Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I
IMECS 2008, 19-21 March, 2008, Hong Kong

A Case Study: Kalman & Alpha-Beta Computation under High Correlation

D. M. Akbar Hussain, *Member IEEE, IAENG*, David Hicks, Daniel Ortiz-Arroyo, Shaiq A. Haq, Zaki Ahmed

Abstract— **The investigation presented here compares the advantage of using a Kalman filter as opposed to an alpha-beta filter for multi-target tracking systems. The former is often used to speed up the computation time. However, it is shown here that due to the difficulty of data association the benefits are not as good as might be expected based on their relative results from single target tracking case. Extensive analyses are performed by selecting various scenarios where the correlation factor affects the performance of alpha-beta filter.**

Index Terms—**Filtering, Kalman filter, alpha-beta filter, Target tracking, State estimation.**

I. Introduction

During the last two decades the improved technology available for surveillance systems has generated a great deal of interest in algorithms capable of tracking large number of objects using information from one or more sensors like radar, sonar etc. Typical sensor systems, such as radar, obtain data returns corrupted with noise from true targets and possibly from other objects. In general the tracking problem requires processing of incoming data to produce accurate position and velocity estimates [1][2][3]. There are two types of uncertainties involved with this incoming data, first the position inaccuracy, as the measurements are corrupted by noise, and second the measurement origin since there may be uncertainty as to which measurement originates from which target [4][5]. These uncertainties lead to a data association problem and the tracking performance depends not only on the measurement noise but also upon the uncertainty in the measurement origin [6][7][8]. Therefore, in a multi-target environment extensive computation may be required to establish the correspondence between measurements and tracks at each radar scan [9][10]. After the data association process, tracks are normally updated using either standard Kalman or alpha-beta filter [11][12][13]. Also tracks whose statistics deviate from the assumed model and shown to be following the same target are normally eliminated [14][15].

Kalman or alpha-beta filters could be ideal choice for a single target case where one noisy measurement is obtained at each radar scan. In the multi-target tracking case, an unknown number of measurements are received at each radar scan and assuming no false measurements, each one has to be associated with an existing or new tracking filter. When the targets are well apart from each other then forming a measurement prediction ellipse around a track to associate the correct measurement with that track is a standard technique [16]. When targets are near to each other, more than one measurement may fall within the prediction ellipse of a filter and prediction ellipses of different filters may interact. The number of measurements accepted by a filter will therefore be quite sensitive in this situation to the accuracy of the prediction ellipse. Several approaches may be used for this situation [17][18], one of which is called the Track Splitting Filter algorithm. In this algorithm, if ***n*** measurements occur inside a prediction ellipse, then the filter branches or splits in to ***n*** tracking filters [16]. This situation, which results in an increased number of filters being used, requires more processing power and storage memory. Some mechanism for restricting the excessive number of tracks that originate from track splitting is required to avoid system crash. Also, this process of track splitting eventually may result in more than one filter tracking the same target. There are two standard techniques to keep this track explosion under control. The first criterion is the support function that uses the likelihood function of a track as the pruning criterion. Second the similarity criterion which uses a distance threshold to prune similar filter tracking the same target [19]. In this paper investigations are restricted to the application of the first criterion only. A typical recursive multi-target tracking system is shown in figure 1. The algorithm is implemented using an AMD Athlon ™ 64, 2.2 GHz microprocessor on a standard PC for convenience. However, real implementation should be on a much powerful processor for example on a DSP for faster computation.

