Chapter 10

Tracking a Sine Wave

Introduction

IN THIS chapter we will attempt to apply extended Kalman filtering to a problem we briefly investigated with a linear Kalman filter. We will revisit the problem of tracking a sinusoidal signal measurement corrupted by noise. We have already shown that if the frequency of the sinusoid was known in advance the signal could be tracked quite well with a linear Kalman filter. However, if the frequency of the sinusoidal signal is unknown either our model of the real world or our measurement model becomes nonlinear, and we must resort to an extended Kalman filter. Several possible extended Kalman filters for this application, each of which has different states, will be explored in this chapter.

Extended Kalman Filter

In Chapter 5 we attempted to estimate the states (i.e., derivatives) of a sinusoidal signal based on noisy measurement of the sinusoidal signal. We showed that a linear polynomial Kalman filter was adequate for estimation purposes, but a much better linear Kalman filter, which made use of the fact that we knew that the true signal was sinusoidal, could be constructed. However, it was also demonstrated that if our a priori information was in error (i.e., knowledge of the frequency of the sinusoid is in error) the performance of the better linear Kalman filter deteriorated to the point where the estimates were no better, and possibly worse, than that of the linear polynomial Kalman filter (i.e., linear Kalman filter required no a priori information at all). In this section we will attempt to build an extended Kalman filter that operates on the sinusoidal measurement but also takes into account that the frequency of the sinusoidal signal is unknown and must also be estimated.[1,2]

In some of the one-dimensional work that follows, we will make use of the concept of negative and positive frequencies. Some of these frequency concepts may be difficult to understand from a physical point of view. These concepts are much easier to visualize in two dimensions. For example, in the two-dimensional x-y plane, circular motion can be described by the equations

$$x = A \sin \omega t$$
$$y = A \cos \omega t$$

In this case a positive frequency means that circular motion will be clockwise, and a negative frequency will describe circular motion in the counterclockwise direction.

In this chapter we consider the one-dimensional sinusoidal signal given by

$$x = A \sin \omega t$$

We can define a new variable ϕ to be

$$\phi = \omega t$$

If the frequency of the sinusoid is constant, we can take the derivative of the preceding equation to obtain

$$\dot{\phi} = \omega$$

Similarly, by assuming that the frequency and amplitude of the sinusoid are constant we are also saying that

$$\dot{\omega} = 0$$
$$\dot{A} = 0$$

We can express the preceding set of scalar equations that model our version of the real world in state-space form as

$$\begin{bmatrix} \dot{\phi} \\ \dot{\omega} \\ \dot{A} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \omega \\ A \end{bmatrix} + \begin{bmatrix} 0 \\ u_{s1} \\ u_{s2} \end{bmatrix}$$

where u_{s1} and u_{s2} are white process noise sources that have been added to the derivatives of frequency and amplitude. These white noise sources may be required later to get the filter to work if we encounter problems. The process noise has been added to the derivatives of the states that are the least certain because we really do not know if the frequency and amplitude of the sinusoid will be constant. The preceding state-space equation is linear for this particular formulation. From the preceding state-space equation the continuous process-noise matrix can be written by inspection as

$$\boldsymbol{Q} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \Phi_{s1} & 0 \\ 0 & 0 & \Phi_{s2} \end{bmatrix}$$

where Φ_{s1} and Φ_{s2} are the spectral densities of the white noise sources u_{s1} and u_{s2}. The systems dynamics matrix also can be written by inspection of the state-space equation as

$$F = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Because F^2 is zero as a result of

$$F^2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

the fundamental matrix can be expressed exactly as the two-term Taylor-series expansion

$$\Phi = I + Ft = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} t = \begin{bmatrix} 1 & t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Therefore, the exact discrete fundamental matrix can be obtained by replacing t with T_s, yielding

$$\Phi_k = \begin{bmatrix} 1 & T_s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Although the state-space equation is linear, the measurement equation is nonlinear for this particular formulation of the problem. In fact, information concerning the sinusoidal nature of the signal is buried in the measurement equation. We have constructed the problem so that the linearized measurement equation (i.e., we are actually measuring x, which is not a state, thus making the measurement nonlinear) is given by

$$\Delta x^* = \begin{bmatrix} \dfrac{\partial x}{\partial \phi} & \dfrac{\partial x}{\partial \omega} & \dfrac{\partial x}{\partial A} \end{bmatrix} \begin{bmatrix} \Delta\phi \\ \Delta\omega \\ \Delta A \end{bmatrix} + v$$

where v is white measurement noise. To calculate the partial derivatives of the preceding equation, we first recall that

$$x = A \sin \omega t = A \sin \phi$$

Therefore, the partial derivatives in the measurement matrix are easily evaluated as

$$\frac{\partial x}{\partial \phi} = A\cos\phi$$
$$\frac{\partial x}{\partial \omega} = 0$$
$$\frac{\partial x}{\partial A} = \sin\phi$$

making the linerized measurement matrix (i.e., matrix of partial differentials)

$$\boldsymbol{H} = [A\,\cos\phi \quad 0 \quad \sin\phi]$$

The measurement matrix is evaluated at the projected estimates of A and ϕ. In this formulation the measurement noise is a scalar. Therefore, the discrete measurement noise matrix will also be a scalar given by

$$\boldsymbol{R}_k = \sigma_x^2$$

where σ_x^2 is the variance of the measurement noise.

Recall that the discrete process-noise matrix can be found from the continuous process-noise matrix according to

$$\boldsymbol{Q}_k = \int_0^{T_2} \Phi(\tau)\boldsymbol{Q}\Phi^T(\tau)\,\mathrm{d}t$$

Substitution of the appropriate matrices into the preceding expression yields

$$\boldsymbol{Q}_k = \int_0^{T_s} \begin{bmatrix} 1 & \tau & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \Phi_{s1} & 0 \\ 0 & 0 & \Phi_{s2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ \tau & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathrm{d}\tau$$

After multiplying out the three matrices we obtain

$$\boldsymbol{Q}_k = \int_0^{T_s} \begin{bmatrix} \tau^2\Phi_{s1} & \tau\Phi_{s1} & 0 \\ \tau\Phi_{s1} & \Phi_{s1} & 0 \\ 0 & 0 & \Phi_{s2} \end{bmatrix} \mathrm{d}\tau$$

Finally, integrating the preceding expression shows the discrete process-noise matrix to be

$$\boldsymbol{Q}_k = \begin{bmatrix} \frac{\Phi_{s1}T_s^3}{3} & \frac{\Phi_{s1}T_s^2}{2} & 0 \\ \frac{\Phi_{s1}T_s^2}{2} & \Phi_{s1}T_s & 0 \\ 0 & 0 & \Phi_{s2}T_s \end{bmatrix}$$

We now have enough information to solve the matrix Riccati equations for the Kalman gains.

Because the fundamental matrix is exact in this application, we can also use it to exactly propagate the state estimates in the extended Kalman filter over the sampling interval. In other words, by using the fundamental matrix we can propagate the states from time $k-1$ to time k or in matrix form

$$\begin{bmatrix} \bar{\phi}_k \\ \bar{\omega}_k \\ \bar{A}_k \end{bmatrix} = \begin{bmatrix} 1 & T_s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\phi}_{k-1} \\ \hat{\omega}_{k-1} \\ \hat{A}_{k-1} \end{bmatrix}$$

We can multiply out the preceding matrix equation to get the three scalar equations

$$\begin{aligned} \bar{\phi}_k &= \hat{\phi}_{k-1} + \hat{\omega}_{k-1}T_s \\ \bar{\omega}_k &= \hat{\omega}_{k-1} \\ \bar{A}_k &= \hat{A}_{k-1} \end{aligned}$$

The linearized measurement matrix is used in the Riccati equations, but we do not have to use that matrix in the computation of the residual for the actual extended Kalman-filtering equations. We can do better by using the actual nonlinear equation for the residual or

$$\text{Res}_k = x_k^* - \bar{A}_k \sin \bar{\phi}_k$$

where the residual has been computed using the projected values of the states (i.e., barred quantities in the preceding equation). Now the extended Kalman-filtering equations can be written as

$$\begin{aligned} \hat{\phi}_k &= \bar{\phi}_k + K_{1_k}\text{Res}_k \\ \hat{\omega}_k &= \bar{\omega}_k + K_{2_k}\text{Res}_k \\ \hat{A}_k &= \bar{A}_k + K_{3_k}\text{Res}_k \end{aligned}$$

Listing 10.1 presents the extended Kalman filter for estimating the states of a noisy sinusoidal signal whose frequency is unknown. We can see that the

Listing 10.1 Extended Kalman filter for sinusoidal signal with unknown frequency

```
C THE FIRST THREE STATEMENTS INVOKE THE ABSOFT RANDOM
  NUMBER GENERATOR ON THE MACINTOSH
        GLOBAL DEFINE
                INCLUDE 'quickdraw.inc'
        END
        IMPLICIT REAL*8(A-H,O-Z)
        REAL*8 P(3,3),Q(3,3),M(3,3),PHI(3,3),HMAT(1,3),HT(3,1),PHIT(3,3)
        REAL*8 RMAT(1,1),IDN(3,3),PHIP(3,3),PHIPPHIT(3,3),HM(1,3)
        REAL*8 HMHT(1,1),HMHTR(1,1),HMHTRINV(1,1),MHT(3,1),K(3,1)
        REAL*8 KH(3,3),IKH(3,3)
        INTEGER ORDER
        A=1.
        W=1.
        TS=.1
        ORDER=3
        PHIS1=0.
        PHIS2=0.
        SIGX=1.
        OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
        OPEN(2,STATUS='UNKNOWN',FILE='COVFIL')
        T=0.
        S=0.
        H=.001
        DO 14 I=1,ORDER
        DO 14 J=1,ORDER
        PHI(I,J)=0.
        P(I,J)=0.
        Q(I,J)=0.
        IDN(I,J)=0.
 14     CONTINUE
        RMAT(1,1)=SIGX**2
        IDN(1,1)=1.
        IDN(2,2)=1.
        IDN(3,3)=1.
        PHIH=0.
        WH=2.
        AH=3.
        P(1,1)=0.
        P(2,2)=(W-WH)**2
        P(3,3)=(A-AH)**2
        XT=0.
        XTD=A*W
        WHILE(T<=20.)
              XTOLD=XT
              XTDOLD=XTD
              XTDD=-W*W*XT
              XT=XT+H*XTD
              XTD=XTD+H*XTDD
              T=T+H
              XTDD=-W*W*XT
```

(*continued*)

Listing 10.1 (*Continued*)

```
          XT=.5*(XTOLD+XT+H*XTD)
          XTD=.5*(XTDOLD+XTD+H*XTDD)
          S=S+H
          IF(S>=(TS-.00001))THEN
               S=0.
               PHI(1,1)=1.
               PHI(1,2)=TS
               PHI(2,2)=1.
               PHI(3,3)=1.
               Q(1,1)=TS*TS*TS*PHIS1/3.
               Q(1,2)=.5*TS*TS*PHIS1
               Q(2,1)=Q(1,2)
               Q(2,2)=PHIS1*TS
               Q(3,3)=PHIS2*TS
               PHIB=PHIH+WH*TS
               HMAT(1,1)=AH*COS(PHIB)
               HMAT(1,2)=0.
               HMAT(1,3)=SIN(PHIB)
               CALL  MATTRN(PHI,ORDER,ORDER,PHIT)
               CALL  MATTRN(HMAT,1,ORDER,HT)
               CALL  MATMUL(PHI,ORDER,ORDER,P,ORDER,
                 ORDER,PHIP)
               CALL  MATMUL(PHIP,ORDER,ORDER,PHIT,ORDER,
1                ORDER,PHIPPHIT)
               CALL  MATADD(PHIPPHIT,ORDER,ORDER,Q,M)
               CALL  MATMUL(HMAT,1,ORDER,M,ORDER,
                 ORDER,HM)
               CALL  MATMUL(HM,1,ORDER,HT,ORDER,1,HMHT)
               CALL  MATADD(HMHT,ORDER,ORDER,RMAT,
                 HMHTR)HMHTRINV(1,1)=1./HMHTR(1,1)
               CALL  MATMUL(M,ORDER,ORDER,HT,ORDER,1,
                 MHT)
               CALL  MATMUL(MHT,ORDER,1,HMHTRINV,1,1,K)
               CALL  MATMUL(K,ORDER,1,HMAT,1,ORDER,KH)
               CALL  MATSUB(IDN,ORDER,ORDER,KH,IKH)
               CALL  MATMUL(IKH,ORDER,ORDER,M,ORDER,
                 ORDER,P)
               CALL  GAUSS(XTNOISE,SIGX)
               XTMEAS=XT+XTNOISE
               RES=XTMEAS-AH*SIN(PHIB)
               PHIH=PHIB+K(1,1)*RES
               WH=WH+K(2,1)*RES
               AH=AH+K(3,1)*RES
               ERRPHI=W*T-PHIH
               SP11=SQRT(P(1,1))
               ERRW=W-WH
               SP22=SQRT(P(2,2))
               ERRA=A-AH
               SP33=SQRT(P(3,3))
```

(*continued*)

Listing 10.1 (*Continued*)

```
                  XTH=AH*SIN(PHIH)
                  XTDH=AH*WH*COS(PHIH)
                  WRITE(9,*)T,XT,XTH,XTD,XTDH,W,WH,A,AH,W*T,
                    PHIH
                  WRITE(1,*)T,XT,XTH,XTD,XTDH,W,WH,A,AH,W*T,
                    PHIH
                  WRITE(2,*)T,ERRPHI,SP11,-SP11,ERRW,SP22,-SP22,
1                   ERRA,SP33,-SP33
              ENDIF
          END DO
          PAUSE
          CLOSE(1)
          CLOSE(2)
          END

C SUBROUTINE GAUSS IS SHOWN IN LISTING 1.8
C SUBROUTINE MATTRN IS SHOWN IN LISTING 1.3
C SUBROUTINE MATMUL IS SHOWN IN LISTING 1.4
C SUBROUTINE MATADD IS SHOWN IN LISTING 1.1
C SUBROUTINE MATSUB IS SHOWN IN LISTING 1.2
```

simulation is initially set up to run without process noise (i.e., PHIS1 = PHIS2 = 0). The actual sinusoidal signal has unity amplitude, and the standard deviation of the measurement noise is also unity. The frequency of the actual sinusoidal signal is unity. The filter's initial state estimates of ϕ is set to zero, whereas the estimate of A has been set to three (rather than unity). The initial estimate of the frequency is set to two (rather than unity). Values are used for the initial covariance matrix to reflect the uncertainties in our initial state estimates. Because the fundamental matrix is exact in this filter formulation, a special subroutine is *not* required in this simulation to integrate the state equations over the sampling interval to obtain the state projections.

