Chapter 6

Continuous Polynomial Kalman Filter

Introduction

SO FAR we have only studied the discrete Kalman filter because that is the filter that is usually implemented in real-world applications. However, there is also a continuous version of the Kalman filter. Although the continuous Kalman filter is usually not implemented, it can be used as an aid in better understanding the properties of the discrete Kalman filter. For small sampling times the discrete Kalman filter in fact becomes a continuous Kalman filter. Because the continuous Kalman filter does not require the derivation of a fundamental matrix for either the Riccati equations or filtering equations, the continuous filter can also be used as a check of the discrete filter. In addition, we shall see that with the continuous polynomial Kalman filter it is possible, under steady-state conditions, to derive transfer functions that exactly represent the filter. These transfer functions cannot only be used to better understand the operation of the Kalman filter (i.e., both continuous and discrete) but can also aid with such mundane issues as helping to choose the correct amount of process noise to use.

First, we will present the theoretical differential equations for the continuous Riccati equations and Kalman filter. Next, we will derive different-order continuous polynomial Kalman filters. We will demonstrate for the case of zero process noise that the Kalman gains and covariance matrix predictions of the continuous polynomial Kalman filter exactly match the formulas we have already derived for the discrete polynomial Kalman filter. We will then show that when process noise is introduced we can derive closed-form solutions for both the gains and covariance matrix predictions of the continuous polynomial Kalman filters. These closed-form solutions can then be used to derive transfer functions for the continuous polynomial Kalman filter. Finally, formulas will be derived showing how the bandwidth of the continuous polynomial Kalman filter transfer function is related to the amount of process and measurement noise we tell the filter.

Theoretical Equations

As was the case with the discrete filter, continuous Kalman filtering theory[1] first requires that our model of the real world be described by a set of differential equations. These equations must be cast in matrix or state-space form as

$$\dot{\boldsymbol{x}} = \boldsymbol{F}\boldsymbol{x} + \boldsymbol{G}\boldsymbol{u} + \boldsymbol{w}$$

where $\boldsymbol{x}$ is a column vector with the states of the system, $\boldsymbol{F}$ the system dynamics matrix, $\boldsymbol{u}$ a known deterministic or control vector, and $\boldsymbol{w}$ a white noise process represented as a column vector. There is a process noise matrix $\boldsymbol{Q}$ that is related to the process noise column vector according to

$$\boldsymbol{Q} = \boldsymbol{E}[\boldsymbol{w}\boldsymbol{w}^T]$$

We have seen in preceding chapters that although the process noise sometimes has no physical meaning it is often used as a device for telling the filter that we understand that the filter's model of the real world may not be precise (i.e., larger values of process noise indicate less confidence in our model of the real world). The continuous Kalman filter also requires that the measurements be linearly related to the states according to

$$\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{v}$$

where $\boldsymbol{z}$ is the measurement vector, $\boldsymbol{H}$ the measurement matrix, and $\boldsymbol{v}$ white measurement noise. The measurement noise matrix $\boldsymbol{R}$ is related to the measurement noise vector $\boldsymbol{v}$ according to

$$\boldsymbol{R} = \boldsymbol{E}[\boldsymbol{v}\boldsymbol{v}^T]$$

where the continuous measurement noise matrix consists of spectral densities describing the measurement noise. In preceding chapters we saw that the discrete measurement noise matrix consisted of variances describing the measurement noise.

For the model of the real world just described, the resultant continuous Kalman filter is described by the matrix differential equation

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{F}\hat{\boldsymbol{x}} + \boldsymbol{G}\boldsymbol{u} + \boldsymbol{K}(\boldsymbol{z} - \boldsymbol{H}\hat{\boldsymbol{x}})$$

The continuous Kalman filter no longer requires the fundamental matrix for the propagation of the states. State propagation is accomplished by numerically integrating the matrix differential equation involving the system dynamics matrix. The Kalman gains $\boldsymbol{K}$, required by the preceding Kalman-filtering differential equation, are now obtained by first integrating the nonlinear matrix differential Riccati equation for the covariance matrix

$$\dot{\boldsymbol{P}} = -\boldsymbol{P}\boldsymbol{H}^T\boldsymbol{R}^{-1}\boldsymbol{H}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{F}^T + \boldsymbol{F}\boldsymbol{P} + \boldsymbol{Q}$$

and then solving the matrix equation for the gain in terms of the covariance matrix or

$$\boldsymbol{K} = \boldsymbol{P}\boldsymbol{H}^T\boldsymbol{R}^{-1}$$

In the next few sections we will derive continuous polynomial Kalman filters for different-order systems.

However, let us first visualize how the filtering equations work in the continuous case. From the continuous Riccati equation we can see that there are two sources of uncertainty involved, represented by the covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$. If we are trying to quantify the correctness of our model of the real world, we must also include its uncertainty as part of its behavior (i.e., characterized by the process noise $\boldsymbol{Q}$). This means we want the process noise to directly influence our estimates. The measurement noise $\boldsymbol{R}$ is a hindrance in estimating the states of the real world, and we would like to reduce its influence in the estimation process. We can see from the filtering equation that the filter gain $\boldsymbol{K}$ operates on the difference between a measurement and the prior estimate of what the measurement should be. Consequently, a larger $\boldsymbol{Q}$ increases $\boldsymbol{P}$, which results in a larger $\boldsymbol{K}$, whereas a larger $\boldsymbol{R}$ tends to decrease $\boldsymbol{K}$. A decreased $\boldsymbol{K}$ does not weight the measurement as much because it is noisy, whereas a larger $\boldsymbol{K}$ (caused by more $\boldsymbol{Q}$) causes more of the measurement to be incorporated in the estimate. It will be seen that this logic leads directly to the concept of how filter bandwidth is determined (i.e., larger $\boldsymbol{Q}$ or smaller $\boldsymbol{R}$ will increase the filter bandwidth, whereas smaller $\boldsymbol{Q}$ or larger $\boldsymbol{R}$ will decrease the filter bandwidth). These considerations apply similarly to the discrete Kalman filter presented earlier.

Zeroth-Order or One-State Continuous Polynomial Kalman Filter

In a zeroth-order system the differential equation representing the real world is given by

$$\dot{x} = u_s$$

where x is the state and u_s is white process noise with spectral density Φ_s. Therefore, for the preceding zeroth-order polynomial case, matrices became scalars, and the continuous process noise matrix is also a scalar given by

$$\boldsymbol{Q} = E(u_s^2) = \Phi_s$$

From the state-space equation or the equation representing the real world the system dynamics matrix can be obtained by inspection and is the scalar

$$\boldsymbol{F} = 0$$

Let us assume that the measurement equation is simply

$$x^* = x + v_n$$

where v_n is white noise with spectral density Φ_n. We can see from the preceding equation that the measurement matrix is a scalar given by

$$\boldsymbol{H} = 1$$

whereas the measurement noise matrix is a scalar spectral density or

$$\boldsymbol{R} = E(v_n^2) = \Phi_n$$

Substitution of the preceding scalar matrices into the nonlinear matrix Riccati differential equation for the covariance yields

$$\dot{\boldsymbol{P}} = -\boldsymbol{P}\boldsymbol{H}^T\boldsymbol{R}^{-1}\boldsymbol{H}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{F}^T + \boldsymbol{F}\boldsymbol{P} + \boldsymbol{Q} = -P\Phi_n^{-1}P + \Phi_s$$

or

$$\dot{P} = \frac{-P^2}{\Phi_n} + \Phi_s$$

whereas the Kalman gain can be obtained from

$$\boldsymbol{K} = \boldsymbol{P}\boldsymbol{H}^T\boldsymbol{R}^{-1} = P\Phi_n^{-1}$$

or

$$K = \frac{P}{\Phi_n}$$

We will often call the continuous Kalman gain K_c.

We would like to see if the formulas we just derived for the gain and covariance of the zeroth-order continuous polynomial Kalman filter are equivalent in some way to their discrete counter parts. We have already shown in Chapter 4 that a discrete polynomial Kalman filter without process noise and infinite initial covariance matrix is equivalent to a recursive least-squares filter. Therefore, we will use our closed-form solutions for the recursive least-squares filter as if they were solutions for the discrete Kalman filter (i.e., without process noise). For the zeroth-order system we have demonstrated in Chapter 3 that the recursive least-squares filter gain (i.e., discrete Kalman gain) is given by

$$K_k = \frac{1}{k} \qquad k = 1, 2, \ldots, n$$

whereas the variance of the error in the estimate of the state (i.e., covariance matrix) is

$$P_k = \frac{\sigma_n^2}{k}$$

In this chapter we will sometimes call the discrete or recursive Kalman gain K_d and the discrete error in the estimate of the state P_d.