II. Target Model

The motion of a target being tracked is assumed to be approximately linear and modeled by the equations;

$$\underline{x}_{n+1} = \Phi \underline{x}_n + \Gamma \underline{w}_n \tag{1}$$

$$\underline{z}_{n+1} = H \underline{x}_{n+1} + \underline{v}_{n+1} \tag{2}$$

Where the state vector

$$\underline{x}_{n+1}^T = [x \quad \dot{x} \quad y \quad \dot{y}]_{n+1} \tag{3}$$

is a four-dimensional vector with x, y positions and their derivatives (velocity), $\underline{w}_n$ the two-dimensional disturbance vector, $\underline{z}_{n+1}$ the two dimensional measurement vector and $\underline{v}_{n+1}$ is the two-dimensional measurement error vector. Also Φ is

Manuscript submitted to IMECS – 2008 Hong Kong 19 – 21 March 2008, on December 07, 2008 for ICSE'08 Conference.

D. M. Akbar Hussain is with the Software Engineering and Media Technology Department, Aalborg University Denmark (e-mail: akbar@aaue.dk). He is a member of prestigious organizations like ***IEEE*** and ***IAENG***.

the assumed (4x4) state transition matrix, Γ (4x2) is the excitation matrix and H (2x4) is the measurement matrix and they are defined respectively,

$$\Phi = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$\Gamma = \begin{bmatrix} \Delta t^2/2 & 0 \\ \Delta t & 0 \\ 0 & \Delta t^2/2 \\ 0 & \Delta t \end{bmatrix} \quad (5)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (6)$$

Here Δt is the sampling interval and corresponds to the time interval (scan interval, 1 second in our investigation) assumed constant, at which radar measurement data is received.

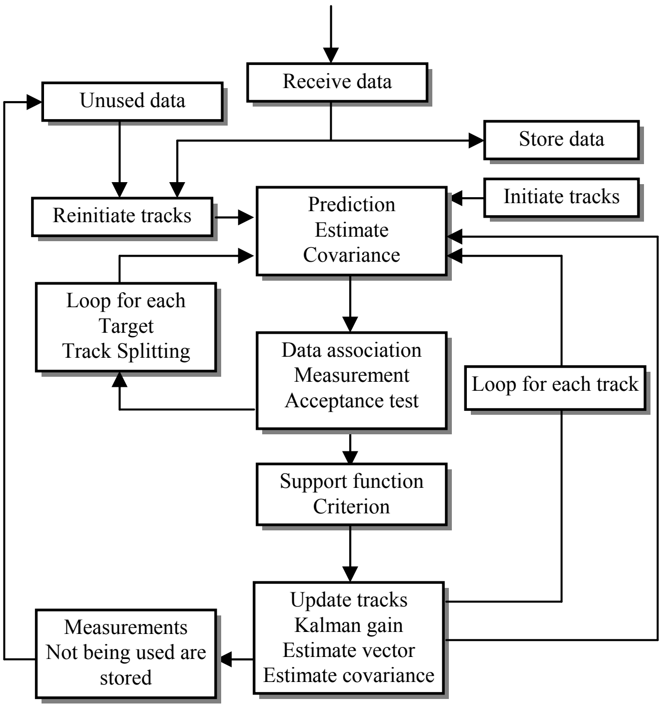

Figure 1: Recursive Multi-Target Tracking System

The system noise sequence $\underline{w}_n$ is a two dimensional Gaussian white sequence for which

$$E(\underline{w}_n) = 0 \quad (7)$$

Where E is the expectation operator. The covariance of $\underline{w}_n$ is

$$E(\underline{w}_n \quad \underline{w}_m^T) = Q_n \delta_{nm} \quad (8)$$

Where Q_n is a positive semi-definite (2x2) diagonal matrix and δ_{nm} is the Kronecker delta defined as

$$\delta_{nm} = \begin{cases} 0 & n \neq m \\ 1 & n = m \end{cases}$$

The measurement noise sequence $\underline{v}_n$ is a two-dimensional zero mean Gaussian white sequence with a covariance of

$$E(\underline{v}_n \quad \underline{v}_m^T) = R_n \delta_{nm} \quad (9)$$

where R_n is a positive semi-definite symmetric (2x2) matrix given by

$$R_n = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} \quad (10)$$

σ_x^2 and σ_y^2 are the variances in the errors of the x, y position measurements, and σ_{xy} is the covariance between the x and y measurement errors. It is assumed that the measurement noise sequence and the system noise sequence are independent of each other, that is

