Tracking optimisation for multifunction radar

M Kanetos, K Woodbridge and C Baker

University College London Department of Electronic and Electrical Engineering

Abstract: The analysis compares the time efficiency of the the aß and aß? filters. It is concluded by simulation on different trajectories that the aß? filter requires fewer updates than the aß filter when tracking accelerating targets, while offering superior tracking accuracy. Furthermore the effect of taking M successive looks at the target and then look again after N sampling intervals is examined

1 Introduction

In conventional track-while-scan radars with mechanically rotated antennas, the radar has to allocate the time in each azimuth section in a pre-determined fashion between the different tasks that must be executed. The advent of the phased array radar with the capability to electronically steer the main beam contributed to overcome many of the problems and limitations inherent to traditional TWS systems.

A Multi-Function Radar (MFR) combines the electronic steering of the antenna beam with the use of computer control to replace a number of conventional sensors. This enables the MFR to carry out volume surveillance, tracking, missile communications and aircraft support and navigation. Thus the MFR attempts to substitute a number of sensors, each of which dedicates all of its time in performing a single function. Therefore the issue of time management acquires a pivotal role in optimising the performance of the MFR.

In recent years a considerable amount of effort has been put in investigating ways to optimise the time efficiency of phased arrays when carrying out tracking. Although many of the algorithms developed to provide a smoothed estimate of the target position employ a fixed time update interval, there have also been attempts to vary the update interval according to the dynamics of each track. For example, when a target travels in a straight line a large update interval is sufficient. When the target is manoeuvring the update interval is reduced to compensate for this.

The issue of calculating the optimum update interval has attracted much attention [1-4]. Cohen demonstrates an algorithm based on the aß filter which varies the update interval according to the residual in the prediction of the target position. It is shown that the update interval reduces at the points where the target manoeuvres, while at parts of smooth motion it is maintained at its maximum value. This leads to improved time efficiency compared with the simple aß filter that uses a fixed update interval. Wilkin [4] extends Cohen's work by introducing additional mechanisms that allow the update interval to be maintained at its maximum allowed value for longer periods.

The aim of this work is to compare the time efficiency of the aß and aß? filter for tracking. Firstly an algorithm is introduced that allows the sampling interval in the two filters to vary. The filters are then compared for five different computer-generated trajectories, by examining the number of updates each filter had to perform and the mean error. Then the performance of the two filters is assessed by taking M successive samples of the target position and then looking back after missing the following N samples. The variation of error for different M/N ratios is examined and the performance of the two filters for the same M/N ratios is compared.

2 aß and aß? filters for constant update time interval

Many tracking algorithms are based on the Kalman filter. Simplifications can be made to the general Kalman predictor-corrector filter which lead, for the second- order Kalman filter that assumes constant velocity between updates, to the alpha-beta filter [3]. The aß filter provides prediction of the position given by:

rp(n) = rs(n-1)+Tvs(n-1) (1)

while for the velocity we have:
vp(n) = vs(n-1) (2)

where rp is the Predicted Position, rs is the Smoothed Position, vp is the Predicted Velocity, vs the Smoothed Velocity and T is the update time interval. The filtering equations of the aß tracker provide smoothed estimates of the target's position and velocity, as given by:

rs(n) = rp(n) + a [rm(n) - rp(n)] (3)
vs(n) = vs(n-1) +(ß/T) [rm(n) - rp(n)] (4)

where a, ß are constants and rm(n) is the measurement of the position at data point n. Bar-Shalom [1] shows that the smoothing coefficients a and ß are given by:

a = v(2ß) - ß/2 (5)
ß = 2(2- a) - 4v(1- a) (6)

It has been shown [2, 3, 4] that the value of a that optimally balances short-term noise reduction and rapid response to manoeuvres is approximately 0.5. Thus from (6) we have ß = 0.172. Similarly by applying the same simplifications to the third order Kalman filter which assumes constant acceleration between updates, one obtains the aß? filter. The aß? filter provides prediction of the position and velocity given by:

rp(n) = rs(n-1) + Tvs(n-1) +0.5(T^2)as(n-1) (7)
vp(n) = vs(n-1) + Tas(n-1) (8)
where as is the Smoothed Acceleration , for which we have: ap(n) = as(n-1) (9)

ap being the Predicted Acceleration. The filtering equations of the aß? filter provide smoothed estimates for the position, velocity and acceleration:

rs(n) = rp(n) + a [rm(n) - rp(n)] (10)
vs(n) = vp(n) + (ß/T) [rm(n) - rp(n)] (11)
as(n)= as(n-1) + (?/T^2) [rm(n) – rp(n)] (12)

Similarly to the aß filter Bar-Shalom shows that (5) and (6) are valid for the aß? filter for calculating a and ß, and that ? is given by:
? = (ß^2)/ a (13)
where by using a = 0.5 and ß = 0.172, then ? = 0.029.