A simulation was written to integrate the continuous differential equation derived from the zeroth-order covariance differential equation

$$\dot{P} = \frac{-P^2}{\Phi_n} + \Phi_s$$

and the resultant program is shown in Listing 6.1. However, to make the comparison with the discrete filter we will initially set the process noise to zero (i.e., PHIS = 0). Also shown in the program are the relationships between the discrete noise variance and continuous noise spectral density. If the sampling interval T_s in the discrete world is small, we can relate the continuous noise spectral density to the discrete noise variance σ_n^2 according to

$$\Phi_n = \sigma_n^2 T_s$$

Similarly, Listing 6.1 also describes the theoretical relationship between the continuous Kalman gain and discrete recursive least-squares gain. One can show that as the sampling interval gets small the continuous Kalman gain K_c and the discrete recursive least-squares gain K_d are related according to[2]

$$K_c = \frac{K_d}{T_s}$$

As also can be seen from Listing 6.1, the simulation uses second-order Runge–Kutta numerical integration and double-precision arithmetic to solve the nonlinear Riccati differential equation for the covariance of the error in the estimate. The program had numerical difficulties when the initial covariance was set to infinity, and so the initial covariance was set to 100 to avoid those difficulties. We have already shown in Chapter 4 that the solution for the steady-state covariance matrix was virtually independent of initial condition. Therefore, it was hypothesized that choosing an initial covariance value to avoid numerical problems should not influence the final result. The integration step size used to integrate the Riccati equation was set to 0.001 s. Using smaller values of the integration step size did not appear to change the answers significantly but did dramatically increase the computer running time.

The nominal case of Listing 6.1 was run. From Fig. 6.1 we can conclude that, for the case considered, the match between the continuous Kalman gain and the discrete recursive least-squares gain (i.e., or discrete Kalman gain) is excellent. From Fig. 6.2 we can conclude that the continuous and discrete covariances are also nearly identical. It therefore appears that we are integrating the nonlinear covariance differential equation correctly because the theoretical relationships have been verified. In other words, for small sampling times we have verified that for the one-state system the continuous Kalman gain evaluated at the sampling

Listing 6.1 Integrating one-state covariance nonlinear Riccati differential equation

```
    IMPLICIT REAL*8(A-H)
    IMPLICIT REAL*8(O-Z)
    REAL*8 F(1,1),P(1,1),Q(1,1),POLD(1,1),HP(1,1)
    REAL*8 PD(1,1)
    REAL*8 HMAT(1,1),HT(1,1),FP(1,1),PFT(1,1),PHT(1,1),K(1,1)
    REAL*8 PHTHP(1,1),PHTHPR(1,1),PFTFP(1,1),PFTFPQ(1,1)
    INTEGER ORDER
    OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
    ORDER=1
    T=0.
    S=0.
    H=.001
    TS=.1
    TF=10.
    PHIS=0.
    XJ=1.
    DO 14 I=1,ORDER
    DO 14 J=1,ORDER
    F(I,J)=0.
    P(I,J)=0.
    Q(I,J)=0.
14  CONTINUE
    DO 11 I=1,ORDER
    HMAT(1,I)=0.
    HT(I,1)=0.
11  CONTINUE
    Q(1,1)=PHIS
    HMAT(1,1)=1.
    HT(1,1)=1.
    SIGN2=1.**2
    PHIN=SIGN2*TS
    P(1,1)=100.
    WHILE(T<=TF)
            DO 20 I=1,ORDER
            DO 20 J=1,ORDER
                    POLD(I,J)=P(I,J)
20          CONTINUE
            CALL MATMUL(F,ORDER,ORDER,P,ORDER,ORDER,FP)
            CALL MATTRN(FP,ORDER,ORDER,PFT)
            CALL MATMUL(P,ORDER,ORDER,HT,ORDER,1,PHT)
            CALL MATMUL(HMAT,1,ORDER,P,ORDER,ORDER,HP)
            CALL MATMUL(PHT,ORDER,1,HP,1,ORDER,PHTHP)
            DO 12 I=1,ORDER
            DO 12 J=1,ORDER
                    PHTHPR(I,J)=PHTHP(I,J)/PHIN
12.         CONTINUE
            CALL MATADD(PFT,ORDER,ORDER,FP,PFTFP)
```

(*continued*)

Listing 6.1 (*Continued*)

```
          CALL MATADD(PFTFP,ORDER,ORDER,Q,PFTFPQ)
          CALL MATSUB(PFTFPQ,ORDER,ORDER,PHTHPR,PD)
          DO 13 I=1,ORDER
                K(I,1)=PHT(I,1)/PHIN
13        CONTINUE
          DO 50 I=1,ORDER
          DO 50 J=1,ORDER
                P(I,J)=P(I,J)+H*PD(I,J)
50        CONTINUE
          T=T+H
          CALL MATMUL(F,ORDER,ORDER,P,ORDER,ORDER,FP)
          CALL MATTRN(FP,ORDER,ORDER,PFT)
          CALL MATMUL(P,ORDER,ORDER,HT,ORDER,1,PHT)
          CALL MATMUL(HMAT,1,ORDER,P,ORDER,ORDER,HP)
          CALL MATMUL(PHT,ORDER,1,HP,1,ORDER,PHTHP)
          DO 15 I=1,ORDER
          DO 15 J=1,ORDER
                PHTHPR(I,J)=PHTHP(I,J)/PHIN
15        CONTINUE
          CALL MATADD(PFT,ORDER,ORDER,FP,PFTFP)
          CALL MATADD(PFTFP,ORDER,ORDER,Q,PFTFPQ)
          CALL MATSUB(PFTFPQ,ORDER,ORDER,PHTHPR,PD)
          DO 16 I=1,ORDER
          K(I,1)=PHT(I,1)/PHIN
16        CONTINUE
          DO 60 I=1,ORDER
          DO 60 J=1,ORDER
                P(I,J)=.5*(POLD(I,J)+P(I,J)+H*PD(I,J))
60        CONTINUE
          S=S+H
          IF(S>=(TS-.00001))THEN
               S=0.
               XK1=1./XJ
               PDISC=SIGN2/XJ;
               WRITE(9,*)T,K(1,1)*TS,XK1,P,PDISC
               WRITE(1,*)T,K(1,1)*TS,XK1,P,PDISC
               XJ=XJ+1.
          ENDIF
  END DO
  PAUSE
  CLOSE(1)
  END

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
```

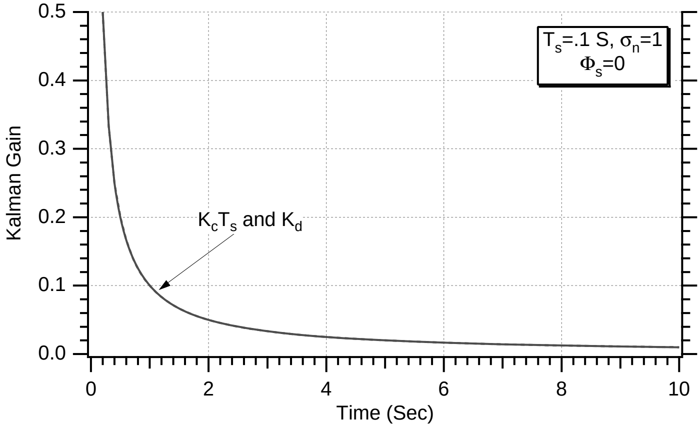

Fig. 6.1 Continuous and discrete Kalman gains are identical for zeroth-order system.

time is simply the discrete recursive least-squares gain (or discrete Kalman gain evaluated at discrete sampling times) divided by the sampling time, or

$$K_c = \frac{K_d}{T_s}$$

In addition we have verified that the continuous and discrete covariances are equal or

$$P_c = P_d$$

We would also get a similarly excellent match between the continuous Riccati equations and discrete Riccati equations when there is process noise. However,

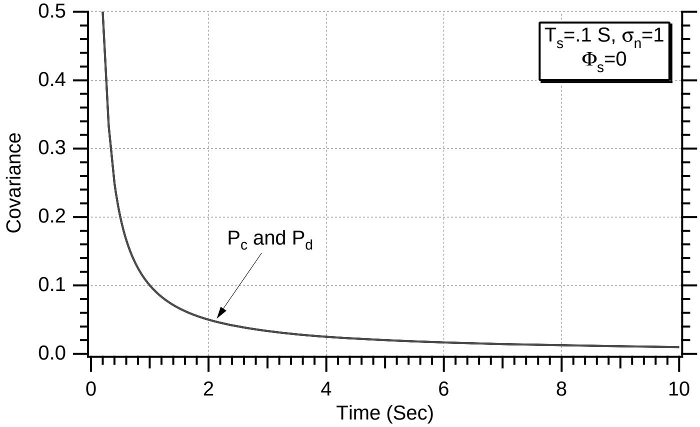

Fig. 6.2 Continuous and discrete covariances are identical for zeroth-order system.

with process noise the gain and covariance will not go to zero as time increases (i.e., more measurements are taken).

