

https://dspace.org/

The system design of a radar track-while-scan facility utilizing modern digital techniques

Henery, Michael

2015-01-26
http://hdl.handle.net/10539/16757

Downloaded from a DSpace repository. https://dspace.org/

$(\alpha+\beta)^2 < 4\beta$ - underdamped

$(\alpha+\beta)^2 = 4\beta$ - critically damped

$(\alpha+\beta)^2 > 4\beta$ - overdamped

Figure 3.6 illustrates the above regions.

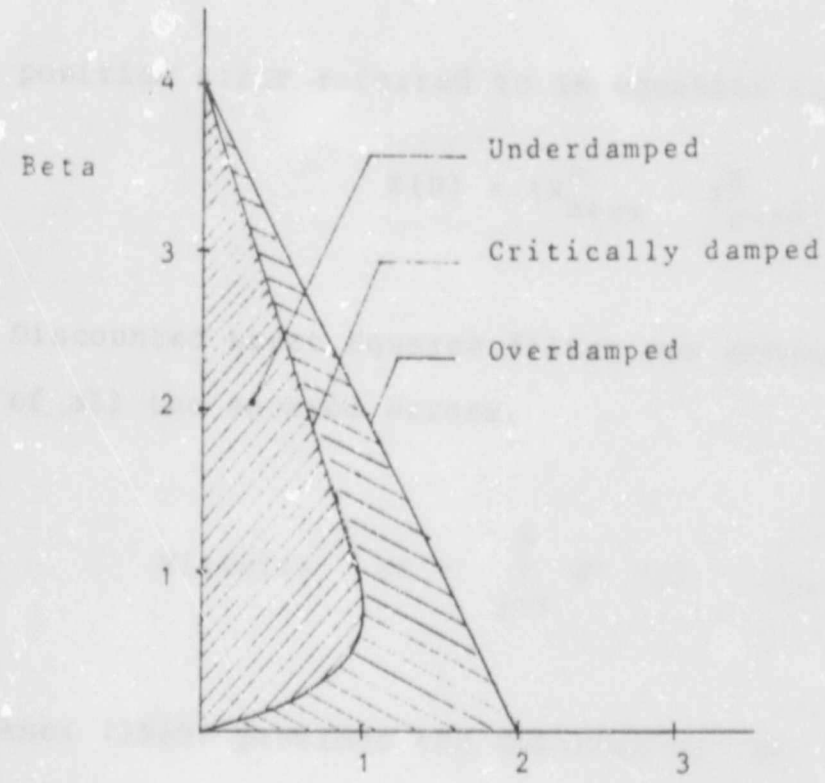


Figure 3.6 The Alpha-Beta Filter Response Regions

From Van Der Merwe (1981) it may be shown that the steady state lag is given by the equation

$$\text{Lag} = \ddot{x}T^2(1-\alpha)/\beta \qquad (3.11)$$

Thus it is concluded that the filter will track a constant velocity target with zero steady state error and will lag a constant acceleration target by a fixed amount in the steady state. The dynamic lag versus noise smoothing compromise is similar to that of the Alpha filter.

3.3.2.1 The Discounted Least Squares Alpha-Beta Filter

The position error referred to in equation (3.9) is

$$E(N) = (X^N_{meas} - X^N_{pred})$$

The Discounted Least Squares filter now attempts to minimise the sum of all the squared errors.

$$\text{Minimise} \quad SS = \sum_{j=1}^{N} E^2(j) \qquad (3.12)$$

Brookner (1980) provides the solution as an Alpha-Beta filter with

$$\alpha = 1-\theta^2$$

$$\beta = (1-\theta)^2 \qquad (3.13)$$

Eliminating θ from the equations one concludes that

$$\alpha = 2\sqrt{\beta} - \beta \qquad (3.14)$$

This indicates that the filter is critically damped. The filter design now depends only on the selection of a single parameter beta.

The locus of the critically damped filter on the alpha-beta plane may be seen in figure 3.7.

