Chapter 11 *Target Tracking*

Part I: Single Target Tracking

Tracking radar systems are used to measure the target's relative position in range, azimuth angle, elevation angle, and velocity. Then, by using and keeping track of these measured parameters the radar can predict their future values. Target tracking is important to military radars as well as to most civilian radars. In military radars, tracking is responsible for fire control and missile guidance; in fact, missile guidance is almost impossible without proper target tracking. Commercial radar systems, such as civilian airport traffic control radars, may utilize tracking as a means of controlling incoming and departing airplanes.

Tracking techniques can be divided into range/velocity tracking and angle tracking. It is also customary to distinguish between continuous single-target tracking radars and multi-target track-while-scan (TWS) radars. Tracking radars utilize pencil beam (very narrow) antenna patterns. It is for this reason that a separate search radar is needed to facilitate target acquisition by the tracker. Still, the tracking radar has to search the volume where the target's presence is suspected. For this purpose, tracking radars use special search patterns, such as helical, T.V. raster, cluster, and spiral patterns, to name a few.

11.1. Angle Tracking

Angle tracking is concerned with generating continuous measurements of the target's angular position in the azimuth and elevation coordinates. The accuracy of early generation angle tracking radars depended heavily on the size of the pencil beam employed. Most modern radar systems achieve very fine angular measurements by utilizing monopulse tracking techniques.

Tracking radars use the angular deviation from the antenna main axis of the target within the beam to generate an error signal. This deviation is normally measured from the antenna's main axis. The resultant error signal describes how much the target has deviated from the beam main axis. Then, the beam position is continuously changed in an attempt to produce a zero error signal. If the radar beam is normal to the target (maximum gain), then the target angular position would be the same as that of the beam. In practice, this is rarely the case.

In order to be able to quickly achieve changing the beam position, the error signal needs to be a linear function of the deviation angle. It can be shown that this condition requires the beam's axis to be squinted by some angle (squint angle) off the antenna's main axis.

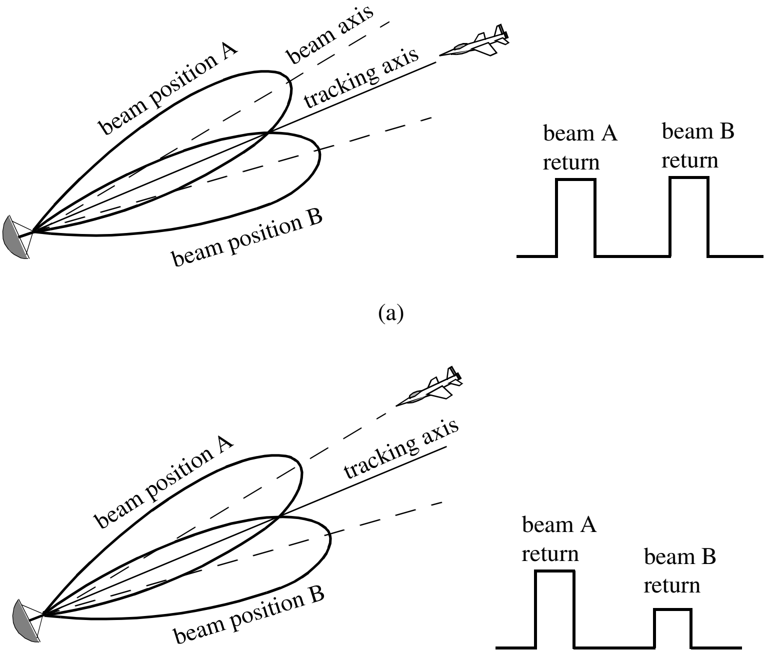
11.1.1. Sequential Lobing

Sequential lobing is one of the first tracking techniques that was utilized by the early generation of radar systems. Sequential lobing is often referred to as lobe switching or sequential switching. It has a tracking accuracy that is limited by the pencil beam width used and by the noise caused by either mechanical or electronic switching mechanisms. However, it is very simple to implement. The pencil beam used in sequential lobing must be symmetrical (equal azimuth and elevation beam widths).

Tracking is achieved (in one coordinate) by continuously switching the pencil beam between two pre-determined symmetrical positions around the antenna's Line of Sight (LOS) axis. Hence, the name sequential lobing is adopted. The LOS is called the radar tracking axis, as illustrated in Fig. 11.1.

As the beam is switched between the two positions, the radar measures the returned signal levels. The difference between the two measured signal levels is used to compute the angular error signal. For example, when the target is tracked on the tracking axis, as the case in Fig. 11.1a, the voltage difference is zero and, hence, is also the error signal. However, when the target is off the tracking axis, as in Fig. 11.1b, a nonzero error signal is produced. The sign of the voltage difference determines the direction in which the antenna must be moved. Keep in mind, the goal here is to make the voltage difference be equal to zero.

In order to obtain the angular error in the orthogonal coordinate, two more switching positions are required for that coordinate. Thus, tracking in two coordinates can be accomplished by using a cluster of four antennas (two for each coordinate) or by a cluster of five antennas. In the latter case, the middle antenna is used to transmit, while the other four are used to receive.



(a)

(b)

Figure 11.1. Sequential lobing. (a) Target is located on track axis. (b) Target is off track axis.

11.1.2. Conical Scan

Conical scan is a logical extension of sequential lobing where, in this case, the antenna is continuously rotated at an offset angle, or has a feed that is rotated about the antenna's main axis. Fig. 11.2 shows a typical conical scan beam. The beam scan frequency, in radians per second, is denoted as ω_s. The angle between the antenna's LOS and the rotation axis is the squint angle φ. The antenna's beam position is continuously changed so that the target will always be on the tracking axis.

Fig. 11.3 shows a simplified conical scan radar system. The envelope detector is used to extract the return signal amplitude and the Automatic Gain Control (AGC) tries to hold the receiver output to a constant value. Since the AGC operates on large time constants, it can hold the average signal level constant and still preserve the signal rapid scan variation. It follows that the tracking error signals (azimuth and elevation) are functions of the target's RCS; they are functions of its angular position with the main beam axis.

In order to illustrate how conical scan tracking is achieved, we will first consider the case shown in Fig. 11.4. In this case, as the antenna rotates around the tracking axis all target returns have the same amplitude (zero error signal). Thus, no further action is required.

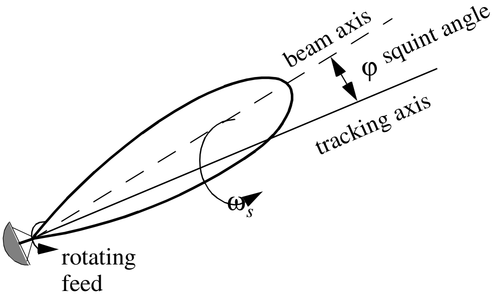


Figure 11.2. Conical scan beam.

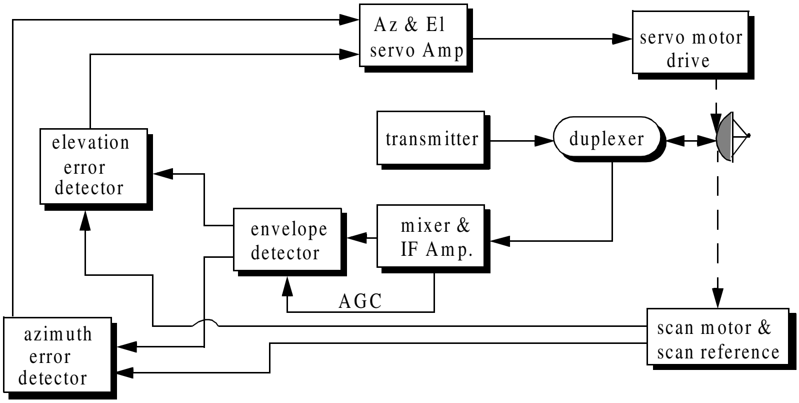


Figure 11.3. Simplified conical scan radar system.

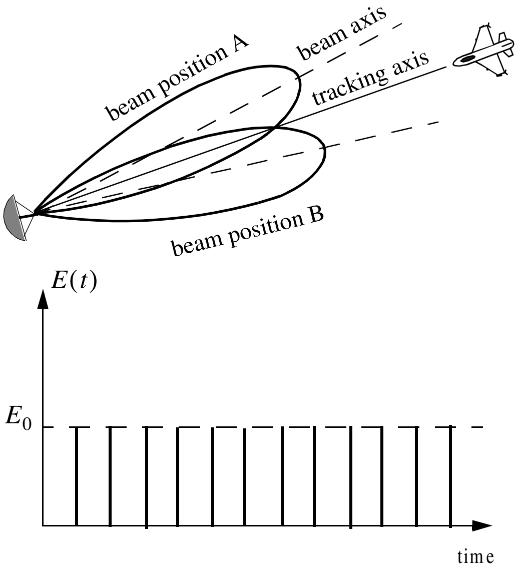


Figure 11.4. Error signal produced when the target is on the tracking axis for conical scan.

Next, consider the case depicted by Fig. 11.5. Here, when the beam is at position B, returns from the target will have maximum amplitude. And when the antenna is at position A, returns from the target have minimum amplitude. Between those two positions, the amplitude of the target returns will vary between the maximum value at position B, and the minimum value at position A. In other words, Amplitude Modulation (AM) exists on top of the returned signal. This AM envelope corresponds to the relative position of the target within the beam. Thus, the extracted AM envelope can be used to derive a servo-control system in order to position the target on the tracking axis.

Now, let us derive the error signal expression that is used to drive the servo-control system. Consider the top view of the beam axis location shown in Fig. 11.6. Assume that $t = 0$ is the starting beam position. The locations for maximum and minimum target returns are also identified. The quantity ε defines the distance between the target location and the antenna's tracking axis. It follows that the azimuth and elevation errors are, respectively, given by

$$\varepsilon_a = \varepsilon \sin\varphi \qquad \textbf{(11.1)}$$

$$\varepsilon_e = \varepsilon \cos\varphi \qquad \textbf{(11.2)}$$

These are the error signals that the radar uses to align the tracking axis on the target.

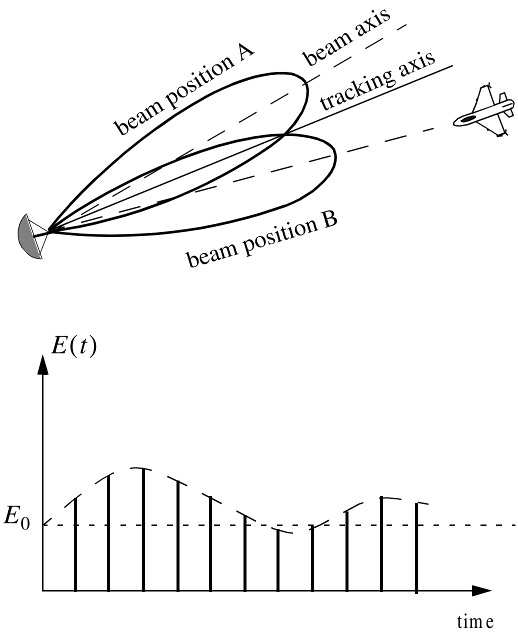


Figure 11.5. Error signal produced when the target is off the tracking axis for conical scan.

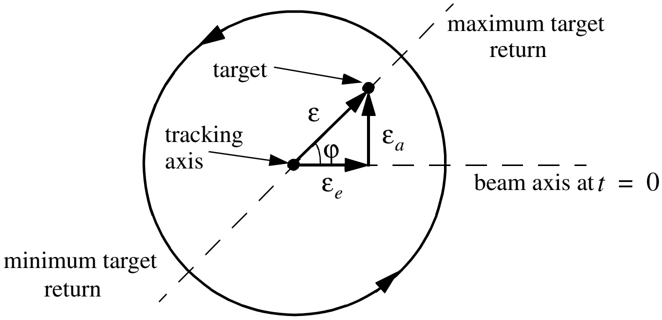


Figure 11.6. Top view of beam axis for a complete scan.

The AM signal $E(t)$ can then be written as

$$E(t) = E_0 \cos(\omega_s t - \varphi) = E_0 \varepsilon_e \cos \omega_s t + E_0 \varepsilon_a \sin \omega_s t \tag{11.3}$$

where E_0 is a constant called the error slope, ω_s is the scan frequency in radians per seconds, and φ is the angle already defined. The scan reference is the signal that the radar generates to keep track of the antenna's position around a complete path (scan). The elevation error signal is obtained by mixing the signal $E(t)$ with $\cos \omega_s t$ (the reference signal) followed by low pass filtering. More precisely,

$$E_e(t) = E_0 \cos(\omega_s t - \varphi) \cos \omega_s t = -\frac{1}{2} E_0 \cos \varphi + \frac{1}{2} \cos(2\omega_s t - \varphi) \tag{11.4}$$

and after low pass filtering we get

$$E_e(t) = -\frac{1}{2} E_0 \cos \varphi \tag{11.5}$$

Negative elevation error drives the antenna beam downward, while positive elevation error drives the antenna beam upward. Similarly, the azimuth error signal is obtained by multiplying $E(t)$ by $\sin \omega_s t$ followed by low pass filtering. It follows that

$$E_a(t) = \frac{1}{2} E_0 \sin \varphi \tag{11.6}$$

The antenna scan rate is limited by the scanning mechanism (mechanical or electronic), where electronic scanning is much faster and more accurate than mechanical scan. In either case, the radar needs at least four target returns to be able to determine the target azimuth and elevation coordinates (two returns per coordinate). Therefore, the maximum conical scan rate is equal to one fourth of the PRF. Rates as high as 30 scans per seconds are commonly used.

The conical scan squint angle needs to be large enough so that a good error signal can be measured. However, due to the squint angle, the antenna gain in the direction of the tracking axis is less than maximum. Thus, when the target is in track (located on the tracking axis), the SNR suffers a loss equal to the drop in the antenna gain. This loss is known as the squint or crossover loss. The squint angle is normally chosen such that the two-way (transmit and receive) crossover loss is less than a few decibels.

11.2. Amplitude Comparison Monopulse

Amplitude comparison monopulse tracking is similar to lobing in the sense that four squinted beams are required to measure the target's angular position. The difference is that the four beams are generated simultaneously rather than sequentially. For this purpose, a special antenna feed is utilized such that the four beams are produced using a single pulse, hence the name "monopulse." Additionally, monopulse tracking is more accurate and is not susceptible to lobing anomalies, such as AM jamming and gain inversion ECM. Finally, in sequential and conical lobing variations in the radar echoes degrade the tracking accuracy; however, this is not a problem for monopulse techniques since a single pulse is used to produce the error signals. Monopulse tracking radars can employ both antenna reflectors as well as phased array antennas.

Fig. 11.7 show a typical monopulse antenna pattern. The four beams A, B, C, and D represent the four conical scan beam positions. Four feeds, mainly horns, are used to produce the monopulse antenna pattern. Amplitude monopulse processing requires that the four signals have the same phase and different amplitudes.

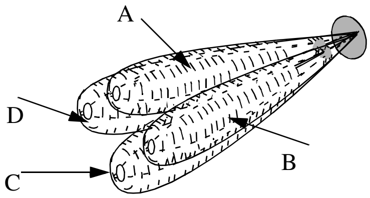


Figure 11.7. Monopulse antenna pattern.

A good way to explain the concept of amplitude monopulse technique is to represent the target echo signal by a circle centered at the antenna's tracking axis, as illustrated by Fig. 11.8a, where the four quadrants represent the four beams. In this case, the four horns receive an equal amount of energy, which indicates that the target is located on the antenna's tracking axis. However, when the target is off the tracking axis (Figs. 11.8b-d), an unbalance of energy occurs in the different beams. This unbalance of energy is used to generate an error signal that drives the servo-control system. Monopulse processing consists of computing a sum Σ and two difference Δ (azimuth and elevation) antenna patterns. Then by dividing a Δ channel voltage by the Σ channel voltage, the angle of the signal can be determined.

The radar continuously compares the amplitudes and phases of all beam returns to sense the amount of target displacement off the tracking axis. It is critical that the phases of the four signals be constant in both transmit and receive modes. For this purpose, either digital networks or microwave comparator circuitry are utilized. Fig. 11.9 shows a block diagram for a typical microwave comparator, where the three receiver channels are declared as the sum channel, elevation angle difference channel, and azimuth angle difference channel.