First-Order or Two-State Continuous Polynomial Kalman Filter

In a first-order system the differential equation representing the real world is given by

$$\ddot{x} = u_s$$

where x is the state and u_s is white process noise with spectral density Φ_s. We can represent the preceding differential equation in state-space form as

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

We can see from the preceding equation that the continuous process noise matrix can be described by

$$\boldsymbol{Q} = E\left[\begin{bmatrix} 0 \\ u_s \end{bmatrix}[0 \quad u_s]\right] = \Phi_s \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

whereas the system dynamics matrix can be obtained by inspection of the state-space equation as

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

We assume that the measurement equation is simply

$$x^* = x + v_n$$

where v_n is white noise with spectral density Φ_n. The preceding equation can be put into the form

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

Therefore, the measurement matrix can be obtained by inspection of the preceding equation as

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

In addition, the measurement noise matrix, which was already described by a scalar variance in the discrete world, now becomes a scalar spectral density in the continuous world or

$$\boldsymbol{R} = \Phi_n$$

Substitution of the preceding matrices into the differential equation for the covariance and numerically integrating

$$\dot{\boldsymbol{P}} = -\boldsymbol{P}\boldsymbol{H}^T\boldsymbol{R}^{-1}\boldsymbol{H}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{F}^T + \boldsymbol{F}\boldsymbol{P} + \boldsymbol{Q}$$

yields the covariance $\boldsymbol{P}$, which is now a two-dimensional matrix. The Kalman gain can be obtained from the covariance matrix according to

$$\boldsymbol{K} = \boldsymbol{P}\boldsymbol{H}^T\boldsymbol{R}^{-1}$$

where $\boldsymbol{K}$ is a two-dimensional column vector.

We would like see if the formulas we just derived for the gain and covariance matrices for the first-order filter are equivalent in some way to their discrete counterparts. To facilitate the comparison, we will again consider the case of zero process noise. We have already shown that a discrete polynomial Kalman filter without process noise and infinite initial covariance matrix is equivalent to a recursive least-squares filter. For the first-order system we have demonstrated in Chapter 3 that the two recursive least-squares filter gains, which are equivalent to the Kalman gains (i.e., for the case of zero process noise), are given by

$$K_{1_k} = \frac{2(2k-1)}{k(k+1)} \qquad k = 1, 2, \ldots, n$$

$$K_{2_k} = \frac{6}{k(k+1)T_s}$$

We have also demonstrated in Chapter 4 that the two diagonal terms of the Kalman or recursive least-squares covariance matrix for the case of zero process noise are given by

$$P_{11_k} = \frac{2(2k-1)\sigma_n^2}{k(k+1)}$$

$$P_{22_k} = \frac{12\sigma_n^2}{k(k^2-1)T_s^2}$$

A simulation was written to integrate the second-order continuous nonlinear matrix differential equation for the covariance matrix, and the resultant program is shown in Listing 6.2. The simulation is initially set up without process noise (PHIS $=$ 0). Also shown in the program are the relationships between the discrete noise variance and continuous noise spectral density. As was the case in the preceding section, if the sampling interval T_s in the discrete world is small we can relate the continuous noise spectral density to the discrete noise variance according to

$$\Phi_n = \sigma_n^2 T_s$$

In addition the program also describes the relationship between the continuous and discrete Kalman (i.e., recursive least squares) gains. We have already demonstrated that as the sampling interval gets small the continuous Kalman gain K_c and the discrete recursive least-squares gain K_d at the sampling times are related according to

$$K_c = \frac{K_d}{T_s}$$

As was the case with Listing 6.1, Listing 6.2 also uses second-order Runge–Kutta numerical integration and double-precision arithmetic to solve the nonlinear second-order matrix differential equation for the covariance matrix. This program also had numerical difficulties when the diagonal elements of the initial covariance matrix were set to infinity, and so the diagonal terms of the initial covariance matrix were set to 100 and the off-diagonal terms were set to zero. Recall that we have already shown in Chapter 4 that the solution for the steady-state covariance matrix was virtually independent of initial condition. Therefore, we again hypothesized that choosing an initial covariance matrix to avoid numerical problems should not influence the final result. As was the case with Listing 6.1, the integration step size in Listing 6.2 was set to 0.001 s. As was the case in the preceding section, using smaller values did not seem to change the answers significantly but did dramatically increase the run time.

The nominal case of Listing 6.2 was run, and the continuous and discrete Kalman gains are displayed in Figs. 6.3 and 6.4. From these figures we can conclude that for the case considered, the match between the continuous and discrete Kalman gains at the sampling times are excellent. We can feel confident that we are numerically integrating the nonlinear covariance matrix differential equation correctly because the theoretical relationships and simulation results agree. Again, for small sampling times we have verified that for the two-state system the continuous Kalman gain is simply the discrete Kalman gain divided by the sampling time or

$$K_{1c} = \frac{K_{1d}}{T_s}$$

$$K_{2c} = \frac{K_{2d}}{T_s}$$

Similarly Figs. 6.5 and 6.6 show that the covariance matrix diagonal terms obtained by numerically integrating the nonlinear matrix Riccati differential equation of Listing 6.2 matches the formulas from the recursive least-squares filter. Therefore, we have confidence that for the first-order polynomial Kalman filter there is also a good match between the continuous and discrete systems. In addition we have verified that the continuous and discrete covariances are equal or

$$P_{11c} = P_{11d}$$

$$P_{22c} = P_{22d}$$

One would also get a similar excellent match between the continuous Riccati equations and discrete Riccati equations when there is process noise for the first-order system. However, as was already mentioned, the gains and covariances will not go to zero as more measurements are taken when there is process noise.

Listing 6.2 Integrating two-state covariance nonlinear Riccati differential equation

```
      IMPLICIT  REAL*8(A-H)
      IMPLICIT  REAL*8(O-Z)
      REAL*8  F(2,2),P(2,2),Q(2,2),POLD(2,2),HP(1,2)
      REAL*8  PD(2,2)
      REAL*8  HMAT(1,2),HT(2,1),FP(2,2),PFT(2,2),PHT(2,1),K(2,1)
      REAL*8  PHTHP(2,2),PHTHPR(2,2),PFTFP(2,2),PFTFPQ(2,2)
      INTEGER  ORDER
      OPEN(1,STATUS=‘UNKNOWN’,FILE=‘DATFIL’)
      ORDER=2
      T=0.
      S=0.
      H=.001
      TS=.1
      TF=10.
      PHIS=0.
      XJ=1.
      DO  14  I=1,ORDER
      DO  14  J=1,ORDER
      F(I,J)=0.
      P(I,J)=0.
      Q(I,J)=0.
14    CONTINUE
      DO  11  I=1,ORDER
      HMAT(1,I)=0.
      HT(I,1)=0.
11    CONTINUE
      F(1,2)=1.
      Q(2,2)=PHIS
      HMAT(1,1)=1.
      HT(1,1)=1.
      SIGN2=1.**2
      PHIN=SIGN2*TS
      P(1,1)=100.
      P(2,2)=100.
      WHILE(T<=TF)
              DO  20  I=1,ORDER
              DO  20  J=1,ORDER
              POLD(I,J)=P(I,J)
20            CONTINUE
              CALL  MATMUL(F,ORDER,ORDER,P,ORDER,ORDER,FP)
              CALL  MATTRN(FP,ORDER,ORDER,PFT)
              CALL  MATMUL(P,ORDER,ORDER,HT,ORDER,1,PHT)
              CALL  MATMUL(HMAT,1,ORDER,P,ORDER,ORDER,HP)
              CALL  MATMUL(PHT,ORDER,1,HP,1,ORDER,PHTHP)
              DO  12  I=1,ORDER
              DO  12  J=1,ORDER
                        PHTHPR(I,J)=PHTHP(I,J)/PHIN
12            CONTINUE
              CALL  MATADD(PFT,ORDER,ORDER,FP,PFTFP)
```

(*continued*)

