, $P_{12} = P1_{12} - P1_{12} G_{11}$

$P_{21} = Pl_{21} - Pl_{11} G_{21}$, $P_{22} = P1_{22} - P1_{12} G_{21}$

$P_{33} = P1_{33} - P1_{33} G_{34}$, $P_{34} = P1_{34} - P1_{34} G_{32}$

$P_{43} = P1_{43} - P1_{33} G_{42}$, $P_{44} = P1_{44} - P1_{34} G_{42}$

$X1 = X1_1 + G_{11} (Y_1 - \dot{Y}_1)$, $X2 = X1_2 + G_{21}(Y_1 - \dot{Y}_1)$

$X3 = X1_3 + G_{32} (Y_2 - \dot{Y}_2)$, $X4 = X1_4 + G_{42}(Y_2 - \dot{Y}_2)$

A. Parallel Structure

Top module of parallel kalman filter as shown in figure 6 is designed using Verilog HDL on Xilinx ISE simulator. The sub-modules like Variance 1, Variance 2, Range, Bearing and designed at RTL level to developed complete module of kalman filter (J. M. Jover at al, 1986). In top module there are two measurement signals Y1 and Y2 at the input having 32 bits each with a clock signal clk. At the output of kalman filter there are four signal which represent the updated states like X1= Range, X2=Rate of change of range. Complete kalman filter has used 18 adders, 8 constant multiplier, 10 sub tractor, 4 divider and 12 multipliers. The RTL level implementation of parallel kalman filter is described in form of functional waveform as in figure 19.

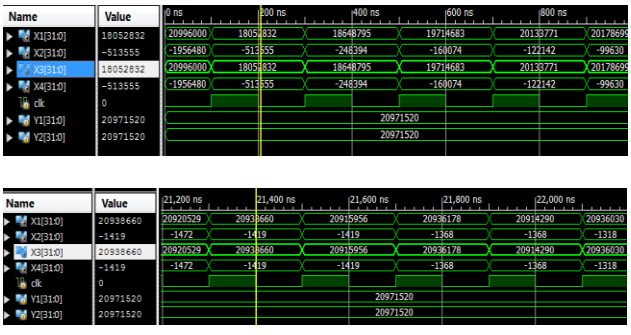

Figure 19 Simulation result of parallel kalman filter

First test bench indicates the first sample and second test bench indicates the 10^{th} sample. Using 32 bit representation where integer bits are left first 12 and rational bits are remaining 20. So the precision is $1/2^{20}$. To get the actual magnitude of a number, number has to be divided by a factor 2^{20} (1048576). Initial value X1 = 10, X2 = 0, X3 = 10, X4 = 0

1st sample X1=17.216522216, X2= -0.489764213, X3=17.216522216, X4= -0.489764213

Second continually increased

10^{th} sample of range X1 = 19.96389389, X2= -0.001353263, X3 = 19.96389389, X4 = -0.001353263

Accuracy = (20-19.96389389)/20 = 0.018 or 1.8%

Its takes 10 sample to reach the 1.8% of accuracy.

This result is showing the recursive optimized result which is needed for determining the exact position of the moving object so by this it is clearly shown that the object has been tracked.

B. Fully Pipelined Structure

Top module of Pipelined kalman filter as shown in figure 12 designed using Verilog HDL on Xilinx ISE simulator. The sub-modules like variance 1/ variance 2, Range/ Bearing are designed separately for complete implementation of pipelined kalman filter and by merging them one top module of kalman filter is obtained. Top module used two measurement signal at the input having 32 bits each with clock (clk) and others 5 control signal cnt1,cnt2,cnt3,cnt4,cnt5. At the output of filter we get two signals (X1 and X2) for odd clock and (X3 and X4) for even clock. Fully pipelined architecture contains 9 adders, 2 dividers, 6 multipliers, 5 sub tractors, 7 shifters and 2 multiplexer which are about 50% of the parallel architecture. The RTL level implementation of pipelined kalman filter is described in form of functional waveform as shown in figure 20.