The nominal case of Listing 10.1 (i.e., in which there is no process noise) was run. We can see from Figs. 10.1 and 10.2 that for this run the extended Kalman filter is able to estimate the frequency and amplitude of the sinusoidal signal quite well when the actual frequency of the sinusoid is 1 rad/s and the filter's initial frequency estimate is 2 rad/s. After approximately 5 s the frequency and amplitude estimates are near perfect.

To test the filter's robustness, another case was run in which the actual frequency of the sinusoid is negative, but the filter is initialized with a positive frequency. We can see from Fig. 10.3 that now the filter is unable to estimate the negative frequency. In fact the final frequency estimate of the extended Kalman filter is approximately zero. Figure 10.4 indicates that under these circumstances the filter is also unable to estimate the signal amplitude. The figure indicates that the extended Kalman filter estimates the amplitude to be three, rather than the true value of unity.

Even though at this point we know the filter is not satisfactory in meeting our original goals, we would like to examine it further to see if we can learn

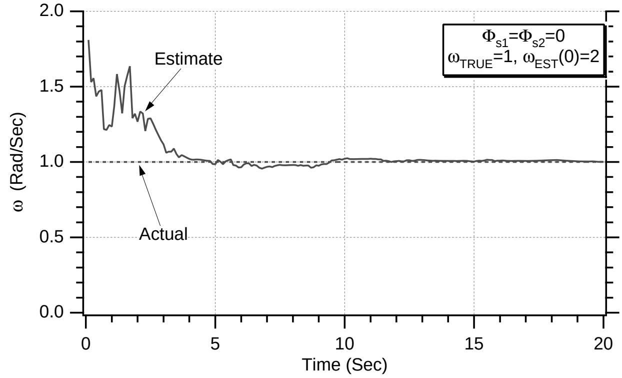

Fig. 10.1 Extended Kalman filter is able to estimate positive frequency when initial frequency estimate is also positive.

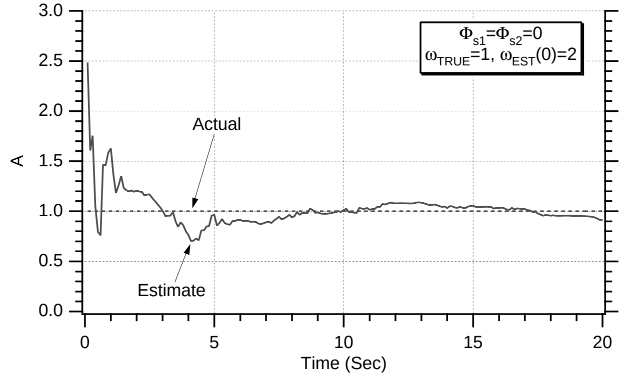

Fig. 10.2 Extended Kalman filter is able to estimate amplitude when actual frequency is positive and initial frequency estimate is also positive.

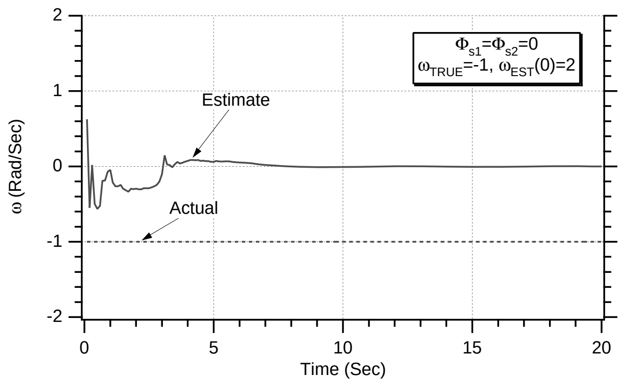

Fig. 10.3 Extended Kalman filter is unable to estimate negative frequency when initial frequency estimate is positive.

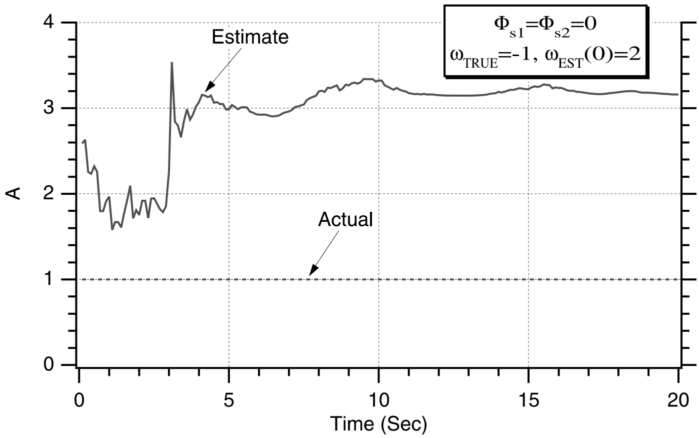

Fig. 10.4 Extended Kalman filter is unable to estimate amplitude when actual frequency is negative and initial frequency estimate is positive.

something that will be of future use. To further test the filter's properties, another case was run in which the actual frequency of the sinusoid is negative, but this time the filter is initialized with a negative frequency. We can see from Fig. 10.5 that the filter is now able to estimate the negative frequency fairly rapidly. Figure 10.6 also indicates that under these circumstances the filter is also able to estimate the signal amplitude accurately.

As a final test of the filter's properties, another case was run in which the actual frequency of the sinusoid is positive while the filter is initialized with a negative frequency. We can see from Fig. 10.7 that after approximately 10 s the filter is now able to estimate the positive frequency. However, Fig. 10.8 indicates that

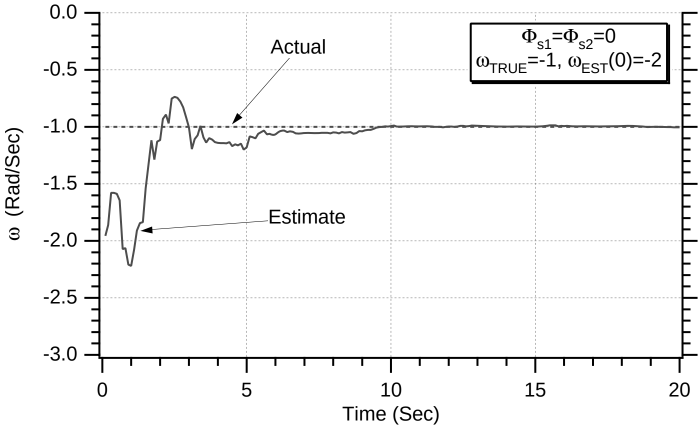

Fig. 10.5 Extended Kalman filter is now able to estimate negative frequency when initial frequency estimate is also negative.

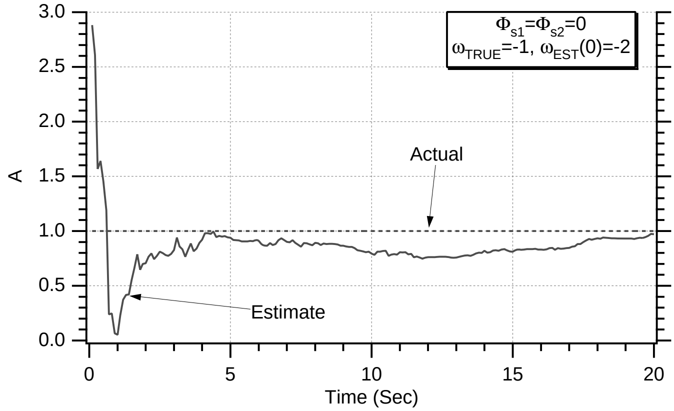

Fig. 10.6 Extended Kalman filter is now able to estimate amplitude when actual frequency is negative and initial frequency estimate is also negative.

under these circumstances the filter is unable to estimate the signal amplitude. The actual amplitude is 1, and the estimated amplitude is approximately 0.6.

From the preceding set of experiments, it appears that the extended Kalman filter only works if the initial estimate of the signal frequency is of the same sign as the actual frequency. We have already seen that when the signs of the actual frequency and initial frequency estimate are mismatched either the signal frequency or amplitude or both could not be estimated.

Sometimes adding process noise is the engineering fix for getting a Kalman filter to perform properly and robustly. Figures 10.9 and 10.10 reexamine the case where the actual frequency of the sinusoid is positive but the initial frequency estimate of the filter is negative. If we now add process noise (i.e.,

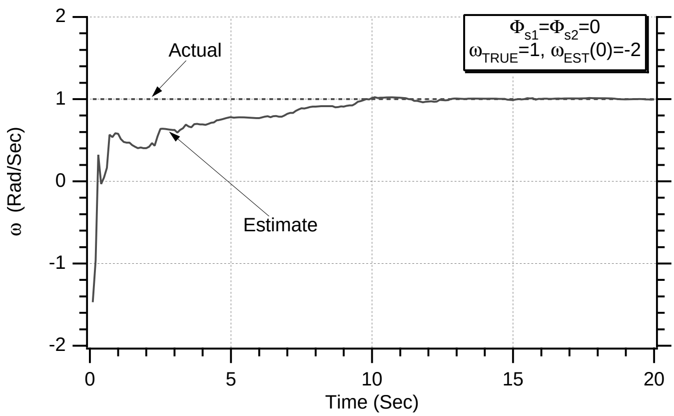

Fig. 10.7 Extended Kalman filter is able to estimate positive frequency when initial frequency estimate is negative.

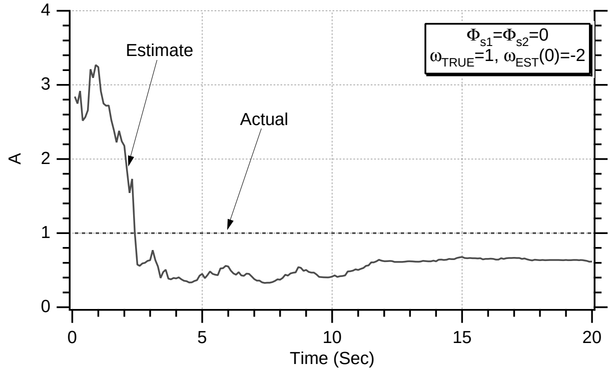

Fig. 10.8 Extended Kalman filter is not able to estimate amplitude when actual frequency is positive and initial frequency estimate is negative.

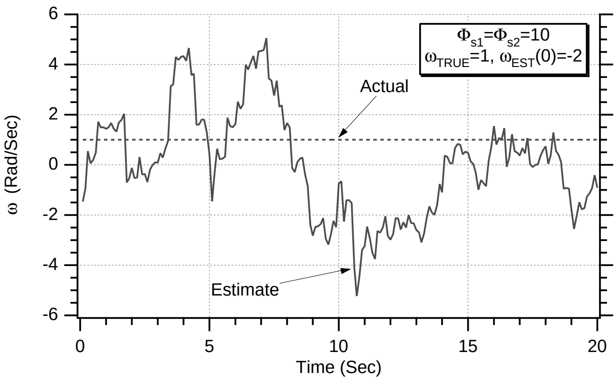

Fig. 10.9 Addition of process noise is not the engineering fix to enable filter to estimate frequency.

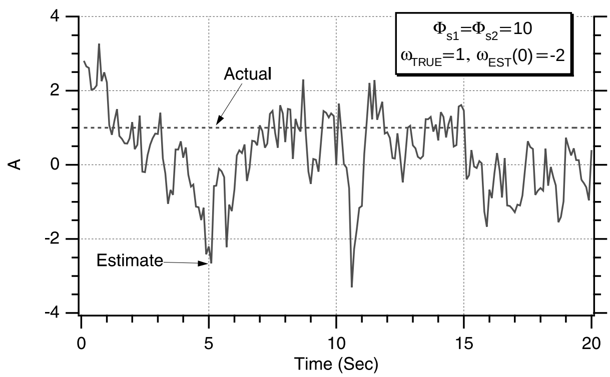

Fig. 10.10 Addition of process noise is not the engineering fix to enable filter to estimate amplitude.

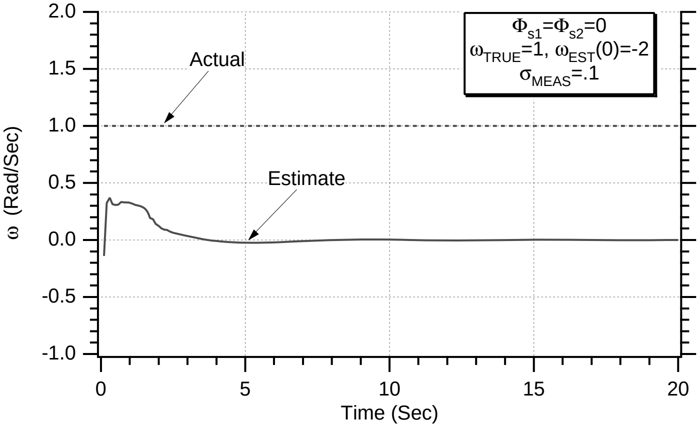

Fig. 10.11 Reducing measurement noise by an order of magnitude does not enable Kalman filter to estimate positive frequency when initial frequency estimate is negative.