$$E(\underline{v}_n \quad \underline{w}_m^T) = 0 \quad (11)$$

The initial state $\underline{x}_0$ is also assumed independent of the $\underline{w}_n$ and $\underline{v}_n$ sequences that is

$$E(\underline{x}_0 \quad \underline{w}_n^T) = 0 \quad (12)$$

$$E(\underline{x}_0 \quad \underline{v}_n^T) = 0 \quad (13)$$

$\underline{x}_0$ is a four dimensional random vector with mean $E(\underline{x}_0) = \hat{\underline{x}}_{0/0}$ and a (4x4) positive semi-definite covariance matrix defined by

$$P_0 = E\left[(\underline{x}_0 - \overline{\underline{x}}_0)(\underline{x}_0 - \overline{\underline{x}}_0)^T\right] \quad (14)$$

where $\overline{\underline{x}}_0$ is the mean of the initial state $\underline{x}_0$. The Kalman filter is an optimal filter as it minimizes the mean squared error between the estimated state and the true (actual) state provided the target dynamics are correctly modeled.

The standard Kalman filter equations for estimating the position and velocity of the target motion described by equations (1) and (2) are;

$$\hat{\underline{x}}_{n+1/n} = \Phi \hat{\underline{x}}_n \quad (15)$$

$$\hat{\underline{x}}_{n+1} = \hat{\underline{x}}_{n+1/n} + K_{n+1} \underline{v}_{n+1} \quad (16)$$

$$K_{n+1} = P_{n+1/n} H^T B_{n+1}^{-1} \quad (17)$$

$$P_{n+1/n} = \Phi P_n \Phi^T + \Gamma Q_n^F \Gamma^T \quad (18)$$

$$B_{n+1} = R_{n+1} + H P_{n+1/n} H^T \quad (19)$$

$$P_{n+1} = (I - K_{n+1} H) P_{n+1/n} \quad (20)$$

$$\underline{v}_{n+1} = \underline{z}_{n+1} - H \hat{\underline{x}}_{n+1/n} \quad (21)$$

Where

$\hat{\underline{x}}_{n+1/n}$, $\hat{\underline{x}}_{n+1}$, K_{n+1}, $P_{n+1/n}$, B_{n+1} and P_{n+1}

are the predicted state, estimated state, the Kalman gain matrix, the prediction covariance matrix, the covariance matrix of innovation, and the covariance matrix of estimation respectively. Q_n^F is the covariance of the measurement noise assumed by the filter that is normally taken equal to Q_n.

In a practical situation, however, the value of $\boldsymbol{Q_n}$ is not known so the choice of $\boldsymbol{Q_n^F}$ should be such that the filter can adequately track any possible motion of the target. To start the computation an initial value is chosen for $\boldsymbol{P_0}$. Even if this is a diagonal matrix, then clearly from the above equations the covariance matrices

$$B_{n+1},\quad P_{n+1},\quad P_{n+1/n}$$

for a given $\boldsymbol{n}$ does not remain diagonal when $\boldsymbol{R_n}$ is not diagonal. When the measurement errors in each co-ordinate are independent, that is $\boldsymbol{R_n}$ is diagonal, the Kalman filter may be de-coupled into two optimal tracking filters, known as alpha-beta filters [20]. This filter configuration simplifies the computational requirements considerably, because the states relating to each of the two co-ordinates can be estimated independently. The equations for an alpha-beta filter to estimate x-position and velocity are:

$$\underline{\hat{x}}_{n+1} = \underline{\hat{x}}_{n+1/n} + \alpha_{n+1}(z\underline{\hat{x}}_{n+1} - \underline{\hat{x}}_{n+1/n}) \qquad (22)$$

$$\underline{\hat{x}}_{n+1} = \underline{\hat{x}}_{n+1/n} + \frac{\beta_{n+1}}{\Delta t}(z\underline{\hat{x}}_{n+1} - \underline{\hat{x}}_{n+1/n}) \qquad (23)$$

Where α, β defines the gain values and Δt is the time interval.