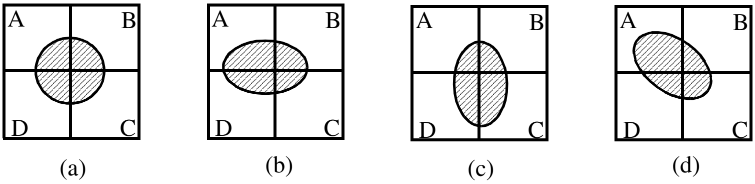


Figure 11.8. Illustration of monopulse concept. (a) Target is on the tracking axis. (b) - (d) Target is off the tracking axis.

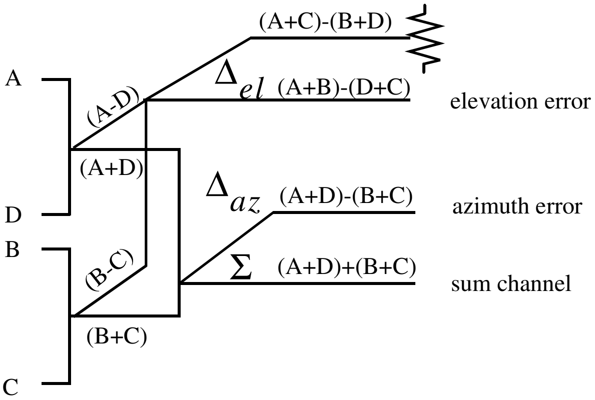


Figure 11.9. Monopulse comparator.

To generate the elevation difference beam, one can use the beam difference (A−D) or (B−C). However, by first forming the sum patterns (A+B) and (D+C) and then computing the difference (A+B)−(D+C), we achieve a stronger elevation difference signal, Δ_{el}. Similarly, by first forming the sum patterns (A+D) and (B+C) and then computing the difference (A+D)−(B+C), a stronger azimuth difference signal, Δ_{az}, is produced.

A simplified monopulse radar block diagram is shown in Fig. 11.10. The sum channel is used for both transmit and receive. In the receiving mode the sum channel provides the phase reference for the other two difference channels. Range measurements can also be obtained from the sum channel. In order to illustrate how the sum and difference antenna patterns are formed, we will assume a $\sin\varphi/\varphi$ single element antenna pattern and squint angle φ_0. The sum signal in one coordinate (azimuth or elevation) is then given by

$$\Sigma(\varphi) = \frac{\sin(\varphi - \varphi_0)}{(\varphi - \varphi_0)} + \frac{\sin(\varphi + \varphi_0)}{(\varphi + \varphi_0)} \tag{11.7}$$

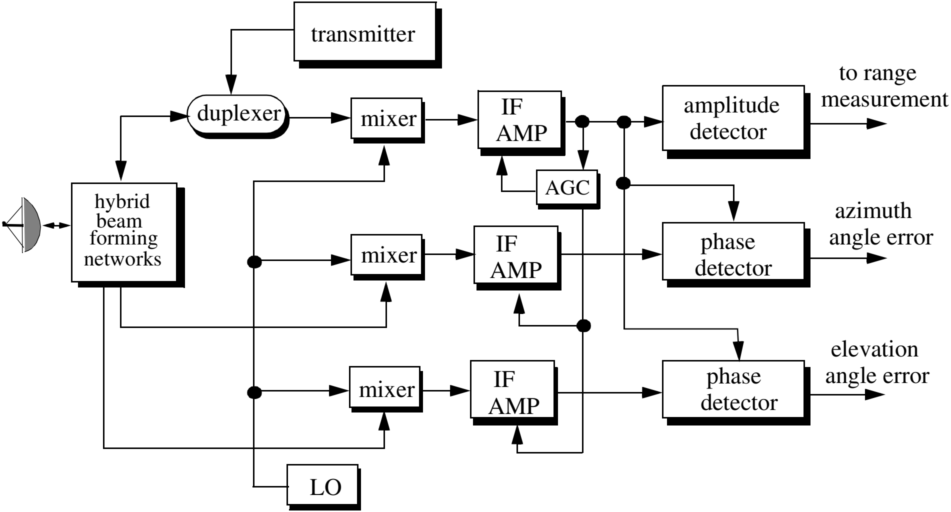


Figure 11.10. Simplified amplitude comparison monopulse radar block diagram.

and a difference signal in the same coordinate is

$$\Delta(\varphi) = \frac{\sin(\varphi - \varphi_0)}{(\varphi - \varphi_0)} - \frac{\sin(\varphi + \varphi_0)}{(\varphi + \varphi_0)} \tag{11.8}$$

MATLAB Function "mono_pulse.m"

The function "mono_pulse.m" implements Eqs. (11.7) and (11.8). Its output includes plots of the sum and difference antenna patterns as well as the difference-to-sum ratio. It is given in Listing 11.1 in Section 11.10. The syntax is as follows:

mono_pulse (phi0)

where *phi0* is the squint angle in radians.

Fig. 11.11 (a-c) shows the corresponding plots for the sum and difference patterns for $\varphi_0 = 0.15$ radians. Fig. 11.12 (a-c) is similar to Fig. 11.11, except in this case $\varphi_0 = 0.75$ radians. Clearly, the sum and difference patterns depend heavily on the squint angle. Using a relatively small squint angle produces a better sum pattern than that resulting from a larger angle. Additionally, the difference pattern slope is steeper for the small squint angle.

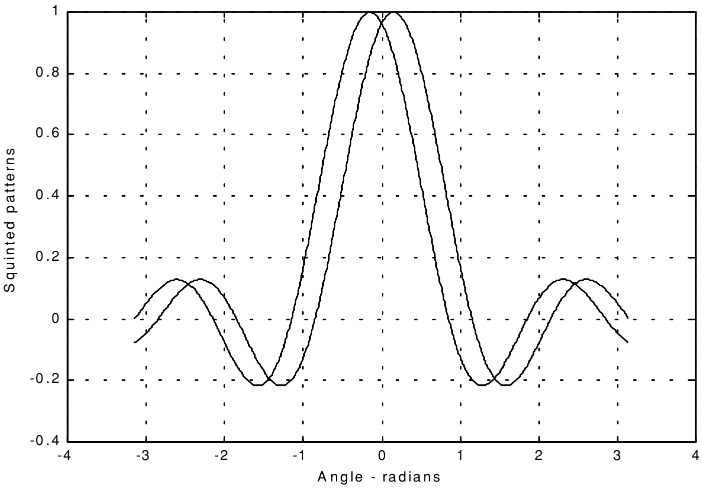


Figure 11.11a. Two squinted patterns. Squint angle is $\varphi_0 = 0.15$ radians.

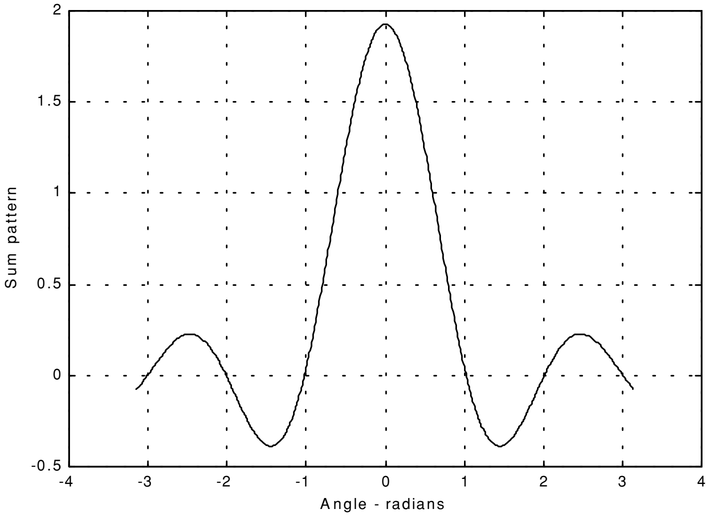


Figure 11.11b. Sum pattern corresponding to Fig. 11.11a.

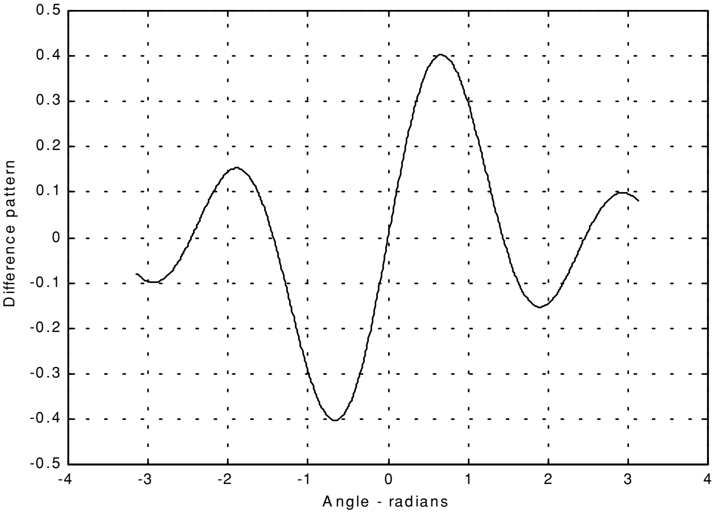


Figure 11.11c. Difference pattern corresponding to Fig. 11.11a.

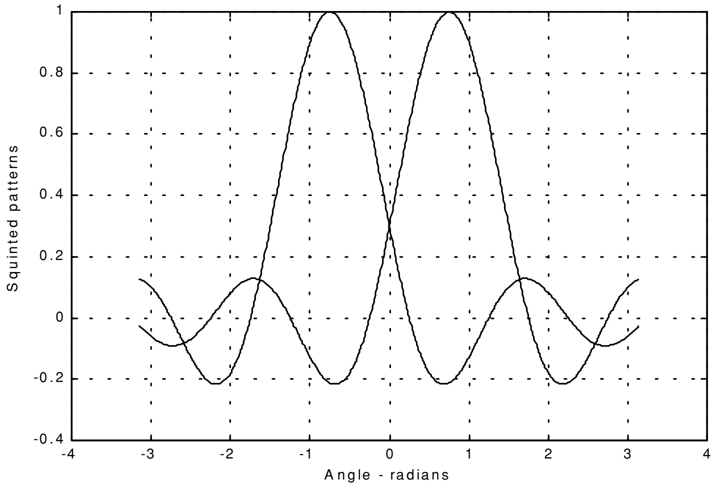


Figure 11.12a. Two squinted patterns. Squint angle is $\varphi_0 = 0.75$ radians.

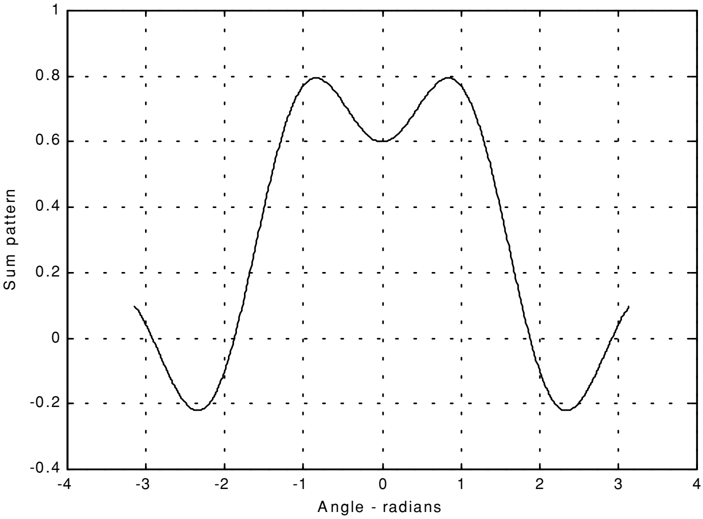


Figure 11.12b. Sum pattern corresponding to Fig. 11.12a.

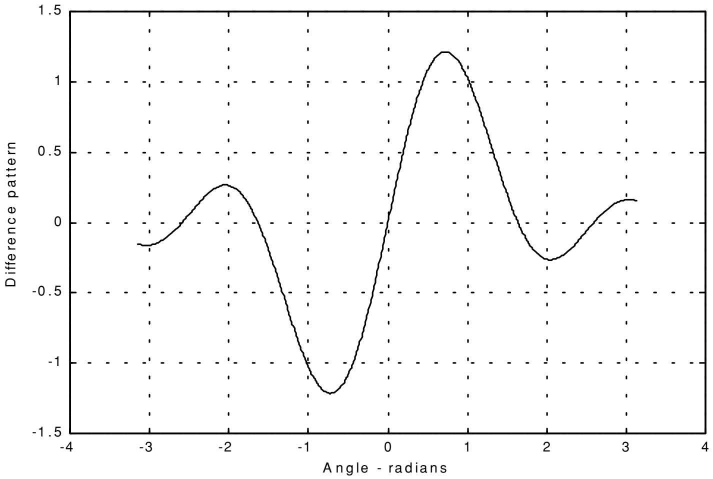


Figure 11.12c. Difference pattern corresponding to Fig. 11.12a.

The difference channels give us an indication of whether the target is on or off the tracking axis. However, this signal amplitude depends not only on the target angular position, but also on the target's range and RCS. For this reason the ratio Δ/Σ (delta over sum) can be used to accurately estimate the error angle that only depends on the target's angular position.

Let us now address how the error signals are computed. First, consider the azimuth error signal. Define the signals S_1 and S_2 as

$$S_1 = A + D \tag{11.9}$$

$$S_2 = B + C \tag{11.10}$$

The sum signal is $\Sigma = S_1 + S_2$, and the azimuth difference signal is $\Delta_{az} = S_1 - S_2$. If $S_1 \geq S_2$, then both channels have the same phase $0°$ (since the sum channel is used for phase reference). Alternatively, if $S_1 < S_2$, then the two channels are $180°$ out of phase. Similar analysis can be done for the elevation channel, where in this case $S_1 = A + B$ and $S_2 = D + C$. Thus, the error signal output is

$$\varepsilon_\varphi = \frac{|\Delta|}{|\Sigma|} \cos\xi \tag{11.11}$$

where ξ is the phase angle between the sum and difference channels and it is equal to $0°$ or $180°$. More precisely, if $\xi = 0$, then the target is on the tracking axis; otherwise it is off the tracking axis. Fig. 11.13 (a,b) shows a plot for the ratio Δ / Σ for the monopulse radar whose sum and difference patterns are in Figs. 11.11 and 11.12.

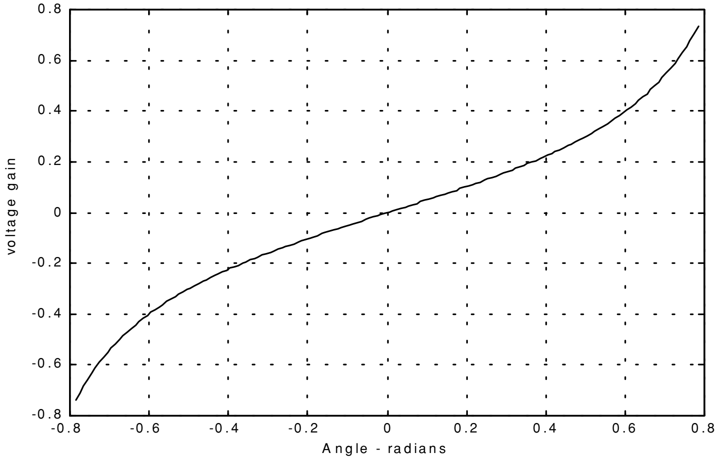


Figure 11.13a. Difference-to-sum ratio corresponding to Fig. 11.11a.

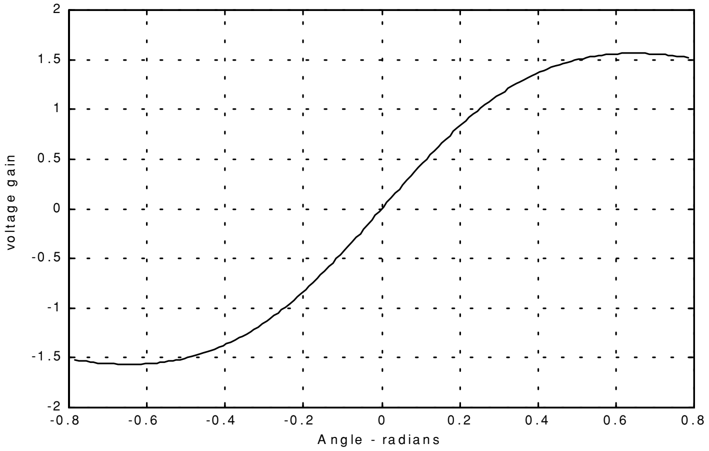


Figure 11.13b. Difference-to-sum ratio corresponding to Fig. 11.12a.