$\Phi_{s1} = 10, \quad \Phi_{s2} = 10$), the extended Kalman filter's frequency and amplitude estimates are considerably worse than they were when there was no process noise at all (see Figs. 10.7–10.8). Therefore, it appears that the extended Kalman filter is not able to estimate frequency and amplitude under all circumstances, even in the presence of process noise.

It also was hypothesized that perhaps there was too much measurement noise for the extended Kalman filter to work properly. Figure 10.11 indicates that even if we reduce the measurement noise by an order of magnitude the extended Kalman filter is still unable to estimate the positive frequency of the sinusoid when the initial filter frequency estimate is negative. In addition, from Fig. 10.12

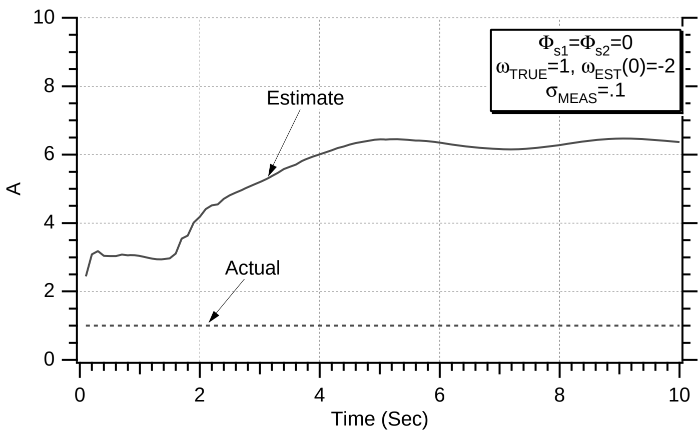

Fig. 10.12 Reducing measurement noise by an order of magnitude does not enable Kalman filter to estimate amplitude when actual frequency is positive and initial frequency estimate is negative.

we can see that the filter is also unable to estimate the amplitude under the same conditions. In other words, the filter's inability to estimate frequency and amplitude properly under a variety of initial conditions is not caused by the fact that the measurement noise is too large.

The conclusion reached is that the extended Kalman filter we have formulated in this section does not appear to be working satisfactorily if the filter is not properly initialized. Is it possible that, for this problem, the frequency of the sinusoid is unobservable?

Two-State Extended Kalman Filter with a Priori Information

So far we have shown that the extended Kalman filter of the preceding section was unable to estimate the frequency and amplitude of the sinusoidal signal unless the filter was properly initialized. With perfect hindsight this now seems reasonable because if we have a signal given by

$$x = A \sin \omega t$$

it is possible to determine if a negative value of x is a result of either a negative frequency ω or a negative amplitude A. Therefore, to gain a deeper understanding of the problem it is hypothesized that if the amplitude of the sinusoid is known in advance we should be able to estimate the frequency based on measurements of x. For the academic purpose of testing our hypothesis, we will derive another extended Kalman filter, which assumes that the amplitude of the sinusoidal signal is known precisely. Under these circumstances the model of the real world can be simplified from that of the preceding section to

$$\begin{bmatrix} \dot{\phi} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ u_s \end{bmatrix}$$

where u_s is white process noise that has been added to the derivative of frequency to account for the fact that we may not be modeling the real world perfectly. Therefore, the continuous process-noise matrix can be written by inspection of the preceding state-space equation as

$$\boldsymbol{Q} = \begin{bmatrix} 0 & 0 \\ 0 & \Phi_s \end{bmatrix}$$

where Φ_s is the spectral density of the white process noise. The systems dynamics matrix also can be obtained by inspection of the state-space equation representing the real world as

$$\boldsymbol{F} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Because $\boldsymbol{F}^2$ is zero, the fundamental matrix can be represented exactly by the two-term Taylor-series expansion

$$\Phi = \boldsymbol{I} + \boldsymbol{F}t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} t = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}$$

which means that the discrete fundamental matrix is given by

$$\Phi_k = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}$$

Although the state-space equation representing our model of the real world is linear, the measurement equation (i.e., we are actually measuring x, which is not a state) is nonlinear. Therefore, the linearized measurement equation is given by

$$\Delta x^* = \begin{bmatrix} \dfrac{\partial x}{\partial \phi} & \dfrac{\partial x}{\partial \omega} \end{bmatrix} \begin{bmatrix} \Delta\phi \\ \Delta\omega \end{bmatrix} + v$$

where v is white measurement noise. We have already shown in the preceding section that

$$x = A \sin \omega t = A \sin \phi$$

the partial derivatives of the measurement matrix can be easily evaluated as

$$\frac{\partial x}{\partial \phi} = A \cos \phi$$

$$\frac{\partial x}{\partial \omega} = 0$$

making the linearized measurement matrix

$$\boldsymbol{H} = [A \cos \phi \quad 0]$$

The measurement matrix will be evaluated at the projected estimate of ϕ. In this example, because the measurement noise is a scalar, the discrete measurement noise matrix will also be a scalar given by

$$\boldsymbol{R}_k = \sigma_x^2$$

where σ_x^2 is the variance of the measurement noise. Recall that the discrete process-noise matrix can be found from the continuous process-noise matrix according to

$$\boldsymbol{Q}_k = \int_0^{T_2} \Phi(\tau)\boldsymbol{Q}\Phi^T(\tau)\,\mathrm{d}t$$

Substitution of the appropriate matrices into the preceding expression yields

$$\boldsymbol{Q}_k = \int_0^{T_s} \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \Phi_s \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \tau & 1 \end{bmatrix} \mathrm{d}\tau$$

After multiplying out the three matrices, we get

$$\boldsymbol{Q}_k = \int_0^{T_s} \begin{bmatrix} \tau^2\Phi_s & \tau\Phi_s \\ \tau\Phi_s & \Phi_s \end{bmatrix} \mathrm{d}\tau$$

Finally, after integration we obtain for the discrete process-noise matrix

$$\boldsymbol{Q}_k = \begin{bmatrix} \dfrac{\Phi_s T_s^3}{3} & \dfrac{\Phi_s T_s^2}{2} \\ \dfrac{\Phi_s T_s^2}{2} & \Phi_s T_s \end{bmatrix}$$

We now have enough information to solve the matrix Riccati equations for the Kalman gains.

Because the fundamental matrix is also exact in this application, we can also use it to propagate the state estimates in the extended Kalman filter over the sampling interval. Using the fundamental matrix, we can propagate the states from time $k-1$ to time k or in matrix form

$$\begin{bmatrix} \bar{\phi}_k \\ \bar{\omega}_k \end{bmatrix} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\phi}_{k-1} \\ \hat{\omega}_{k-1} \end{bmatrix}$$

We can multiply out the preceding matrix equation to get the two scalar equations

$$\bar{\phi}_k = \hat{\phi}_{k-1} + \hat{\omega}_{k-1} T_s$$
$$\bar{\omega}_k = \hat{\omega}_{k-1}$$

Again, as was the case in the preceding section, because the measurement matrix is linearized we do not have to use it in the calculation of the residual for the actual extended Kalman-filtering equations. Instead, we can do better by using the actual nonlinear equation for the residual or

$$\mathrm{Res}_k = x_k^* - A \sin \bar{\phi}_k$$

In the preceding equation the amplitude A is not estimated but is assumed to be known a prior. The extended Kalman-filtering equations can now be written as

$$\hat{\phi}_k = \bar{\phi}_k + K_{1_k} \mathrm{Res}_k$$
$$\hat{\omega}_k = \bar{\omega}_k + K_{2_k} \mathrm{Res}_k$$

where the barred quantities represent projected states that have been already defined.

Listing 10.2 presents the two-state extended Kalman filter for estimating the states of a noisy sinusoidal signal whose frequency is unknown but whose amplitude is known. As was the case with Listing 10.1, we can see that the simulation is initially set up to run without process noise (i.e., PHIS = 0). The actual sinusoidal signal still has unity amplitude, and the standard deviation of the measurement noise is also unity. We can see from Listing 10.2 that the frequency of the actual sinusoidal signal is also unity. The filter's initial state estimate of ϕ is set to zero because time is initially zero (i.e., $\phi = \omega t$), while the initial estimate of the frequency is set to two (rather than unity). Values are used for the initial covariance matrix to reflect the uncertainties in our initial state estimates. Again, because the fundamental matrix is exact, a special subroutine is not required in this simulation to integrate the state equations over the sampling interval in order to obtain the state projections.

Listing 10.2 Two-state extended Kalman filter with a priori information for estimating frequency of sinusoidal signal

```
C THE FIRST THREE STATEMENTS INVOKE THE ABSOFT RANDOM
  NUMBER GENERATOR ON THE MACINTOSH
        GLOBAL DEFINE
              INCLUDE 'quickdraw.inc'
        END
        IMPLICIT REAL*8(A-H,O-Z)
        REAL*8 P(2,2),Q(2,2),M(2,2),PHI(2,2),HMAT(1,2),HT(2,1),PHIT(2,2)
        REAL*8 RMAT(1,1),IDN(2,2),PHIP(2,2),PHIPPHIT(2,2),HM(1,2)
        REAL*8 HMHT(1,1),HMHTR(1,1),HMHTRINV(1,1),MHT(2,1),K(2,1)
        REAL*8 KH(2,2),IKH(2,2)
        INTEGER ORDER
        TS=.1
        A=1.
        W=1.
        PHIS=0.
        SIGX=1.
        ORDER=2
        OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
        OPEN(2,STATUS='UNKNOWN',FILE='COVFIL')
        T=0.
        S=0.
        H=.001
        DO 14 I=1,ORDER
        DO 14 J=1,ORDER
        PHI(I,J)=0.
        P(I,J)=0.
        Q(I,J)=0.
        IDN(I,J)=0.
 14     CONTINUE
        RMAT(1,1)=SIGX**2
```

(*continued*)

Listing 10.2 (*Continued*)

```
      IDN(1,1)=1.
      IDN(2,2)=1.
      PHIH=0.
      WH=2.
      P(1,1)=0.**2
      P(2,2)=(W-WH)**2
      XT=0.
      XTD=A*W
      WHILE(T<=20.)
            XTOLD=XT
            XTDOLD=XTD
            XTDD=-W*W*XT
            XT=XT+H*XTD
            XTD=XTD+H*XTDD
            T=T+H
            XTDD=-W*W*XT
            XT=.5*(XTOLD+XT+H*XTD)
            XTD=.5*(XTDOLD+XTD+H*XTDD)
            S=S+H
            IF(S>=(TS-.00001))THEN
                  S=0.
                  PHI(1,1)=1.
                  PHI(1,2)=TS
                  PHI(2,2)=1.
                  Q(1,1)=TS*TS*TS*PHIS/3.
                  Q(1,2)=.5*TS*TS*PHIS
                  Q(2,1)=Q(1,2)
                  Q(2,2)=PHIS*TS
                  PHIB=PHIH+WH*TS
                  HMAT(1,1)=COS(PHIB)
                  HMAT(1,2)=0.
                  CALL MATTRN(PHI,ORDER,ORDER,PHIT)
                  CALL MATTRN(HMAT,1,ORDER,HT)
                  CALL MATMUL(PHI,ORDER,ORDER,P,ORDER,
                    ORDER,PHIP)
                  CALL MATMUL(PHIP,ORDER,ORDER,PHIT,ORDER,
     1              ORDER,PHIPPHIT)
                  CALL MATADD(PHIPPHIT,ORDER,ORDER,Q,M)
                  CALL MATMUL(HMAT,1,ORDER,M,ORDER,ORDER,
                    HM)
                  CALL MATMUL(HM,1,ORDER,HT,ORDER,1,HMHT)
                  CALL MATADD(HMHT,ORDER,ORDER,RMAT,
                    HMHTR)
                  HMHTRINV(1,1)=1./HMHTR(1,1)
                  CALL MATMUL(M,ORDER,ORDER,HT,ORDER,1,
                    MHT)
                  CALL MATMUL(MHT,ORDER,1,HMHTRINV,1,1,K)
                  CALL MATMUL(K,ORDER,1,HMAT,1,ORDER,KH)
                  CALL MATSUB(IDN,ORDER,ORDER,KH,IKH)
                  CALL MATMUL(IKH,ORDER,ORDER,M,ORDER,
```

(*continued*)

Listing 10.2 (*Continued*)

```
                 ORDER,P)
                CALL GAUSS(XTNOISE,SIGX)
                XTMEAS=XT+XTNOISE
                RES=XTMEAS-A*SIN(PHIB)
                PHIH=PHIB+K(1,1)*RES
                WH=WH+K(2,1)*RES
                PHIREAL=W*T
                ERRPHI=PHIREAL-PHIH
                SP11=SQRT(P(1,1))
                ERRW=W-WH
                SP22=SQRT(P(2,2))
                XTH=A*SIN(PHIH)
                XTDH=A*WH*COS(PHIH)
                WRITE(9,*)T,XT,XTH,XTD,XTDH,W,WH,PHI,PHIH
                WRITE(1,*)T,XT,XTH,XTD,XTDH,W,WH,PHI,PHIH
                WRITE(2,*)T,ERRPHI,SP11,-SP11,ERRW,SP22,-SP22
            ENDIF
        END DO
        PAUSE
        CLOSE(1)
        CLOSE(2)
        END

C SUBROUTINE GAUSS IS SHOWN IN LISTING 1.8
C SUBROUTINE MATTRN IS SHOWN IN LISTING 1.3
C SUBROUTINE MATMUL IS SHOWN IN LISTING 1.4
C SUBROUTINE MATADD IS SHOWN IN LISTING 1.1
C SUBROUTINE MATSUB IS SHOWN IN LISTING 1.2
```

The nominal case of Listing 10.2 was run, and we can see from Fig. 10.13 that the new two-state extended Kalman filter appears to be working because it is able to estimate the positive frequency when the filter frequency estimate is initialized positive. However, under these idealized initialization conditions the preceding extended Kalman filter also had similar performance (see Fig 10.1).