III. Track Maintenance/update

As shown on the previous page in figure 1, following the initial track formation incoming observations are considered for the continuation of existing tracks. The continuation procedure consists of prediction, measurement association and state estimation (i.e. updating). At each radar scan, the target position is predicted using eq. (15) and the uncertainty associated (eq. (21)) with this is used to place a measurement acceptance ellipse around the predicted position. If the dynamics of the assumed target model is correct then each measurement from that particular target will fall inside the predicted ellipse (measurement acceptance ellipse). However, at times incorrect measurements (that are not original returns from that particular target) may have been used to update the target. In such case the target dynamic does not remain correct and true measurement may fall outside the prediction ellipse. On the other hand, when targets are very close together, more than one measurement may fall within the prediction ellipse of a particular target. Therefore, one has to resolve such situations through various data association techniques [17][18]. The track splitting filter algorithm is one such technique in which the filter is allowed to split into the total number of measurements inside the ellipse [16]. This approach assumes that all the measurements falling inside the ellipse are equally probable for that particular target; therefore, all of them are used to update its state. Once the filter determines that a measurement has fallen inside its prediction ellipse, it uses a measurement acceptance test and if the test is satisfied then that particular measurement is used for update.

The measurement acceptance criterion uses a simple criterion i.e., if the dimension of the measurement vector $\boldsymbol{Z_n}$ eq. 2 is M, then the norm $\boldsymbol{d_n^2}$ of the innovation vector $\underline{\boldsymbol{v}}_{\boldsymbol{n}}$ at scan $\boldsymbol{n}$ for a filter is given by

$$d_n^2 = \underline{v}_n^T B_n^{-1} \underline{v}_n \qquad (24)$$

where the M-dimensional Gaussian probability density for the innovation is

$$f(\underline{v}) = \frac{e^{-\frac{d^2}{2}}}{(2\pi)^{M/2}\sqrt{|B_n|}} \qquad (25)$$

with $\boldsymbol{B_n}$ being the innovations covariance matrix for the specific filter and | $\boldsymbol{B_n}$ | its determinant. Provided that the filter model for the track dynamics is accurate and that all the measurements used to update the track did indeed originate from one particular target, the quantity $\boldsymbol{d_n^2}$ is a sum of squares of M-independent zero mean and unit standard deviation Gaussian random variables. Thus $\boldsymbol{d_n^2}$ will have a χ2 distribution with M degrees of freedom. The measurement acceptance criterion for a track is thus defined that if $\boldsymbol{d_n^2}$ is less than a threshold $\boldsymbol{J^2}$ (with some known probability) then that particular measurement at scan $\boldsymbol{n}$ can be used for update [23].

IV. Track pruning

A mechanism for restricting the excessive tracks that originate from the track splitting filter algorithm under measurement ambiguity is necessary because not only it will produce inaccurate tracking but also the computational and storage requirements increases exponentially if the interaction of ellipses occur for several scans. One such mechanism is called track support function and is given by the following relationship [23].

$$S_n = -\frac{1}{2}\sum_{i=1}^{n} d_i^2 \qquad (26)$$

Where $\boldsymbol{d_i^2}$ is given by eq. (24). The support function can be calculated recursively from

$$S_{n+1} = S_n - \frac{1}{2}(\underline{v}_{n+1}^T B_{n+1}^{-1} \underline{v}_{n+1}) \qquad (27)$$

If the support function of a track is smaller than a threshold value [23], it may not represent a true target in the sense that the measurements it has been using are inconsistent with the assumed target model. Therefore, all those tracks whose support function is lower than a given threshold are pruned.