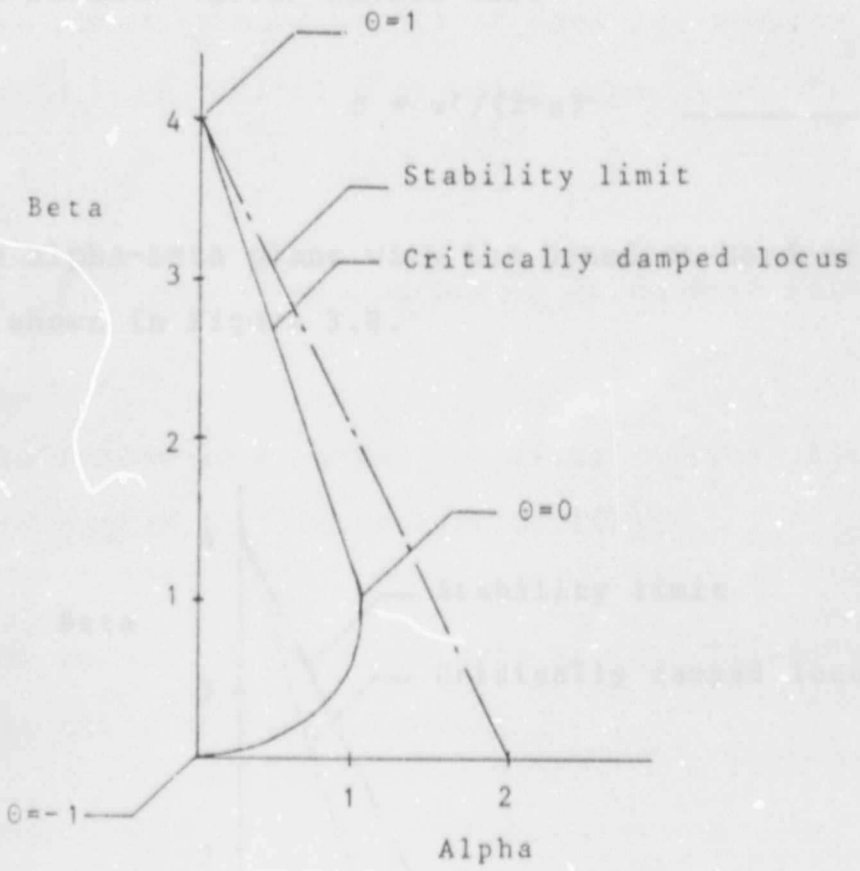


Figure 3.7 Locus for the Critically Damped Alpha-Beta Filter

For this filter Alpha cannot exceed unity, at which point $\theta=0$, and the filter has zero memory (no noise smoothing). If $\theta=1$ the filter has infinite memory (very heavy noise smoothing with large dynamic lag).

3.3.2.2 The Benedict - Bordner Alpha-Beta Filter

In order for the alpha-beta filter to provide the best transient following capability for a constant velocity target while simultaneously providing the best minimum variance estimate of position and velocity of any fixed parameter filter, Benedict and Bordner (1962) showed that

$$\beta = \alpha^2/(2-\alpha) \qquad (3.15)$$

The alpha-beta plane with the Benedict-Bordner locus filled in is shown in figure 3.8.

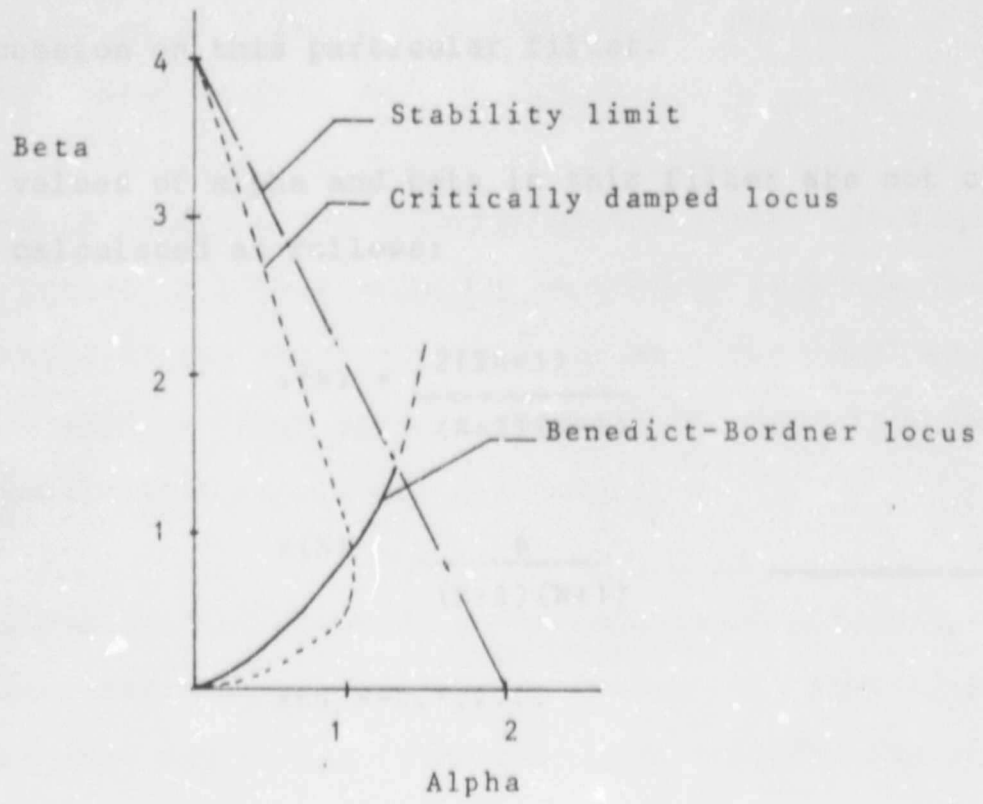


Figure 3.8 Locus for the Benedict-Bordner Condition

For values of alpha>1 the filter is overdamped and for values of alpha<1 the filter is underdamped. This particular filter does not offer better performance than the critically damped filter. However, it is optimal with respect to all first order constant gain filters and it is thus not necessary to look for more exotic filters of the same order.