11.3. Phase Comparison Monopulse

Phase comparison monopulse is similar to amplitude comparison monopulse in the sense that the target angular coordinates are extracted from one sum and two difference channels. The main difference is that the four signals produced in amplitude comparison monopulse will have similar phases but different amplitudes; however, in phase comparison monopulse the signals have the same amplitude and different phases. Phase comparison monopulse tracking radars use a minimum of a two-element array antenna for each coordinate (azimuth and elevation), as illustrated in Fig. 11.14. A phase error signal (for each coordinate) is computed from the phase difference between the signals generated in the antenna elements.

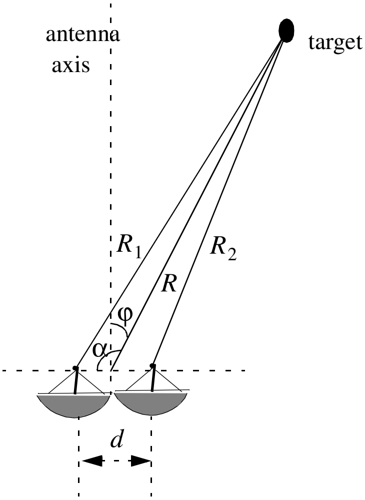


Figure 11.14. Single coordinate phase comparison monopulse antenna.

Consider Fig. 11.14; since the angle α is equal to $\varphi + \pi/2$, it follows that

$$R_1^2 = R^2 + \left(\frac{d}{2}\right)^2 - 2\frac{d}{2}R\cos\left(\varphi + \frac{\pi}{2}\right) \qquad \textbf{(11.12)}$$

$$= R^2 + \frac{d^2}{4} - dR\sin\varphi$$

and since $d \ll R$ we can use the binomial series expansion to get

$$R_1 \approx R\left(1 + \frac{d}{2R}\sin\varphi\right) \tag{11.13}$$

Similarly,

$$R_2 \approx R\left(1 - \frac{d}{2R}\sin\varphi\right) \tag{11.14}$$

The phase difference between the two elements is then given by

$$\phi = \frac{2\pi}{\lambda}(R_1 - R_2) = \frac{2\pi}{\lambda}d\sin\varphi \tag{11.15}$$

where λ is the wavelength. The phase difference ϕ is used to determine the angular target location. Note that if $\phi = 0$, then the target would be on the antenna's main axis. The problem with this phase comparison monopulse technique is that it is quite difficult to maintain a stable measurement of the off boresight angle φ, which causes serious performance degradation. This problem can be overcome by implementing a phase comparison monopulse system as illustrated in Fig. 11.15.

The (single coordinate) sum and difference signals are, respectively, given by

$$\Sigma(\varphi) = S_1 + S_2 \tag{11.16}$$

$$\Delta(\varphi) = S_1 - S_2 \tag{11.17}$$

where the S_1 and S_2 are the signals in the two elements. Now, since S_1 and S_2 have similar amplitude and are different in phase by ϕ, we can write

$$S_1 = S_2 e^{-j\phi} \tag{11.18}$$

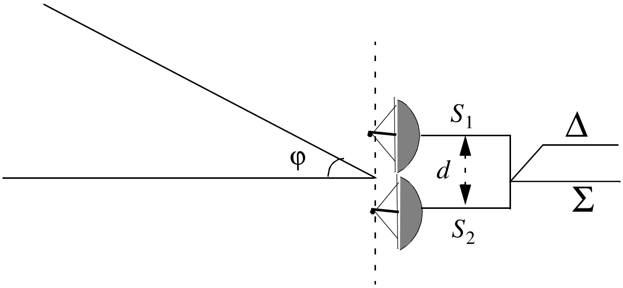


Figure 11.15. Single coordinate phase monopulse antenna, with sum and difference channels.

It follows that

$$\Delta(\varphi) = S_2(1 - e^{-j\phi}) \tag{11.19}$$

$$\Sigma(\varphi) = S_2(1 + e^{-j\phi}) \tag{11.20}$$

The phase error signal is computed from the ratio Δ/Σ. More precisely,

$$\frac{\Delta}{\Sigma} = \frac{1 - e^{-j\phi}}{1 + e^{-j\phi}} = j\tan\left(\frac{\phi}{2}\right) \tag{11.21}$$

which is purely imaginary. The modulus of the error signal is then given by

$$\frac{|\Delta|}{|\Sigma|} = \tan\left(\frac{\phi}{2}\right) \tag{11.22}$$

This kind of phase comparison monopulse tracker is often called the half-angle tracker.

11.4. Range Tracking

Target range is measured by estimating the round-trip delay of the transmitted pulses. The process of continuously estimating the range of a moving target is known as range tracking. Since the range to a moving target is changing with time, the range tracker must be constantly adjusted to keep the target locked in range. This can be accomplished using a split gate system, where two range gates (early and late) are utilized. The concept of split gate tracking is illustrated in Fig. 11.16, where a sketch of a typical pulsed radar echo is shown in the figure. The early gate opens at the anticipated starting time of the radar echo and lasts for half its duration. The late gate opens at the center and closes at the end of the echo signal. For this purpose, good estimates of the echo duration and the pulse centertime must be reported to the range tracker so that the early and late gates can be placed properly at the start and center times of the expected echo. This reporting process is widely known as the “designation process.”

The early gate produces positive voltage output while the late gate produces negative voltage output. The outputs of the early and late gates are subtracted, and the difference signal is fed into an integrator to generate an error signal. If both gates are placed properly in time, the integrator output will be equal to zero. Alternatively, when the gates are not timed properly, the integrator output is not zero, which gives an indication that the gates must be moved in time, left or right depending on the sign of the integrator output.

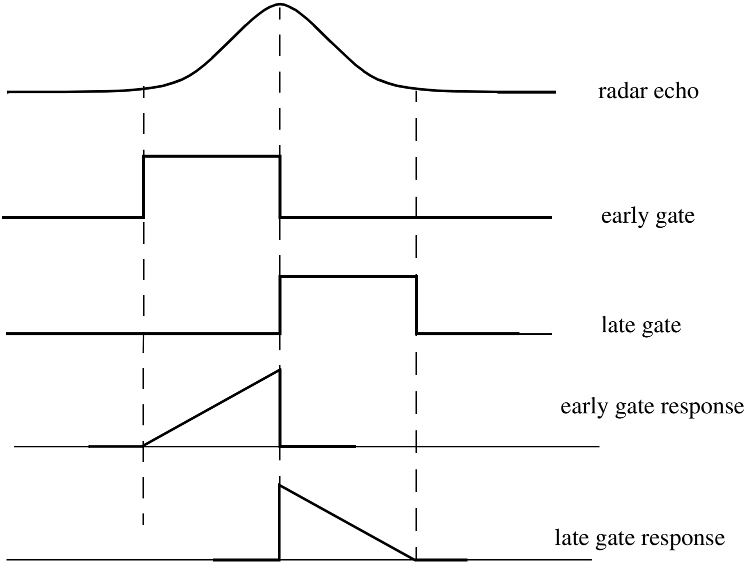


Figure 11.16. Illustration of split-range gate.

Part II: Multiple Target Tracking

Track-while-scan radar systems sample each target once per scan interval, and use sophisticated smoothing and prediction filters to estimate the target parameters between scans. To this end, the Kalman filter and the Alpha-Beta-Gamma ($\alpha\beta\gamma$) filter are commonly used. Once a particular target is detected, the radar may transmit up to a few pulses to verify the target parameters, before it establishes a track file for that target. Target position, velocity, and acceleration comprise the major components of the data maintained by a track file.

The principles of recursive tracking and prediction filters are presented in this part. First, an overview of state representation for Linear Time Invariant (LTI) systems is discussed. Then, second and third order one-dimensional fixed gain polynomial filter trackers are developed. These filters are, respectively, known as the $\alpha\beta$ and $\alpha\beta\gamma$ filters (also known as the g-h and g-h-k filters). Finally, the equations for an n-dimensional multi-state Kalman filter is introduced and analyzed. As a matter of notation, small case letters, with an underneath bar, are used.

11.5. Track-While-Scan (TWS)

Modern radar systems are designed to perform multi-function operations, such as detection, tracking, and discrimination. With the aid of sophisticated computer systems, multi-function radars are capable of simultaneously tracking many targets. In this case, each target is sampled once (mainly range and angular position) during a dwell interval (scan). Then, by using smoothing and prediction techniques future samples can be estimated. Radar systems that can perform multi-tasking and multi-target tracking are known as Track-While-Scan (TWS) radars.

Once a TWS radar detects a new target it initiates a separate track file for that detection; this ensures that sequential detections from that target are processed together to estimate the target's future parameters. Position, velocity, and acceleration comprise the main components of the track file. Typically, at least one other confirmation detection (verify detection) is required before the track file is established.

Unlike single target tracking systems, TWS radars must decide whether each detection (observation) belongs to a new target or belongs to a target that has been detected in earlier scans. And in order to accomplish this task, TWS radar systems utilize correlation and association algorithms. In the correlation process each new detection is correlated with all previous detections in order to avoid establishing redundant tracks. If a certain detection correlates with more than one track, then a pre-determined set of association rules are exercised so that the detection is assigned to the proper track. A simplified TWS data processing block diagram is shown in Fig. 11.17.

Choosing a suitable tracking coordinate system is the first problem a TWS radar has to confront. It is desirable that a fixed reference of an inertial coordinate system be adopted. The radar measurements consist of target range, velocity, azimuth angle, and elevation angle. The TWS system places a gate around the target position and attempts to track the signal within this gate. The gate dimensions are normally azimuth, elevation, and range. Because of the uncertainty associated with the exact target position during the initial detections, a gate has to be large enough so that targets do not move appreciably from scan to scan; more precisely, targets must stay within the gate boundary during successive scans. After the target has been observed for several scans the size of the gate is reduced considerably.

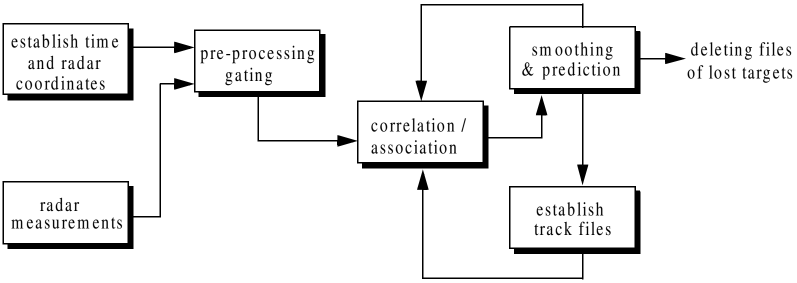

establish time and radar coordinates
pre-processing gating
radar measurements
correlation / association
smoothing & prediction
deleting files of lost targets
establish track files

Figure. 11.17. Simplified block diagram of TWS data processing.

Gating is used to decide whether an observation is assigned to an existing track file, or to a new track file (new detection). Gating algorithms are normally based on computing a statistical error distance between a measured and an estimated radar observation. For each track file, an upper bound for this error distance is normally set. If the computed difference for a certain radar observation is less than the maximum error distance of a given track file, then the observation is assigned to that track.

All observations that have an error distance less than the maximum distance of a given track are said to correlate with that track. For each observation that does not correlate with any existing tracks, a new track file is established accordingly. Since new detections (measurements) are compared to all existing track files, a track file may then correlate with no observations or with one or more observations. The correlation between observations and all existing track files is identified using a correlation matrix. Rows of the correlation matrix represent radar observations, while columns represent track files. In cases where several observations correlate with more than one track file, a set of predetermined association rules can be utilized so that a single observation is assigned to a single track file.

11.6. State Variable Representation of an LTI System

Linear time invariant system (continuous or discrete) can be describe mathematically using three variables. They are the input, output, and the state variables. In this representation, any LTI system has observable or measurable objects (abstracts). For example, in the case of a radar system, range may be an object measured or observed by the radar tracking filter. States can be derived in many different ways. For the scope of this book, states of an object or an abstract are the components of the vector that contains the object and its time derivatives. For example, a third-order one-dimensional (in this case range) state vector representing range can be given by

$$\underline{x} = \begin{bmatrix} R \\ \dot{R} \\ \ddot{R} \end{bmatrix} \tag{11.23}$$

where R, $\dot{R}$, and $\ddot{R}$ are, respectively, the range measurement, range rate (velocity), and acceleration. The state vector defined in Eq. (11.23) can be representative of continuous or discrete states. In this book, the emphasis is on discrete time representation, since most radar signal processing is executed using digital computers. For this purpose, an n-dimensional state vector has the following form:

$$\underline{x} = \begin{bmatrix} x_1 \; \dot{x}_1 \; \ldots \; x_2 \; \dot{x}_2 \; \ldots \; x_n \; \dot{x}_n \; \ldots \end{bmatrix}^t \tag{11.24}$$

where the superscript indicates the transpose operation.

The LTI system of interest can be represented using the following state equations:

$$\underline{\dot{x}}(t) = \underline{A} \ \underline{x}(t) + \underline{B}\underline{w}(t) \tag{11.25}$$

$$\underline{y}(t) = \underline{C} \ \underline{x}(t) + \underline{D}\underline{w}(t) \tag{11.26}$$

where: $\underline{\dot{x}}$ is the value of the $n \times 1$ state vector; $\underline{y}$ is the value of the $p \times 1$ output vector; $\underline{w}$ is the value of the $m \times 1$ input vector; $\underline{A}$ is an $n \times n$ matrix; $\underline{B}$ is an $n \times m$ matrix; $\underline{C}$ is $p \times n$ matrix; and $\underline{D}$ is an $p \times m$ matrix. The homogeneous solution (i.e., $\underline{w} = \underline{0}$) to this linear system, assuming known initial condition $\underline{x}(0)$ at time t_0, has the form

$$\underline{x}(t) = \underline{\Phi}(t - t_0)\underline{x}(t - t_0) \tag{11.27}$$

The matrix $\underline{\Phi}$ is known as the state transition matrix, or fundamental matrix, and is equal to

$$\underline{\Phi}(t - t_0) = e^{\underline{A}(t - t_0)} \tag{11.28}$$

Eq. (11.28) can be expressed in series format as

$$\underline{\Phi}(t - t_0)\Big|_{t_0 = 0} = e^{\underline{A}(t)} = \underline{I} + \underline{A}t + \underline{A}^2\frac{t^2}{2!} + \ldots = \sum_{k=0} \underline{A}^k \frac{t^k}{k!} \tag{11.29}$$

Example 11.1: Compute the state transition matrix for an LTI system when

$$\underline{A} = \begin{bmatrix} 0 & 1 \\ -0.5 & -1 \end{bmatrix}$$

Solution:

The state transition matrix can be computed using Eq. (11.29). For this purpose, compute $\underline{A}^2$ and $\underline{A}^3$.... It follows

$$\underline{A}^2 = \begin{bmatrix} -\frac{1}{2} & -1 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \qquad \underline{A}^3 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{4} & 0 \end{bmatrix} \qquad \ldots$$

Therefore,

$$\underline{\Phi} = \begin{bmatrix} 1 + 0t - \dfrac{\frac{1}{2}t^2}{2!} + \dfrac{\frac{1}{2}t^3}{3!} + \ldots & 0 + t - \dfrac{t^2}{2!} + \dfrac{\frac{1}{2}t^3}{3!} + \ldots \\ 0 - \dfrac{1}{2}t + \dfrac{\frac{1}{2}t^2}{2!} - \dfrac{\frac{1}{4}t^3}{3!} + \ldots & 1 - t + \dfrac{\frac{1}{2}t^2}{2!} + \dfrac{0t^3}{3!} + \ldots \end{bmatrix}$$

The state transition matrix has the following properties (the proof is left as an exercise):

1. *Derivative property*

$$\frac{\partial}{\partial t}\underline{\Phi}(t-t_0) = \underline{A}\,\underline{\Phi}(t-t_0) \tag{11.30}$$

2. *Identity property*

$$\underline{\Phi}(t_0-t_0) = \underline{\Phi}(0) = \underline{I} \tag{11.31}$$

3. *Initial value property*

$$\left.\frac{\partial}{\partial t}\underline{\Phi}(t-t_0)\right|_{t=t_0} = \underline{A} \tag{11.32}$$

4. *Transition property*

$$\underline{\Phi}(t_2-t_0) = \underline{\Phi}(t_2-t_1)\underline{\Phi}(t_1-t_0) \qquad ;\ t_0 \le t_1 \le t_2 \tag{11.33}$$

5. *Inverse property*

$$\underline{\Phi}(t_0-t_1) = \underline{\Phi}^{-1}(t_1-t_0) \tag{11.34}$$

6. *Separation property*

$$\underline{\Phi}(t_1-t_0) = \underline{\Phi}(t_1)\underline{\Phi}^{-1}(t_0) \tag{11.35}$$

The general solution to the system defined in Eq. (11.25) can be written as

$$\underline{x}(t) = \underline{\Phi}(t-t_0)\underline{x}(t_0) + \int_{t_0}^{t}\underline{\Phi}(t-\tau)\underline{B}\,\underline{w}(\tau)d\tau \tag{11.36}$$

The first term of the right-hand side of Eq. (11.36) represents the contribution from the system response to the initial condition. The second term is the contribution due to the driving force $\underline{w}$. By combining Eqs. (11.26) and (11.36) an expression for the output is computed as

$$\underline{y}(t) = \underline{C}e^{\underline{A}(t-t_0)}\underline{x}(t_0) + \int_{t_0}^{t}[\underline{C}e^{\underline{A}(t-\tau)}\underline{B} - \underline{D}\delta(t-\tau)]\underline{w}(\tau)d\tau \tag{11.37}$$

Note that the system impulse response is equal to $\underline{C}e^{\underline{A}t}\underline{B} - \underline{D}\delta(t)$.