Another case was run in which the actual frequency was negative and the initial frequency estimate was positive. We can see from Fig. 10.14 that now the new two-state extended Kalman filter is able to estimate the negative frequency quite well under these circumstances, whereas the preceding three-state extended Kalman filter was unable to estimate the negative frequency under similar circumstances (see Fig. 10.3). Thus, it appears that when the amplitude of the sinusoid is known in advance it is possible to estimate the frequency of the sinusoid.

Figures 10.15 and 10.16 complete further experiments in which we are attempting to estimate the frequency for various filter frequency initialization conditions. As expected, Fig. 10.15 shows that a negative frequency can be easily estimated when the initial filter frequency estimate is negative. This is not

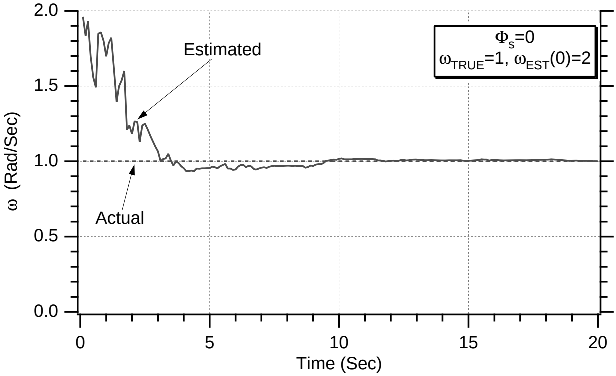

Fig. 10.13 Two-state extended Kalman filter estimates positive frequency when initial frequency estimate is also positive.

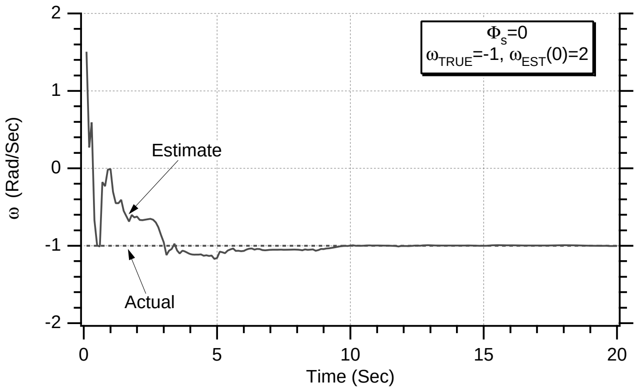

Fig. 10.14 New two-state extended Kalman filter estimates negative frequency when initialized positive.

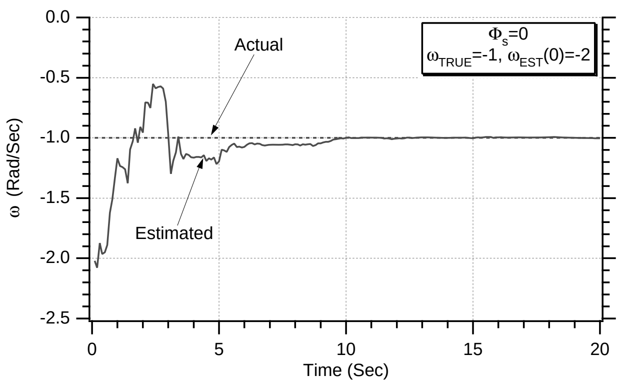

Fig. 10.15 New two-state extended Kalman filter estimates negative frequency when initialized negative.

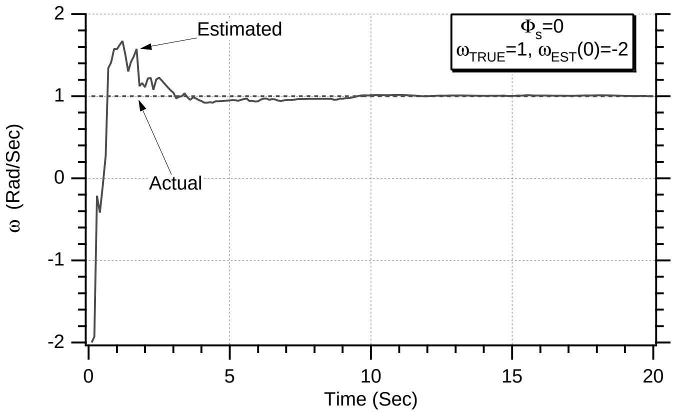

Fig. 10.16 New two-state extended Kalman filter estimates positive frequency when initialized negative.

surprising because the preceding three-state extended Kalman filter also could have accomplished this task. However, Fig. 10.16 shows that a positive frequency also can be estimated when the initial filter frequency estimate is negative. The preceding three-state extended Kalman filter would have failed at this attempt.

To see if the new two-state extended Kalman filter was really working all of the time, even more runs were made. Listing 10.2 was modified slightly so that it could be run in the Monte Carlo mode. Code was changed so that 10 runs could be made, one after another, each with a different noise stream. The first case that was examined in detail was the one in which the actual frequency of the sinusoid was negative and the filter's initial estimate of the frequency was positive. The single-run simulation results of Fig. 10.14 indicated that the filter was working. However, we can see from Fig. 10.17 that one case out of 10 failed to estimate the

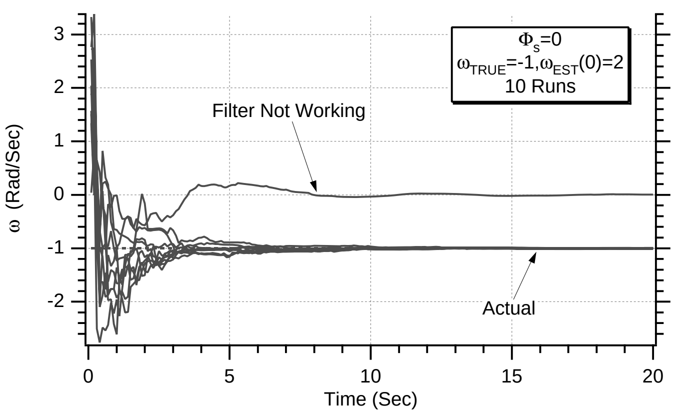

Fig. 10.17 New two-state extended Kalman filter does not estimate negative frequency all of the time when filter is initialized positive.

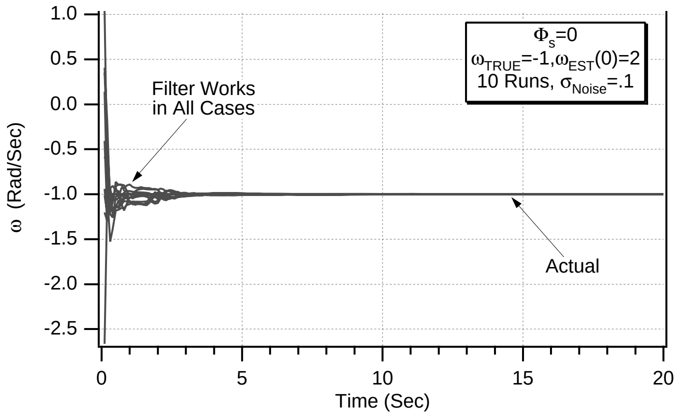

Fig. 10.18 New two-state extended Kalman filter works all of the time when the measurement noise is reduced by an order of magnitude.

negative frequency. In that particular case the filter thought the frequency was zero.

Because the filter should work all of the time, it was hypothesized that perhaps there was too much measurement noise. The standard deviation of the measurement noise was unity, and the amplitude of the sinusoidal signal was also unity. Figure 10.18 shows that when the measurement noise is reduced by an order of magnitude to 0.1 the filter is now successful in estimating the negative frequency in all 10 cases.

In the preceding figure the frequency estimates looked excellent. However, the real performance of the two-state extended Kalman filter is determined by the error in the estimate of frequency. Figure 10.19 displays the error in the estimate

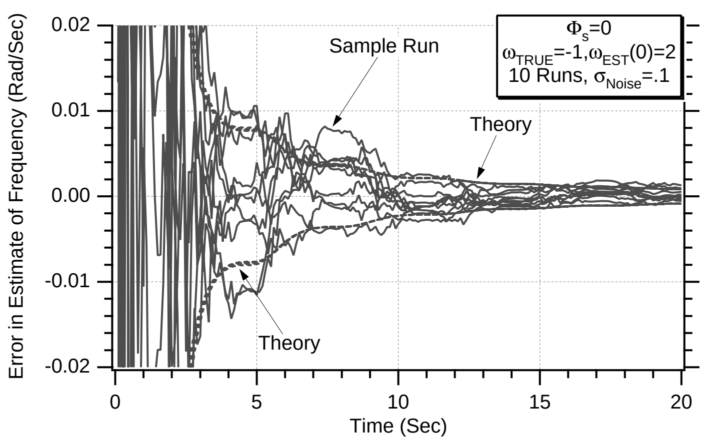

Fig. 10.19 Error in the estimate results indicate that the new two-state extended Kalman filter is able to estimate frequency.

of frequency for the case in which the actual frequency is negative, but the filter's initial estimate of frequency is positive when the measurement noise is 0.1. The theoretical error bounds for the error in the estimate of frequency, obtained by taking the square root of the second diagonal element of the covariance matrix, are also displayed in Fig. 10.19. We can see that for all 10 runs the simulated error in the estimate of frequency lies within the theoretical bounds most of the time, indicating that the new two-state extended Kalman filter is working properly.

The two-state extended Kalman filter derived in this section was an academic exercise in explaining the failure of the three-state extended Kalman filter of the preceding section. We still want to find out if an extended Kalman filter can be designed to estimate the signal states and frequency under poor initialization conditions.

Alternate Extended Kalman Filter for Sinusoidal Signal

In this section we will try again to build an extended Kalman filter that makes use of the fact that the signal is sinusoidal but also takes into account that both the frequency and amplitude of the sinusoid are also unknown.

Recall that the actual signal is given by

$$x = A \sin \omega t$$

As was just mentioned, in this model we will assume that both the amplitude A and the frequency ω of the sinusoid are unknown. If we take the derivative of the preceding equation, we also get a sinusoid or

$$\dot{x} = A\omega \cos \omega t$$

Taking the derivative again yields

$$\ddot{x} = -A\omega^2 \sin \omega t$$

Thus, we can see that the preceding second-order differential equation can be expressed in terms of the first equation or

$$\ddot{x} = -\omega^2 x$$

In other words, integrating the preceding equation twice, with the appropriate initial conditions, will yield the original sinusoidal signal. The preceding equation is especially useful because the sinusoidal term and signal amplitude have been eliminated and the second derivative of the signal or third state has been expressed in terms of the signal or first state. This is precisely what we desire for expressing the problem in state-space notation.

If we assume that the sinusoidal frequency is an unknown constant, its derivative must be zero. To account for the fact that the sinusoidal frequency may not be a constant, it is prudent to add process noise to the derivative of frequency. Therefore, the two differential equations representing the real world in this example are

$$\ddot{x} = -\omega^2 x$$
$$\dot{\omega} = u_s$$

where u_s is white process noise with spectral density Φ_s. The preceding differential equations can be expressed as three first-order differential equations (i.e., one of the equations is redundant) in state-space form as

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -\omega^2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ u_s \end{bmatrix}$$

The preceding equation is nonlinear because a state also shows up in the 3×3 matrix multiplying the state vector. From the preceding equation we can see that the systems dynamics matrix turns out to be

$$\boldsymbol{F} = \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{bmatrix} \dfrac{\partial \dot{x}}{\partial x} & \dfrac{\partial \dot{x}}{\partial \dot{x}} & \dfrac{\partial \dot{x}}{\partial \omega} \\ \dfrac{\partial \ddot{x}}{\partial x} & \dfrac{\partial \ddot{x}}{\partial \dot{x}} & \dfrac{\partial \ddot{x}}{\partial \omega} \\ \dfrac{\partial \dot{\omega}}{\partial x} & \dfrac{\partial \dot{\omega}}{\partial \dot{x}} & \dfrac{\partial \dot{\omega}}{\partial \omega} \end{bmatrix}$$

where the partial derivatives are evaluated at the current estimates. Taking the partial derivatives in this example can be done by inspection, and the resultant systems dynamics matrix is given by

$$\boldsymbol{F} = \begin{bmatrix} 0 & 1 & 0 \\ -\hat{\omega}^2 & 0 & -2\hat{\omega}\hat{x} \\ 0 & 0 & 0 \end{bmatrix}$$

where the terms in the systems dynamics matrix have been evaluated at the current state estimates. In this example the exact fundamental matrix will be difficult, if not impossible, to find. If we assume that the elements of the systems dynamics matrix are approximately constant between sampling instants, then we use a two-term Taylor-series approximation for the fundamental matrix, yielding

$$\Phi(t) \approx I + \boldsymbol{F}t = \begin{bmatrix} 1 & t & 0 \\ -\hat{\omega}^2 t & 1 & -2\hat{\omega}\hat{x}t \\ 0 & 0 & 1 \end{bmatrix}$$

Therefore, the discrete fundamental matrix can be obtained by substituting T_s for t or

$$\Phi_k \approx \begin{bmatrix} 1 & T_s & 0 \\ -\hat{\omega}_{k-1}^2 T_s & 1 & -2\hat{\omega}_{k-1}\hat{x}_{k-1}T_s \\ 0 & 0 & 1 \end{bmatrix}$$

The continuous process-noise matrix can by found from the original state-space equation to be

$$\boldsymbol{Q} = E(ww^T) = E\left[\begin{bmatrix} 0 \\ 0 \\ u_s \end{bmatrix}[0 \quad 0 \quad u_s]\right] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Phi_s \end{bmatrix}$$

