International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 9 (2015) pp. 24521-24527

http://www.ripublication.com

Implemention of Kalman Filter For Object Tracking Applications

[1]Sista Raghava Rama Sripad, [2]R. Parameshwaran
[1,2]School of Computing, SASTRA University,Thanjavur-613401, India
[1]Email: raghavaramasripad@gmail.com

Abstract

Kalman filter is recursive estimating filter. It is used for predicting the state of a system in association to a suite of sensors and other measurement devices.It estimates the future behavior of a system based on a series of past behaviors. It is used to its full potential mainly in object tracking and guidance of vehicles and inertial navigation systems. Several object tracking methods focus only on improving the detecting mechanisms but not on the performance parameters of the kalman algorithm itself.In this paper, a new method to increase the overall speed of the kalman algorithm is proposed where the multipliers are reduced by employing multiplexers. This increases the speed of the system since the total number of multiplications are reduced considerably. This also decreases the amount of energy consumed making it suitable for implementing it in hardware creating a suitable. Then this device is implemented on hardware to create a dedicated kalman filter module

Keywords: Kalman filter, multiplexers, estimation, objects tracking, VLSI architecture.

Introduction

Kalman Filter is an estimation algorithm used for predicting the future steps of any given system. It uses a recursive method to predict and update the system states based upon the measurement received by an external source. Based on this updated measurement the system updates the object's trajectory and guides it accordingly or predicts the next position where the target is most likely to occur. This filter was proposed by R. E. Kalman in 1960 as a way to increase the guidance capability of Apollo space crafts. Since then it has found various uses in tracking, guidance, inertial navigation of space crafts, unmanned vehicles, cruise missiles, vehicle tracking, economics sensor fusion, etc. The necessity of using Kalman filter in object tracking is to remove the noise accumulated in measurements and to give an accurate estimation of a target's position over time.

The filter predicts and corrects its state estimation for tracking an object using mathematical procedures. This algorithm predicts the position of the object by adding its previous position and velocity measurements and a correction estimate which is proportional to the predicted error. This minimizes the error statistically producing a better estimate as to where the object will be present in the next time frame. Fig .1.shows the basic block diagram of a kalman algorithm module. As shown, it consists of two blocks for predicting and updating the values respectively. The process is recursive in nature i, e., it is in an indefinite loop of predicting and then updating the values using new measurements.

Existing Method

The kalman estimator is a recursive algorithm that operates on noisy input data to produce an optimal statistical estimate of the object. It takes the measurement of the surrounding environment that affects the system and based on the previous inputs, estimates the position of the target in the future. The kalman filter governs a state *m* goverened by the linear difference equation

$$m_k = Am_{k-1} + Bn_{k-1} + c_{k-1} \tag{1}$$

Where

A is a matrix that represents the difference between previous time step *k-1* to current time step *k*.

B is a matrix that represents the control input *n* to the state *m* with a measurement *z* that is given by

$$o_k = Jm_k + f_k \tag{2}$$

Where

The random variables c_{k-1} and f_k represent process and system noise respectively.

J represents the matrix that relates the state to measurement o_k.

Basically a simple kalman algorithm works in two steps

1. Time update(prediction step)
2. Measurement update(correction step)

During the time update the algorithm projects how the current state estimate of the system behaves ahead of time. Consider a_k a system state where 'k' represents the present system state and 'k-1' represents the previous system state. The equations representing the time update step in kalman filter is given by

$$\hat{a}_k^- = X\hat{a}_{k-1} + Ys_{k-1} \tag{3}$$

$$B_k^- = XB_{k-1}X^T + Q \tag{4}$$

Where

X is an n×n matrix that relates the previous state with the present state

$\hat{a}_k^-$ is the *a priori* system state

B is the estimate error covariance

Q is the process noise co variance

s_{k-1} is the optional control input with a related n × l matrix Y

After the algorithm predicts the state of the system for the next time step, it collects the measurement from outside sensors to update the filter by using a set of calculations to update the system and to calculate as to how much deviation has the system suffered due to various noises. Using this it calculates a gain which tells as to how much correction is needed and the system corrects itself to that amount from the next iteration. The equations relating to measurement step are given as follows

$$K_k = B_k^- H^T (H B_k^- H^T + R)^{-1} \tag{5}$$

$$\hat{a}_k = \hat{a}_k^- + K_k (z_k - H \hat{a}_k^-) \tag{6}$$

$$B_k = (I - K_k H) P_k^- \tag{7}$$

Where

K_k represents the kalman gain

R is the measurement error variance matrix

H is the measurement noise covariance

Here equation (3) represents the state prediction which informs as to where the systems might end up. Equation (4) predicts the covariance which informs as to how much error may end inside the system after the prediction. Equation (5) gives the kalman gain which gives the amount of correction we need to incorporate into the system to match the prediction to the obtained measurement. Equation (6) updates the estimate with measurement f_k. Equation (7) updates the error covariance.