Figure 20 Simulation result of pipelined kalman filter

By analysis of both types of structure and after comparing proposed architecture with previous implementation (Song Ci et al, 2005), it is noticed that the area get reduced in fully pipelined structure with the increment in the speed. Both implementations gave the same result because combinational logic and accuracy is also same, only difference comes in the way of design how input has been handled. It could be seen from the test bench that here is only two output signal X3 and X4 while there was four output signals in the parallel design. To handle that extra control signal cnt1, cnt2, cnt3, cnt4, cnt5 has been used. The results were compared and examined on FPGA virtex5 (Xilinx Virtex5, 2012).

Table 1 Combinational Delay in Parallel and Pipelined Implementation

	Parallel	Pipelined
Total Delay	65.234ns	46.712ns
Logic delay	27.589ns	20.228ns
Routing Delay	37.645ns	26.484ns

Table 2 Comparison of proposed architecture with previous implementations for 12 bit architecture

	Parallel Implementation on flex 8000	Pipelined Implementation on virtex 800	Proposed Pipelined Implementation on virtex 5
Total Delay	260.67ns	41.85ns	14.556ns
BELs	6180	3653	1711
Max operating frequency	3.836MHz	23.89MHz	68.7MHz

The proposed pipelined architecture is also tested at 21.4 MHz clock speed for 32 bit operation.

Conclusion

Kalman filter is an optimal estimator in comparison to other estimator. It requires large implementation area and for that there is a need of an efficient method to implement it on hardware. The architecture proposed to design the kalman filter has been optimized in terms of speed and area compared to previous works as shown in table 2. Problem of speed had been removed by parallel structure but area utilization increased which is the main problem for sensor's application. This problem has been reduced by implementing the pipelined structure. This has been done more efficiently using efficient method for implementing division and other arithmetic operation. High radix method has been introduced for implementing division process, which reduces the combinational delay from 39.919 ns to 20.717 ns about half compare to simple non restoring method. Architectures have been thoroughly verified on simulator and FPGA. Thorough analysis of kalman algorithm leads to a novel technique of hardware reduction by parallel and pipelined approaches. Reduction in the chip area reduces the power consumption and dissipation which motivates the further utilization of kalman filter in several signal processing systems, simultaneously reducing the area and increasing the speed will allow us to embed it with large variety of applications. This work purpose an efficient kalman filter using 32 bit input which has a great resolution $1/2^{32}$ of the full scale value.

References

F.A. Faruqi and R.C. Davis (1980), Kalman Filter design for target tracking, *IEEE Trans*. Aerosp. *Electron. Syst*., *AES-16*, pp. 500-508.

Y. Bar-Shalom and X. R. Lin (1993), Estimation and tracking: principles, techniques, and software, *Artech House*, pp. 417-483.

S. Y. Kung and J. N. Hwang (1991), Systolic array designs for Kalman Filtering, *IEE Trans., Signal Processing*, pp. 171-182

L. P. Maguire and G. W. Irwin (1991), Transputer implementation of Kalman Filter, *IEEE Proc.*, Vol. 138, pp. 355-362.

M. Munu, I. Harrison, D. Wilkin, and M. S. Woolfson (1992), Comparison of adaptive target-tracking algorithms for phased-array radar, *IEE Proc. F. Commun, Radar and Signal*, Vol. 139, pp. 336-342.

Z. Mehri, M. Ghantous, M. Bayoumi, M. Bayoumi, A El-Desouki (2006), A Fully-Pipelined Parallel Architecture for Kalman Tracking Filter, *IEEE Proc. Of CAMP*, pp. 81-86.

C.R. Lee, Z. Salcic (1997), A Fully-hardware-type Maximum-parallel Architecture for Kalman tracking, *ICICS, Vol. 148, pp. 348-382.*

C.P. Atherton, H. J. Lin (1994), Parallel implementation of IMM tracking algorithm using transputers, *IEE Proc.-Radar, Sonar Navig.*, Vol. 141, pp. 325-332.

J. M. Jover, T. Kailath (1986), A parallel architecture for Kalman filter measurement updata and parameter estimation, *Automatica*, Vol. 22, pp. 32-57.

Song Ci, Sharif, H (2005), Performance comparison of Kalman filter based approaches for energy efficiency in wireless sensor networks" *IEEE conf: on Computer Systems and applications*.

Xilinx Virtex-5 (2012) FPGA User Guide ,UG190 (v5.4) March 16.