3 aß and aß? filters with variable update time

Cohen [2] shows that instead of maintaining a constant update interval for the aß filter it is more efficient to allow for it to vary. For the aß filter the update time interval T(n) is shown to be given by:
T(n) = T(n-1)/ve(n) (16)
where T(n-1) is the previous update time interval and e(n) the error in the position prediction . For the aß? filter one has:
T(n) = T(n-1) / ? e(n) (18)
Also by normalising the error e(n) in (16) and (18) to the noise standard deviation s the update time interval remains unchanged when the error is equal to s [2]. Thus:
en(n) = e(n)/s, (19) ?
For the aß filter we have: T(n) = T(n-1)/ v(en(n)) (20)
And for the aß? filter: T(n) = T(n-1)/? en(n) (21)
Furthermore, Cohen argues that T(n) as given by (20) and (21) will vary rapidly even in the non-manoeuvring sections of a trajectory due to the randomness of the noise. To overcome this, Cohen suggests the use of a first order filter to smooth the residual as follows:
es(n) = es(n-1) +ar [e(n) – es(n-1)] (22)

This is used in (19) to find the normalised residual. It is also shown that smoothing the residual as in (22) leads to a faster increase of T(n) after the manoeuvre has ended. The value of the gain ar depends

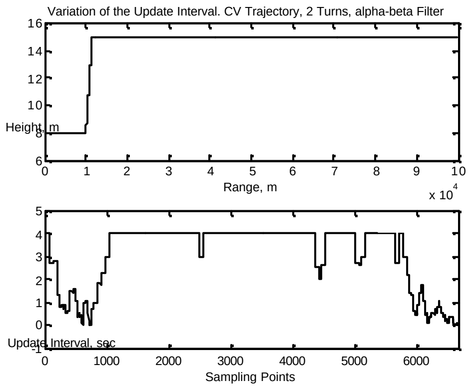

Figure 1 Demonstration of the time-varying algorithm for the aß filter.

on the specific track, but generally Wilkins [4] suggests values between 0.5 and 0.9. A value of ar=0.75 is used for the purposes of this work. Also Wilkin demonstrates that the residual in (22) should be computed as:

er(n)=v([rmx(n)-rpx(n)]^2+[rmy(n)-rpy(n)]^2) (24)

Equations (24), (22), (19), (20), (21) and (23) are used to calculate the update time interval in the programs written. Figure 1 demonstrates the variation of the update interval during a constant velocity trajectory with two turns. T(n) is small initially, then it decreases again due to noise and then again at the manoeuvring part.

4 Simulation to assess the performance of the aß and aß? filters

4.1 Comparison of the aß and aß? filters with variable update time

In order to investigate the time efficiency of the two filters simulation was performed using five computer-generated trajectories. The first is for constant velocity, while the remaining four are for constant acceleration. Details of the trajectories are included in Table 1.The results obtained for the aß filter are summarised in Table 2 and those for the aß? filter in Table 3. Measurement noise was added directly on the x and y co-ordinates, as Cohen suggests. Also the error is measured by (24). The noise standard deviation was 1.098m. The results shown in Tables 2 and 3 are the average of multiple runs for each case.

Trajectory	Description	Initial Velocity (m/sec)	Acceleration (g)	Duration (sec)
1	Straight line	750	0	135
2	Straight line	20	2	500
3	Straight line	20	0.2	500
4	Circular 180° turn	10	1	254
5	Circular 180° turn	10	2	537

Table 1: Details of the trajectories used for simulation

Trajectory	1	2	3	4	5
Number of updates	63	622	249	214	544
Error (m)	1.3096	64.4844	9.2734	796.2702	1.4993*10^3

Table 2: Results for the aß filter

Trajectory	1	2	3	4	5
Number of updates	73	176	154	130	305
Error (m)	1.8877	13.9293	2.3758	324.0662	287.3181

Table 3: Results for the aß? filter

For Trajectory 1 the aß tracker, as expected, performed better with fewer updates and smaller error. However for the rest of the trajectories, where acceleration is present, the aß? filter performs considerably better that the aß filter. For example, for Trajectory 2 it required a three times fewer updates while at the same time the error is maintained at significantly lower levels. Similarly for Trajectories 3, 4 and 5 the aß? filter achieves a lower number of updates, which are combined with improved tracking accuracy.