The difference equations describing a discrete time system, equivalent to Eqs. (11.25) and (11.26), are

$$\underline{x}(n+1) = \underline{A}\ \underline{x}(n) + \underline{B}\underline{w}(n) \tag{11.38}$$

$$\underline{y}(n) = \underline{C}\ \underline{x}(n) + \underline{D}\underline{w}(n) \tag{11.39}$$

where n defines the discrete time nT and T is the sampling interval. All other vectors and matrices were defined earlier. The homogeneous solution to the system defined in Eq. (11.38), with initial condition $\underline{x}(n_0)$, is

$$\underline{x}(n) = \underline{A}^{n-n_0}\underline{x}(n_0) \tag{11.40}$$

In this case the state transition matrix is an $n \times n$ matrix given by

$$\underline{\Phi}(n, n_0) = \underline{\Phi}(n - n_0) = \underline{A}^{n-n_0} \tag{11.41}$$

The following is the list of properties associated with the discrete transition matrix

$$\underline{\Phi}(n+1-n_0) = \underline{A}\underline{\Phi}(n-n_0) \tag{11.42}$$

$$\underline{\Phi}(n_0-n_0) = \underline{\Phi}(0) = \underline{I} \tag{11.43}$$

$$\underline{\Phi}(n_0+1-n_0) = \underline{\Phi}(1) = \underline{A} \tag{11.44}$$

$$\underline{\Phi}(n_2-n_0) = \underline{\Phi}(n_2-n_1)\underline{\Phi}(n_1-n_0) \tag{11.45}$$

$$\underline{\Phi}(n_0-n_1) = \underline{\Phi}^{-1}(n_1-n_0) \tag{11.46}$$

$$\underline{\Phi}(n_1-n_0) = \underline{\Phi}(n_1)\underline{\Phi}^{-1}(n_0) \tag{11.47}$$

The solution to the general case (i.e., non-homogeneous system) is given by

$$\underline{x}(n) = \underline{\Phi}(n-n_0)\underline{x}(n_0) + \sum_{m=n_0}^{n-1} \underline{\Phi}(n-m-1)\underline{B}\underline{w}(m) \tag{11.48}$$

It follows that the output is given by

$$\underline{y}(n) = \underline{C}\underline{\Phi}(n-n_0)\underline{x}(n_0) + \sum_{m=n_0}^{n-1} \underline{C}\ \underline{\Phi}(n-m-1)\underline{B}\underline{w}(m) + \underline{D}\underline{w}(n) \tag{11.49}$$

where the system impulse response is given by

$$\underline{h}(n) = \sum_{m=n_0}^{n-1} \underline{C}\ \underline{\Phi}(n-m-1)\underline{B}\underline{\delta}(m) + \underline{D}\underline{\delta}(n) \tag{11.50}$$

Taking the Z-transform for Eqs. (11.38) and (11.39) yields

$$z\underline{x}(z) = \underline{A}\underline{x}(z) + \underline{B}\underline{w}(z) + z\underline{x}(0) \tag{11.51}$$

$$\underline{y}(z) = \underline{C}\underline{x}(z) + \underline{D}\underline{w}(z) \tag{11.52}$$

Manipulating Eqs. (11.51) and (11.52) yields

$$\underline{x}(z) = [z\underline{I} - \underline{A}]^{-1}\underline{B}\underline{w}(z) + [z\underline{I} - \underline{A}]^{-1} z\underline{x}(0) \tag{11.53}$$

$$\underline{y}(z) = \{\underline{C}[z\underline{I} - \underline{A}]^{-1}\underline{B} + \underline{D}\}\underline{w}(z) + \underline{C}[z\underline{I} - \underline{A}]^{-1} z\underline{x}(0) \tag{11.54}$$

It follows that the state transition matrix is

$$\underline{\Phi}(z) = z[z\underline{I} - \underline{A}]^{-1} = [\underline{I} - z^{-1}\underline{A}]^{-1} \tag{11.55}$$

and the system impulse response in the z-domain is

$$\underline{h}(z) = \underline{C}\underline{\Phi}(z)z^{-1}\underline{B} + \underline{D} \tag{11.56}$$

11.7. The LTI System of Interest

For the purpose of establishing the framework necessary for the Kalman filter development, consider the LTI system shown in Fig. 11.18. This system (which is a special case of the system described in the previous section) can be described by the following first order differential vector equations

$$\dot{\underline{x}}(t) = \underline{A}\ \underline{x}(t) + \underline{u}(t) \tag{11.57}$$

$$\underline{y}(t) = \underline{G}\ \underline{x}(t) + \underline{v}(t) \tag{11.58}$$

where $\underline{y}$ is the observable part of the system (i.e., output), $\underline{u}$ is a driving force, and $\underline{v}$ is the measurement noise. The matrices $\underline{A}$ and $\underline{G}$ vary depending on the system. The noise observation $\underline{v}$ is assumed to be uncorrelated. If the initial condition vector is $\underline{x}(t_0)$, then from Eq. (11.36) we get

$$\underline{x}(t) = \underline{\Phi}(t-t_0)\underline{x}(t_0) + \int_{t_0}^{t} \underline{\Phi}(t-\tau)\underline{u}(\tau)d\tau \tag{11.59}$$

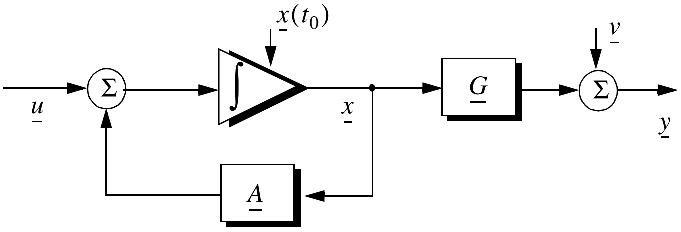


Figure 11.18. An LTI system.

The object (abstract) is observed only at discrete times determined by the system. These observation times are declared by discrete time nT where T is the sampling interval. Using the same notation adopted in the previous section, the discrete time representations of Eqs. (11.57) and (11.58) are

$$\underline{x}(n) = \underline{A}\ \underline{x}(n-1) + \underline{u}(n) \tag{11.60}$$

$$\underline{y}(n) = \underline{G}\ \underline{x}(n) + \underline{v}(n) \tag{11.61}$$

The homogeneous solution to this system is given in Eq. (11.27) for continuous time, and in Eq. (11.40) for discrete time.

The state transition matrix corresponding to this system can be obtained using Taylor series expansion of the vector $\underline{x}$. More precisely,

$$\begin{aligned} x &= x + T\dot{x} + \frac{T^2}{2!}\ddot{x} + \dots \\ \dot{x} &= \dot{x} + T\ddot{x} + \dots \\ \ddot{x} &= \ddot{x} + \dots \end{aligned} \tag{11.62}$$

It follows that the elements of the state transition matrix are defined by

$$\underline{\Phi}[ij] = \left\{ \begin{matrix} T^{j-i} \div (j-i)! & 1 \le i, j \le n \\ 0 & j < i \end{matrix} \right\} \tag{11.63}$$

Using matrix notation, the state transition matrix is then given by

$$\underline{\Phi} = \begin{bmatrix} 1 & T & \frac{T^2}{2!} & \ldots \\ 0 & 1 & T & \ldots \\ 0 & 0 & 1 & \ldots \\ \ldots & \ldots & \ldots & \ldots \end{bmatrix} \tag{11.64}$$

The matrix given in Eq. (11.64) is often called the Newtonian matrix.

11.8. Fixed-Gain Tracking Filters

This class of filters (or estimators) is also known as “Fixed-Coefficient” filters. The most common examples of this class of filters are the $\alpha\beta$ and $\alpha\beta\gamma$ filters and their variations. The $\alpha\beta$ and $\alpha\beta\gamma$ trackers are one-dimensional second and third order filters, respectively. They are equivalent to special cases of the one-dimensional Kalman filter. The general structure of this class of estimators is similar to that of the Kalman filter.

The standard $\alpha\beta\gamma$ filter provides smoothed and predicted data for target position, velocity (Doppler), and acceleration. It is a polynomial predictor/corrector linear recursive filter. This filter can reconstruct position, velocity, and constant acceleration based on position measurements. The $\alpha\beta\gamma$ filter can also provide a smoothed (corrected) estimate of the present position which can be used in guidance and fire control operations.

Notation:

For the purpose of the discussion presented in the remainder of this chapter, the following notation is adopted: $x(n|m)$ represents the estimate during the nth sampling interval, using all data up to and including the mth sampling interval; y_n is the nth measured value; and e_n is the nth residual (error).

The fixed-gain filter equation is given by

$$\underline{x}(n|n) = \underline{\Phi}\underline{x}(n-1|n-1) + \underline{K}[y_n - \underline{G}\underline{\Phi}\underline{x}(n-1|n-1)] \tag{11.65}$$

Since the transition matrix assists in predicting the next state,

$$\underline{x}(n+1|n) = \underline{\Phi}\underline{x}(n|n) \tag{11.66}$$

Substituting Eq. (11.66) into Eq. (11.65) yields

$$\underline{x}(n|n) = \underline{x}(n|n-1) + \underline{K}[y_n - \underline{G}\underline{x}(n|n-1)] \tag{11.67}$$

The term enclosed within the brackets on the right hand side of Eq. (11.67) is often called the residual (error) which is the difference between the measured input and predicted output. Eq. (11.67) means that the estimate of $\underline{x}(n)$ is the sum of the prediction and the weighted residual. The term $\underline{G}\underline{x}(n|n-1)$ represents the prediction state. In the case of the $\alpha\beta\gamma$ estimator, $\underline{G}$ is row vector given by

$$\underline{G} = \begin{bmatrix} 1 & 0 & 0 & \ldots \end{bmatrix} \tag{11.68}$$

and the gain matrix $\underline{K}$ is given by

$$\underline{K} = \begin{bmatrix} \alpha \\ \beta/T \\ \gamma/T^2 \end{bmatrix} \tag{11.69}$$

One of the main objectives of a tracking filter is to decrease the effect of the noise observation on the measurement. For this purpose the noise covariance matrix is calculated. More precisely, the noise covariance matrix is

$$\underline{C}(n|n) = E\{(\underline{x}(n|n)\ )\underline{x}^t(n|n)\} \qquad ; \ y_n = v_n \tag{11.70}$$

where E indicates the expected value operator. Noise is assumed to be a zero mean random process with variance equal to σ_v^2. Additionally, noise measurements are also assumed to be uncorrelated,

$$E\{v_n v_m\} = \begin{cases} \delta\sigma_v^2 & n = m \\ 0 & n \neq m \end{cases} \tag{11.71}$$

Eq. (11.65) can be written as

$$\underline{x}(n|n) = \underline{A}\underline{x}(n-1|n-1) + \underline{K}y_n \tag{11.72}$$

where

$$\underline{A} = (\underline{I} - \underline{K}\underline{G})\underline{\Phi} \tag{11.73}$$

Substituting Eqs. (11.72) and (11.73) into Eq. (11.70) yields

$$\underline{C}(n|n) = E\{(\underline{A}\underline{x}(n-1|n-1) + \underline{K}y_n)(\underline{A}\underline{x}(n-1|n-1) + \underline{K}y_n)^t\} \tag{11.74}$$

Expanding the right hand side of Eq. (11.74) and using Eq. (11.71) give

$$\underline{C}(n|n) = \underline{A}\,\underline{C}(n-1|n-1)\underline{A}^t + \underline{K}\sigma_v^2\underline{K}^t \tag{11.75}$$

Under the steady state condition, Eq. (11.75) collapses to

$$\underline{C}(n|n) = \underline{A}\,\underline{C}\,\underline{A}^t + \underline{K}\sigma_v^2\underline{K}^t \tag{11.76}$$

where $\underline{C}$ is the steady state noise covariance matrix. In the steady state,

$$\underline{C}(n|n) = \underline{C}(n-1|n-1) = \underline{C} \qquad for\ \ any\ \ n \tag{11.77}$$

Several criteria can be used to establish the performance of fixed-gain tracking filter. The most commonly used technique is to compute the Variance Reduction Ratio (VRR). The VRR is defined only when the input to the tracker is noise measurements. It follows that in the steady state case, the VRR is the steady state ratio of the output variance (auto-covariance) to the input measurement variance.

In order to determine the stability of the tracker under consideration, consider the Z-transform for Eq. (11.72),

$$\underline{x}(z) = \underline{A}z^{-1}\underline{x}(z) + \underline{K}y_n(z) \tag{11.78}$$

Rearranging Eq. (11.78) yields the following system transfer functions:

$$\underline{h}(z) = \frac{\underline{x}(z)}{y_n(z)} = (\underline{I} - \underline{A}z^{-1})^{-1}\underline{K} \tag{11.79}$$

where $(\underline{I} - \underline{A}z^{-1})$ is called the characteristic matrix. Note that the system transfer functions can exist only when the characteristic matrix is a non-singular matrix. Additionally, the system is stable if and only if the roots of the characteristic equation are within the unit circle in the z-plane,

$$\left|(\underline{I} - \underline{A}z^{-1})\right| = 0 \tag{11.80}$$

The filter's steady state errors can be determined with the help of Fig. 11.19. The error transfer function is

$$\underline{e}(z) = \frac{\underline{y}(z)}{1 + \underline{h}(z)} \tag{11.81}$$

and by using Abel's theorem, the steady state error is

$$\underline{e}_\infty = \lim_{t \to \infty} \underline{e}(t) = \lim_{z \to 1}\left(\frac{z-1}{z}\right)\underline{e}(z) \tag{11.82}$$

Substituting Eq. (11.82) into (11.81) yields

$$\underline{e}_{\infty} = \lim_{z \to 1} \frac{z-1}{z} \frac{\underline{y}(z)}{1+\underline{h}(z)} \tag{11.83}$$

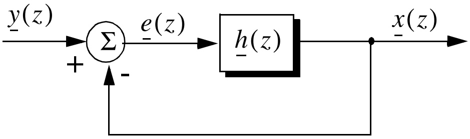


Figure 11.19. Steady state errors computation.