V. Analyses

As mentioned earlier, when the track splitting filter algorithm is used, the tracking filter splits into branches which are equal to the number of measurements found within the predicted acceptance ellipse. This means the shape of the measurement ellipse is very important in the case of neighboring or crossing targets. A four crossing target scenario is considered for our investigation as shown in figure 2 for low and high correlation factors of 0.1 and 0.9 respectively.

These four targets are moving from a fixed location with same velocity and they cross each other after 30 seconds. The correlation factor is approximately kept constant throughout the run (100 Seconds) by maintaining the relative positions of the target and the platform (sensor is onboard a ship). To obtain the two correlation factors only the initial position of the platform is changed and all other parameters remain the same. The scenario was run with ten different random seeds; table 1 gives the number of average tracks present for the two filters with low and high correlation factors. As anticipated, because of the inferior measurement prediction ellipse [6], alpha-beta filter has considerably more branching near the crossing point (30^{th} seconds) for the high correlation factor. For low correlation the numbers of branches are almost the same.

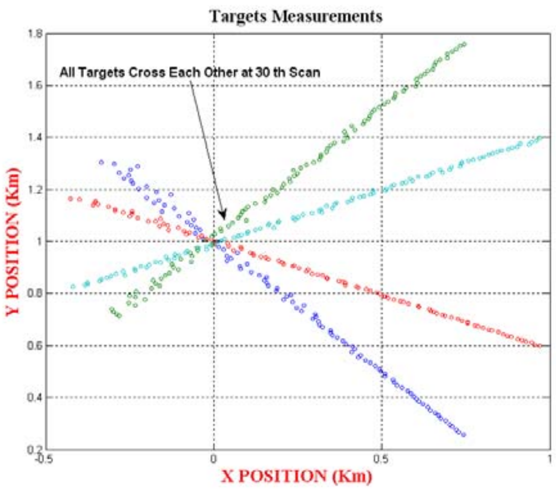

Figure 2: Crossing Target Scenario

Table 1

Scan No.	Low correlation (0.1) Kalman	High correlation (0.9) Kalman	Low correlation (0.1) Alpha-Beta	High correlation (0.9) Alpha-Beta
22	4	4	4	4
23	4	4	4	4
24	4	4	4	8
25	4	5	5	10
26	5	10	6	30
27	7	24	7	40
28	12	29	13	51
29	15	32	18	45
30	16	28	19	42
31	16	26	20	42
32	14	17	19	35
33	13	12	18	30
34	10	9	15	22
35	8	5	10	15
36	6	4	7	13
37	4	4	6	12
38	4	4	5	9
39	4	4	4	7
40	4	4	4	4

The fact of the matter is, this increased branching requires extra overhead computation for data association and track maintenance. The computation ratio for Kalman vis a v alpha-beta filter is in the order of 1 to 7 approximately. We selected a number of similar scenarios to compare the speed-up between the two filters and the average speed-up is plotted in figure 3 against various numbers of targets. It can be seen from the figure that as the ambiguity increases, the speed-up deteriorates to a ratio of 1 to 3. Therefore, the advantage of using an alpha-beta filter under high correlation is not really great. One of the main reasons for excessive branching, in the case of alpha-beta filter under high correlation, is due to the shape of the prediction ellipse which is almost like a circle around the predicted position of the track. However, the shape of prediction ellipse in case of Kalman filter is like a true ellipse aligned in the direction of the target heading as shown in figure 4. For our second part of investigation the scenario geometry was modified and the measurement data was generated corresponding to one of the target before crossing (30^{th} seconds) and the second after the crossing as shown in figure 5, duration of the tracking is 100 seconds. In this investigation we want to find how effective is the support function criterion for the two filters when correlation is high.