The Benedict-Bordner filter does not specify an optimal value of alpha, the selection of which depends on the system application.

3.3.2.3 The Expanding Memory Alpha-Beta Filter

The reader is referred to Miller (1981) for a more detailed discussion on this particular filter.

The values of alpha and beta in this filter are not constant but are calculated as follows:

$$\alpha(N) = \frac{2(2N+1)}{(N+2)(N+1)}$$

$$\beta(N) \quad \frac{6}{(N+2)(N+1)} \qquad \text{(3.16)}$$

and $N=0,1,2\ldots$

A plot of alpha and beta versus N is given in figure 3.9.

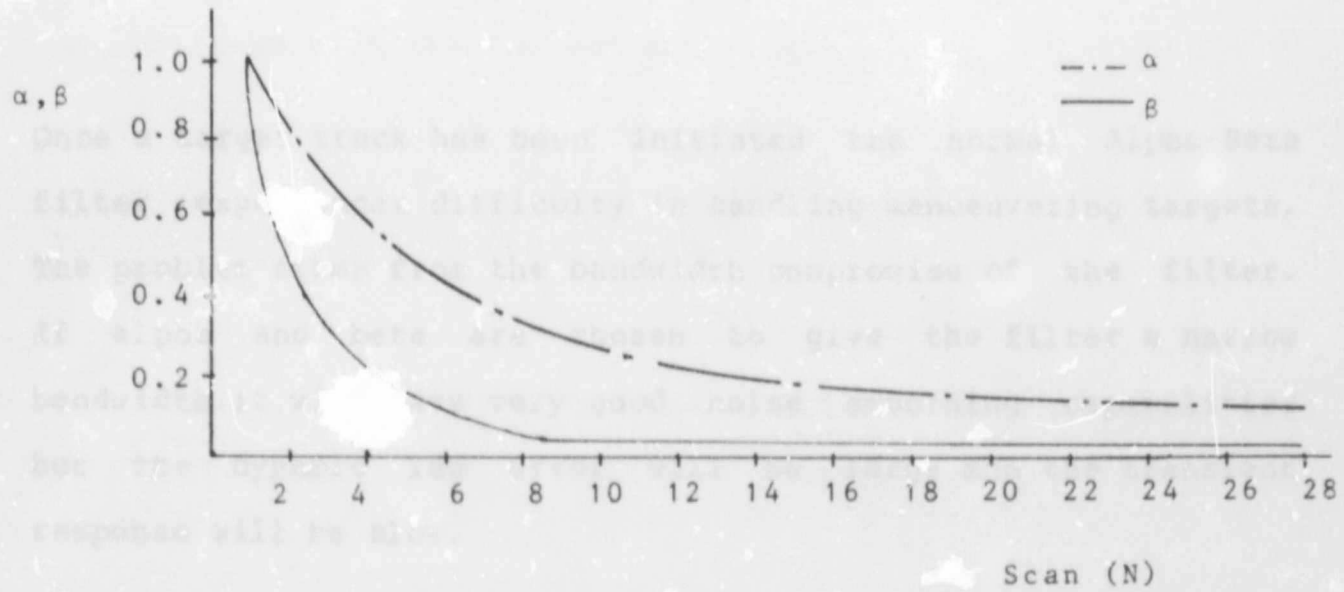


Figure 3.9 Plot of α,β Versus N for the Expanding Memory Filter

As the value of N increases the gains decrease: the filter memory increases.

One would normally use this filter for track initiation since one places a higher emphasis on dynamic response than on noise smoothing during the acquisition phase. The ideal would be to start with a high dynamic response and gradually reduce it as the acquisition transients die down.

The gains are not allowed to decrease indefinitely but will be frozen after a predetermined number of iterations so as to ensure that the filter retains a measure of dynamic response in the event of target manoeuvres at a later stage.

3.3.2.4 Adaptive Alpha-Beta Filters

Once a target track has been initiated the normal Alpha-Beta filter experiences difficulty in handling manoeuvering targets. The problem stems from the bandwidth compromise of the filter. If alpha and beta are chosen to give the filter a narrow bandwidth it will have very good noise smoothing capabilities but the dynamic lag error will be large and the transient response will be slow.