11.8.1. The αβ Filter

The αβ tracker produces, on the *nth* observation, smoothed estimates for position and velocity, and a predicted position for the $(n+1)th$ observation. Fig. 11.20 shows an implementation of this filter. Note that the subscripts "*p*" and "*s*" are used to indicate, respectively, the predicated and smoothed values. The αβ tracker can follow an input ramp (constant velocity) with no steady state errors. However, a steady state error will accumulate when constant acceleration is present in the input. Smoothing is done to reduce errors in the predicted position through adding a weighted difference between the measured and predicted values to the predicted position, as follows:

$$x_s(n) = x(n|n) = x_p(n) + \alpha(x_0(n) - x_p(n)) \tag{11.84}$$

$$\dot{x}_s(n) = x'(n|n) = \dot{x}_s(n-1) + \frac{\beta}{T}(x_0(n) - x_p(n)) \tag{11.85}$$

x_0 is the position input samples. The predicted position is given by

$$x_p(n) = x_s(n|n-1) = x_s(n-1) + T\dot{x}_s(n-1) \tag{11.86}$$

The initialization process is defined by

$$x_s(1) = x_p(2) = x_0(1)$$

$$\dot{x}_s(1) = 0$$

$$\dot{x}_s(2) = \frac{x_0(2) - x_0(1)}{T}$$

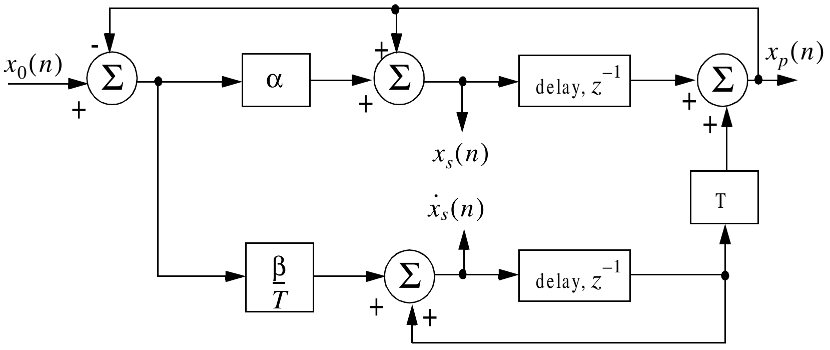


Figure 11.20. An implementation for an $\alpha\beta$ tracker.

A general form for the covariance matrix was developed in the previous section, and is given in Eq. (11.75). In general, a second order one-dimensional covariance matrix (in the context of the $\alpha\beta$ filter) can be written as

$$\underline{C}(n|n) = \begin{bmatrix} C_{xx} & C_{x\dot{x}} \\ C_{\dot{x}x} & C_{\dot{x}\dot{x}} \end{bmatrix} \tag{11.87}$$

where, in general, C_{xy} is

$$C_{xy} = E\{xy^t\} \tag{11.88}$$

By inspection, the $\alpha\beta$ filter has

$$\underline{A} = \begin{bmatrix} 1-\alpha & (1-\alpha)T \\ -\beta/T & (1-\beta) \end{bmatrix} \tag{11.89}$$

$$\underline{K} = \begin{bmatrix} \alpha \\ \beta/T \end{bmatrix} \tag{11.90}$$

$$\underline{G} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{11.91}$$

$$\underline{\Phi} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \tag{11.92}$$

Finally, by using Eqs. (11.89) through (11.92) in Eq. (11.72) yields the steady state noise covariance matrix,

$$\underline{C} = \frac{\sigma_v^2}{\alpha(4-2\alpha-\beta)}\begin{bmatrix} 2\alpha^2 - 3\alpha\beta + 2\beta & \frac{\beta(2\alpha-\beta)}{T} \\ \frac{\beta(2\alpha-\beta)}{T} & \frac{2\beta^2}{T^2} \end{bmatrix} \tag{11.93}$$

It follows that the position and velocity VRR ratios are, respectively, given by

$$(VRR)_x = C_{xx}/\sigma_v^2 = \frac{2\alpha^2 - 3\alpha\beta + 2\beta}{\alpha(4-2\alpha-\beta)} \tag{11.94}$$

$$(VRR)_{\dot{x}} = C_{\dot{x}\dot{x}}/\sigma_v^2 = \frac{1}{T^2}\,\frac{2\beta^2}{\alpha(4-2\alpha-\beta)} \tag{11.95}$$

The stability of the $\alpha\beta$ filter is determined from its system transfer functions. For this purpose, compute the roots for Eq. (11.80) with $\underline{A}$ from Eq. (11.89),

$$\left|\underline{I} - \underline{A}z^{-1}\right| = 1 - (2-\alpha-\beta)z^{-1} + (1-\alpha)z^{-2} = 0 \tag{11.96}$$

Solving Eq. (11.96) for z yields

$$z_{1,2} = 1 - \frac{\alpha+\beta}{2} \pm \frac{1}{2}\sqrt{(\alpha-\beta)^2 - 4\beta} \tag{11.97}$$

and in order to guarantee stability

$$|z_{1,2}| < 1 \tag{11.98}$$

Two cases are analyzed. First, $z_{1,2}$ are real. In this case (the details are left as an exercise),

$$\beta > 0 \qquad ; \ \alpha > -\beta \tag{11.99}$$

The second case is when the roots are complex; in this case we find

$$\alpha > 0 \tag{11.100}$$

The system transfer functions can be derived by using Eqs. (11.79), (11.89), and (11.90),

$$\begin{bmatrix} h_x(z) \\ h_{x'}(z) \end{bmatrix} = \frac{1}{z^2 - z(2-\alpha-\beta) + (1-\alpha)} \begin{bmatrix} \alpha z\left(z - \frac{(\alpha-\beta)}{\alpha}\right) \\ \frac{\beta z(z-1)}{T} \end{bmatrix} \tag{11.101}$$

Up to this point all relevant relations concerning the $\alpha\beta$ filter were made with no regard to how to choose the gain coefficients (α and β). Before considering the methodology of selecting these coefficients, consider the main objective behind using this filter. The purpose of the $\alpha\beta$ tracker can be described twofold:

1. *The tracker must reduce the measurement noise as much as possible.*
2. *The filter must be able to track maneuvering targets, with as little residual (tracking error) as possible.*

The reduction of measurement noise reduction is normally determined by the VRR ratios. However, the maneuverability performance of the filter depends heavily on the choice of the parameters α and β.

A special variation of the $\alpha\beta$ filter was developed by Benedict and Bordner[1], and is often referred to as the Benedict-Bordner filter. The main advantage of the Benedict-Bordner is reducing the transient errors associated with the $\alpha\beta$ tracker. This filter uses both the position and velocity VRR ratios as measure of performance. It computes the sum of the squared differences between the input (position) and the output when the input has a unit step velocity at time zero. Additionally, it computes the squared differences between the real velocity and the velocity output when the input is as described earlier. Both error differences are minimized when

$$\beta = \frac{\alpha^2}{2-\alpha} \tag{11.102}$$

In this case, the position and velocity VRR ratios are, respectively, given by

$$(VRR)_x = \frac{\alpha(6-5\alpha)}{\alpha^2 - 8\alpha + 8} \tag{11.103}$$

$$(VRR)_{\dot{x}} = \frac{2}{T^2}\ \frac{\alpha^3/(2-\alpha)}{\alpha^2 - 8\alpha + 8} \tag{11.104}$$

Another important sub-class of the $\alpha\beta$ tracker is the critically damped filter, often called the fading memory filter. In this case, the filter coefficients are chosen on the basis of a smoothing factor ξ, where $0 \le \xi \le 1$. The gain coefficients are given by

$$\alpha = 1 - \xi^2 \tag{11.105}$$

[1] Benedict, T. R. and Bordner, G. W., Synthesis of an Optimal Set of Radar Track-While-Scan Smoothing Equations. *IRE Transaction on Automatic Control, AC-7.* July 1962, pp. 27-32.

$$\beta = (1-\xi)^2 \tag{11.106}$$

Heavy smoothing means $\xi \rightarrow 1$ and little smoothing means $\xi \rightarrow 0$. The elements of the covariance matrix for a fading memory filter are

$$C_{xx} = \frac{1-\xi}{(1+\xi)^3}\ (1+4\xi+5\xi^2)\ \sigma_v^2 \tag{11.107}$$

$$C_{x\dot{x}} = C_{\dot{x}x} = \frac{1}{T}\ \frac{1-\xi}{(1+\xi)^3}\ (1+2\xi+3\xi^2)\ \sigma_v^2 \tag{11.108}$$

$$C_{\dot{x}\dot{x}} = \frac{2}{T^2}\ \frac{1-\xi}{(1+\xi)^3}\ (1-\xi)^2\ \sigma_v^2 \tag{11.109}$$

11.8.2. The $\alpha\beta\gamma$ Filter

The $\alpha\beta\gamma$ tracker produces, for the nth observation, smoothed estimates of position, velocity, and acceleration. It also produces predicted position and velocity for the $(n+1)th$ observation. An implementation of the $\alpha\beta\gamma$ tracker is shown in Fig. 11.21.

The $\alpha\beta\gamma$ tracker will follow an input whose acceleration is constant with no steady state errors. Again, in order to reduce the error at the output of the tracker, a weighted difference between the measured and predicted values is used in estimating the smoothed position, velocity, and acceleration as follows:

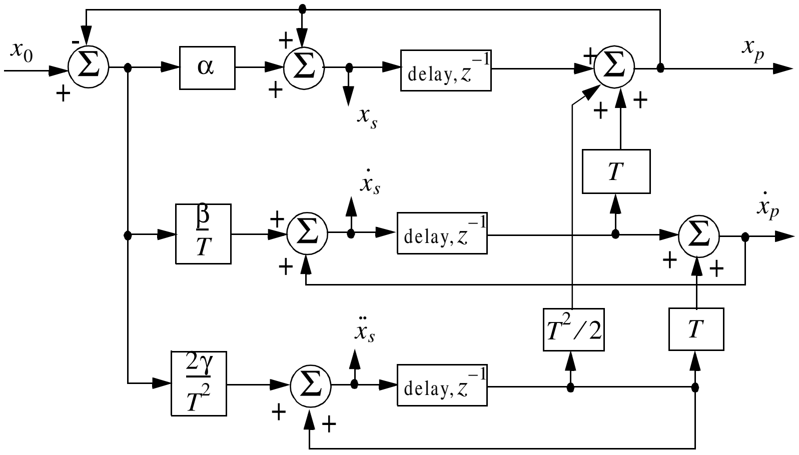


Figure 11.21. An implementation for an $\alpha\beta\gamma$ tracker.

$$x_s(n) = x_p(n) + \alpha(x_0(n) - x_p(n)) \tag{11.110}$$

$$\dot{x}_s(n) = \dot{x}_s(n-1) + T\ddot{x}_s(n-1) + \frac{\beta}{T}\ (x_0(n) - x_p(n)) \tag{11.111}$$

$$\ddot{x}_s(n) = \ddot{x}_s(n-1) + \frac{2\gamma}{T^2}\ (x_0(n) - x_p(n)) \tag{11.112}$$

$$x_p(n+1) = x_s(n) + T\ \dot{x}_s(n) + \frac{T^2}{2}\ \ddot{x}_s(n) \tag{11.113}$$

and the initialization process is

$$x_s(1) = x_p(2) = x_0(1)$$

$$\dot{x}_s(1) = \ddot{x}_s(1) = \ddot{x}_s(2) = 0$$

$$\dot{x}_s(2) = \frac{x_0(2) - x_0(1)}{T}$$

$$\ddot{x}_s(3) = \frac{x_0(3) + x_0(1) - 2x_0(2)}{T^2}$$

Using Eq. (11.63) the state transition matrix for the $\alpha\beta\gamma$ filter is

$$\underline{\Phi} = \begin{bmatrix} 1 & T & \dfrac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \tag{11.114}$$

The covariance matrix (which is symmetric) can be computed from Eq. (11.76). For this purpose, note that

$$\underline{K} = \begin{bmatrix} \alpha \\ \beta/T \\ \gamma/T^2 \end{bmatrix} \tag{11.115}$$

$$\underline{G} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{11.116}$$

and

$$\underline{A} = (\underline{I} - \underline{K}\underline{G})\underline{\Phi} = \begin{bmatrix} 1-\alpha & (1-\alpha)T & (1-\alpha)T^2/2 \\ -\beta/T & -\beta+1 & (1-\beta/2)T \\ -2\gamma/T^2 & -2\gamma/T & (1-\gamma) \end{bmatrix} \tag{11.117}$$

Substituting Eq. (11.117) into (11.76) and collecting terms the VRR ratios are computed as

$$(VRR)_x = \frac{2\beta(2\alpha^2 + 2\beta - 3\alpha\beta) - \alpha\gamma(4 - 2\alpha - \beta)}{(4 - 2\alpha - \beta)(2\alpha\beta + \alpha\gamma - 2\gamma)} \tag{11.118}$$

$$(VRR)_{\dot{x}} = \frac{4\beta^3 - 4\beta^2\gamma + 2\gamma^2(2 - \alpha)}{T^2(4 - 2\alpha - \beta)(2\alpha\beta + \alpha\gamma - 2\gamma)} \tag{11.119}$$

$$(VRR)_{\ddot{x}} = \frac{4\beta\gamma^2}{T^4(4 - 2\alpha - \beta)(2\alpha\beta + \alpha\gamma - 2\gamma)} \tag{11.120}$$

As in the case of any discrete time system, this filter will be stable if and only if all of its poles fall within the unit circle in the z-plane.

The $\alpha\beta\gamma$ characteristic equation is computed by setting

$$\left|\underline{I} - \underline{A}z^{-1}\right| = 0 \tag{11.121}$$

Substituting Eq. (11.117) into (11.121) and collecting terms yield the following characteristic function:

$$f(z) = z^3 + (-3\alpha + \beta + \gamma)z^2 + (3 - \beta - 2\alpha + \gamma)z - (1 - \alpha) \tag{11.122}$$

The $\alpha\beta\gamma$ becomes a Benedict-Bordner filter when

$$2\beta - \alpha\left(\alpha + \beta + \frac{\gamma}{2}\right) = 0 \tag{11.123}$$

Note that for $\gamma = 0$ Eq. (11.123) reduces to Eq. (11.102). For a critically damped filter the gain coefficients are

$$\alpha = 1 - \xi^3 \tag{11.124}$$

$$\beta = 1.5(1 - \xi^2)(1 - \xi) = 1.5(1 - \xi)^2(1 + \xi) \tag{11.125}$$

$$\gamma = (1 - \xi)^3 \tag{11.126}$$

Note that heavy smoothing takes place when $\xi \rightarrow 1$, while $\xi = 0$ means that no smoothing is present.

MATLAB Function "ghk_tracker.m"

The function *"ghk_tracker.m"*[1] implements the steady state $\alpha\beta\gamma$ filter. It is given in Listing 11.2 in Section 11.10. The syntax is as follows:

[residual, estimate] = ghk_tracker (X0, smoocof, inp, npts, T, nvar)

where

Symbol	Description	Status
X0	*initial state vector*	*input*
smoocof	*desired smoothing coefficient*	*input*
inp	*array of position measurements*	*input*
npts	*number of points in input position*	*input*
T	*sampling interval*	*input*
nvar	*desired noise variance*	*input*
residual	*array of position error (residual)*	*output*
estimate	*array of predicted position*	*output*

Note that *"ghk_tracker.m"* uses MATLAB's function *"normrnd.m"* to generate zero mean Gaussian noise, which is part of MATLAB's Statistics Toolbox. If this toolbox is not available to the user, then *"ghk_tracker.m"* function-call must be modified to

[residual, estimate] = ghk_tracker1 (X0, smoocof, inp, npts, T)

which is also part of Listing 11.2. In this case, noise measurements are either to be considered unavailable or are part of the position input array.

To illustrate how to use the functions *ghk_tracker.m* and *ghk_tracker.m1,* consider the inputs shown in Figs. 11.22 and 11.23. Fig. 11.22 assumes an input with lazy maneuvering, while Fig. 11.23 assumes an aggressive maneuvering case. For this purpose, the program called *"fig11_21.m"* was written. It is given in Listing 11.3 in Section 11.10.