Listing 6.2 (*Continued*)

```
            CALL MATADD(PFTFP,ORDER,ORDER,Q,PFTFPQ)
            CALL MATSUB(PFTFPQ,ORDER,ORDER,PHTHPR,PD)
            DO 13 I=1,ORDER
                    K(I,1)=PHT(I,1)/PHIN
13          CONTINUE
            DO 50 I=1,ORDER
            DO 50 J=1,ORDER
                    P(I,J)=P(I,J)+H*PD(I,J)
50          CONTINUE
            T=T+H
            CALL MATMUL(F,ORDER,ORDER,P,ORDER,ORDER,FP)
            CALL MATTRN(FP,ORDER,ORDER,PFT)
            CALL MATMUL(P,ORDER,ORDER,HT,ORDER,1,PHT)
            CALL MATMUL(HMAT,1,ORDER,P,ORDER,ORDER,HP)
            CALL MATMUL(PHT,ORDER,1,HP,1,ORDER,PHTHP)
            DO 15 I=1,ORDER
            DO 15 J=1,ORDER
                    PHTHPR(I,J)=PHTHP(I,J)/PHIN
15          CONTINUE
            CALL MATADD(PFT,ORDER,ORDER,FP,PFTFP)
            CALL MATADD(PFTFP,ORDER,ORDER,Q,PFTFPQ)
            CALL MATSUB(PFTFPQ,ORDER,ORDER,PHTHPR,PD)
            DO 16 I=1,ORDER
                    K(I,1)=PHT(I,1)/PHIN
16          CONTINUE
            DO 60 I=1,ORDER
            DO 60 J=1,ORDER
                    P(I,J)=.5*(POLD(I,J)+P(I,J)+H*PD(I,J))
60          CONTINUE
            S=S+H
            IF(S>=(TS-.00001))THEN
                    S=0.
                    XK1=2.*(2.*XJ-1.)/(XJ*(XJ+1))
                    XK2=6./(XJ*(XJ+1)*TS)
                    P11DISC=2.*(2.*XJ-1)*SIGN2/(XJ*(XJ+1.))
                    IF(XJ.EQ.1)THEN
                            P22DISC=0.
                    ELSE
                            P22DISC=12*SIGN2/(XJ*(XJ*XJ-1)*TS*TS)
                    ENDIF
                    WRITE(9,*)T,K(1,1)*TS,XK1,K(2,1)*TS,XK2,P(1,1)
 1                          ,P11DISC,P(2,2),P22DISC
                    WRITE(1,*)T,K(1,1)*TS,XK1,K(2,1)*TS,XK2,P(1,1)
 1                          ,P11DISC,P(2,2),P22DISC
                    XJ=XJ+1.
            ENDIF
    END DO
    PAUSE
    CLOSE(1)
    END
```

(*continued*)

Listing 6.2 (*Continued*)

```
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
```

Second-Order or Three-State Continuous Polynomial Kalman Filter

Using techniques similar to that of the preceding two sections, one can show that for the second-order case the continuous process noise matrix is described by

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

whereas the system dynamics and measurement matrices are given by

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

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

The measurement noise matrix remains a scalar spectral density in the continuous world or

$$\boldsymbol{R} = \Phi_n$$

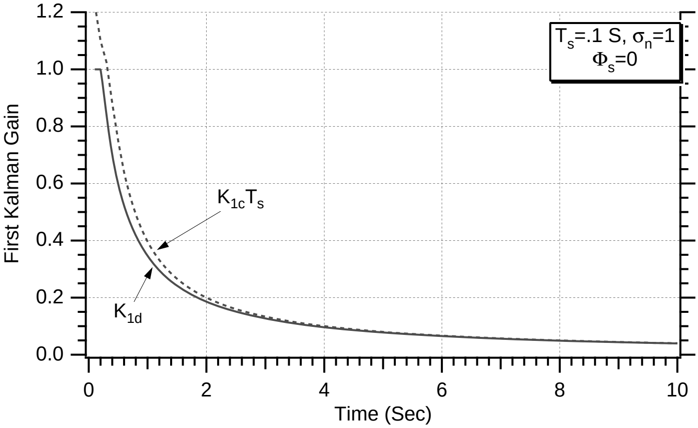

Fig. 6.3 Integrating two-state nonlinear matrix Riccati differential equation yields good match with formula for first gain.

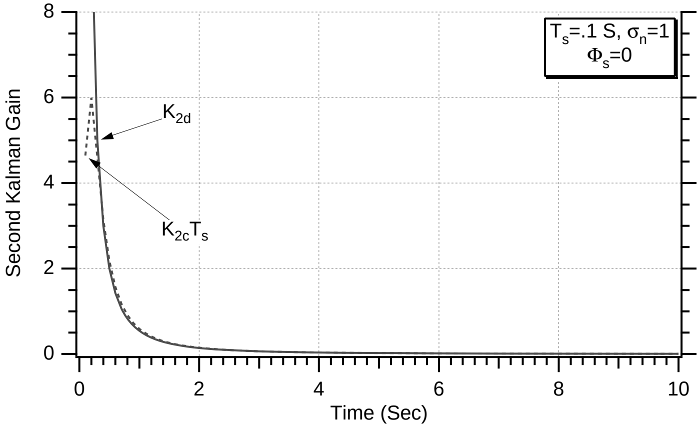

Fig. 6.4 Integrating two-state nonlinear matrix Riccati differential equation yields good match with formula for second gain.

Substitution of the preceding matrices into the differential equation for the covariance yields

$$\dot{P} = -PH^TR^{-1}HP + PF^T + FP + Q$$

where P is now a three-dimensional matrix and the Kalman gain can be obtained from

$$K = PH^TR^{-1}$$

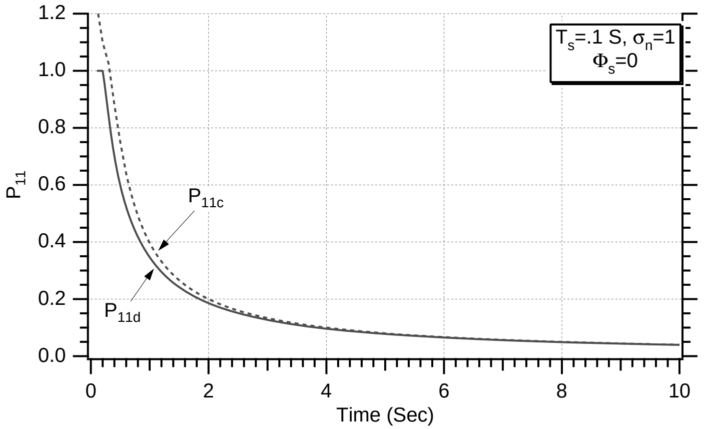

Fig. 6.5 Integrating two-state nonlinear matrix Riccati differential equation yields good match for first diagonal element of covariance matrix.

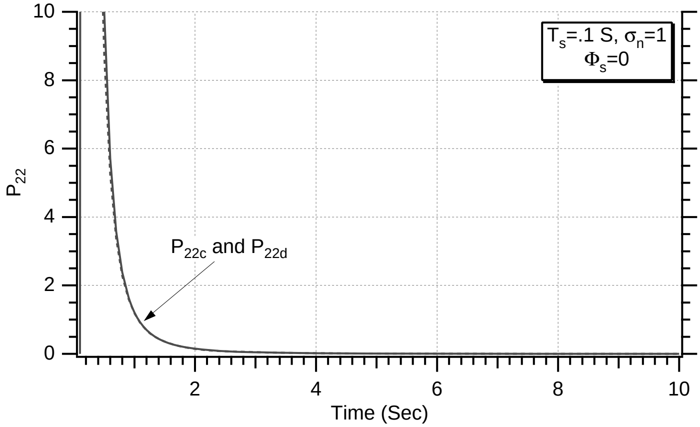

Fig. 6.6 Integrating two-state nonlinear matrix Riccati differential equation yields good match for second diagonal element of covariance matrix.

where $\boldsymbol{K}$ is a three-dimensional column vector.

We have already shown that a discrete polynomial Kalman filter without process noise and infinite initial covariance matrix is equivalent to a recursive least-squares filter. In Chapter 3 we have demonstrated that for the second-order system the three recursive least-squares gains are given by

$$K_{1_k} = \frac{3(3k^2 - 3k + 2)}{k(k+1)(k+2)} \qquad k = 1, 2, \ldots, n$$

$$K_{2_k} = \frac{18(2k-1)}{k(k+1)(k+2)T_s}$$

$$K_{3_k} = \frac{60}{k(k+1)(k+2)T_s^2}$$

and that the three diagonal terms of the covariance matrix are given by

$$P_{11_k} = \frac{3(3k^2 - 3k + 2)\sigma_n^2}{k(k+1)(k+2)}$$

$$P_{22_k} = \frac{12(16k^2 - 30k + 11)\sigma_n^2}{k(k^2-1)(k^2-4)T_s^2}$$

$$P_{33_k} = \frac{720\sigma_n^2}{k(k^2-1)(k^2-4)T_s^4}$$

A simulation was written to integrate the preceding second-order nonlinear continuous matrix differential equation, and the resultant program is shown in Listing 6.3. The process noise in the simulation is initially set to zero (i.e., PHIS = 0). The simulation of Listing 6.3 uses the second-order Runge–Kutta numerical integration technique of Chapter 1 and double-precision arithmetic to solve the nonlinear matrix differential equation. This program also had numerical difficulties when the diagonal elements of the initial covariance matrix were set to infinity, and so the diagonal elements of the initial covariance matrix were each set to 100. Again, we have already shown in Chapter 4 that the solution for the steady-state covariance matrix was virtually independent of initial condition. Therefore, we again hypothesize that choosing an initial covariance matrix to avoid numerical problems should not influence the final result. The integration step size of Listing 6.3 was set to 0.001 s. Again, using smaller values of the integration step size did not seem to change the answers significantly but did dramatically increase the computer run time.