A possible solution to this shortcoming is to adapt the values of alpha and beta during filter operation to suit the target manoeuvres and thereby to ensure the best possible compromise at all times.

Van der Merwe (1988) suggests an Alpha-Beta filter that provides narrow bandwidth tracking for non-accelerating targets and hence good noise smoothing is available, while it also provides wide bandwidth tracking for manoeuvering targets. Essentially the filter is allowed to choose between discrete bandwidths. The bandwidth selection is based upon an estimate of the tracking filter lag calculated at set intervals, say every ten or twenty filter updates. If the lag exceeds a certain upper threshold the bandwidth is increased and conversely if it falls below a lower threshold the bandwidth is decreased. If the lag remains within the two thresholds the bandwidth is unaffected. Everytime the lag causes the change to a new bandwidth new values of alpha and beta are calculated. The bandwidth may only

be increased or decreased by one level at a time and the filter is initiated with the largest bandwidth.

3.3.3 The Alpha-Beta-Gamma Filter

The reader is referred to Brookner (1980) and Skolnik (1980) for more information on this particular filter.

This filter is analogous to the Alpha-Beta filter except that it assumes a constant acceleration target.

The transition matrix is as given in equation 3.3.

The alpha-beta-gamma tracking equations may be written as

$$\text{Smoothing:} \quad X^N_{smooth} = X^N_{pred} + \alpha(X^N_{meas} - X^N_{pred})$$

$$V^N_{smooth} = V^{N-1}_{pred} + \beta/T(X^N_{meas} - X^N_{pred})$$

$$A^N_{smooth} = A^{N-1}_{pred} + 2\gamma/T^2(X^N_{meas} - X^N_{pred})$$

$$\text{Predicting:} \quad X^N_{pred} = X^N_{smooth} + TV^N_{smooth}$$

$$V^N_{pred} = V^N_{smooth} + TA^N_{smooth}$$

$$A^N_{pred} = A^N_{smooth} \qquad \text{(3.17)}$$

where A is the acceleration.

Again, by analogy to Benedict-Bordner's work it can be shown that for the filter to be optimal the following conditions should be valid:

$$\alpha(\alpha+\beta+\gamma/2) = 2\beta$$

$$\beta^2 = 2\alpha\gamma \qquad (3.18)$$

Hence filter design entails the choice of a single parameter, the remaining parameters then being calculated using the above equations. In practice the Alpha-Beta-Gamma filter seems to be avoided, some authors being of the opinion that it offers worse performance than the Alpha- Beta filter. The main reason for this is that no realistic target can be expected to perform uniform acceleration for any appreciable length of time.

3.3.4 The Kalman Filter

The reader is referred to works by Stark and Tuteur (1979), Bozic (1979) and Costella and Dunnebacke (1974) for more information on the filter.

The Kalman filter is inherently adaptive and can be applied to any system that is Markoffian in the wide sense. To a large degree Kalman filtering represents the state of the art as far as linear mean square estimation is concerned. Not only does the Kalman filter minimise the mean square error, but if Gaussian distributions are assumed, then the Kalman filter is

also a maximum liklihood filter and a minimum covariance filter.

Like the Alpha-Beta filter, the Kalman filter is a recursive fading memory filter, but in addition it possesses the significant advantage that its parameters adapt to the changing target dynamics and also provide a measure of how well it is estimating the target position and velocity. The Kalman filter is the optimum filter when the statistical properties of both the target dynamics and the corrupting noise are known. The Kalman filter provides a recursive, adaptive solution to

$$X(N) = a_1 X^N_{meas} + a_2 X^{N-1}_{meas} + \ldots \qquad (3.19)$$

Being recursive the Kalman filter has an infinite impulse response.

A form of the Kalman filter is shown in figure 3.10

From the diagram one can see that:

$$X(k) = AX(k-1) + W(k-1)$$

$$Y(k) = CX(k) + n(k) \qquad (3.20)$$

Note that these equations, while being written in first order matrix form, describe the system and measurement models for an N order system as

$$X(N) = \begin{bmatrix} X_1(N) \\ X_2(N-1) \\ \vdots \\ X_3(N-K) \end{bmatrix} \qquad (3.21)$$