Figs. 11.24 and 11.25 show the residual error and predicted position corresponding (generated using the program *"fig11_21.m")* to Fig. 11.22 for two cases: heavy smoothing and little smoothing with and without noise. The noise is white Gaussian with zero mean and variance of $\sigma_v^2 = 0.05$. Figs. 11. 26 and 11.27 show the residual error and predicted position corresponding (generated using the program *"fig11_20.m")* to Fig. 11.23 with and without noise.

[1] This function was written by Mr. Edward Shamsi of COLSA Corporation in Huntsville, AL.

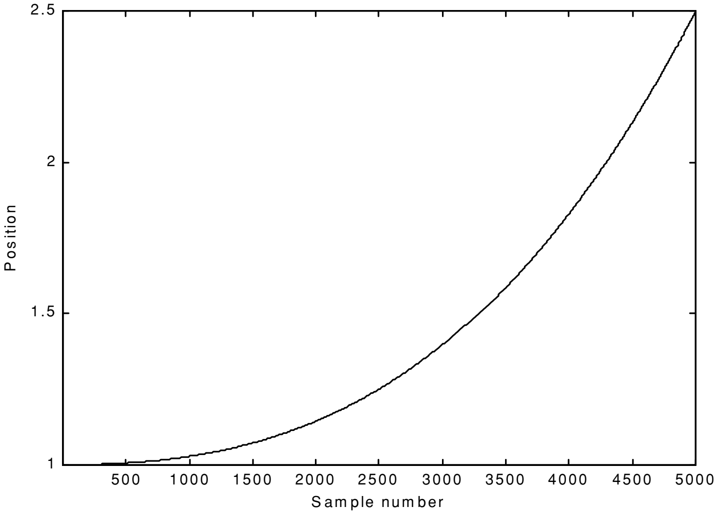


Figure 11.22. Position (truth-data); lazy maneuvering.

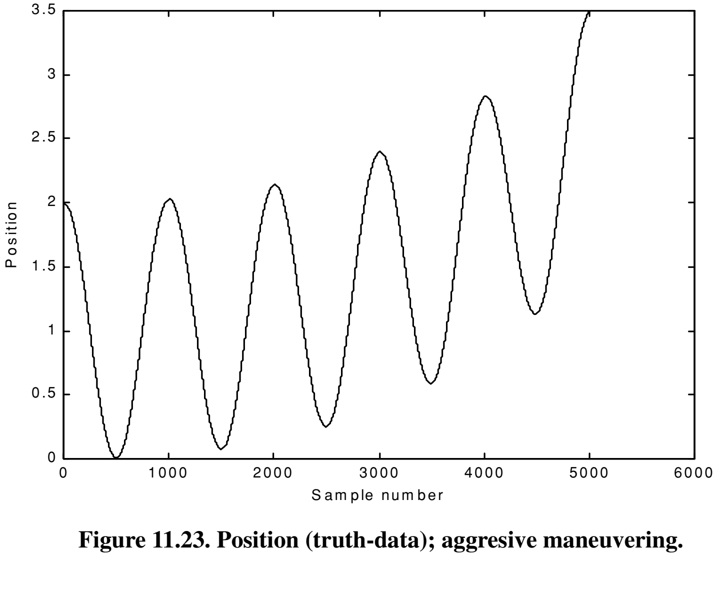


Figure 11.23. Position (truth-data); aggresive maneuvering.

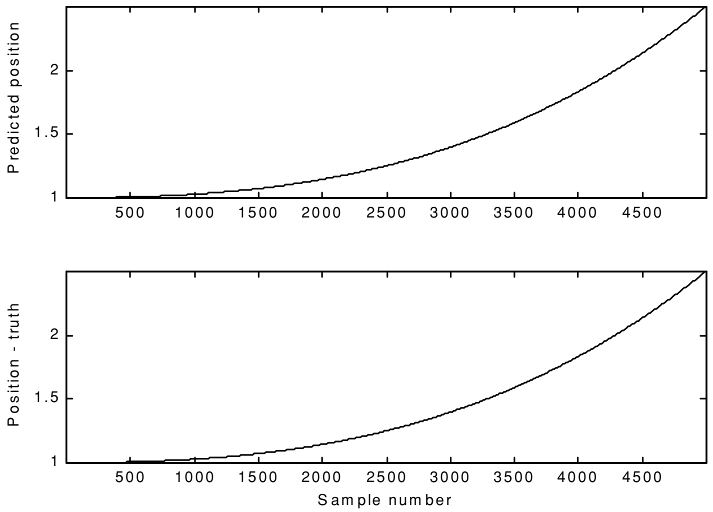


Figure 11.24a-1. Predicted and true position. $\xi = 0.1$ (i.e., large gain coefficients). No noise present.

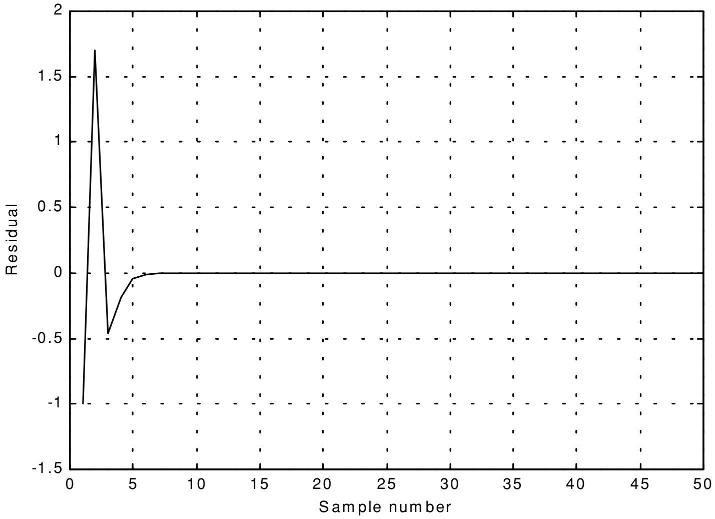


Figure 11.24a-2. Position residual (error). Large gain coefficients. No noise. The error settles to zero fairly quickly.

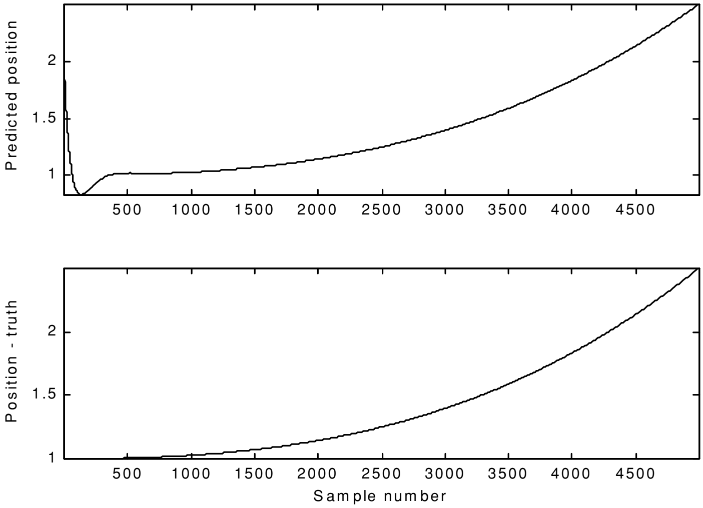


Figure 11.24b-1. Predicted and true position. $\xi = 0.9$ (i.e., small gain coefficients). No noise present.

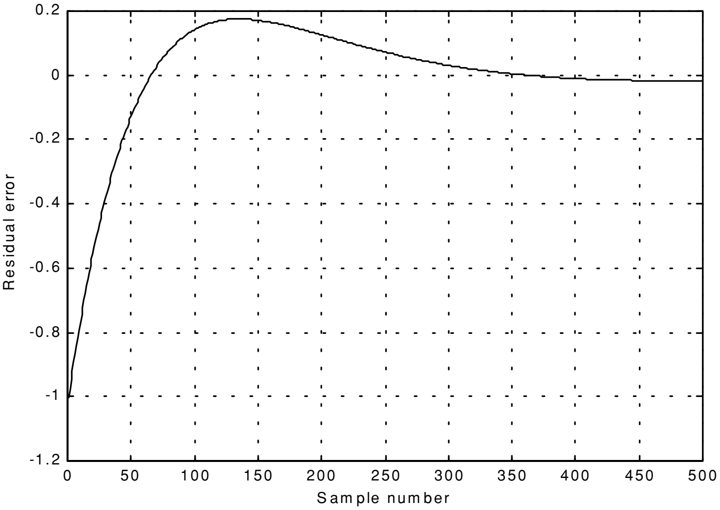


Figure 11.24b-2. Position residual (error). Small gain coefficients. No noise. It takes the filter longer time for the error to settle down.

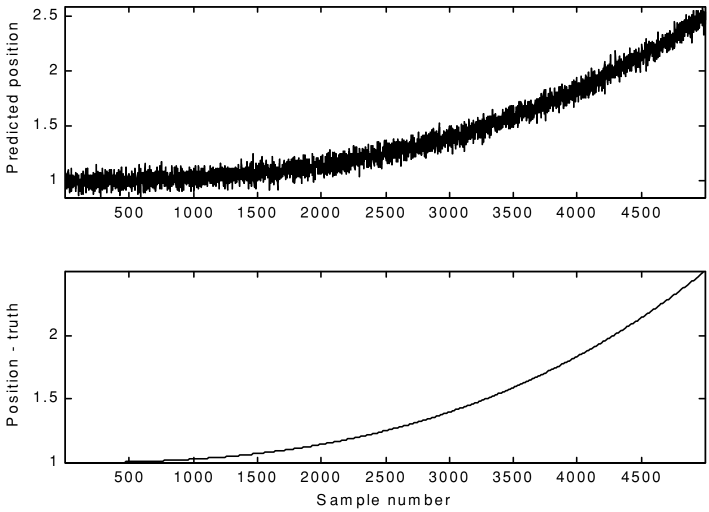


Figure 11.25a-1. Predicted and true position. $\xi = 0.1$ (i.e., large gain coefficients). Noise is present.

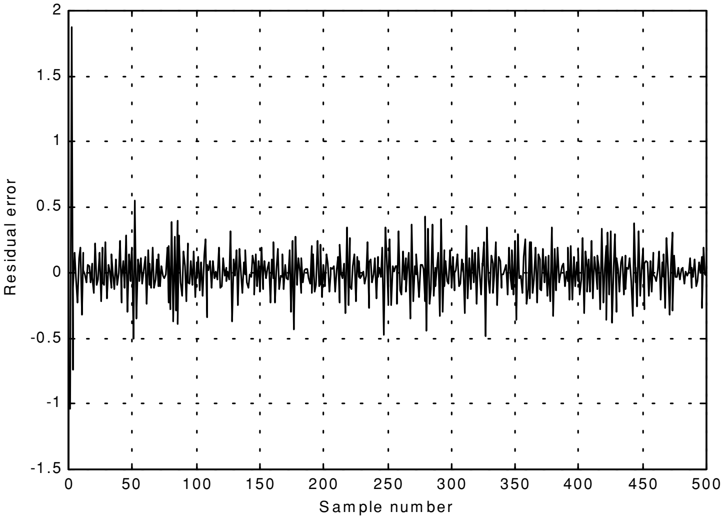


Figure 11.25a-2. Position residual (error). Large gain coefficients. Noise present. The error settles down quickly. The variation is due to noise.

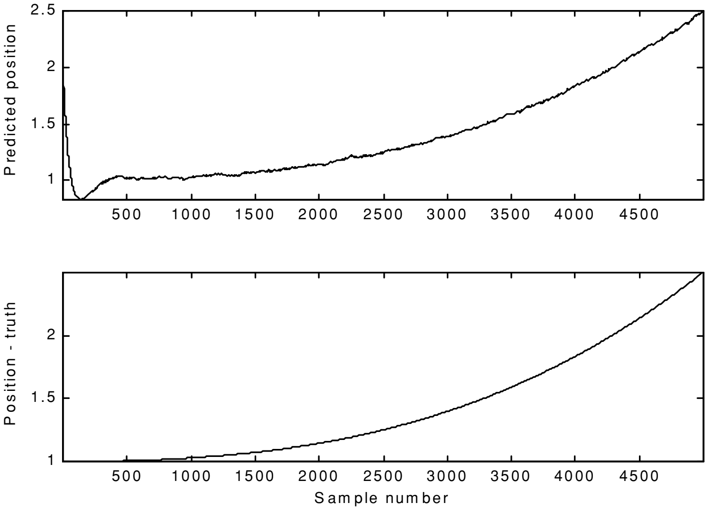


Figure 11.25b-1. Predicted and true position. $\xi = 0.9$ (i.e., small gain coefficients). Noise is present.

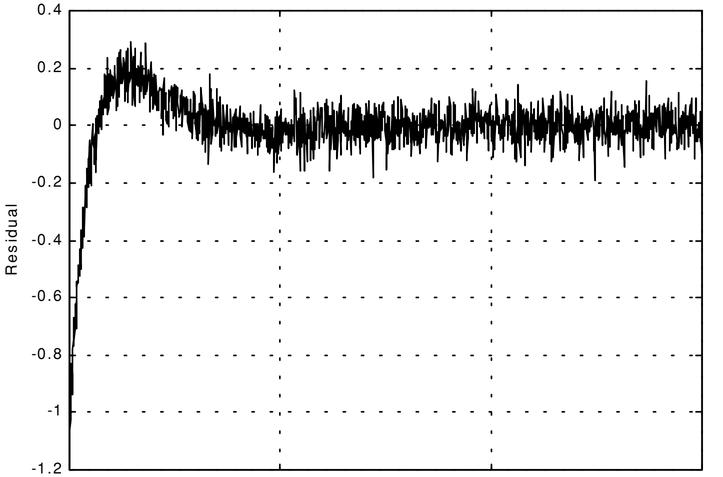


Figure 11.25b-2. Position residual (error). Small gain coefficients. Noise present. The error requires more time before settling down. The variation is due to noise.

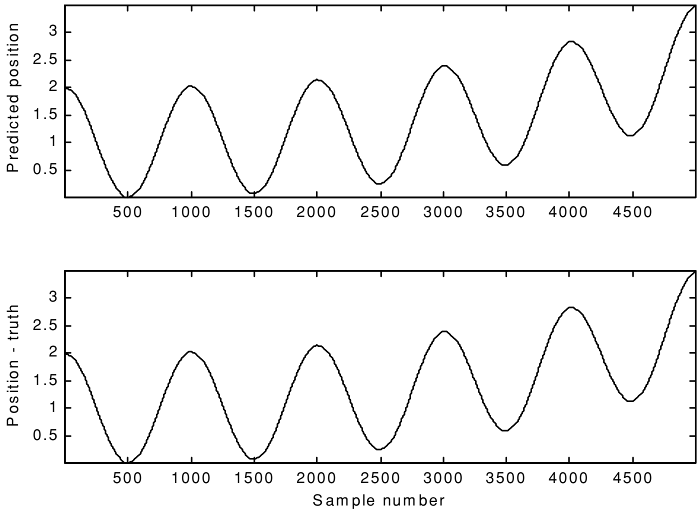


Figure 11.26a. Predicted and true position. $\xi = 0.1$ (i.e., large gain coefficients). Noise is present.

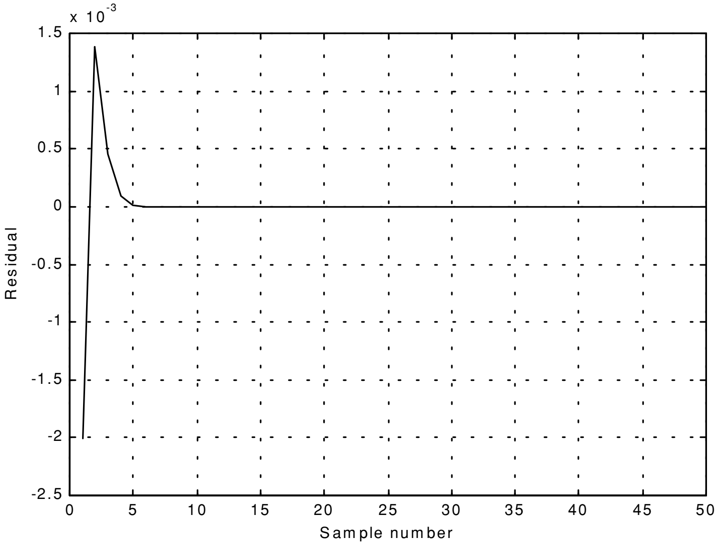


Figure 11.26b. Position residual (error). Large gain coefficients. No noise. The error settles down quickly.