The nominal case of Listing 6.3 was run, and the three Kalman gains and their discrete equivalents appear in Figs. 6.7–6.9. From these figures we can see that, for the case considered, the match between the continuous and discrete Kalman gains is close. We can again conclude that we are integrating the nonlinear covariance matrix differential equation correctly because the theoretical relationships have been verified. Again, for small sampling times we have verified that the continuous Kalman gains of the three-state system are simply the discrete Kalman gains divided by the sampling time or

$$K_{1c} = \frac{K_{1d}}{T_s}$$

$$K_{2c} = \frac{K_{2d}}{T_s}$$

$$K_{3c} = \frac{K_{3d}}{T_s}$$

Similarly Figs. 6.10–6.12 show that the covariance matrix diagonal terms obtained by integration of the continuous matrix Riccati differential equation match the formulas from the recursive least-squares filter. Therefore, we have confidence that for the second-order polynomial Kalman filter there is also a good match between the continuous and discrete systems. In addition we have verified that the continuous and discrete covariances are equal or

$$P_{11c} = P_{11d}$$

$$P_{22c} = P_{22d}$$

$$P_{33c} = P_{33d}$$

Again, one would also get a similar excellent match between the continuous Riccati equations and discrete Riccati equations when there is process noise for the second-order system. As was mentioned before, the gains and covariances will not go to zero as more measurements are taken when process noise is present.

Listing 6.3 Integrating three-state covariance nonlinear Riccati differential equation

```
      IMPLICIT REAL*8(A-H)
      IMPLICIT REAL*8(O-Z)
      REAL*8 F(3,3),P(3,3),Q(3,3),POLD(3,3),HP(1,3)
      REAL*8 PD(3,3)
      REAL*8 HMAT(1,3),HT(3,1),FP(3,3),PFT(3,3),PHT(3,1),K(3,1)
      REAL*8 PHTHP(3,3),PHTHPR(3,3),PFTFP(3,3),PFTFPQ(3,3)
      INTEGER ORDER
      OPEN(1,STATUS=‘UNKNOWN’,FILE=‘DATFIL’)
      ORDER=3
      T=0.
      S=0.
      H=.001
      TS=.1
      TF=10.
      PHIS=0.
      XJ=1.
      DO 14 I=1,ORDER
      DO 14 J=1,ORDER
      F(I,J)=0.
      P(I,J)=0.
      Q(I,J)=0.
14    CONTINUE
      DO 11 I=1,ORDER
      HMAT(1,I)=0.
      HT(I,1)=0.
11    CONTINUE
      F(1,2)=1.
      F(2,3)=1.
      Q(3,3)=PHIS
      HMAT(1,1)=1.
      HT(1,1)=1.
      SIGN2=1.**2
      PHIN=SIGN2*TS
      P(1,1)=100.
      P(2,2)=100.
      P(3,3)=100.
      WHILE(T<=TF)
              DO 20 I=1,ORDER
              DO 20 J=1,ORDER
                      POLD(I,J)=P(I,J)
20            CONTINUE
              CALL MATMUL(F,ORDER,ORDER,P,ORDER,ORDER,FP)
              CALL MATTRN(FP,ORDER,ORDER,PFT)
              CALL MATMUL(P,ORDER,ORDER,HT,ORDER,1,PHT)
              CALL MATMUL(HMAT,1,ORDER,P,ORDER,ORDER,HP)
              CALL MATMUL(PHT,ORDER,1,HP,1,ORDER,PHTHP)
              DO 12 I=1,ORDER
              DO 12 J=1,ORDER
```

(*continued*)

Listing 6.3 (*Continued*)

```
                 PHTHPR(I,J)=PHTHP(I,J)/PHIN
12.      CONTINUE
         CALL  MATADD(PFT,ORDER,ORDER,FP,PFTFP)
         CALL  MATADD(PFTFP,ORDER,ORDER,Q,PFTFPQ)
         CALL  MATSUB(PFTFPQ,ORDER,ORDER,PHTHPR,PD)
         DO  13  I=1,ORDER
                 K(I,1)=PHT(I,1)/PHIN
13       CONTINUE
         DO  50  I=1,ORDER
         DO  50  J=1,ORDER
                 P(I,J)=P(I,J)+H*PD(I,J)
50       CONTINUE
         T=T+H
         CALL  MATMUL(F,ORDER,ORDER,P,ORDER,ORDER,FP)
         CALL  MATTRN(FP,ORDER,ORDER,PFT)
         CALL  MATMUL(P,ORDER,ORDER,HT,ORDER,1,PHT)
         CALL  MATMUL(HMAT,1,ORDER,P,ORDER,ORDER,HP)
         CALL  MATMUL(PHT,ORDER,1,HP,1,ORDER,PHTHP)
         DO  15  I=1,ORDER
         DO  15  J=1,ORDER
                 PHTHPR(I,J)=PHTHP(I,J)/PHIN
15       CONTINUE
         CALL  MATADD(PFT,ORDER,ORDER,FP,PFTFP)
         CALL  MATADD(PFTFP,ORDER,ORDER,Q,PFTFPQ)
         CALL  MATSUB(PFTFPQ,ORDER,ORDER,PHTHPR,PD)
         DO  16  I=1,ORDER
                 K(I,1)=PHT(I,1)/PHIN
16       CONTINUE
         DO  60  I=1,ORDER
         DO  60  J=1,ORDER
                 P(I,J)=.5*(POLD(I,J)+P(I,J)+H*PD(I,J))
60       CONTINUE
         S=S+H
         IF(S>=(TS-.00001))THEN
                S=0.
                XK1=3.*(3*XJ*XJ-3.*XJ+2.)/(XJ*(XJ+1)*(XJ+2))
                XK2=18.*(2.*XJ-1.)/(XJ*(XJ+1)*(XJ+2)*TS)
                XK3=60./(XJ*(XJ+1)*(XJ+2)*TS*TS)
                P11DISC=3*(3*XJ*XJ-3*XJ+2)*SIGN2/(XJ*(XJ+1)*
 1                     (XJ+2))
                IF(XJ.EQ.1.OR.XJ.EQ.2)THEN
                       P22DISC=0.
                       P33DISC=0.
                ELSE
                       P22DISC=12*(16*XJ*XJ-30*XJ+11)*SIGN2/
 1                            (XJ*(XJ*XJ-1)*(XJ*XJ-2)*TS*TS)
                       P33DISC=720*SIGN2/(XJ*(XJ*XJ-1)*(XJ*XJ
 1                            -2)*TS**4)
                ENDIF
```

(*continued*)

Listing 6.2 (*Continued*)

```
                  WRITE(9,*)T,K(1,1)*TS,XK1,K(2,1)*TS,XK2,
 1                        K(3,1)*TS,XK3,P(1,1),P11DISC,P(2,2),
 2                        P22DISC,P(3,3),P33DISC
                  WRITE(1,*)T,K(1,1)*TS,XK1,K(2,1)*TS,XK2,
 1                        K(3,1)*TS,XK3,P(1,1),P11DISC,P(2,2),
 2                        P22DISC,P(3,3),P33DISC

                  XJ=XJ+1.
         ENDIF
   END DO
   PAUSE
   CLOSE(1)
   END

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
```

Transfer Function for Zeroth-Order Filter

We saw in Chapter 5 that when process noise was present the gain and covariance matrix approached steady state. We can derive this steady-state equation for the continuous Kalman filter by noting that in the covariance nonlinear differential equation steady state is reached when the covariance no longer changes. For the zeroth-order continuous polynomial Kalman filter this means that the steady-state solution can be found by setting the derivative of the covariance to zero or

$$\dot{P} = \frac{-P^2}{\Phi_n} + \Phi_s = 0$$

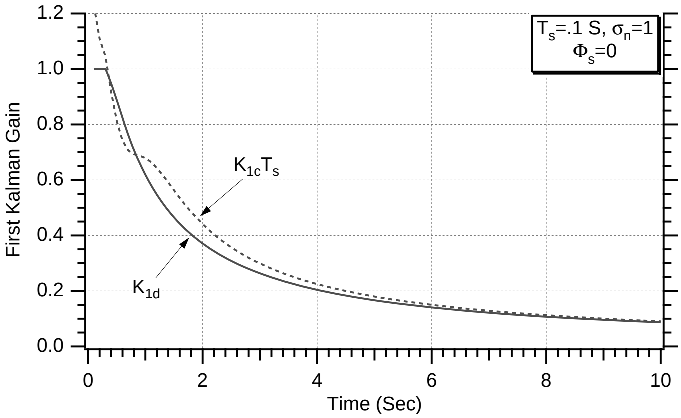

Fig. 6.7 Integrating three-state nonlinear matrix Riccati differential equation yields good match with formula for first gain.