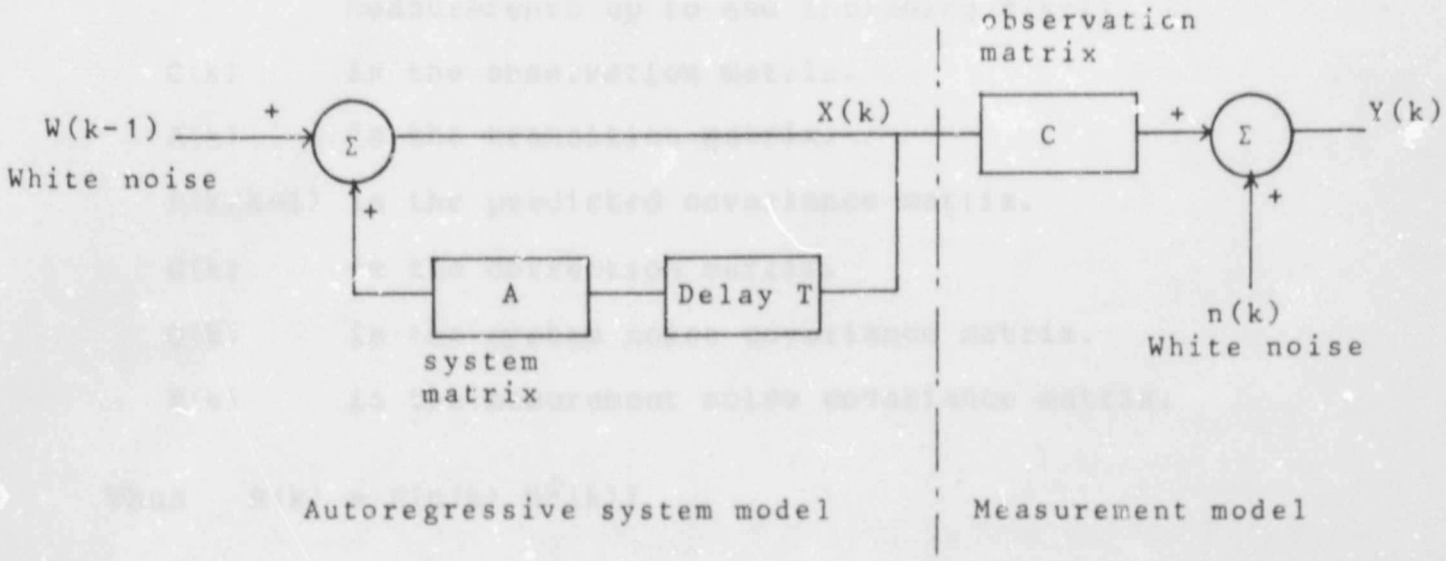


Figure 3.10 A General Form of the Kalman Filter

Sheats (1975) and Bozic (1979) show that the (k+1)th derivative of the state variable $X_1(n)$ is modelled as being white noise.

The Kalman filter algorithm is as follows:

$$\hat{X}(k/k-1) = A\hat{X}(k-1/k-1)$$

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

$$\hat{X}(k/k) = \hat{X}(k/k-1) + G(k)(V(k)-C\hat{X}(k/k-1))$$

$$P(k/k) = P(k/k-1) - G(k)PC(k/k-1)$$

$$G(k) = P(k/k-1)C^T(CP(k/k-1)C^T + R(k))^{-1} \qquad (3.22)$$

where:

X(k/k) is the estimate of the state vector based on measurements up to and including Y(k)

X(k/k-1) is the prediction of the state vector based on

measurements up to and including Y(k-1)

C(k) is the observation matrix.

A(k) is the transition matrix.

P(k/k-1) is the predicted covariance matrix.

G(k) is the correction matrix.

Q(k) is the system noise covariance matrix.

R(k) is the mesurement noise covariance matrix.

Thus $R(k) = E[n(k)\ n^T(k)]$

$Q(k) = E[W(k)\ W^T(k)]$ from Bozic (1979).

It may be seen from the above equations that:

a) It is necessary to only store the previous estimate of the state vector rather than all previous outputs.

b) Only the physical model A(k) and C(k) and the statistics of the random processes Q(k) and R(k) need be defined.

Figure 3.11 illustrates the computational stages in the Kalman Filter.

Figure 3.12 details the computational steps in the Kalman filter.

The filter operates in a "predict-correct" fashion. The error term is added to the predicted estimate in order to determine the filtered estimate.