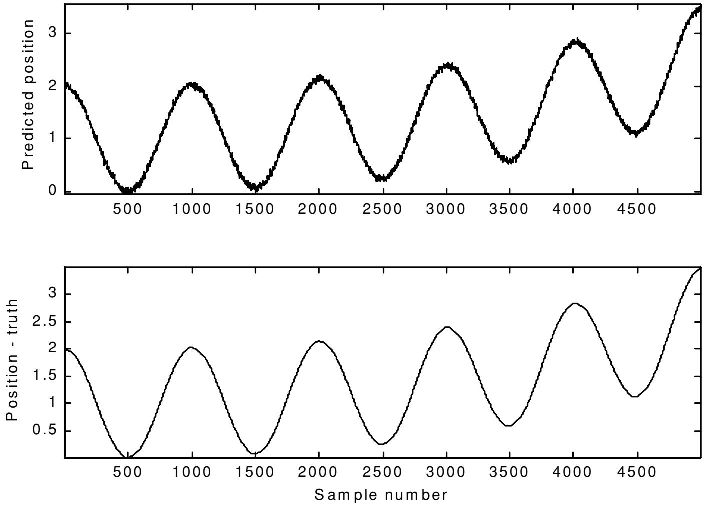


Figure 11.27a. Predicted and true position. $\xi = 0.8$ (i.e., small gain coefficients). Noise is present.

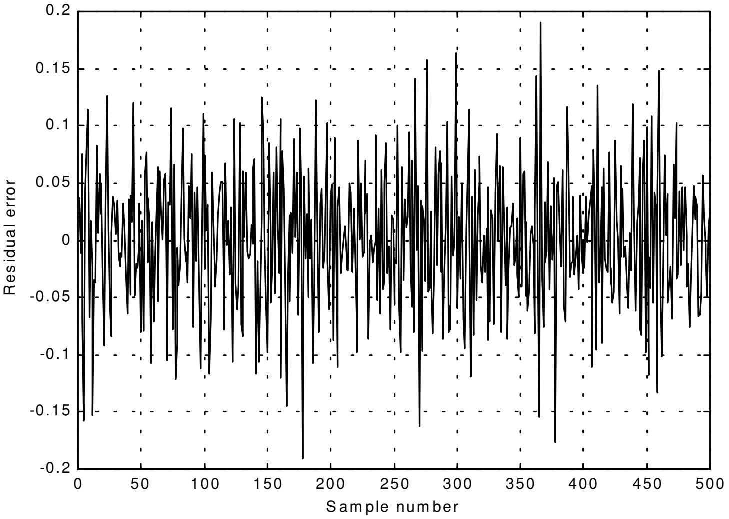


Figure 11.27b. Position residual (error). Small gain coefficients. Noise present. The error stays fairly large; however, its average is around zero. The variation is due to noise.

11.9. The Kalman Filter

The Kalman filter is a linear estimator that minimizes the mean squared error as long as the target dynamics are modeled accurately. All other recursive filters, such as the $\alpha\beta\gamma$ and the Benedict-Bordner filters, are special cases of the general solution provided by the Kalman filter for the mean squared estimation problem. Additionally, the Kalman filter has the following advantages:

1. *The gain coefficients are computed dynamically. This means that the same filter can be used for a variety of maneuvering target environments.*
2. *The Kalman filter gain computation adapts to varying detection histories, including missed detections.*
3. *The Kalman filter provides an accurate measure of the covariance matrix. This allows for better implementation of the gating and association processes.*
4. *The Kalman filter makes it possible to partially compensate for the effects of miss-correlation and miss-association.*

Many derivations of the Kalman filter exist in the literature; only results are provided in this chapter. Fig. 11.28 shows a block diagram for the Kalman filter. The Kalman filter equations can be deduced from Fig. 11.28. The filtering equation is

$$\underline{x}(n|n) = \underline{x}_s(n) = \underline{x}(n|n-1) + K(n)[\underline{y}(n) - \underline{G}\underline{x}(n|n-1)] \tag{11.127}$$

The measurement vector is

$$\underline{y}(n) = \underline{G}\underline{x}(n) + \underline{v}(n) \tag{11.128}$$

where $\underline{v}(n)$ is zero mean, white Gaussian noise with covariance $\Re_c$,

$$\Re_c = E\{\underline{y}(n) \ \ \underline{y}^t(n)\} \tag{11.129}$$

The gain (weights) vector is dynamically computed as

$$\underline{K}(n) = \underline{P}(n|n-1)\underline{G}^t[\underline{G}\underline{P}(n|n-1)\underline{G}^t + \Re_c]^{-1} \tag{11.130}$$

where the measurement noise matrix $\underline{P}$ represents the predictor covariance matrix, and is equal to

$$\underline{P}(n+1|n) = E\{\underline{x}_s(n+1)\underline{x}^*{}_s(n)\} = \underline{\Phi}\underline{P}(n|n)\underline{\Phi}^t + \underline{Q} \tag{11.131}$$

where $\underline{Q}$ is the covariance matrix for the input $\underline{u}$,

$$\underline{Q} = E\{\underline{u}(n)\ \underline{u}^t(n)\} \tag{11.132}$$

The corrector equation (covariance of the smoothed estimate) is

$$\underline{P}(n|n) = [\underline{I} - \underline{K}(n)\underline{G}]\underline{P}(n|n-1) \tag{11.133}$$

Finally, the predictor equation is

$$\underline{x}(n+1|n) = \underline{\Phi}\underline{x}(n|n) \tag{11.134}$$

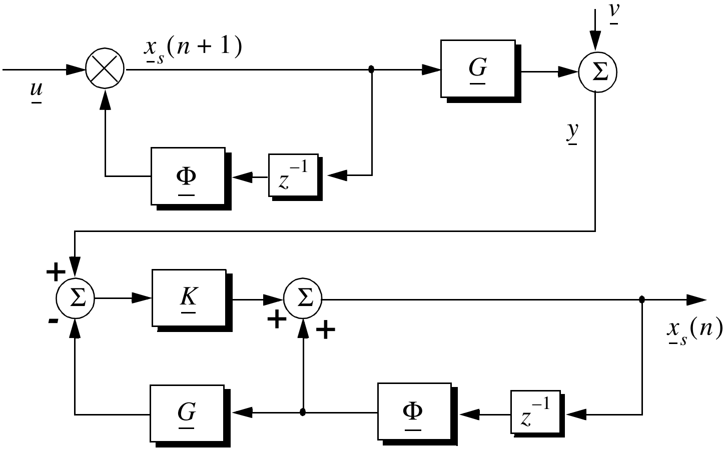


Figure 11.28. Structure of the Kalman filter.

11.9.1. The Singer $\alpha\beta\gamma$-Kalman Filter

The Singer[1] filter is a special case of the Kalman where the filter is governed by a specified target dynamic model whose acceleration is a random process with autocorrelation function given by

$$E\{\ddot{x}(t)\ \ddot{x}(t+t_1)\} = \sigma_a^2\ e^{-\frac{|t_1|}{\tau_m}} \tag{11.135}$$

[1] Singer, R. A., Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets, *IEEE Transaction on aerospace and Electronics, AES-5*, July, 1970. pp. 473-483.

where τ_m is the correlation time of the acceleration due to target maneuver or atmospheric turbulence. The correlation time τ_m may vary from as low as 10 seconds for aggressive maneuvering to as large as 60 seconds for lazy maneuver cases.

Singer defined the random target acceleration model by a first order Markov process given by

$$\ddot{x}(n+1) = \rho_m \ \ddot{x}(n) + \sqrt{1-\rho_m^2} \ \sigma_m \ w(n) \tag{11.136}$$

where $w(n)$ is a zero mean, Gaussian random variable with unity variance, σ_m is the maneuver standard deviation, and the maneuvering correlation coefficient ρ_m is given by

$$\rho_m = e^{-\frac{T}{\tau_m}} \tag{11.137}$$

The continuous time domain system that corresponds to these conditions is as the Wiener-Kolmogorov whitening filter which is defined by the differential equation

$$\frac{d}{dt}v(t) = -\beta_m v(t) + w(t) \tag{11.138}$$

where β_m is equal to $1/\tau_m$. The maneuvering variance using Singer's model is given by

$$\sigma_m^2 = \frac{A_{max}^2}{3}[1 + 4P_{max} - P_0] \tag{11.139}$$

A_{max} is the maximum target acceleration with probability P_{max} and the term P_0 defines the probability that the target has no acceleration.

The transition matrix that corresponds to the Singer filter is given by

$$\underline{\Phi} = \begin{bmatrix} 1 & T & \frac{1}{\beta_m^2}(-1 + \beta_m T + \rho_m) \\ 0 & 1 & \frac{1}{\beta_m}(1 - \rho_m) \\ 0 & 0 & \rho_m \end{bmatrix} \tag{11.140}$$

Note that when $T\beta_m = T/\tau_m$ is small (the target has constant acceleration), then Eq. (11.140) reduces to Eq. (11.114). Typically, the sampling interval T is much less than the maneuver time constant τ_m; hence, Eq. (11.140) can be accurately replaced by its second order approximation. More precisely,

$$\underline{\Phi} = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T(1 - T/2\tau_m) \\ 0 & 0 & \rho_m \end{bmatrix} \tag{11.141}$$

The covariance matrix was derived by Singer, and it is equal to

$$\underline{C} = \frac{2\sigma_m^2}{\tau_m} \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \tag{11.142}$$

where

$$C_{11} = \sigma_x^2 = \frac{1}{2\beta_m^5}\left[1 - e^{-2\beta_m T} + 2\beta_m T + \frac{2\beta_m^3 T^3}{3} - 2\beta_m^2 T^2 - 4\beta_m T e^{-\beta_m T}\right] \tag{11.143}$$

$$C_{12} = C_{21} = \frac{1}{2\beta_m^4}[e^{-2\beta_m T} + 1 - 2e^{-\beta_m T} + 2\beta_m T e^{-\beta_m T} - 2\beta_m T + \beta_m^2 T^2] \tag{11.144}$$

$$C_{13} = C_{31} = \frac{1}{2\beta_m^3}[1 - e^{-2\beta_m T} - 2\beta_m T e^{-\beta_m T}] \tag{11.145}$$

$$C_{22} = \frac{1}{2\beta_m^3}[4e^{-\beta_m T} - 3 - e^{-2\beta_m T} + 2\beta_m T] \tag{11.146}$$

$$C_{23} = C_{32} = \frac{1}{2\beta_m^2}[e^{-2\beta_m T} + 1 - 2e^{-\beta_m T}] \tag{11.147}$$

$$C_{33} = \frac{1}{2\beta_m}[1 - e^{-2\beta_m T}] \tag{11.148}$$

Two limiting cases are of interest:

1. *The short sampling interval case ($T \ll \tau_m$),*

$$\lim_{\beta_m T \to 0} \underline{C} = \frac{2\sigma_m^2}{\tau_m} \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 \\ T^4/8 & T^3/3 & T^2/2 \\ T^3/6 & T^2/2 & T \end{bmatrix} \tag{11.149}$$

and the state transition matrix is computed from Eq. (11.141) as

$$\lim_{\beta_m T \to 0} \underline{\Phi} = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \tag{11.150}$$

which is the same as the case for the $\alpha\beta\gamma$ filter (constant acceleration).

2. *The long sampling interval ($T \gg \tau_m$). This condition represents the case when acceleration is a white noise process. The corresponding covariance and transition matrices are, respectively, given by*

$$\lim_{\beta_m T \to \infty} \underline{C} = \sigma_m^2 \begin{bmatrix} \frac{2T^3\tau_m}{3} & T^2\tau_m & \tau_m^2 \\ T^2\tau_m & 2T\tau_m & \tau_m \\ \tau_m^2 & \tau_m & 1 \end{bmatrix} \tag{11.151}$$

$$\lim_{\beta_m T \to \infty} \underline{\Phi} = \begin{bmatrix} 1 & T & T\tau_m \\ 0 & 1 & \tau_m \\ 0 & 0 & 0 \end{bmatrix} \tag{11.152}$$

Note that under the condition that $T \gg \tau_m$, the cross correlation terms C_{13} and C_{23} become very small. It follows that estimates of acceleration are no longer available, and thus a two state filter model can be used to replace the three state model. In this case,

$$\underline{C} = 2\sigma_m^2\tau_m \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{bmatrix} \tag{11.153}$$

$$\underline{\Phi} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \tag{11.154}$$

11.9.2. Relationship between Kalman and $\alpha\beta\gamma$ Filters

The relationship between the Kalman filter and the $\alpha\beta\gamma$ filters can be easily obtained by using the appropriate state transition matrix $\underline{\Phi}$, and gain vector $\underline{K}$ corresponding to the $\alpha\beta\gamma$ in Eq. (11.127). Thus,

$$\begin{bmatrix} x(n|n) \\ \dot{x}(n|n) \\ \ddot{x}(n|n) \end{bmatrix} = \begin{bmatrix} x(n|n-1) \\ \dot{x}(n|n-1) \\ \ddot{x}(n|n-1) \end{bmatrix} + \begin{bmatrix} k_1(n) \\ k_2(n) \\ k_3(n) \end{bmatrix} [x_0(n) - x(n|n-1)] \tag{11.155}$$

with (see Fig. 11.21)

$$x(n|n-1) = x_s(n-1) + T\ \dot{x}_s(n-1) + \frac{T^2}{2}\ \ddot{x}_s(n-1) \tag{11.156}$$

$$\dot{x}(n|n-1) = \dot{x}_s(n-1) + T\ \ddot{x}_s(n-1) \tag{11.157}$$

$$\ddot{x}(n|n-1) = \ddot{x}_s(n-1) \tag{11.158}$$

Comparing the previous three equations with the $\alpha\beta\gamma$ filter equations yields,

$$\begin{bmatrix} \alpha \\ \frac{\beta}{T} \\ \frac{\gamma}{T^2} \end{bmatrix} = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} \tag{11.159}$$

Additionally, the covariance matrix elements are related to the gain coefficients by

$$\begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \frac{1}{C_{11} + \sigma_v^2} \begin{bmatrix} C_{11} \\ C_{12} \\ C_{13} \end{bmatrix} \tag{11.160}$$

Eq. (11.160) indicates that the first gain coefficient depends on the estimation error variance to the total residual variance, while the other two gain coefficients are calculated through the covariances between the second and third states and the first observed state.

MATLAB Function "kalman_filter.m"

The function *"kalman_filter.m"*[1] implements the Singer-$\alpha\beta\gamma$ Kalman filter. It is given in Listing 11.4 in Section 11.10. The syntax is as follows:

[residual, estimate] = kalman_filter(npts, T, X0, inp, R, nvar)

[1] This function was written by Mr. Edward Shamsi of COLSA Corporation in Huntsville, AL.

where

Symbol	Description	Status
npts	*number of points in input position*	*input*
T	*sampling interval*	*input*
X0	*initial state vector*	*input*
inp	*input array*	*input*
R	*noise variance see Eq. (11-129)*	*input*
nvar	*desired state noise variance*	*input*
residual	*array of position error (residual)*	*output*
estimate	*array of predicted position*	*output*

Note that *"kalman_filter.m"* uses MATLAB's function *"normrnd.m"* to generate zero mean Gaussian noise, which is part of MATLAB's Statistics Toolbox.

To illustrate how to use the functions "*kalman_filter.m*", consider the inputs shown in Figs. 11.22 and 11.23. Figs. 11.29 and 11.30 show the residual error and predicted position corresponding to Figs. 11.22 and 11.23. These plots can be reproduced using the program *"fig11_28.m"* given in Listing 11.5 in Section 11.10.