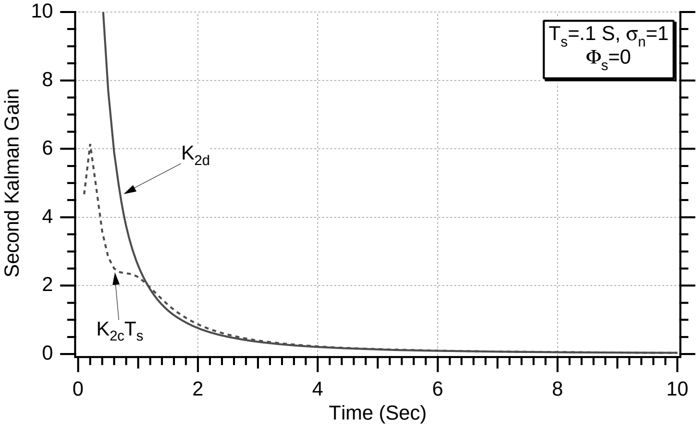

Fig. 6.8 Integrating three-state nonlinear matrix Riccati differential equation yields good match with formula for second gain.

The preceding equation can be solved algebraically to yield the covariance in terms of the measurement and process noise spectral densities or

$$P = (\Phi_s \Phi_n)^{1/2}$$

Because the continuous Kalman gain is computed from the covariance, we can find the closed-form solution for the gain to be given by

$$K = \frac{P}{\Phi_n} = \frac{(\Phi_s \Phi_n)^{1/2}}{\Phi_n}$$

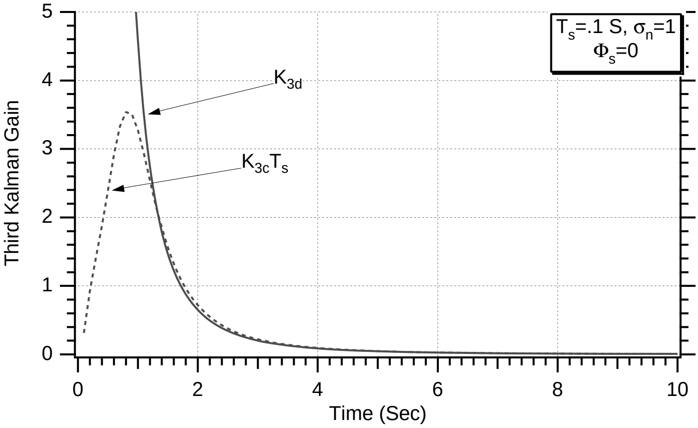

Fig. 6.9 Integrating three-state nonlinear matrix Riccati differential equation yields good match with formula for third gain.

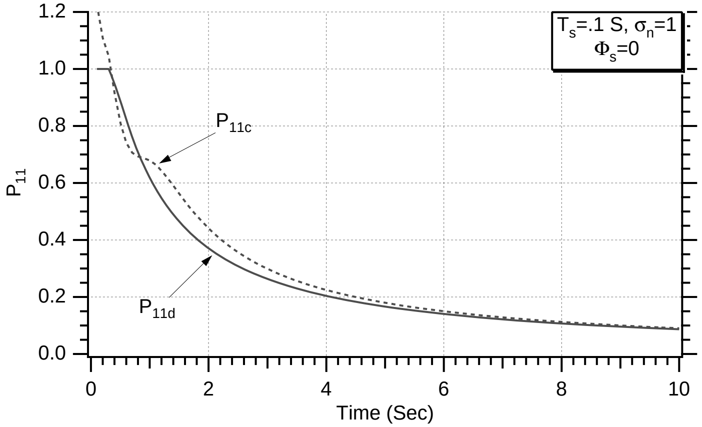

Fig. 6.10 Integrating three-state nonlinear matrix Riccati differential equation yields good match for first diagonal element of covariance matrix.

or

$$K = \left(\frac{\Phi_s}{\Phi_n}\right)^{1/2}$$

Thus, the continuous steady-state Kalman gain only depends on the ratio of the process and measurement noise spectral densities for the zeroth-order continuous polynomial Kalman filter. The interpretation of the preceding formula follows the same logic that was already described. The Kalman filter gain increases with increasing uncertainty of the real world (i.e., Φ_s increases), and new measure-

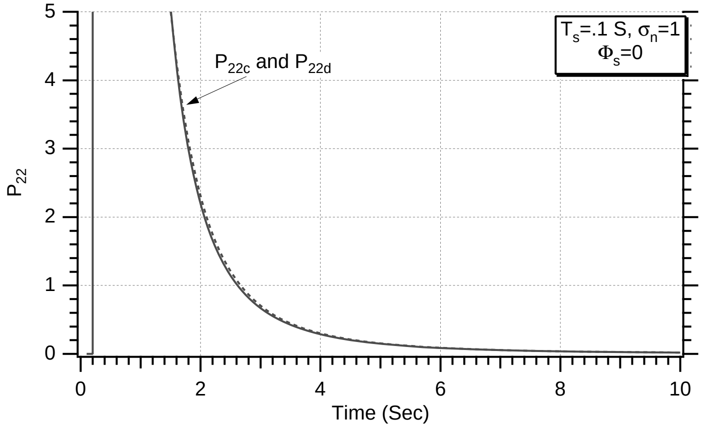

Fig. 6.11 Integrating three-state nonlinear matrix Riccati differential equation yields good match for second diagonal element of covariance matrix.

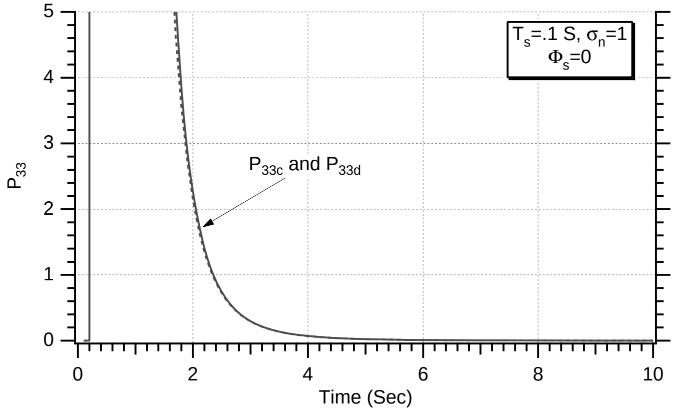

Fig. 6.12 Integrating three-state nonlinear matrix Riccati differential equation yields good match for third diagonal element of covariance matrix.

ments are weighted more heavily. The gain decreases with more uncertainty in the measurements (i.e., Φ_n increases) or with the belief that our model of the real world is near perfect (i.e., Φ_s decreases), and therefore we pay less attention to new measurements.

To check the preceding steady-state expressions, Listing 6.1 was rerun with process noise (i.e., $\Phi_s = 10$ or PHIS = 10 in Listing 6.1). We can see from Figs. 6.13 and 6.14 that after an initial transient period numerically integrating the nonlinear covariance differential equation yields exactly the same results as the steady-state formulas for both the covariance and Kalman gain. Thus, we can conclude that the steady-state formulas for both the gain and covariance for the zeroth-order continuous polynomial Kalman filter are correct.

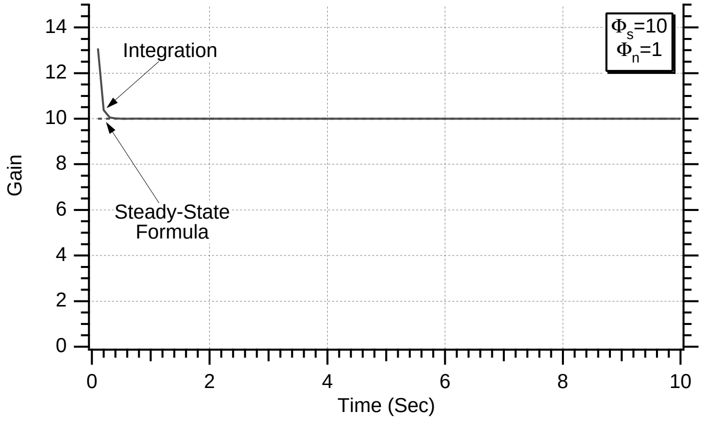

Fig. 6.13 Steady-state formula accurately predicts Kalman gain for zeroth-order continuous polynomial Kalman filter.

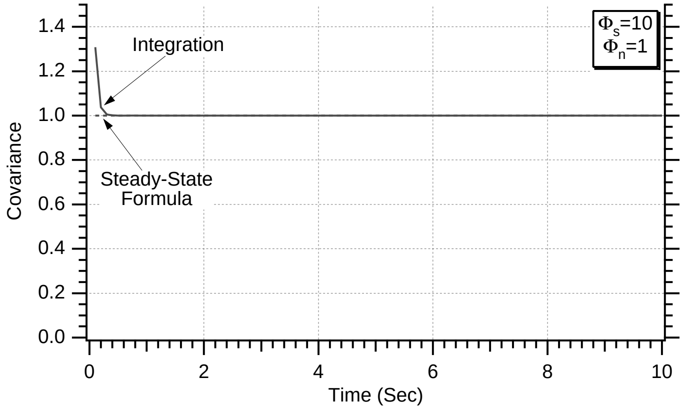

Fig. 6.14 Steady-state formula accurately predicts covariance for zeroth-order continuous polynomial Kalman filter.