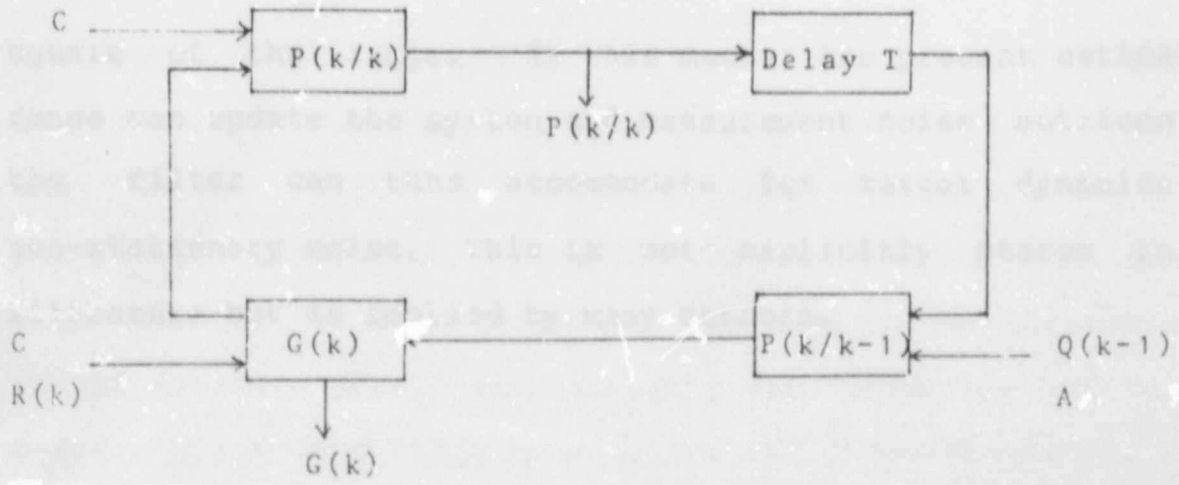


Figure 3.11 Comp[illegible]l Stages in the Kalman Filter

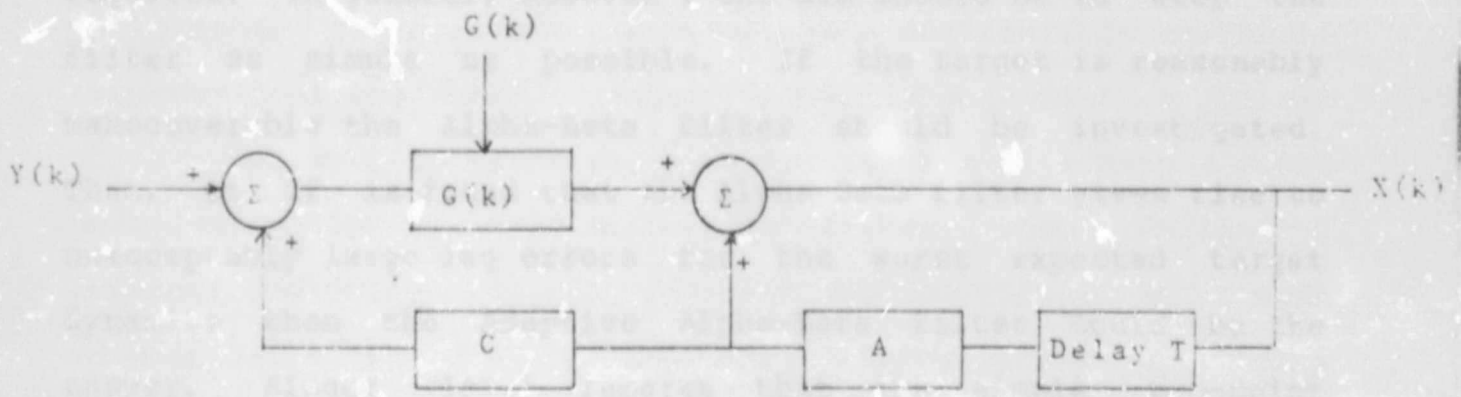


Figure 3.12 Computational Steps in the Kalman Filter

In a practical system the observation and transition matrices will be time invariant. However the system and measurement noise matrices are likely to be a function of range. The system noise will largely be due to geometric target dynamics which decrease with increasing range. Barton (1964) showed that the range measurement noise is inversely proportional to the square root of the signal to noise ratio which is proportional to the

square of the range. In this manner the present estimate of range can update the system and measurement noise matrices and the filter can thus accommodate for target dynamics and non-stationary noise. This is not explicitly stated in the literature but is implied by many authors.

3.4 Tracking Filter Design

Perhaps the most important design consideration in the realisation of tracking filters is deciding what exactly is required. In general, however , the aim should be to keep the filter as simple as possible. If the target is reasonably manoeuverable the Alpha-Beta filter should be investigated. Then, it if is found that the Alpha-Beta filter gives rise to unacceptably large lag errors for the worst expected target dynamics then the Adaptive Alpha-Beta filter could be the answer. Singer (1970) remarks that the simple two-point extrapolator is in many cases justified.

From the previous section it is also noted that if the target is expected to accelerate uniforming for extended periods of time then it may be necessary to investigate the use of an Alpha-Beta-Gamma filter.