Recall that the discrete process-noise matrix can be found from the continuous process-noise matrix according to

$$\boldsymbol{Q}_k = \int_0^{T_s} \Phi(\tau)\boldsymbol{Q}\Phi^T(\tau)\,\mathrm{d}\boldsymbol{t}$$

By substitution of the appropriate matrices into the preceding equation, we obtain

$$\boldsymbol{Q}_k = \int_0^{T_s} \begin{bmatrix} 1 & \tau & 0 \\ -\hat{\omega}^2\tau & 1 & -2\hat{\omega}\hat{x}\tau \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Phi_s \end{bmatrix}\begin{bmatrix} 1 & -\hat{\omega}^2\tau & 0 \\ \tau & 1 & 0 \\ 0 & -2\hat{\omega}\hat{x}\tau & 1 \end{bmatrix}\mathrm{d}\tau$$

Multiplying out the three matrices yields

$$\boldsymbol{Q}_k = \int_0^{T_s} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4\hat{\omega}^2\hat{x}^2\tau^2\Phi_s & -2\hat{\omega}\hat{x}\tau\Phi_s \\ 0 & -2\hat{\omega}\hat{x}\tau\Phi_s & \Phi_s \end{bmatrix}\mathrm{d}\tau$$

If we again assume that the states are approximately constant between the sampling intervals, then the preceding integral can easily be evaluated as

$$\boldsymbol{Q}_k = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.333\hat{\omega}^2\hat{x}^2T_s^3\Phi_s & -\hat{\omega}\hat{x}T_s^2\Phi_s \\ 0 & -\hat{\omega}\hat{x}T_s^2\Phi_s & T_s\Phi_s \end{bmatrix}$$

In this problem we are assuming that the measurement is of the first state plus noise or

$$x_k^* = x_k + v_k$$

Therefore, the measurement is linearly related to the states according to

$$x_k^* = [1 \quad 0 \quad 0]\begin{bmatrix} x \\ \dot{x} \\ \omega \end{bmatrix} + v_k$$

The measurement matrix can be obtained from the preceding equation by inspection as

$$\boldsymbol{H} = [1 \quad 0 \quad 0]$$

In this example the discrete measurement noise matrix is a scalar and is given by

$$\boldsymbol{R}_k = E(\boldsymbol{v}_k \boldsymbol{v}_k^T) = \sigma_x^2$$

We now have enough information to solve the matrix Riccati equations for the Kalman gains.

For this example the projected states in the actual extended Kalman-filtering equations do not have to use the approximation for the fundamental matrix. Instead the state projections, indicated by an overbar, can be obtained by numerically integrating the nonlinear differential equations over the sampling interval. Therefore, the extended Kalman-filtering equations can then be written as

$$\hat{x}_k = \bar{x}_k + K_{1_k}(x_k^* - \bar{x}_k)$$
$$\hat{\dot{x}}_k = \bar{\dot{x}}_k + K_{2_k}(x_k^* - \bar{x}_k)$$
$$\hat{\omega}_k = \hat{\omega}_{k-1} + K_{3_k}(x_k^* - \bar{x}_k)$$

Listing 10.3 presents the alternate three-state extended Kalman filter for estimating the states of a noisy sinusoidal signal whose frequency is unknown. We can see once again that the simulation is initially set up to run without process noise (i.e., PHIS = 0). The actual sinusoidal signal has unity amplitude, and the standard deviation of the measurement noise is also unity. We can see from Listing 10.3 that the frequency of the actual sinusoidal signal is also unity. The filter's initial state estimates of x and $\dot{x}$ are set to zero, while the initial estimate of the frequency is set to two (rather than unity). In other words, the second and third states are mismatched from the real world. Values are used for the initial covariance matrix to reflect the uncertainties in our initial state estimates. As was the case in Chapter 9, subroutine PROJECT is used to integrate the nonlinear equations using Euler integration over the sampling interval to obtain the state projections for x and $\dot{x}$.

A case was run using the nominal values of Listing 10.3 (i.e., no process noise), and we can see from Figs. 10.20–10.22 that the actual quantities and their estimates are quite close. An additional comparison of the results of our alternate three-state extended Kalman filter with the results of the first- and second-order polynomial Kalman filters of Chapter 5 indicates that we are estimating the states more accurately than those imperfect linear filters. However, our state estimates

Listing 10.3 Three-state alternate extended Kalman filter for sinusoidal signal with unknown frequency

```
C THE FIRST THREE STATEMENTS INVOKE THE ABSOFT RANDOM
  NUMBER GENERATOR ON THE MACINTOSH
      GLOBAL DEFINE
            INCLUDE 'quickdraw.inc'
      END
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 P(3,3),Q(3,3),M(3,3),PHI(3,3),HMAT(1,3),HT(3,1),PHIT(3,3)
      REAL*8 RMAT(1,1),IDN(3,3),PHIP(3,3),PHIPPHIT(3,3),HM(1,3)
      REAL*8 HMHT(1,1),HMHTR(1,1),HMHTRINV(1,1),MHT(3,1),K(3,1),
     1  F(3,3)
      REAL*8 KH(3,3),IKH(3,3)
      INTEGER ORDER
      HP=.001
      W=1.
      A=1.
      TS=.1
      ORDER=3
      PHIS=0.
      SIGX=1.
      OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
      OPEN(2,STATUS='UNKNOWN',FILE='COVFIL')
      T=0.
      S=0.
      H=.001
      DO 14 I=1,ORDER
      DO 14 J=1,ORDER
      F(I,J)=0.
      PHI(I,J)=0.
      P(I,J)=0.
      Q(I,J)=0.
      IDN(I,J)=0.
 14   CONTINUE
      RMAT(1,1)=SIGX**2
      IDN(1,1)=1.
      IDN(2,2)=1.
      IDN(3,3)=1.
      P(1,1)=SIGX**2
      P(2,2)=2.**2
      P(3,3)=2.**2
      XTH=0.
      XTDH=0.
      WH=2.
      XT=0.
      XTD=A*W
      WHILE(T<=20.)
           XTOLD=XT
           XTDOLD=XTD
```

(*continued*)

Listing 10.3 (*Continued*)

```
            XTDD=-W*W*XT
            XT=XT+H*XTD
            XTD=XTD+H*XTDD
            T=T+H
            XTDD=-W*W*XT
            XT=.5*(XTOLD+XT+H*XTD)
            XTD=.5*(XTDOLD+XTD+H*XTDD)
            S=S+H
            IF(S>=(TS-.00001))THEN
                  S=0.
                  F(1,2)=1.
                  F(2,1)=-WH**2
                  F(2,3)=-2.*WH*XTH
                  PHI(1,1)=1.
                  PHI(1,2)=TS
                  PHI(2,1)=-WH*WH*TS
                  PHI(2,2)=1.
                  PHI(2,3)=-2.*WH*XTH*TS
                  PHI(3,3)=1.
                  Q(2,2)=4.*WH*WH*XTH*XTH*TS*TS*TS*PHIS/3.
                  Q(2,3)=-2.*WH*XTH*TS*TS*PHIS/2.
                  Q(3,2)=Q(2,3)
                  Q(3,3)=PHIS*TS
                  HMAT(1,1)=1.
                  HMAT(1,2)=0.
                  HMAT(1,3)=0.
                  CALL  MATTRN(PHI,ORDER,ORDER,PHIT)
                  CALL  MATTRN(HMAT,1,ORDER,HT)
                  CALL  MATMUL(PHI,ORDER,ORDER,P,ORDER,
                    ORDER,PHIP)
                  CALL  MATMUL(PHIP,ORDER,ORDER,PHIT,ORDER,
 1                  ORDER,PHIPPHIT)
                  CALL  MATADD(PHIPPHIT,ORDER,ORDER,Q,M)
                  CALL  MATMUL(HMAT,1,ORDER,M,ORDER,ORDER,
                    HM)
                  CALL  MATMUL(HM,1,ORDER,HT,ORDER,1,HMHT)
                  CALL  MATADD(HMHT,ORDER,ORDER,RMAT,
                    HMHTR)
                  HMHTRINV(1,1)=1./HMHTR(1,1)
                  CALL  MATMUL(M,ORDER,ORDER,HT,ORDER,1,
                    MHT)
                  CALL  MATMUL(MHT,ORDER,1,HMHTRINV,1,1,K)
                  CALL  MATMUL(K,ORDER,1,HMAT,1,ORDER,KH)
                  CALL  MATSUB(IDN,ORDER,ORDER,KH,IKH)
                  CALL  MATMUL(IKH,ORDER,ORDER,M,ORDER,
                    ORDER,P)
                  CALL  GAUSS(XTNOISE,SIGX)
                  XTMEAS=XT+XTNOISE
                  CALL  PROJECT(T,TS,XTH,XTDH,XTB,XTDB,HP,WH)
                  RES=XTMEAS-XTB
```

(*continued*)

Listing 10.3 (*Continued*)

```
                    XTH=XTB+K(1,1)*RES
                    XTDH=XTDB+K(2,1)*RES
                    WH=WH+K(3,1)*RES
                    ERRX=XT-XTH
                    SP11=SQRT(P(1,1))
                    ERRXD=XTD-XTDH
                    SP22=SQRT(P(2,2))
                    ERRW=W-WH
                    SP33=SQRT(P(3,3))
                    WRITE(9,*)T,XT,XTH,XTD,XTDH,W,WH
                    WRITE(1,*)T,XT,XTH,XTD,XTDH,W,WH
                    WRITE(2,*)T,ERRX,SP11,-SP11,ERRXD,SP22,-SP22,
   1                  ERRW,SP33,-SP33
                ENDIF
        END DO
        PAUSE
        CLOSE(1)
        CLOSE(2)
        END

        SUBROUTINE PROJECT(TP,TS,XTP,XTDP,XTH,XTDH,HP,W)
        IMPLICIT REAL*8 (A-H)
        IMPLICIT REAL*8 (O-Z)
        T=0.
        XT=XTP
        XTD=XTDP
        H=HP
        WHILE(T<=(TS-.0001))
            XTDD=-W*W*XT
            XTD=XTD+H*XTDD
            XT=XT+H*XTD
            T=T+H
        END DO
        XTH=XT
        XTDH=XTD
        RETURN
        END

C  SUBROUTINE GAUSS IS SHOWN IN LISTING 1.8
C  SUBROUTINE MATTRN IS SHOWN IN LISTING 1.3
C  SUBROUTINE MATMUL IS SHOWN IN LISTING 1.4
C  SUBROUTINE MATADD IS SHOWN IN LISTING 1.1
C  SUBROUTINE MATSUB IS SHOWN IN LISTING 1.2
```

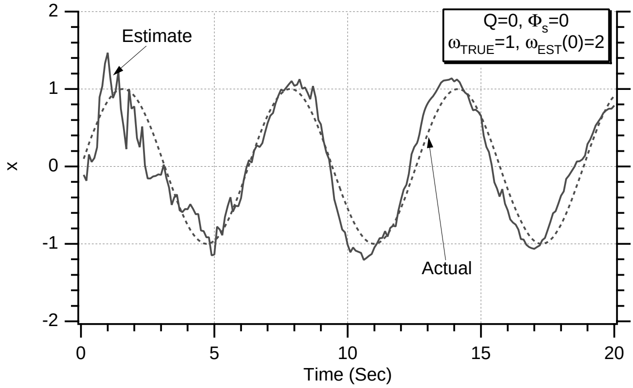

Fig. 10.20 Alternate three-state extended Kalman filter estimates first state well.

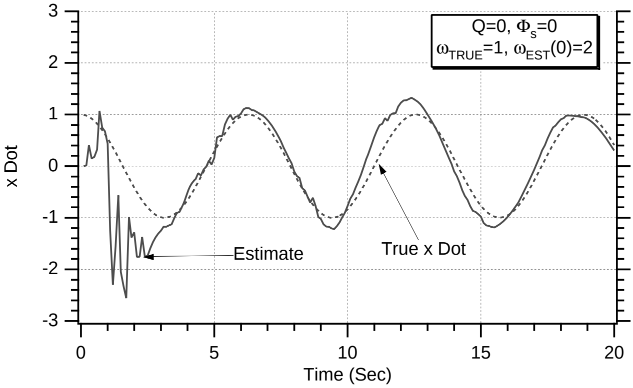

Fig. 10.21 Alternate three-state extended Kalman filter estimates second state well, even though filter is not initialized correctly.

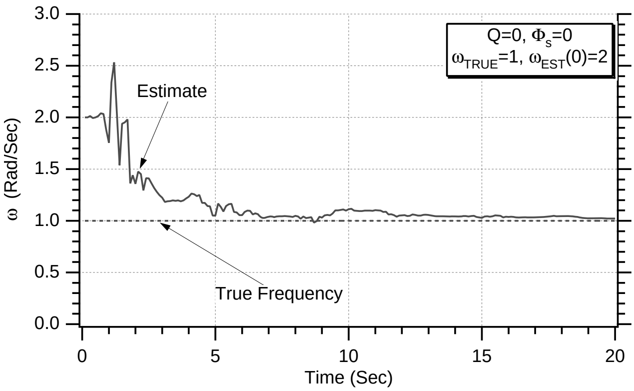

Fig. 10.22 Alternate three-state extended Kalman filter appears able to estimate the frequency of the sinusoid even though filter is not initialized correctly.

are not quite as good as those obtained from the linear sinusoidal Kalman filter of Chapter 5 (i.e., the one in which the fundamental matrix is exact) in which the frequency is known a priori. However, we are estimating the states far more accurately than the linear sinusoidal Kalman filter of Chapter 5, when the frequency is in error. From Fig. 10.22 we can see that it takes approximately 5 s to obtain an accurate estimate of the frequency for the parameters considered.