Figure 3: Kalman & Alpha-Beta Speed-up Comparison

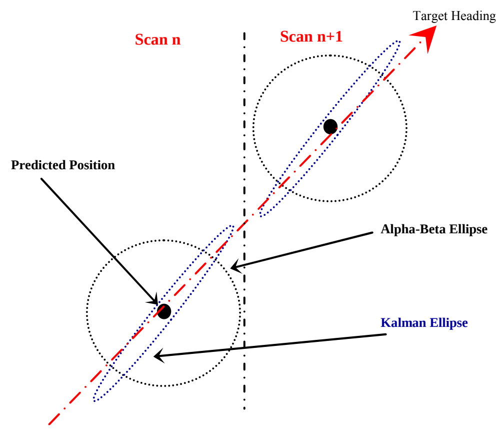

Figure 4: Prediction ellipse for Kalman & Alpha-Beta Filter

Tables 2 and 3 show the initial angles when the target starts its motion and the intersection angles when it changes its direction. The correlation ranges during tracking period for these angles are also shown. The idea is to feed different kinds of data to analyze filter's behavior. Both filters Kalman and alpha-beta filters were used to track these scenarios for the two values of correlation. Basically each run consists of 10 iterations and a new seed is selected for every iteration. The support function and the measurement acceptance values are obtained during these iterations and finally the average is computed. Figure 6 shows the value of these parameters with low correlation factors corresponding to table 2 and it can be seen that after initial track formation the target is following its path. This actually means that the measurement acceptance criterion remains less than a given probability threshold value, which in our case is 99 %. At the intersection point where a new target appears the track is lost, which would be the case by a tracking filter as the measurement from the second target will fall outside the prediction ellipse. However, in figure 7 where the correlation factor is high (corresponding to table 3), the behavior of the two filters is totally different. Kalman filter is consistent by losing the track at the intersection point but alpha-beta filter kept on tracking the target assuming it is the best supported track.

VI. Conclusions

In this paper we have compared the relative merits of the optimal Kalman filter with the sub-optimal alpha-beta filter. The interesting point for investigation is the amount of correlation during tracking period. Much investigation has been carried out in determining the position error accuracy of these two filters that reveals there is not much difference between the two estimates. However, one aspect which has not been given much attention in the past is the shape of the prediction ellipse under different correlation factors. It has been demonstrated by our investigation that in multi-target environments containing neighboring as well as crossing targets, more branching occur in the case of the alpha-beta filter due to the shape of the prediction ellipse. It has been shown that in a high correlation scenario, the de-coupled alpha-beta filter is more likely to accept unrealistic measurements compared with the Kalman filter. Therefore, the speed of computation when using an alpha-beta filter in a multi-target scenario is not high as one would predict from single target considerations. In fact it was found to be only 3 to 4 times faster than a standard Kalman filter for crossing target scenarios containing up to 10 targets. However, one important aspect must be kept in mind that number of branches depends on couple of factors, for example the time the target cross each other and their angle of intersection at the time of crossing. In future we would like to carry out our research work for more in-depth analyses of these two filters considering the results obtained in our investigation here plus other more realistic scenarios.

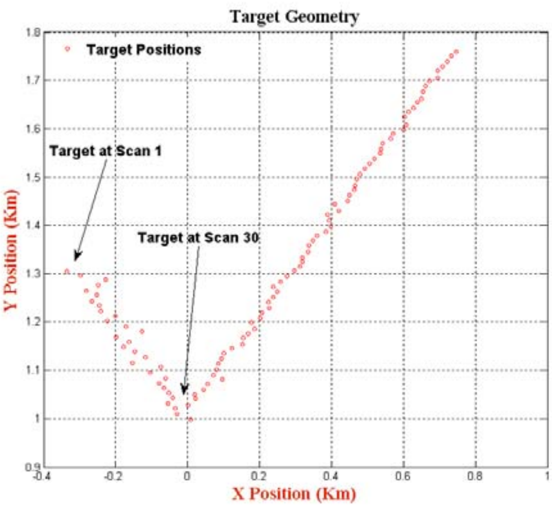

Figure 5: Scenario Geometry

Table 2

Scenario no.	Initial angle	Intersection angle	Correlation range	Average correlation
1	45°	90°	0.02-0.20	0.17
2	90°	135°	0.01-0.18	0.05
3	90°	180°	0.02-0.50	0.30
4	45°	135°	0.02-0.33	0.27