Only if the filter performance specifications are very tight and the target dynamics highly variable should the Kalman filter be used. It should, however, be kept in mind that in order to

realise the filter one must know the statistics of both the target dynamics and the corrupting noise. The price paid for the Kalman filter's superior performance consists of a formidable computational load and that the filter may diverge, thus causing a loss of track. Divergence may occur when the target dynamics depart significantly from those assumed in the design stage and this usually escapes discovery in the design stage but crops up in the field. This is due to the fact that the filter is exercised via a Monte Carlo simulation against target dynamics for which the filter was designed.

Implementation of the simplest filter which meets the systems performance requirements can result in a considerable saving in computer time and memory. Singer (1970) shows that a Kalman filter designed to track manoeuvering targets will have a worse performance than a simple Alpha-Beta filter, when tracking a constant velocity target. One must thus be certain as to what is preferable in the system: not losing track of highly manoeuvering targets or good performance against most other targets.

4.0 HOW THE TARGET IS TRACKED

4.1 The Tracking System

The preferred display for most two-dimensional surveillance radars is the PPI display. All targets on this display are indicated with repect to slant range and azimuth relative to the radar. The data thus displayed on the PPI display is in R-θ coordinates. Morrison (1978) has explained that the tracking algorithms are best implemented if the aircraft trajectory is described in Catesian co-ordinates. It is therefore necessary to convert tne radar R-θ representation of the target position to X- Y coordinates. Two tracking filters are then required to track the aircraft: one filter implements position and velocity smoothing and position prediction in the X-direction, while the other filter implements the smoothing and prediction in the Y-direction. The outputs of the filters are then combined and converted to R-θ coordinates and the predicted position is fed to the PPI display as the tracking symbol position. This predicted position is also used to form the position around which the tracking window is formed.

4.2 Target Designation

The radar operator would typically have in front of him a roller ball or joystick which would allow him to manipulate the

designation marker position on the PPI display. This symbol position would typically be manipulated in an X-Y fashion.

The control of the designation marker via the joystick or roller ball would either be direct positioning of the marker in response to the input position, or else an incremental positioning where the designation marker would move in the direction required and motion would stop if the input device were to be released.

When the operator detects a target on the PPI display he would position the designation symbol over the target and signal to the processor that a track is to be initiated. The accuracy with which he is required to position the symbol is determined by the size of the acquisition window set up by the filter. Section 4.3.2 considers the derivation of this window size. The accuracy also affects the rate at which the filter is able to reduce the tracking error and accurately start predicting the target position. These requirements place fairly limited possibilities on the shape and the size of the designation marker and care should be exercised when choosing the marker dimensions.

4.3 Track Initiation

Once the processor has been instructed that a track is to be initiated two operations are required. Firstly the track file is to be opened and secondly the acquisition window is to be set

up.

4.3.1 Opening The Track File

When the track file is set up the tracking filter has to be initialised. If the data starts to arrive as of time N=0, then because of the recursive structure of the filter the initial estimates of the target position and velocity at N-1 need to be known. A common approach is to label the first two measurements as X(-1) and X(0) and then to set

$$X_{pred}^{N=0} = (X_{meas}^{N=N-1} + X_{meas}^{N=0})/2$$

$$\dot{X}_{pred}^{N=0} = (X_{meas}^{N=0} - X_{meas}^{N=N-1})/T \qquad \text{(4.1)}$$

and to co mence the estimate from the third measurement onwards.

Although the initial estimates are likely to be poor, particularly the velocity estimate, if stable the filter will progressively reduce the errors until they have negligible effect on the filter outputs. In some cases it may be wiser to initialise the filter with zero velocity and the designated target position as the first predicted position.

4.3.2 Setting Up The Acquisition Window

The range and azimuth extent of the acquisition window about the designated target position depend on the expected target manoeuvres within the following sampling period.

Appendix C contains a review of the expected target dynamics.

One can see that the maximum aircraft speeds do not vary much for the jet aircraft. However, the maximum accelerations that may be encountered for civilian aircraft fall way below those of the military aircraft. Typical figures for jet aircraft are velocities of between 150 m/s and 250 m/s with maximum accelerations of 2g and for propeller aircraft velocities of between 30 m/s and 120 m/s with maximum accelerations of 2g.

Appendix D contains a mathematical derivation of the acquisition window size.

It should be apparent that for the typical aircraft expected, be they either military or civilian, the size of the window is only affected by the expected velocity of the target in any direction if a constant sample rate is assumed.

For a radar not using pulse compression techniques the frequency of the system clock is related to the width of the transmitted pulse by the relationship

$$\text{Clock frequency} = \frac{1}{\text{pulse width}} \text{ Hz.} \qquad (4.2)$$

So, as the pulse length decreases, the radar system clock frequency increases. If the processor cannot handle the data rate at which the echoes are received while the beam scans through the tracking window, the tracking window should then be set up using hardware and the data captured in memory. When the beam has swept through the window the processor may then leisurely access and process the data ready for the next scan. Also, if a single processor is being used to execute multiple tracks then it is essential to have a means of setting up the individual windows in hardware and to have a memory associated with each window. When the beam has scanned across the window, the hardware may then signal to the processor that the data is ready.