Now that we have confidence in the steady-state gain calculation for the zeroth-order continuous polynomial Kalman filter, let us see if we can develop a transfer function for the filter. For our problem there are no deterministic inputs (i.e., no known deterministic or control vector), and so the zeroth-order continuous polynomial Kalman filter can be found from the Kalman filtering differential equation

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{F}\hat{\boldsymbol{x}} + \boldsymbol{K}(\boldsymbol{z} - \boldsymbol{H}\hat{\boldsymbol{x}})$$

to yield

$$\dot{\hat{x}} = K(x^* - \hat{x})$$

where x^* is the measurement. We can convert the preceding scalar differential equation to Laplace transform notation and get

$$s\hat{x} = K(x^* - \hat{x})$$

After some algebraic manipulations we get the Kalman filter transfer function relating the estimate to the measurement to be

$$\frac{\hat{x}}{x^*} = \frac{K}{s + K}$$

If we define a natural frequency ω_0 to be the square root of the ratio of the process and measurement noise spectral densities or

$$\omega_0 = \left(\frac{\Phi_s}{\Phi_n}\right)^{1/2}$$

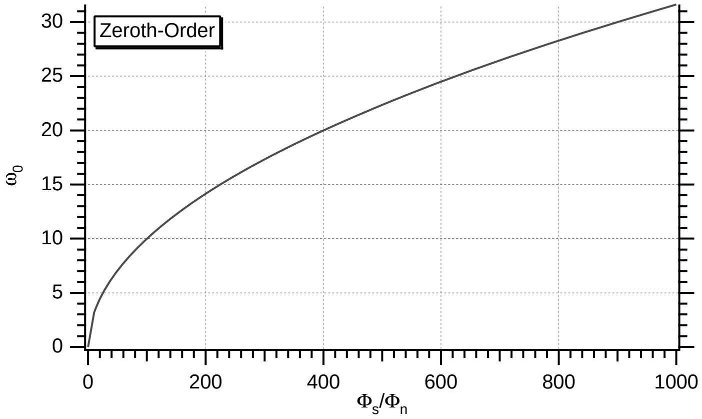

Fig. 6.15 Zeroth-order continuous polynomial Kalman filter's natural frequency increases as the ratio of process to measurement noise increases.

then we can redefine the Kalman gain in terms of the natural frequency

$$K = \left(\frac{\Phi_s}{\Phi_n}\right)^{1/2} = \omega_0$$

and substitute the expression for the steady-state Kalman gain into the filter transfer function and obtain

$$\frac{\hat{x}}{x*} = \frac{1}{1 + \dfrac{s}{\omega_0}}$$

In other words, in the steady state the continuous zeroth-order polynomial Kalman filter is simply a first-order transfer function with a natural frequency that depends on the ratio of the spectral densities of the process noise to the measurement noise. Having more process noise or less measurement noise will tend to increase the bandwidth of the Kalman filter. Figure 6.15 shows more precisely how the bandwidth of the filter varies with the spectral density ratio.

Transfer Function for First-Order Filter

As was the case in the preceding section, we can also derive the steady-state equation for the covariance matrix by noting that in the covariance matrix differential equation steady state is reached when the covariance no longer changes or

$$\begin{bmatrix} \dot{P}_{11} & \dot{P}_{12} \\ \dot{P}_{12} & \dot{P}_{22} \end{bmatrix} = -\begin{bmatrix} P_{11} & P_{12} \\ P_{12} & P_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \Phi_n^{-1} [1 \quad 0] \begin{bmatrix} P_{11} & P_{12} \\ P_{12} & P_{22} \end{bmatrix}$$
$$+ \begin{bmatrix} P_{11} & P_{12} \\ P_{12} & P_{22} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$
$$+ \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{12} & P_{22} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \Phi_s \end{bmatrix} = 0$$

Because the preceding matrix equation is symmetric, it simplifies to the following three scalar equations after multiplying out the terms or

$$0 = 2P_{12} - \frac{P_{11}^2}{\Phi_n}$$
$$0 = P_{22} - \frac{P_{11}P_{12}}{\Phi_n}$$
$$0 = \frac{-P_{12}^2}{\Phi_n} + \Phi_s$$

The preceding equations can be solved algebraically to yield the covariance in terms of the measurement and process noise spectral densities or

$$P_{11} = \sqrt{2}\Phi_s^{1/4}\Phi_n^{3/4}$$
$$P_{22} = \sqrt{2}\Phi_s^{3/4}\Phi_n^{1/4}$$
$$P_{12} = \Phi_s^{1/2}\Phi_n^{1/2}$$

Because the continuous Kalman gains are computed directly from the covariance matrix, they turn out to be

$$K_1 = \frac{P_{11}}{\Phi_n} = \sqrt{2}\left(\frac{\Phi_s}{\Phi_n}\right)^{1/4}$$
$$K_2 = \frac{P_{12}}{\Phi_n} = \left(\frac{\Phi_s}{\Phi_n}\right)^{1/2}$$

To check the preceding steady-state expressions, Listing 6.2 was rerun with process noise (i.e., $\Phi_s = 10$). We can see from Figs. 6.16 and 6.17 that the steady-state gain formulas match the gains obtained by numerical integration after a brief transient period.

We can also see from Figs. 6.18 and 6.19 that the steady-state formulas for the diagonal elements of the covariance matrix match numerical integration experiments. Thus, the formulas for the covariance matrix elements can help us predict the filtering properties of the first-order continuous or discrete polynomial

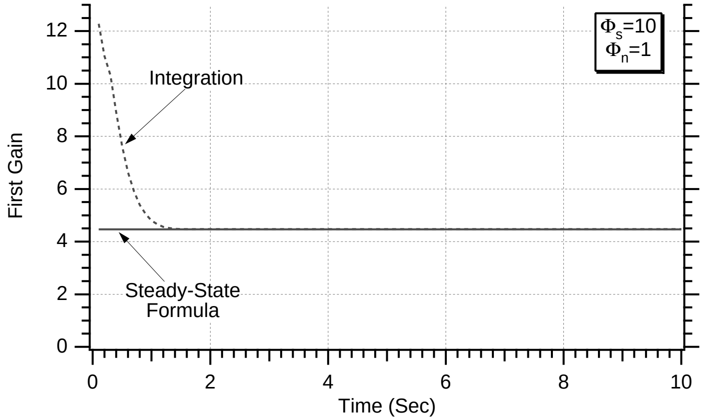

Fig. 6.16 Steady-state gain formula is accurate for first gain in continuous first-order polynomial Kalman filter.

Kalman filters from only knowledge of the process and measurement noise spectral densities.

Now that we have confidence in the steady-state gain calculation for the first-order continuous polynomial Kalman filter let us see if we can develop a transfer function for the filter. For our problem, in which there is no known deterministic input or control vector, we have shown that the polynomial Kalman filter equation in matrix form can be expressed as

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{F}\hat{\boldsymbol{x}} + \boldsymbol{K}(\boldsymbol{z} - \boldsymbol{H}\hat{\boldsymbol{x}})$$

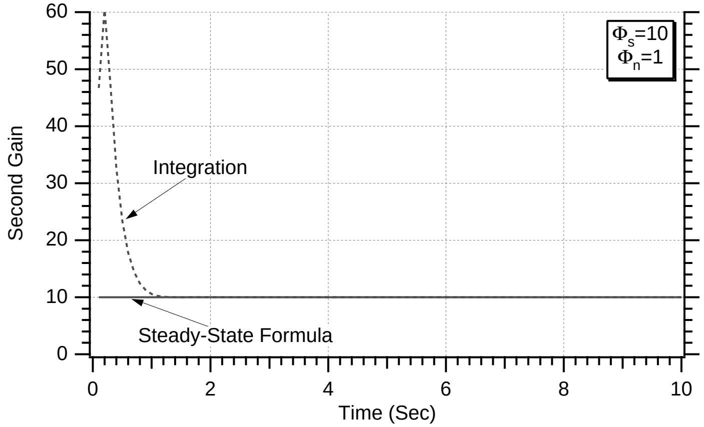

Fig. 6.17 Steady-state gain formula is accurate for second gain in continuous first-order polynomial Kalman filter.

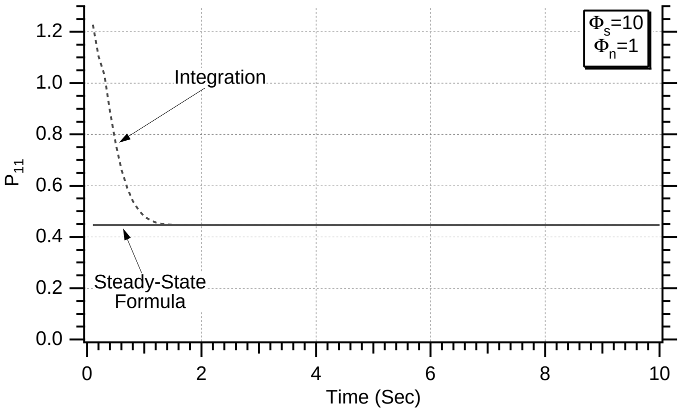

Fig. 6.18 Steady-state formula for first diagonal element of covariance matrix is accurate for continuous first-order polynomial Kalman filter.