Table 3

Scenario no.	Initial angle	Intersection angle	Correlation range	Average correlation
1	45°	90°	0.90-0.99	0.99
2	90°	135°	0.97-0.99	0.99
3	90°	180°	0.93-0.99	0.99
4	45°	135°	0.98-0.99	0.99

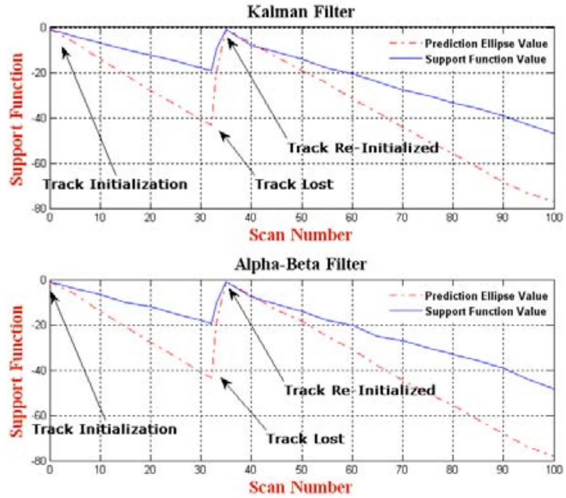

Figure 6: Low Correlation Behavior

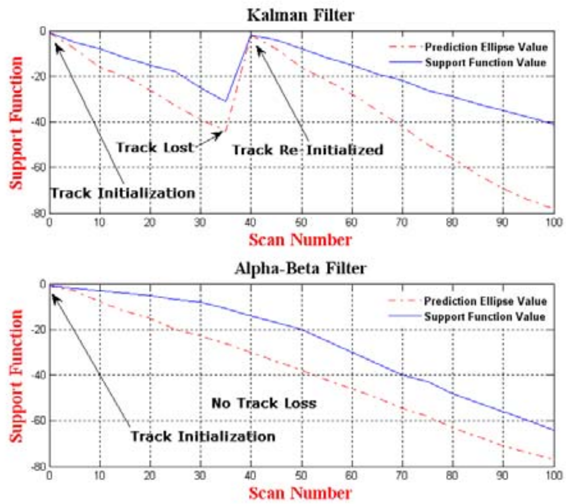

Figure 7: High Correlation Behavior

References

[1] Glen W. Mabey and Jacob Gunther: A Robust Motion-Estimation Algorithm for Multiple-Target Tracking at Close Proximity Based on Hexagonal Partitioning, July 2003, Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance AVSS '03, Publisher: IEEE Computer Society.

[2] David Hall: Lectures in Multi-sensor Data Fusion and Target Tracking, March 2001, Book, Publisher: Artech House, Inc.

[3] Lawrence D. Stone, Thomas L. Corwin, and Carl A. Barlow: Bayesian Multiple Target Tracking, August 1999 Book Publisher: Artech House, Inc.

[4] Peter Nillius, Josephine Sullivan, and Stefan Carlsson: Multi-Target Tracking - Linking Identities using Bayesian Network Inference, June 2006, Proceedings of the 2006 IEEE Computer Society Conference on

Computer Vision and Pattern Recognition - Volume 2 CVPR '06, Publisher: IEEE Computer Society.

[5] Ahmet G. Pakfiliz and Murat Efe: Multi-Target Tracking in Clutter with Histogram Probabilistic Multi-Hypothesis Tracker, August 2005, Proceedings of the 18th International Conference on Systems Engineering ICSENG '05, Publisher: IEEE Computer Society.

[6] Junji Satake and Takeshi Shakunaga: Multiple Target Tracking by Appearance-Based Condensation Tracker using Structure Information, August 2004, Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03 ICPR '04 , Publisher: IEEE Computer Society.