4.4 Track Prediction

Once the radar beam has scanned through the window, the hardware signals to the processor that the data is ready. This data will be a record of the range and azimuth position of each and every echo within the window. If the signal processor has efficiently done its job then the echoes should only be from the target. Small patches of clutter do occassionally break through however. The first stage then is to sort out a particular position, in range and azimuth, which the target is measured to occupy. This sorting procedure is the resolution of track ambiguity problem addressed in section 2.5.1.

Figure 4.1 illustrates a possible set of echo returns within the window.

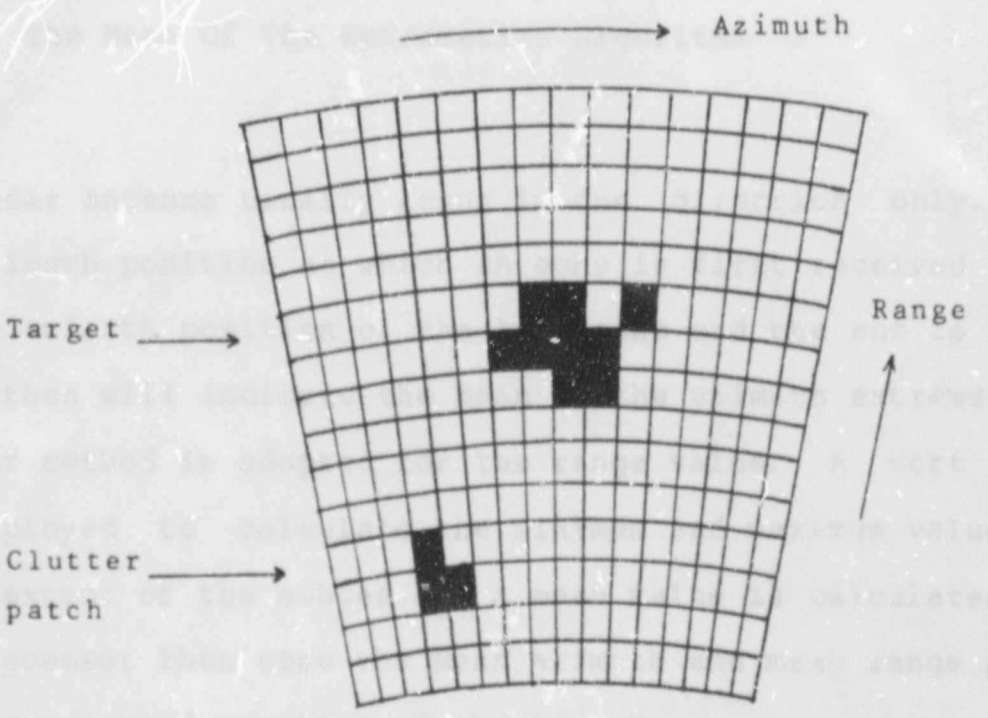


Figure 4.1 A Possible Set of Echo Returns Within the Window

Appendix E contains a derivation of the possible number of echoes received for a given radar system.

One can see that for a typical target size the number of echoes received is of the order of tens of pulses. This indicates that the storage areas for the target position do not need to be vast and that the sort processes will not occupy a large proportion of the processor time.

Various schemes have been suggested to resolve track ambiguity, three of them being discussed below.

4.4.1 The Mean Of The Extremeties Algorithm

The radar antenna usually scans in one direction only. Hence the azimuth position at which an echo is first received is added to the azimuth position of the last echo and the sum is halved. This then will indicate the mean of the azimuth extremeties. A similar method is adopted for the range value. A sort routine is employed to calculate the minimum and maximum value of the range extent of the echoes and a mean value is calculated. The TWS processor then uses the mean azimuth and mean range position for the measured position of the target.

Mathematically this may be expressed as:

$$\text{Mean azimuth} = (\text{Azimuth}_{min} + \text{Azimuth}_{max})/2$$

$$\text{Mean range} = (\text{Range}_{min} + \text{Range}_{max})/2 \quad \text{——} \quad (4.3)$$

Figure 4.2 illustrates the measured position using the above scheme for the echoes received as in figure 4.1.

Author Henery Michael
Name of thesis The System Design Of A Radar Track-while-scan Facility Utilizing Modern Digital Techniques. 1985

PUBLISHER:

University of the Witwatersrand, Johannesburg
©2013

LEGAL NOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed, or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page) for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.