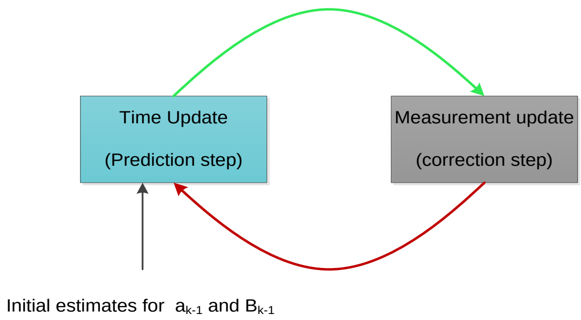

Figure 1: Basic Block Diagram of Kalman Filter

In normal kalman filter applications regarding object tracking several researchers are focussing on the target acquisition methods only and less on the tracking mechanism which is the kalman algorithm. This although produces better detecting methods, considerably reduces the target retainabiltiy in the tracking systems. In [1] a particle filter is used to detect and track the object. This method is useful in detecting several objects at once but lacks credibility in single object tracking. This is a serious drawback in a track and destroy missile systems which after being fired detect, select and track a single target. This reduces the target handling credibilty which can cost lives. In [2] the system uses a servo device to keep track of the object in the center of the camera's field of view and employs a model matching strategy to detect and track the object. This system lacks the abiltiy to track the object if it moves faster than 0.96 m/sec and if the target is closer than ten meters from the camera. This gives a serious drawback for tracking high speed moving objects such as aircrafts and missiles. In [3] the algorithm detects a target using point matching technique. It uses prominent features of the target and then tracks it using kalman filter. However here also the basic kalman algorithm is used without any improvizations. This has serious traacking issues when the corner features of the object are not prominent or inconsistent. In [4] the system uses a shift slgorithm for estimating the values which are inconsistent for use in a hardware implementation. It employs a lot of components in the hardware module making it complex and cubersome. In [5], to detect the object a correlation weighted histogram intersection method is used. Even though this a bit accurate, it is highly complex and a hardware implmentation of both the kalman filter aided by this method [5] increases implementation complexities. So a new method to first reduce the computational complexity and increase the ease of implementation of a basic kalman filter module is required.

Proposed Method

In the present day kalman filter applications, the researchers are focusing only on the target detecting algorithms but not on the kalman filter module. The kalman filter is used as it is which results in an inefficient usage of resources. Basically a kalman filter is always realized in a software format only. There is no dedicated hardware implementation for this estimation algorithm. This seriously limits its usage capabilities on the field and also renders it more complex. This proposed system deals with a dedicated hardware implementation with a substantial increase in speed with a decrease in area, computational complexity and power reduction. This is achieved by altering the matrix multiplication elements. In a normal kalman filter the total number of multiplication needed for the matrices are twenty seven. However a normal FPGA kit provides a total number of thirty two multipliers only. This provides a serious hindrance for the architecture since more number of multiplier modules are being utilized. This reduces the speed of the system and occupies more area. This results in slower outputs and more power consumption. So the main point is to focus on reducing the total number of multipliers used. One of the methods is to use multiplexers to reduce the number of matrix multiplications. The matrix multiplications can be multiplexed to increase the total number of inputs. Consider two matrices X and Y given by

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} \tag{8}$$

$$Y = \begin{bmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \\ y_{31} & y_{32} & y_{33} \end{bmatrix} \quad (9)$$

Consider a matrix multiplcation between (8) and (9). Normally this uses a total of twenty seven multiplications. In the proposed system, multiplexers are employed to reduce the number of multiplications to nine only.