4.2 Varying the time looking at the data and the time looking away.

For the second part of this work the target is tracked by taking M successive looks, spaced 1 sec apart, and then take the next measurement after N sec. The M/N ratio was varied to investigate the effect on tracking accuracy. Figure 2 includes the results obtained for different values of M and N. Here the aß filter is used to track Trajectory 2. From Figure 2 it appears that the error reduces drastically when the M/N ratio is increased from 1 to 3. However maintaining M/N higher than 3 appears to offer little improvement.

It is also interesting to note that for the same M/N ratio, the lower the values of M and N the lower the error is. Thus if for example one is tracking two targets, then a lower error is achieved by sharing the available time to look at each target for 10 sec, rather than for 20 or 30. Figure 3 shows that for the same M/N ratio there can be large differences in the error for different trajectories. Figure 4 attempts to compare the performance of the aß and the aß? filters for the same M/N ratio. For this comparison Trajectory 3 is used where acceleration is present and it appears that the aß? filter has considerably smaller error.

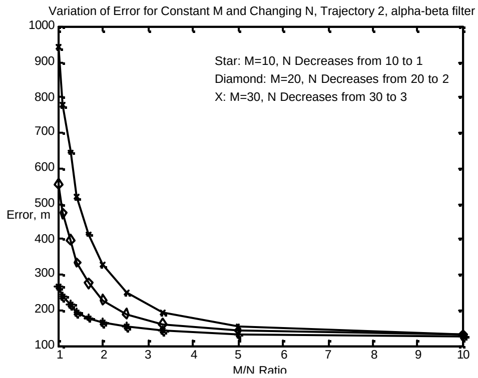

Figure 2: Effect of Different M/N ratios on tracking accuracy

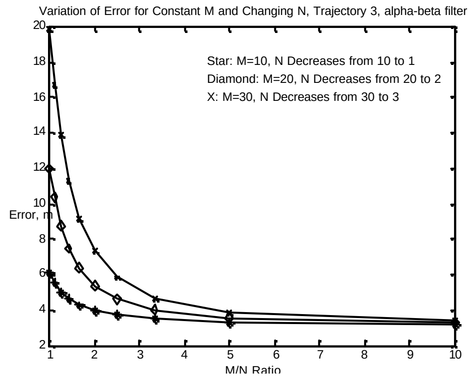

Figure 3: Tracking Trajectory 3 with the aß filter and the same M/N ratio as in Figure 2

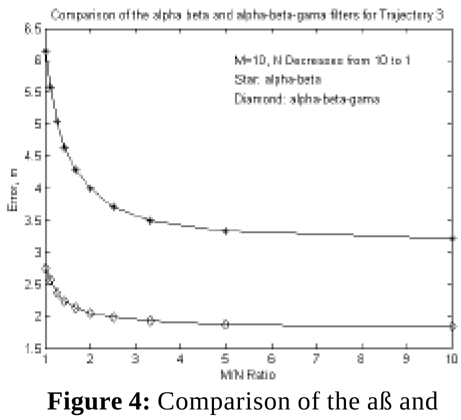

Figure 4: Comparison of the aß and the aß? filters

5 Conclusion

The time efficiency of the aß and the aß? filters has been compared, by using variable update time for both filters. The aß? filter uses fewer updates than the the aß filter when tracking accelerating targets. This is combined with significantly improved tracking accuracy. Also the tracking error has been investigated by taking M successive looks at the target and looking back after N update intervals. It was found that the error reduces considerably when the M/N ratio is increased from 1 to 3, while little improvement is achieved from values of M/N higher than 3. Also, for the same M/N ratio a smaller error is obtained when M and N are kept small. As an example, M=10 and N=3 generates a smaller error than using M=20 and N=6. Finally the aß? filter was found to have a smaller error than the aß filter when tracking accelerating targets with the same M/N ratio.

References

[1]BAR-SHALOM, Y. Xiao-Rong Li, :'Estimation and tracking: Principles, techniques and software', Boston, Artech House, 1993

[2]COHEN, S.A, :'Adaptive variable update rate algorithm for tracking targets with a phased array radar', IEE Proc. Vol. 133, Pt F, No. 3 JUNE 1986

[3]DAWSON, A, :'Use of time varying update time for target tracking using both position and radial velocity measurements', 1996, Nottingham University Theses, George Green Library

[4]Wilkin, D.J, Harisson, I. and Woolfson, M.S, :"Target Tracking Algorithms for Phased Array Radar", IEE Proceedings-F, Vol. 138, No.3, June 1991, pp. 255-262