[7] Hyong-suk Kim, Hong-rak Son, Young-jae Lim and Jae-chul Chung: Target Tracking via Region-Based Confidence Computation with the CNN-UM, December 2002, Proceedings of the Third IEEE Pacific Rim Conference on Multimedia: Advances in Multimedia Information Processing PCM '02, Publisher: Springer-Verlag.

[8] Duncan Smith and Sameer Singh: Approaches to Multi-sensor Data Fusion in Target Tracking, A Survey, December 2006, IEEE Transactions on Knowledge and Data Engineering, Volume 18 Issue 12, Publisher: IEEE Educational Activities Department.

[9] Bum-Jik Lee, Jin-Bae Park and Young-Hoon Joo: IMM Algorithm Using Intelligent Input Estimation for Maneuvering Target Tracking, May 2005, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Volume E88-A Issue 5 Publisher: Oxford University Press.

[10] QingE Wu, Tuo Wang, YongXuan Huang and JiSheng Li: Application of Fuzzy Automata on Target Tracking, Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007) Volume 02, Pages: 437-441, 2007, ISBN:0-7695-2874-0.

[11] David Choi and Benjamin Roy: A Generalized Kalman Filter for Fixed Point Approximation and Efficient Temporal-Difference Learning, April 2006 Discrete Event Dynamic Systems, Volume 16 Issue 2, Publisher: Kluwer Academic.

[12] Jon Whittle and Johann Schumann: Automating the implementation of Kalman filter algorithms, December 2004, ACM Transactions on Mathematical Software (TOMS), Volume 30 Issue 4 Publisher: ACM Press.

[13] R. Vasquez and J. Mayora: Estimation of motion and position of a rigid object using a sequence of images (tridimensional Kalman filter approach), April 1991, Proceedings of the 19th annual conference on Computer Science CSC '91 Publisher: ACM Press.

[14] Jae-Chern Yoo and Young-Soo Kim: Alpha-beta-tracking index (α-β-Λ) tracking filter, January 2003, Signal Processing, Volume 83 Issue 1 Publisher: Elsevier North-Holland, Inc.

[15] P. R. Kalata and K. M. Murphy: Alpha-Beta Target Tracking and Track Rate Variations, Proceedings of the 29th Southeastern Symposium on System Theory (SSST '97) SSST '97, Publisher: IEEE Computer Society.

[16] P. L. Smith and G. Buechler: A branching algorithm for discriminating and tracking multiple objects, IEEE Transactions Automatic Control, Vol. AC-20, February 1975, pp 101-104.

[17] Y. Bar-Shalom and T. E. Fortmann: Tracking and Data Association, Academic Press, Inc. 1988.

[18] S. S. Blackman: Multiple Target Tracking with Radar Applications, Artech House, Inc. 1986.

[19] D. P. Atherton, E. Gul, A. Kountzeris and M. Kharbouch: Tracking Multiple Targets using parallel processing ", Proc. IEE, Part D, No. 4, July 1990, pp 225-234.

[20] A. W. Bridgewater: Analysis of second and third order steady state tracking filters, AGARD conference proceedings No. 252, Moterey, CA, Oct. 1978, pp 9-1 to 9-11.

[21] D. M. Akbar Hussain, Zhenyu Yang, Youmin Zhang: Improved strategy to reduce bottleneck in distributed systems, Proceedings of International Conference on Information, INFO-2004, November 29 - December 2, Hosei University Tokyo Japan (ISBN: 4-901329-02-2).

[22] D. M. Akbar Hussain: Motion Model Employment using interacting Motion Model Algorithm, Proceedings of International Control Conference "Control2006" Glasgow Scotland 30 August – 1 September 2006.

[23] D. P. Atherton, D. M. Akbar Hussain and E. Gul: Target tracking using transputers as parallel processors, 9th IFAC symposium on identification and system parameter estimation, Budapest Hungary July 1991.