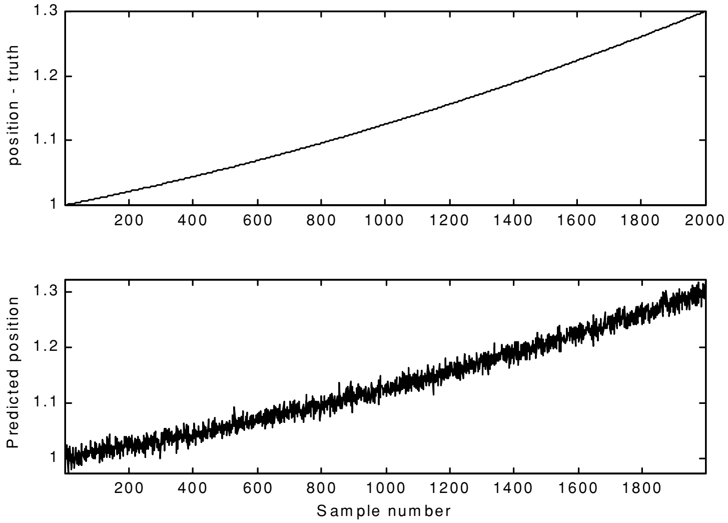


Figure 11.29a. True and predicted positions. Lazy maneuvering. Plot produced using the function *"kalman_filter.m"*.

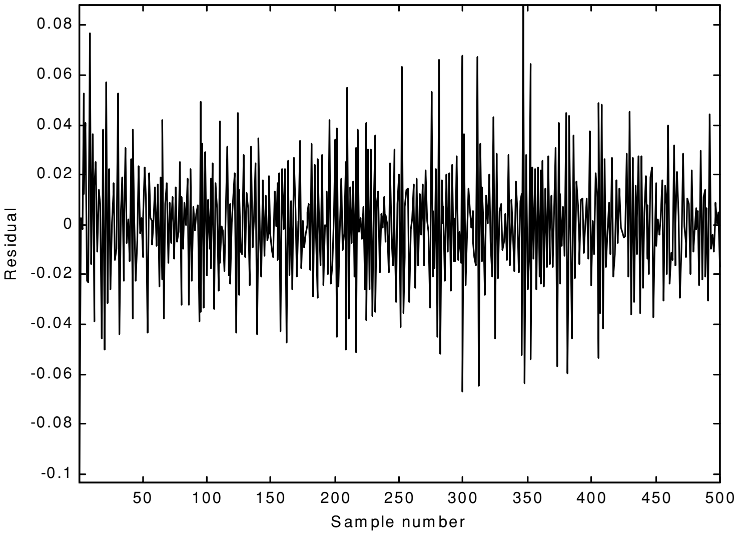


Figure 11.29b. Residual corresponding to Fig. 11.29a.

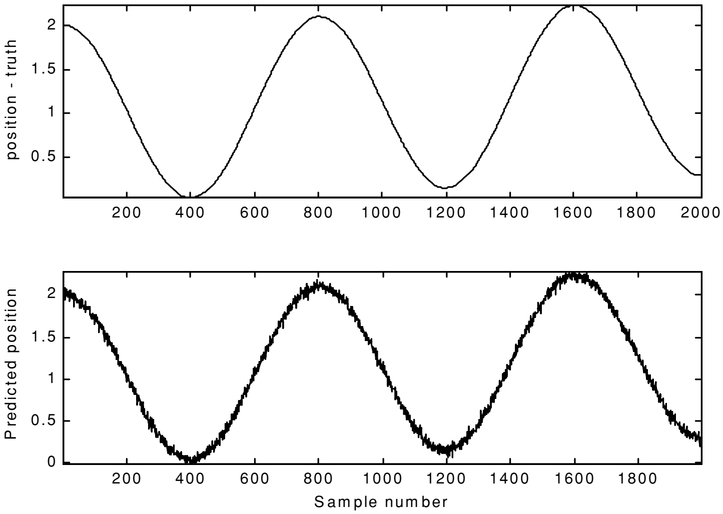


Figure 11.30a. True and predicted positions. Aggressive maneuvering. Plot produced using the function *"kalman_filter.m"*.

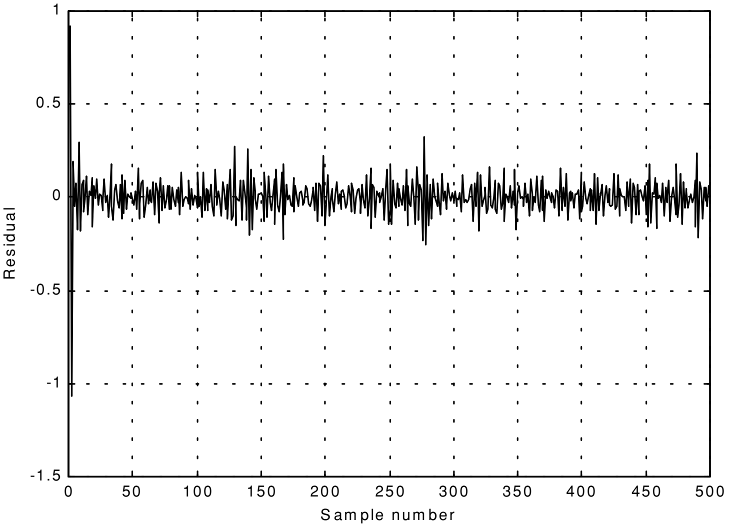


Figure 11.30b. Residual corresponding to Fig. 11.30a.

11.10. MATLAB Programs and Functions

This section contains listings of all MATLAB programs and functions used in this chapter. Users are encouraged to rerun these codes with different inputs in order to enhance their understanding of the theory.

Listing 11.1. MATLAB Function "mono_pulse.m"

```
function mono_pulse(phi0)
eps = 0.0000001;
angle = -pi:0.01:pi;
y1 = sinc(angle + phi0);
y2 = sinc((angle - phi0));
ysum = y1 + y2;
ydif = -y1 + y2;
figure (1)
plot (angle,y1,'k',angle,y2,'k');
grid;
xlabel ('Angle - radians')
ylabel ('Squinted patterns')
figure (2)
plot(angle,ysum,'k');
```

```
grid;
xlabel ('Angle - radians')
ylabel ('Sum pattern')
figure (3)
plot (angle,ydif,'k');
grid;
xlabel ('Angle - radians')
ylabel ('Difference pattern')
angle = -pi/4:0.01:pi/4;
y1 = sinc(angle + phi0);
y2 = sinc((angle - phi0));
ydif = -y1 + y2;
ysum = y1 + y2;
dovrs = ydif ./ ysum;
figure(4)
plot (angle,dovrs,'k');
grid;
xlabel ('Angle - radians')
ylabel ('voltage gain')
```

Listing 11.2. MATLAB Function “ghk_tracker.m”

```
function [residual, estimate] = ghk_tracker (X0, smoocof, inp, npts, T, nvar)
rn = 1.;
% read the initial estimate for the state vector
X = X0;
theta = smoocof;
%compute values for alpha, beta, gamma
w1 = 1. - (theta^3);
w2 = 1.5 * (1. + theta) * ((1. - theta)^2) / T;
w3 = ((1. - theta)^3) / (T^2);
% setup the transition matrix PHI
PHI = [1. T (T^2)/2.;0. 1. T;0. 0. 1.];
while rn < npts ;
  %use the transition matrix to predict the next state
  XN = PHI * X;
  error = (inp(rn) + normrnd(0,nvar)) - XN(1);
  residual(rn) = error;
  tmp1 = w1 * error;
  tmp2 = w2 * error;
  tmp3 = w3 * error;
  % compute the next state
  X(1) = XN(1) + tmp1;
  X(2) = XN(2) + tmp2;
  X(3) = XN(3) + tmp3;
  estimate(rn) = X(1);
  rn = rn + 1.;
end
```

```
return
```

MATLAB Function "ghk_traker1.m"

```
function [residual, estimate] = ghk_tracker1 (X0, smoocof, inp, npts, T)
rn = 1.;
% read the initial estimate for the state vector
X = X0;
theta = smoocof;
%compute values for alpha, beta, gamma
w1 = 1. - (theta^3);
w2 = 1.5 * (1. + theta) * ((1. - theta)^2) / T;
w3 = ((1. - theta)^3) / (T^2);
% setup the transition matrix PHI
PHI = [1. T (T^2)/2.;0. 1. T;0. 0. 1.];
while rn < npts ;
  %use the transition matrix to predict the next state
  XN = PHI * X;
  error = inp(rn)  - XN(1);
  residual(rn) = error;
  tmp1 = w1 * error;
  tmp2 = w2 * error;
  tmp3 = w3 * error;
  % compute the next state
  X(1) = XN(1) + tmp1;
  X(2) = XN(2) + tmp2;
  X(3) = XN(3) + tmp3;
  estimate(rn) = X(1);
  rn = rn + 1.;
end
return
```

Listing 11.3. MATLAB Program "fig11_21.m"

```
clear all
eps = 0.0000001;
npts = 5000;
del = 1./ 5000.;
t = 0. : del : 1.;
% generate input sequence
inp = 1.+ t.^3 + .5 .*t.^2 + cos(2.*pi*10 .* t) ;
% read the initial estimate for the state vector
X0 = [2,.1,.01]';
% this is the update interval in seconds
T = 100. * del;
% this is the value of the smoothing coefficient
xi = .91;
[residual, estimate] = ghk_tracker (X0, xi, inp, npts, T, .01);
```

```
figure(1)
plot (residual(1:500))
xlabel ('Sample number')
ylabel ('Residual error')
grid
figure(2)
NN = 4999.;
n = 1:NN;
plot (n,estimate(1:NN),'b',n,inp(1:NN),'r')
xlabel ('Sample number')
ylabel ('Position')
legend ('Estimated','Input')
```

Listing 11.4. MATLAB Function "kalman_filter.m"

```
function [residual, estimate] = kalman_filter(npts, T, X0, inp, R, nvar)
N = npts;
rn=1;
% read the initial estimate for the state vector
X = X0;
% it is assumed that the measurmeny vector H=[1,0,0]
% this is the state noise variance
VAR = nvar;
% setup the initial value for the predication covariance.
S = [1. 1. 1.; 1. 1. 1.; 1. 1. 1.];
% setup the transition matrix PHI
PHI = [1. T (T^2)/2.; 0. 1. T; 0. 0. 1.];
% setup the state noise covariance matrix
Q(1,1) = (VAR * (T^5)) / 20.;
Q(1,2) = (VAR * (T^4)) / 8.;
Q(1,3) = (VAR * (T^3)) / 6.;
Q(2,1) = Q(1,2);
Q(2,2) = (VAR * (T^3)) / 3.;
Q(2,3) = (VAR * (T^2)) / 2.;
Q(3,1) = Q(1,3);
Q(3,2) = Q(2,3);
Q(3,3) = VAR * T;
while rn < N ;
  %use the transition matrix to predict the next state
  XN = PHI * X;
  % Perform error covariance extrapolation
  S = PHI * S * PHI' + Q;
  % compute the Kalman gains
  ak(1) = S(1,1) / (S(1,1) + R);
  ak(2) = S(1,2) / (S(1,1) + R);
  ak(3) = S(1,3) / (S(1,1) + R);
  %perform state estimate update:
  error = inp(rn) + normrnd(0,R) - XN(1);
  residual(rn) = error;
  tmp1 = ak(1) * error;
  tmp2 = ak(2) * error;
  tmp3 = ak(3) * error;
  X(1) = XN(1) + tmp1;
  X(2) = XN(2) + tmp2;
  X(3) = XN(3) + tmp3;
  estimate(rn) = X(1);
  % update the error covariance
  S(1,1) = S(1,1) * (1. -ak(1));
  S(1,2) = S(1,2) * (1. -ak(1));
  S(1,3) = S(1,3) * (1. -ak(1));
  S(2,1) = S(1,2);
  S(2,2) = -ak(2) * S(1,2) + S(2,2);
  S(2,3) = -ak(2) * S(1,3) + S(2,3);
  S(3,1) = S(1,3);
  S(3,3) = -ak(3) * S(1,3) + S(3,3);
  rn = rn + 1.;
end
```

Listing 11.5. MATLAB Program "fig11_28.m"

```
clear all
npts = 2000;
del = 1/2000;
t = 0:del:1;
inp = (1+.2 .* t + .1 .*t.^2) + cos(2. * pi * 2.5 .* t);
X0 = [1,.1,.01]';
% it is assumed that the measurmeny vector H=[1,0,0]
% this is the update interval in seconds
T = 1.;
% enter the measurement noise variance
R = .035;
% this is the state noise variance
nvar = .5;
[residual, estimate] = kalman_filter(npts, T, X0, inp, R, nvar);
figure(1)
plot(residual)
xlabel ('Sample number')
ylabel ('Residual')
figure(2)
subplot(2,1,1)
plot(inp)
axis tight
ylabel ('position - truth')
subplot(2,1,2)
plot(estimate)
axis tight
```

```
xlabel ('Sample number')
ylabel ('Predicted position')
```

Problems

11.1. Show that in order to be able to quickly achieve changing the beam position the error signal needs to be a linear function of the deviation angle.

11.2. Prepare a short report on the vulnerability of conical scan to amplitude modulation jamming. In particular consider the self-protecting technique called “Gain Inversion.”

11.3. Consider a conical scan radar. The pulse repetition interval is $10\mu s$. Calculate the scan rate so that at least ten pulses are emitted within one scan.

11.4. Consider a conical scan antenna whose rotation around the tracking axis is completed in 4 seconds. If during this time 20 pulses are emitted and received, calculate the radar PRF and the unambiguous range.

11.5. Reproduce Fig. 11.11 for $\varphi_0 = 0.05, 0.1, 0.15$ radians.

11.6. Reproduce Fig. 11.13 for the squint angles defined in the previous problem.

11.7. Derive Eq. (11.33) and Eq. (11.34).

11.8. Consider a monopulse radar where the input signal is comprised of both target return and additive white Gaussian noise. Develop an expression for the complex ratio Σ/Δ.

11.9. Consider the sum and difference signals defined in Eqs. (11.7) and (11.8). What is the squint angle φ_0 that maximizes $\Sigma(\varphi = 0)$?

11.10. A certain system is defined by the following difference equation:

$$y(n) + 4y(n-1) + 2y(n-2) = w(n)$$

Find the solution to this system for $n > 0$ and $w = \delta$.

11.11. Prove the state transition matrix properties (i.e., Eqs. (11.30) through (11.36)).

11.12. Suppose that the state equations for a certain discrete time LTI system are

$$\begin{bmatrix} x_1(n+1) \\ x_2(n+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(n)$$

If $y(0) = y(1) = 1$, find $y(n)$ when the input is a step function.

11.13. Derive Eq. (11.55).

11.14. Derive Eq. (11.75).

11.15. Using Eq. (11.83), compute a general expression (in terms of the transfer function) for the steady state errors when the input sequence is:

$$u1 = \{0, 1, 1, 1, 1, \ldots\}$$

$$u2 = \{0, 1, 2, 3, \ldots\}$$

$$u3 = \{0, 1^2, 2^2, 3^2, \ldots\}$$

$$u4 = \{0, 1^3, 2^3, 3^3, \ldots\}$$

11.16. Verify the results in Eqs. (11.99) and (11.100).

11.17. Develop an expression for the steady state error transfer function for an $\alpha\beta$ tracker.

11.18. Using the result of the previous problem and Eq. (11.83), compute the steady-state errors for the $\alpha\beta$ tracker with the inputs defined in Problem 11.13.

11.19. Design a critically damped $\alpha\beta$, when the measurement noise variance associated with position is $\sigma_v^2 = 50m$ and when the desired standard deviation of the filter prediction error is $5.5m$.

11.20. Derive Eqs. (11.118) through (11.120).

11.21. Derive Eq. (11.122).

11.22. Consider a $\alpha\beta\gamma$ filter. We can define six transfer functions: $H_1(z)$, $H_2(z)$, $H_3(z)$, $H_4(z)$, $H_5(z)$, and $H_6(z)$ (predicted position, predicted velocity, predicted acceleration, smoothed position, smoothed velocity, and smoothed acceleration). Each transfer function has the form

$$H(z) = \frac{a_3 + a_2 z^{-1} + a_1 z^{-2}}{1 + b_2 z^{-1} + b_1 z^{-2} + b_0 z^{-3}}$$

The denominator remains the same for all six transfer functions. Compute all the relevant coefficients for each transfer function.

11.23. Verify the results obtained for the two limiting cases of the Singer-Kalman filter.

11.24. Verify Eq. (11.160).