The fact that the new alternate three-state extended Kalman filter appears to be working under the benign initialization condition means that the filter is now ready for further testing. We will try to make the filter fail, as was done in the preceding two sections of this chapter. We will now consider the more difficult condition where the actual frequency is negative but the initial filter frequency estimate is positive. At first glance Fig. 10.23 appears to be telling us that the new filter also has failed in estimating the frequency. However, a closer examination of Fig. 10.23 indicates that the magnitude of the frequency has been estimated correctly, but the sign has not been estimated correctly.

To see if it is of any practical utility to only be able to estimate the magnitude of the frequency, more information is displayed in Figs. 10.24 and 10.25. Figure 10.24 compares the actual and estimated signal x, whereas Fig. 10.25 compares the actual and estimated derivative of the signal or $\dot{x}$. We can see that in both figures the estimates and actual quantities are very close even though the sign of the estimated frequency magnitude is not correct, indicating that the filter is working properly.

Another experiment was conducted in which the actual frequency of the signal was minus one, but the initial estimate of the frequency was minus two. Under these conditions we do not expect any problems, and Fig. 10.26 confirms that the alternate three-state extended Kalman filter estimates the frequency quite well.

Another experiment was conducted in which the actual frequency of the signal was plus one, but the initial estimate of the frequency was minus two. Under these conditions we expect problems, and Fig. 10.27 indicates that the alternate three-

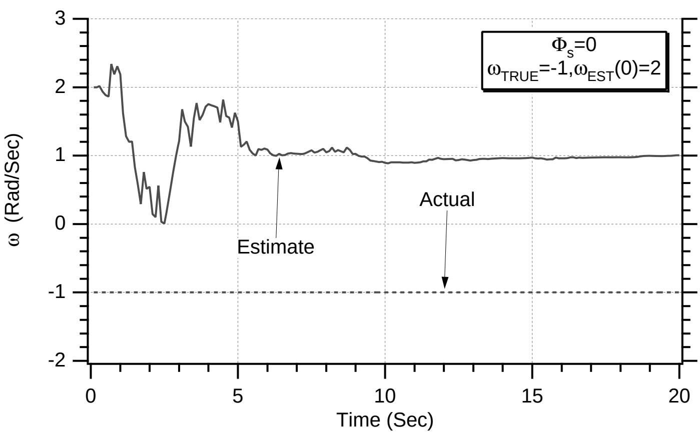

Fig. 10.23 Alternate three-state extended Kalman filter appears able to estimate the magnitude of the frequency but not its sign when the filter is not initialized correctly.

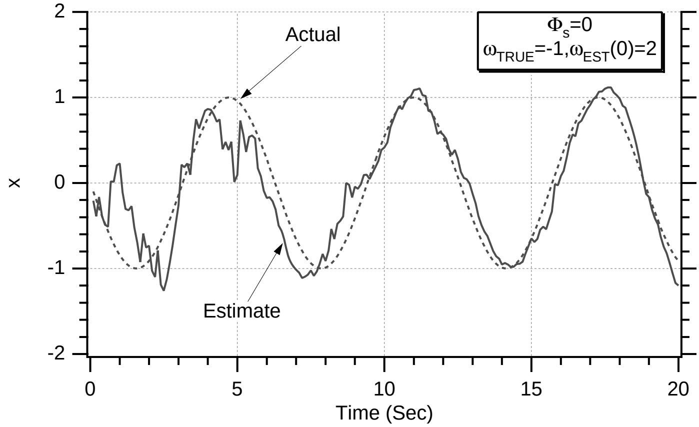

Fig. 10.24 Alternate three-state extended Kalman filter appears able to estimate the signal when the filter is not initialized correctly.

state extended Kalman filter estimates the magnitude of the frequency quite well but estimates the wrong sign.

To again see if it is of any practical utility to be able to estimate the magnitude of the frequency, more information is displayed in Figs. 10.28 and 10.29. Figure 10.28 compares the actual and estimated signal x, whereas Fig. 10.29 compares the actual and estimated derivative of the signal or $\dot{x}$. We can see that in both figures the estimates and actual quantities are very close, indicating that the filter is working properly even though we are not able to estimate the sign of the frequency.

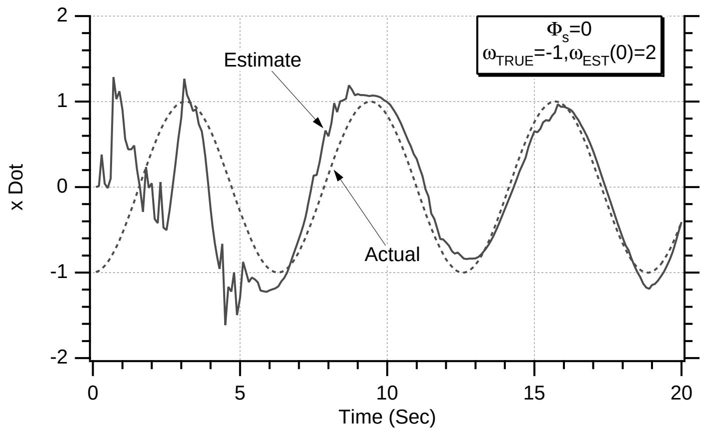

Fig. 10.25 Alternate three-state extended Kalman filter appears able to estimate the derivative of the signal when filter is not initialized correctly.

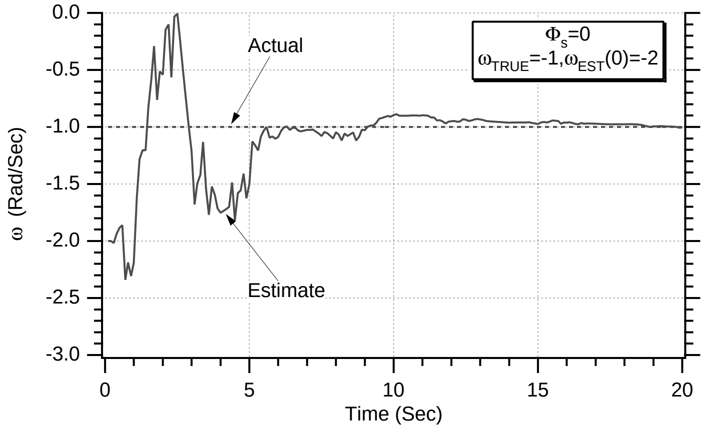

Fig. 10.26 Alternate three-state extended Kalman filter appears able to estimate the frequency correctly when frequency and initial estimate are both negative.

Figures 10.30–10.32 display the theoretical errors in the state estimates (i.e., obtained by taking the square root of the appropriate diagonal element of the covariance matrix) and the single-run simulation errors for each of the three states. The single-run errors in the estimates appear to be within the theoretical bounds approximately 68% of the time, indicating that the extended Kalman filter seems to be behaving properly. Because this example has zero process noise, the errors in the estimates diminish as more measurements are taken.

To verify that the filter was actually working all of the time under a variety of initialization conditions, Listing 10.3 was slightly modified to give it Monte Carlo

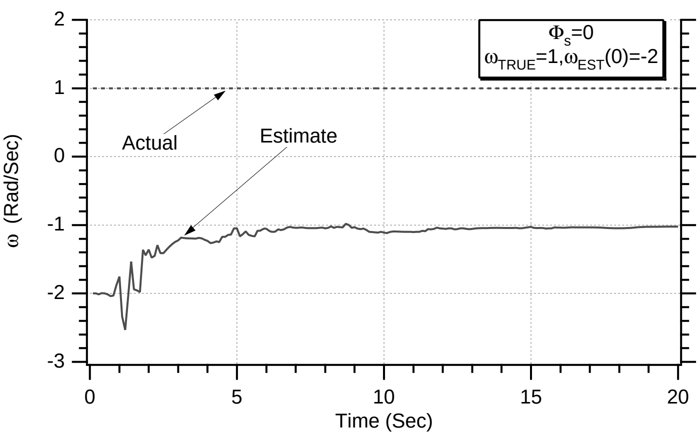

Fig. 10.27 Alternate three-state extended Kalman filter appears able to estimate the magnitude of the frequency but not its sign when the filter is not initialized correctly.

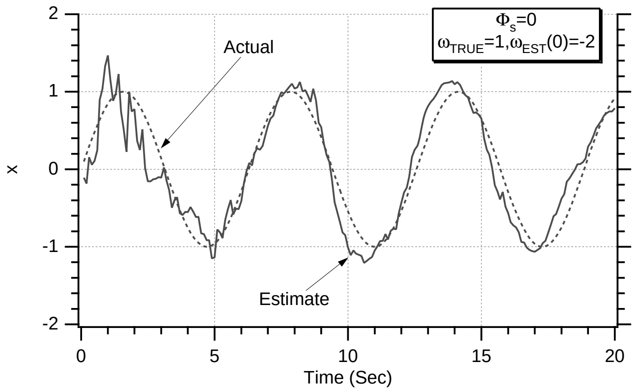

Fig. 10.28 Alternate three-state extended Kalman filter appears able to estimate the signal when the filter is not initialized correctly.

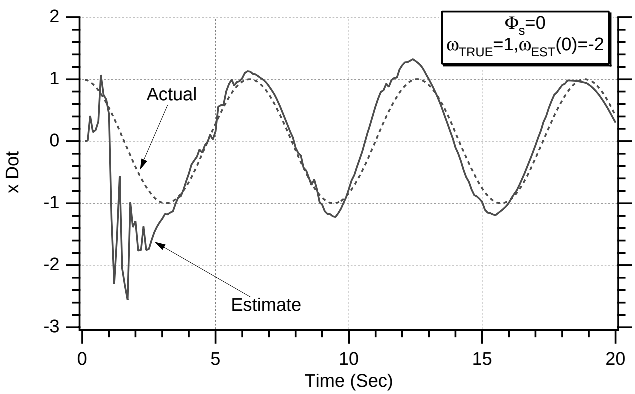

Fig. 10.29 Alternate three-state extended Kalman filter appears able to estimate the derivative of the signal when the filter is not initialized correctly.

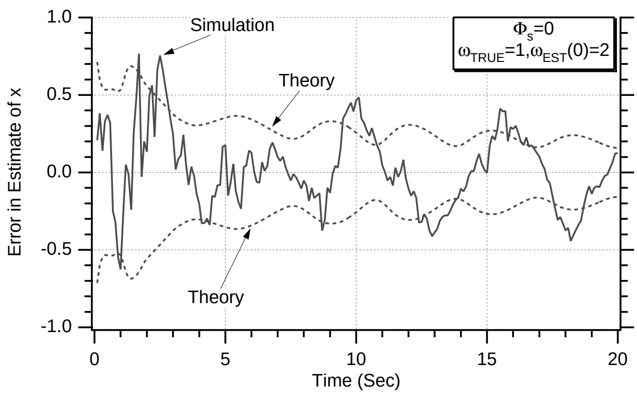

Fig. 10.30 Error in the estimate of first state agrees with theory.

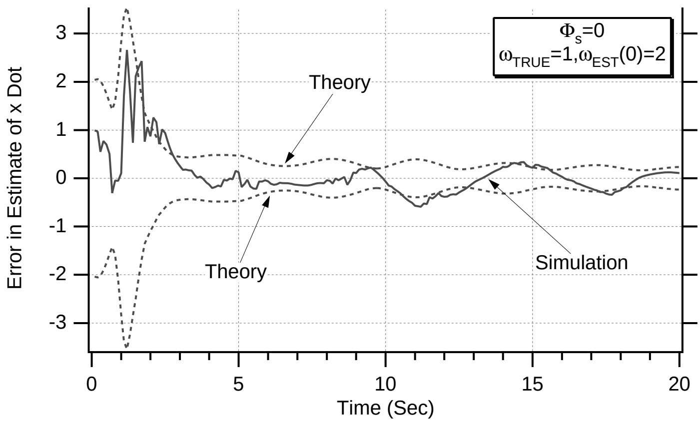

Fig. 10.31 Error in the estimate of second state agrees with theory.

capabilities. Ten-run Monte Carlo sets were run with the alternate three-state extended Kalman filter for the four different frequency and estimated frequency initialization conditions. The standard deviation of the measurement noise for these experiments was unity. The results, which are presented in Figs. 10.33–10.36, show that the correct magnitude of the frequency is always estimated accurately, regardless of initialization. We have already seen that when the correct frequency magnitude is estimated the other states also will be accurate. Thus, we can conclude that the alternate three-state extended Kalman filter is effective in estimating the states of a sinusoid.

We have seen that the alternate three-state extended Kalman filter is only able to estimate both the magnitude and sign of the frequency exactly when the initial

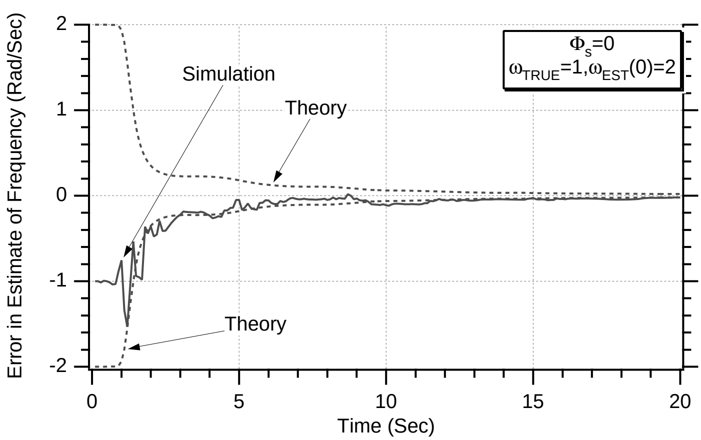

Fig. 10.32 Error in the estimate of third state agrees with theory.