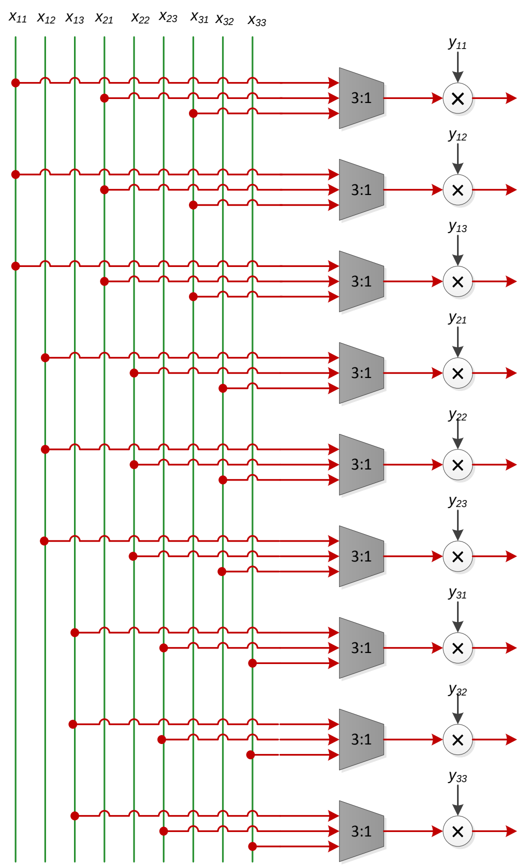

Figure 2: Layout of Multiplexer Based Matrix Mulitplication

This is done by using nine dedicated multipliers which are connected to the inputs using the same number of multiplexers which contain a whole row as input as shown in fig.2 . This decreases the total number of multipliers from twenty seven to nine only. Outside the matrix multiplications we still need a static seven multipliers resulting in a multiplier count of sixteen. The way we approached this was to use 9 multipliers at once. An input of three valued rows as well as an entire 3x3 matrix can be given. Using this,an entire row could be calculated with each set of the inputs of the multiplexer. Any one entire matrix remains the same at the multiplier side while the other matrix is given as input in the form of a row of 3. The entire 3x3 matrix remains the same for these calculations but the row of 3 is what is getting multiplexed from the other 3x3matrix. Consider the multiplication of two 3x3 matrices. As shown in fig.2 a set of 3 elements from one matrix is given as input to a multiplexer which has the corresponding element that is multiplied to each of the three inputs. This process is repeated a total of three times for each set of inputs. This gives rise to a module corresponding fig.2. based on this a module was created to perform the 9 multiplications between a 1x3 matrix and a 3x3 matrix which is shown in fig.2. This procedure is applied to the implementation part which results in a reduced number of multipliers used for matrix multiplications. This reduction of multipliers results in the increase of computational speed and decrease in time taken for estimating the values. A reduction in area is also achieved.

Experimental Results

Table 1: Area Report

Logic Utilization	Used	Available	Utilization
No: of slices	1274	6144	20%
No: of slice flip flops	1243	12288	10%
No: of 4 input LUTs	2000	12288	16%
No: of bonded IOBs	8	240	3%
No: of GCLKs	1	32	3%
No: of DSP48s	16	32	50%

Conclusion

In the proposed method the number of multipliers used for matrix multiplication has been reduced to nine and the total multipliers employed has been reduced to sixteen. This increases the overall speed of the module substantially with a minimum timing given around 11.740 ns. The use of sixteen multipliers reduces the area utilization to 50% making the module less complex with a reduction in power, area and increase in speed of estimation. This implies that the speed of tracking a target in real time increases to a level where the implementation of the kalman algorithm module becomes a indespesable tool for object tracking.

References

[1] M. Marron, J. C. Garcia, M. A. Sotelo, M. Cabello, D. Pizarro, F. Huerta, J. Cerro "Comparing a Kalman Filter and a Particle Filter in a Multiple Objects Tracking Application" IEEE (2007).

[2] Dae-Sik Jang, we-Young Kim, Hyung-Il Choi "Kalman Filter incorporated Model Updating for Real-time Tracking", IEEE TENCON - Digital Signal Processing Applications (1996).

[3] Wei Zeng, Guibin Zhu and Yao Li "Point Matching Estimation For Moving Object Tracking Based On Kalman Filter" Eight IEEE/ACIS International Conference on Computer and Information Science (2009).

[4] Ahmad Ali, Dr. Sikander Majid Mirza "Object Tracking using Correlation, Kalman Filter and Fast Means Shift Algorithms" IEEE-ICET 2nd International Conference on Emerging Technologies (2006).

[5] Saira Saleem Pathan, Ayoub Al-Hamadi and Bernd Michaels "Intelligent Feature-guided Multi-object Tracking Using Kalman Filter" IEEE Catalog (2009).

[6] Auger, François Hilairet, Mickael Guerrero, Josep M. Monmasson, Eric Orlowska- Kowalska, Teresa Katsura, Seiichiro "Industrial applications of the Kalman filter" IEEE Transactions on Industrial Electronics (2013).

[7] Madhur Mehta, Chandni Goyal, M. C. Srivastava, R. C. Jain "Real Time Object Detection and Tracking: Histogram Matching and Kalman Filter Approach" IEEE Transactions (2010).

[8] Caius Suliman, Cristina Cruceru, Florin Moldoveanu "Kalman Filter Based Tracking in an Video Surveillance System" 10th International Conference on Development and Application Systems (2010).

[9] Yushan Sun, Junling Sun, Lei Wan, Chengtao Li And Yinghao Zhang "An Improved Self Adaptive Kalman Filter For Underwater Integrated Navigation System Based On DR" 2nd International Control And Information Processing IEEE (2011).

[10] Xin Li Kejun Wang,Wei Wang and Yang Li "A Multiple Object Tracking Method Using Kalman Filter" Proceedings of the IEEE International Conference on Information and Automation (2010).