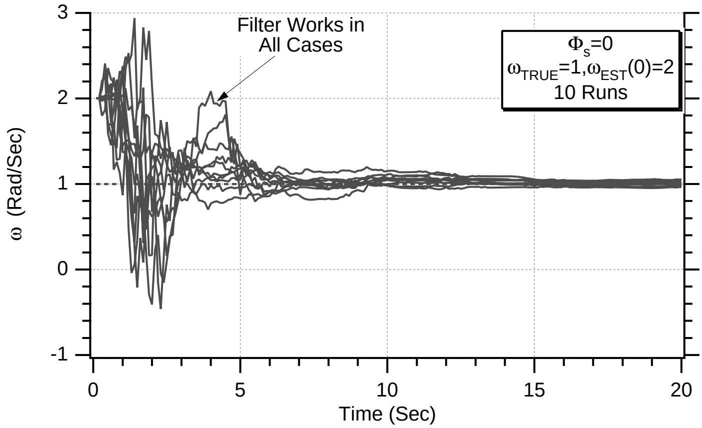

Fig. 10.33 Alternate three-state extended Kalman filter estimates correct frequency when initial frequency estimate is of correct sign.

frequency estimate of the filter is of the same sign as the actual frequency. When this condition is not met, the filter is only able to estimate the correct magnitude of the frequency but not the sign. It is hypothesized that the reason for the filter's lack of ability in always distinguishing between positive and negative frequencies is caused by the formulation of the state-space equations where an ω^2 term appears. Apparently, we cannot distinguish between positive and negative frequencies with only the ω^2 term (i.e., there is no ω term). However, if we are only interested in obtaining estimates of x and $\dot{x}$, knowing the magnitude of the frequency is sufficient (i.e., the sign of the frequency is not important).

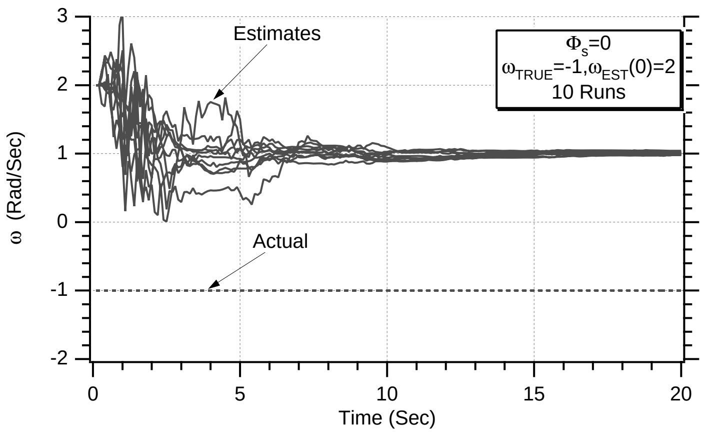

Fig. 10.34 Alternate three-state extended Kalman filter estimates correct frequency magnitude when initial frequency estimate is of the wrong sign.

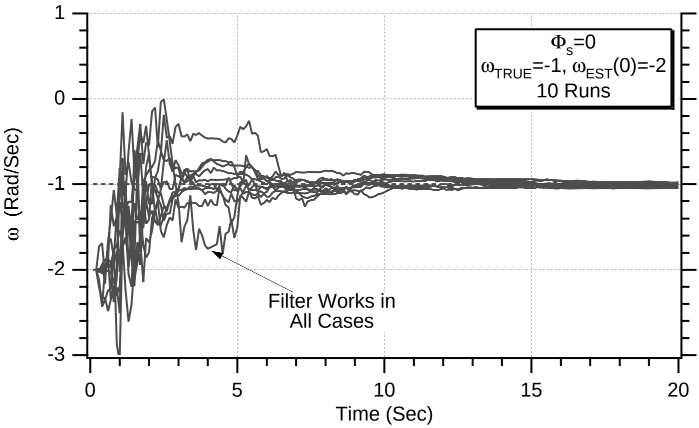

Fig. 10.35 Alternate three-state extended Kalman filter estimates correct frequency when initial frequency estimate is of correct sign.

Another Extended Kalman Filter for Sinusoidal Model

We have just derived an extended Kalman filter in the preceding section in which the sinusoidal frequency ω was a state. We demonstrated that in order to obtain accurate estimates of the states of a sinusoidal signal it was not important to estimate the sign of the sinusoidal frequency but only its magnitude. With that information we will see if we can do even better in the estimation process by reforming the problem to match our new experience base. Let us define a new state z, which is simply the square of the frequency or

$$z = \omega^2$$

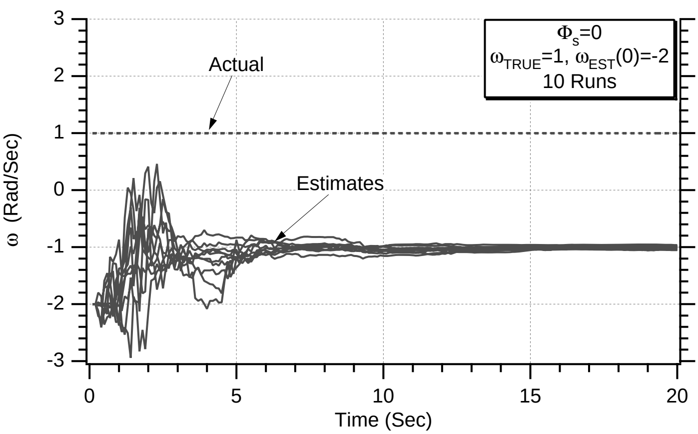

Fig. 10.36 Alternate three-state extended Kalman filter estimates correct frequency magnitude when initial frequency estimate is of the wrong sign.

Under these new circumstances the differential equations representing the real world in this new model are now given by

$$\ddot{x} = -zx$$
$$\dot{z} = u_s$$

where u_s is white process noise with spectral density Φ_s. The preceding differential equations can be expressed as three first-order differential equations (i.e., one of the equations is redundant) in state-space form as

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -z & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ u_s \end{bmatrix}$$

The preceding equation is also nonlinear because a state also shows up in the 3×3 matrix multiplying the state vector. From the preceding equation we know that the systems dynamics matrix is given by

$$\boldsymbol{F} = \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{bmatrix} \dfrac{\partial \dot{x}}{\partial x} & \dfrac{\partial \dot{x}}{\partial \dot{x}} & \dfrac{\partial \dot{x}}{\partial z} \\ \dfrac{\partial \ddot{x}}{\partial x} & \dfrac{\partial \ddot{x}}{\partial \dot{x}} & \dfrac{\partial \ddot{x}}{\partial z} \\ \dfrac{\partial \dot{z}}{\partial x} & \dfrac{\partial \dot{z}}{\partial \dot{x}} & \dfrac{\partial \dot{z}}{\partial z} \end{bmatrix}$$

where the partial derivatives are evaluated at the current estimates. Taking the partial derivatives in this example can be done by inspection, and the resultant systems dynamics matrix is given by

$$\boldsymbol{F} = \begin{bmatrix} 0 & 1 & 0 \\ -\hat{z} & 0 & -\hat{x} \\ 0 & 0 & 0 \end{bmatrix}$$

where the terms in the systems dynamics matrix have been evaluated at the current state estimates. As was the case in the last section, the exact fundamental matrix will be difficult, if not impossible, to find. If we assume that the elements of the systems dynamics matrix are approximately constant between sampling instants, then we use a two-term Taylor-series approximation for the fundamental matrix, yielding

$$\Phi(t) \approx I + \boldsymbol{F}t = \begin{bmatrix} 1 & t & 0 \\ -\hat{z}t & 1 & -\hat{x}t \\ 0 & 0 & 1 \end{bmatrix}$$

Therefore, the discrete fundamental matrix can be obtained by substituting T_s for t or

$$\Phi_k \approx \begin{bmatrix} 1 & T_s & 0 \\ -\hat{z}_{k-1}T_s & 1 & -\hat{x}_{k-1}T_s \\ 0 & 0 & 1 \end{bmatrix}$$

As was the case in the preceding section, the continuous process-noise matrix can by found from the original state-space equation to be

$$\boldsymbol{Q} = E(ww^T) = E\left[\begin{bmatrix} 0 \\ 0 \\ u_s \end{bmatrix}[0 \quad 0 \quad u_s]\right] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Phi_s \end{bmatrix}$$

Recall that the discrete process-noise matrix can be found from the continuous process-noise matrix according to

$$\boldsymbol{Q}_k = \int_0^{T_s} \Phi(\tau)\boldsymbol{Q}\Phi^T(\tau)\,\mathrm{d}\boldsymbol{t}$$

By substitution of the appropriate matrices into the preceding equation, we obtain

$$\boldsymbol{Q}_k = \int_0^{T_s} \begin{bmatrix} 1 & \tau & 0 \\ -\hat{z}\tau & 1 & -\hat{x}\tau \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Phi_s \end{bmatrix} \begin{bmatrix} 1 & -\hat{z}\tau & 0 \\ \tau & 1 & 0 \\ 0 & -\hat{x}\tau & 1 \end{bmatrix} \mathrm{d}\tau$$

Multiplying out the three matrices yields

$$\boldsymbol{Q}_k = \int_0^{T_s} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \hat{x}^2\tau^2\Phi_s & -\hat{x}\tau\Phi_s \\ 0 & -\hat{x}\tau\Phi_s & \Phi_s \end{bmatrix} \mathrm{d}\tau$$

If we again assume that the states are approximately constant between the sampling intervals, then the preceding integral can easily be evaluated as

$$\boldsymbol{Q}_k = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.333\hat{x}^2T_s^3\Phi_s & -0.5\hat{x}T_s^2\Phi_s \\ 0 & -0.5\hat{x}T_s^2\Phi_s & T_s\Phi_s \end{bmatrix}$$

In this problem we are also assuming that the measurement is of the first state plus noise or

$$x_k^* = x_k + v_k$$

Therefore, the measurement is linearly related to the states according to

$$x_k^* = [1 \quad 0 \quad 0]\begin{bmatrix} x \\ \dot{x} \\ z \end{bmatrix} + v_k$$

The measurement matrix can be obtained from the preceding equation by inspection as

$$\boldsymbol{H} = [1 \quad 0 \quad 0]$$

In this example the discrete measurement noise matrix is a scalar and is given by

$$\boldsymbol{R}_k = E(\boldsymbol{v}_k \boldsymbol{v}_k^T) = \sigma_x^2$$

We now have enough information to solve the matrix Riccati equations for the Kalman gains.

As was the case in the preceding section, the projected states in the actual extended Kalman-filtering equations do not have to use the approximation for the fundamental matrix. Instead, the state projections, indicated by an overbar, can be obtained by numerically integrating the nonlinear differential equations over the sampling interval. Therefore, the extended Kalman-filtering equations can then be written as

$$\hat{x}_k = \bar{x}_k + K_{1_k}(x_k^* - \bar{x}_k)$$
$$\hat{\dot{x}}_k = \bar{\dot{x}}_k + K_{2_k}(x_k^* - \bar{x}_k)$$
$$\hat{z}_k = \hat{z}_{k-1} + K_{3_k}(x_k^* - \bar{x}_k)$$

Listing 10.4 presents another three-state extended Kalman filter for estimating the states of a noisy sinusoidal signal whose frequency is unknown. As was the case in the last section, the simulation is initially set up to run without process noise (i.e., PHIS = 0). The actual sinusoidal signal has unity amplitude, and the standard deviation of the measurement noise is also unity. We can see from Listing 10.4 that the frequency of the actual sinusoidal signal is unity. The filter's initial state estimates of x and $\dot{x}$ are set to zero. Because the initial estimate of the frequency in the previous section was two (rather than unity), the initial value of the estimate of z is 4 (i.e., $z = \omega^2 = 2^2 = 4$). In other words, as was the case in the preceding section, the second and third states are mismatched from the real world. Values are used for the initial covariance matrix to reflect the uncertainties in our initial state estimates. Subroutine PROJECT is still used to integrate the nonlinear equations, using Euler integration over the sampling interval to obtain the state projections for x and $\dot{x}$.

A case was run using the nominal values of Listing 10.4 (i.e., no process noise), and we can see from a comparison that the actual quantities and their estimates in Figs. 10.37–10.39 are quite close. If we compare the estimates of x and $\dot{x}$ (i.e., Figs. 10.37 and 10.38) with the preceding results from our alternate

Listing 10.4 Another three-state extended Kalman filter for sinusoidal signal with unknown frequency

```
C THE FIRST THREE STATEMENTS INVOKE THE ABSOFT RANDOM
  NUMBER GENERATOR ON THE MACINTOSH
      GLOBAL DEFINE
            INCLUDE 'quickdraw.inc'
      END
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 P(3,3),Q(3,3),M(3,3),PHI(3,3),HMAT(1,3),HT(3,1),PHIT(3,3)
      REAL*8 RMAT(1,1),IDN(3,3),PHIP(3,3),PHIPPHIT(3,3),HM(1,3)
      REAL*8 HMHT(1,1),HMHTR(1,1),HMHTRINV(1,1),MHT(3,1),K(3,1),
        F(3,3)
      REAL*8 KH(3,3),IKH(3,3)
      INTEGER ORDER
      HP=.001
      W=1.
      WH=2.
      A=1.
      TS=.1
      ORDER=3
      PHIS=0.
      SIGX=1.
      OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
      OPEN(2,STATUS='UNKNOWN',FILE='COVFIL')
      T=0.
      S=0.
      H=.001
      DO 14 I=1,ORDER
      DO 14 J=1,ORDER
      F(I,J)=0.
      PHI(I,J)=0.
      P(I,J)=0.
      Q(I,J)=0.
      IDN(I,J)=0.
 14   CONTINUE
      RMAT(1,1)=SIGX**2
      IDN(1,1)=1.
      IDN(2,2)=1.
      IDN(3,3)=1.
      P(1,1)=SIGX**2
      P(2,2)=2.**2
      P(3,3)=4.**2
      XTH=0.
      XTDH=0.
      ZH=WH**2
      XT=0.
      XTD=A*W
      WHILE(T<=20.)
           XTOLD=XT
           XTDOLD=XTD
```

(*continued*)

Listing 10.4 (*Continued*)

```
          XTDD=-W*W*XT
          XT=XT+H*XTD
          XTD=XTD+H*XTDD
          T=T+H
          XTDD=-W*W*XT
          XT=.5*(XTOLD+XT+H*XTD)
          XTD=.5*(XTDOLD+XTD+H*XTDD)
          S=S+H
          IF(S>=(TS-.00001))THEN
               S=0.
               F(1,2)=1.
               F(2,1)=-WH**2
               F(2,3)=-2.*WH*XTH
               PHI(1,1)=1.
               PHI(1,2)=TS
               PHI(2,1)=-ZH*TS
               PHI(2,2)=1.
               PHI(2,3)=-XTH*TS
               PHI(3,3)=1.
               Q(2,2)=XTH*XTH*TS*TS*TS*PHIS/3.
               Q(2,3)=-XTH*TS*TS*PHIS/2.
               Q(3,2)=Q(2,3)
               Q(3,3)=PHIS*TS
               HMAT(1,1)=1.
               HMAT(1,2)=0.
               HMAT(1,3)=0.
               CALL MATTRN(PHI,ORDER,ORDER,PHIT)
               CALL MATTRN(HMAT,1,ORDER,HT)
               CALL MATMUL(PHI,ORDER,ORDER,P,ORDER,
                 ORDER,PHIP)
               CALL MATMUL(PHIP,ORDER,ORDER,PHIT,ORDER,
1                ORDER,PHIPPHIT)
               CALL MATADD(PHIPPHIT,ORDER,ORDER,Q,M)
               CALL MATMUL(HMAT,1,ORDER,M,ORDER,ORDER,
                 HM)
               CALL MATMUL(HM,1,ORDER,HT,ORDER,1,HMHT)
               CALL MATADD(HMHT,ORDER,ORDER,RMAT,
                 HMHTR)HMHTRINV(1,1)=1./HMHTR(1,1)
               CALL MATMUL(M,ORDER,ORDER,HT,ORDER,1,
                 MHT)
               CALL MATMUL(MHT,ORDER,1,HMHTRINV,1,1,K)
               CALL MATMUL(K,ORDER,1,HMAT,1,ORDER,KH)
               CALL MATSUB(IDN,ORDER,ORDER,KH,IKH)
               CALL MATMUL(IKH,ORDER,ORDER,M,ORDER,
                 ORDER,P)
               CALL GAUSS(XTNOISE,SIGX)
               XTMEAS=XT+XTNOISE
               CALL PROJECT(T,TS,XTH,XTDH,XTB,XTDB,HP,ZH)
               RES=XTMEAS-XTB
```

(*continued*)

Listing 10.4 (*Continued*)

```
                    XTH=XTB+K(1,1)*RES
                    XTDH=XTDB+K(2,1)*RES
                    ZH=ZH+K(3,1)*RES
                    ERRX=XT-XTH
                    SP11=SQRT(P(1,1))
                    ERRXD=XTD-XTDH
                    SP22=SQRT(P(2,2))
                    Z=W**2
                    ERRZ=Z-ZH
                    SP33=SQRT(P(3,3))
                    WH=SQRT(ZH)

                    WRITE(9,*)T,XT,XTH,XTD,XTDH  ,Z,ZH
                    WRITE(1,*)T,XT,XTH,XTD,XTDH  ,Z,ZH
                    WRITE(2,*)T,ERRX,SP11,-SP11,ERRXD,SP22,-SP22,
 1                     ERRZ,SP33,-SP33
              ENDIF
        END DO
        PAUSE
        CLOSE(1)
        CLOSE(2)
        END

        SUBROUTINE PROJECT(TP,TS,XTP,XTDP,XTH,XTDH,HP,Z)
        IMPLICIT REAL*8 (A-H)
        IMPLICIT REAL*8 (O-Z)
        T=0.
        XT=XTP
        XTD=XTDP
        H=HP
        WHILE(T<=(TS-.0001))
              XTDD=-Z*XT
              XTD=XTD+H*XTDD
              XT=XT+H*XTD
              T=T+H
        END DO
        XTH=XT
        XTDH=XTD
        RETURN
        END

C  SUBROUTINE GAUSS IS SHOWN IN LISTING 1.8
C  SUBROUTINE MATTRN IS SHOWN IN LISTING 1.3
C  SUBROUTINE MATMUL IS SHOWN IN LISTING 1.4
C  SUBROUTINE MATADD IS SHOWN IN LISTING 1.1
C  SUBROUTINE MATSUB IS SHOWN IN LISTING 1.2
```

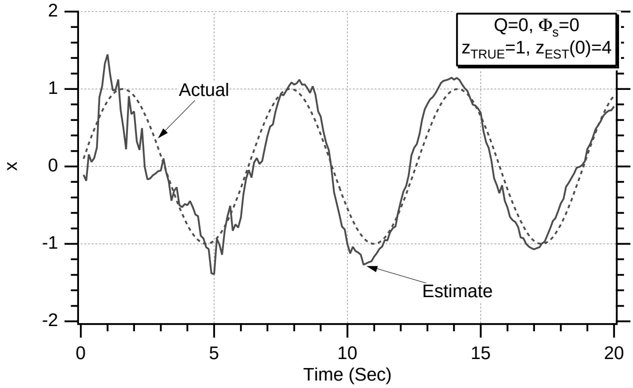

Fig. 10.37 New extended Kalman filter estimates first state quite well.

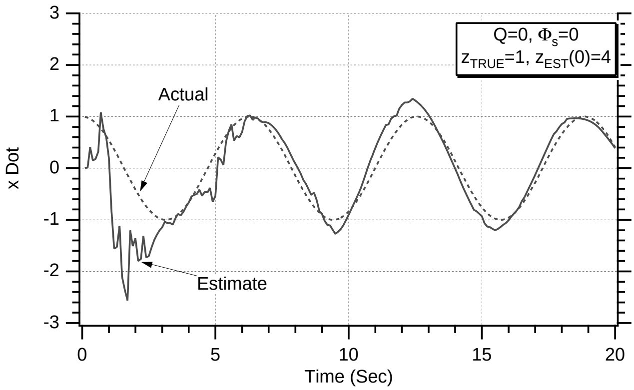

Fig. 10.38 New extended Kalman filter estimates second state well, even though that state is not correctly initialized.

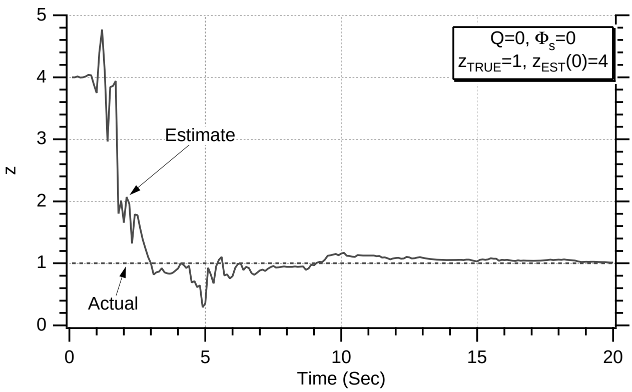

Fig. 10.39 After 5 s the new extended Kalman filter is able to estimate the square of frequency.

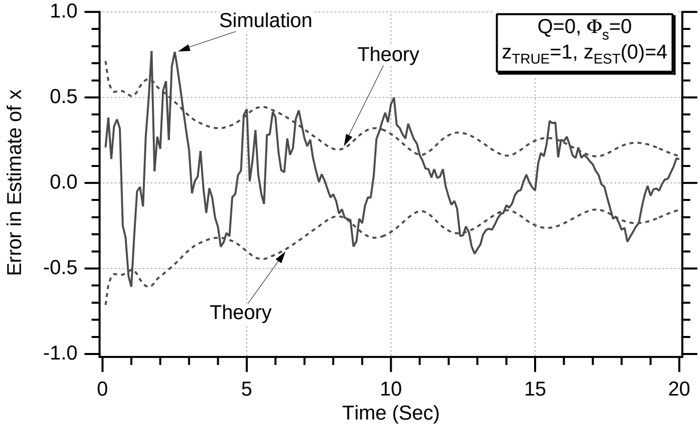

Fig. 10.40 Error in the estimate of first state agrees with theory.

extended Kalman filter (ie., Figs. 10.20 and 10.21), we can see that the new results are virtually identical to the preceding results. Our new filter does not appear to be any better or worse than the preceding filter. From Fig. 10.39 we can see that it takes approximately 5 s to obtain an accurate estimate of z (i.e., the square of the frequency) for the parameters considered.

Figures 10.40–10.42 display the theoretical errors in the state estimates (i.e., obtained by taking the square root of the appropriate diagonal element of the covariance matrix) and the single-run errors for each of the three states obtained from the simulation. The single-run errors in the estimates appear to be within the theoretical bounds approximately 68% of the time, indicating that the extended Kalman filter seems to be behaving properly. Because this example has zero process noise, the errors in the estimates diminish as more measurements are taken. The results of Figs. 10.40 and 10.41 are virtually identical to the results of

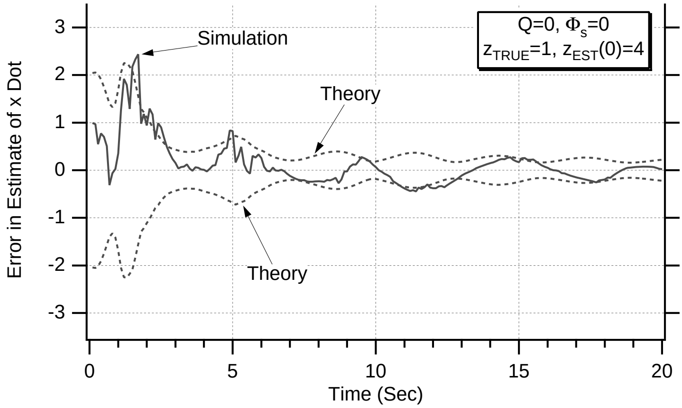

Fig. 10.41 Error in the estimate of second state agrees with theory.

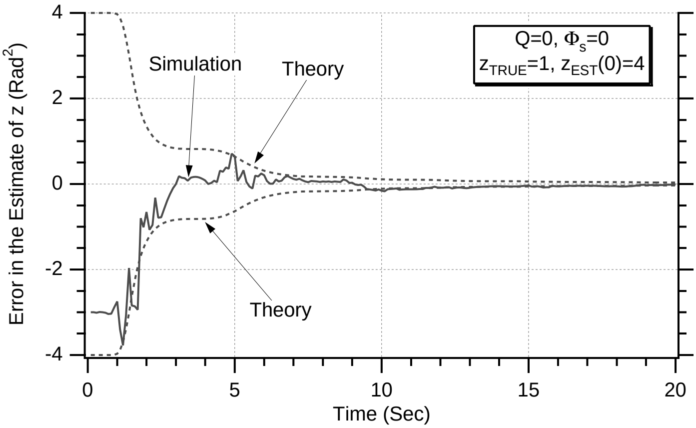

Fig. 10.42 Error in the estimate of third state agrees with theory.

Figs. 10.30 and 10.31. Thus, it appears that there is virtually no difference between the extended Kalman filter of this section and the one of the preceding section.

To verify that the filter was actually working all of the time, Listing 10.4 was slightly modified to give it Monte Carlo capabilities. A ten-run Monte Carlo set was run with the new three-state extended Kalman filter. Figure 10.43 shows that we are able to accurately estimate z in each of the 10 runs. Thus, we can conclude that the new three-state extended Kalman filter is also effective in estimating the states of a sinusoid.

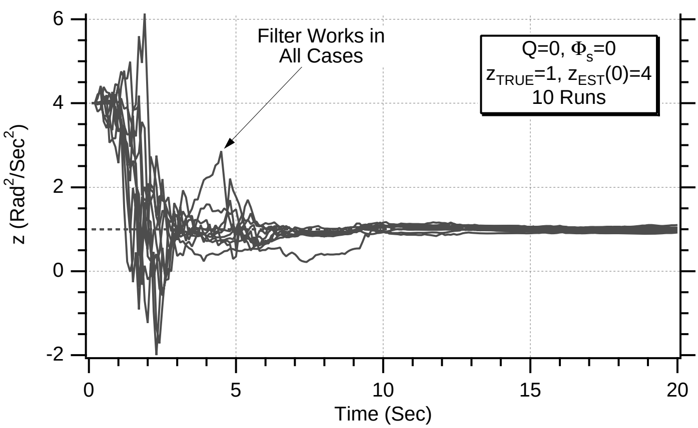

Fig. 10.43 New three-state extended Kalman filter is also effective in estimating the square of the frequency of a sinusoid.

Summary

We have developed several extended Kalman filters that attempt to estimate the states of a sinusoidal signal based upon noisy measurements of the signal. We have demonstrated in this chapter that simply choosing states and building an extended Kalman filter does not guarantee that it will actually work as expected if programmed correctly. Sometimes, some states that are being estimated are not observable. Numerous experiments were conducted, and various extended Kalman filters were designed, each of which had different states, to highlight some of the issues. An extended Kalman filter was finally built that could estimate all of the states under a variety of initialization conditions, but it could not always distinguish between positive and negative frequencies (i.e., always a correct estimate of frequency magnitude but not always a correct estimate of the frequency sign). Monte Carlo experiments confirmed that this particular extended Kalman filter appeared to be very robust. An alternate form of the extended Kalman filter, in which the square of the frequency was being estimated, was shown to be equally as robust.

References

[1]Gelb., A., *Applied Optimal Estimation*, Massachusetts Inst. of Technology Press, Cambridge, MA, 1974, pp. 180–228.

[2]Zarchan, P., *Tactical and Strategic Missile Guidance* 3rd ed., Progress in Astronautics and Aeronautics, AIAA, Reston, VA, 1998, pp. 373–387.

This page intentionally left blank