Substitution of the appropriate matrices into the preceding equation yields for the first-order system the two scalar differential equations for the Kalman filter state estimates

$$\dot{\hat{x}} = \hat{\dot{x}} + K_1(x^* - \hat{x})$$
$$\dot{\hat{\dot{x}}} = K_2(x^* - \hat{x})$$

where x^* is the measurement. We can convert the preceding two scalar differential equations to Laplace transform notation and get

$$s\hat{x} = \hat{\dot{x}} + K_1(x^* - \hat{x})$$
$$s\hat{\dot{x}} = K_2(x^* - \hat{x})$$

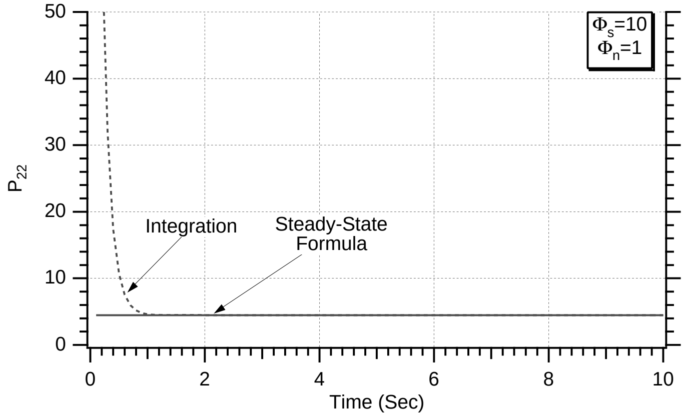

Fig. 6.19 Steady-state formula for second diagonal element of covariance matrix is accurate for continuous first-order polynomial Kalman filter.

After some algebraic manipulation we get the Kalman filter transfer function from the measurement to the state estimate to be

$$\frac{\hat{x}}{x^*} = \frac{K_2 + K_1 s}{s^2 + K_2 + K_1 s}$$

If we define a new natural frequency for the first-order filter (i.e., this natural frequency is different from the natural frequency of zeroth-order filter) as

$$\omega_0 = \left(\frac{\Phi_s}{\Phi_n}\right)^{1/4}$$

then we can express the Kalman gains in terms of the natural frequency as

$$K_1 = \frac{P_{11}}{\Phi_n} = \sqrt{2}\left(\frac{\Phi_s}{\Phi_n}\right)^{1/4} = \sqrt{2}\omega_0$$

$$K_2 = \frac{P_{12}}{\Phi_n} = \left(\frac{\Phi_s}{\Phi_n}\right)^{1/2} = \omega_0^2$$

Substitution of the new gain relationships into the transfer function yields a new transfer function expressed in terms of a natural frequency or

$$\frac{\hat{x}}{x^*} = \frac{1 + \sqrt{2}s/\omega_0}{1 + \sqrt{2}s/\omega_0 + s^2/\omega_0^2}$$

In other words, in the steady state the first-order continuous polynomial Kalman filter is simply a second-order quadratic transfer function with a damping of 0.7 and a natural frequency that depends on the ratio of the spectral densities of the process noise to the measurement noise. Having more process noise or less measurement noise will tend to increase the bandwidth of the

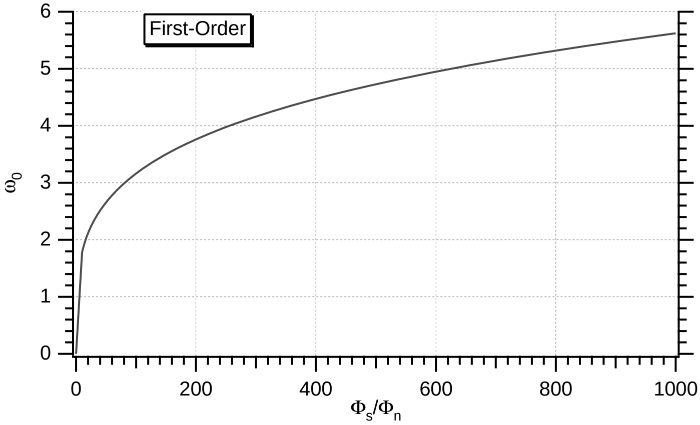

Fig. 6.20 Filter natural frequency increases as the ratio of the process to measurement noise spectral densities increases.

Kalman filter. Figure 6.20 shows more precisely how the bandwidth of the filter varies with the spectral density ratio.

Transfer Function for Second-Order Filter[3]

We can derive the steady-state equation for the second-order filter by noting that in the covariance matrix differential equation steady state is reached when the covariance no longer changes or

$$\begin{bmatrix} \dot{P}_{11} & \dot{P}_{12} & \dot{P}_{13} \\ \dot{P}_{12} & \dot{P}_{22} & \dot{P}_{23} \\ \dot{P}_{13} & \dot{P}_{23} & \dot{P}_{33} \end{bmatrix} = -\begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{12} & P_{22} & P_{23} \\ P_{13} & P_{23} & P_{33} \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\Phi_n^{-1}[1 \quad 0 \quad 0]$$

$$\times\begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{12} & P_{22} & P_{23} \\ P_{13} & P_{23} & P_{33} \end{bmatrix}$$

$$+\begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{12} & P_{22} & P_{23} \\ P_{13} & P_{23} & P_{33} \end{bmatrix}\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\times\begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{12} & P_{22} & P_{23} \\ P_{13} & P_{23} & P_{33} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Phi_s \end{bmatrix} = 0$$

Because the preceding matrix equation is symmetric, the matrix equation simplifies to the following six scalar algebraic equations after multiplying out the terms or

$$P_{11}^2 = 2P_{12}\Phi_n$$

$$P_{12}^2 = 2P_{23}\Phi_n$$

$$P_{13}^2 = \Phi_s\Phi_n$$

$$P_{11}P_{12} = \Phi_n(P_{22} + P_{13})$$

$$P_{11}P_{13} = P_{23}\Phi_n$$

$$P_{12}P_{13} = P_{33}\Phi_n$$

The preceding equations can be solved algebraically to yield the covariance matrix elements in terms of the measurement and process noise spectral densities.

After some algebra we obtain

$$
\begin{aligned}
P_{11} &= 2\Phi_s^{1/6}\Phi_n^{5/6} \\
P_{12} &= 2\Phi_s^{1/3}\Phi_n^{2/3} \\
P_{13} &= \Phi_s^{1/2}\Phi_n^{1/2} \\
P_{22} &= 3\Phi_s^{1/2}\Phi_n^{1/2} \\
P_{23} &= 2\Phi_s^{2/3}\Phi_n^{1/3} \\
P_{33} &= 2\Phi_s^{5/6}\Phi_n^{1/6}
\end{aligned}
$$

The continuous gains are computed from some of the covariance matrix elements. The three Kalman gains turn out to be

$$
\begin{aligned}
K_1 &= \frac{P_{11}}{\Phi_n} = 2\left(\frac{\Phi_s}{\Phi_n}\right)^{1/6} \\
K_2 &= \frac{P_{12}}{\Phi_n} = 2\left(\frac{\Phi_s}{\Phi_n}\right)^{1/3} \\
K_3 &= \frac{P_{13}}{\Phi_n} = \left(\frac{\Phi_s}{\Phi_n}\right)^{1/2}
\end{aligned}
$$

To check the preceding steady-state expressions, Listing 6.3 was rerun with process noise. We can see from Figs. 6.21–6.23 that the steady-state formulas for the Kalman gains are in close agreement with the gains obtained by numerical integration.

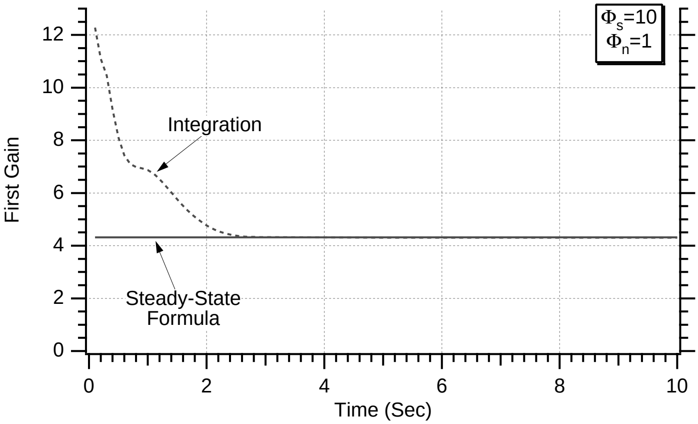

Fig. 6.21 Steady-state gain formula is accurate for first gain in continuous second-order polynomial Kalman filter.

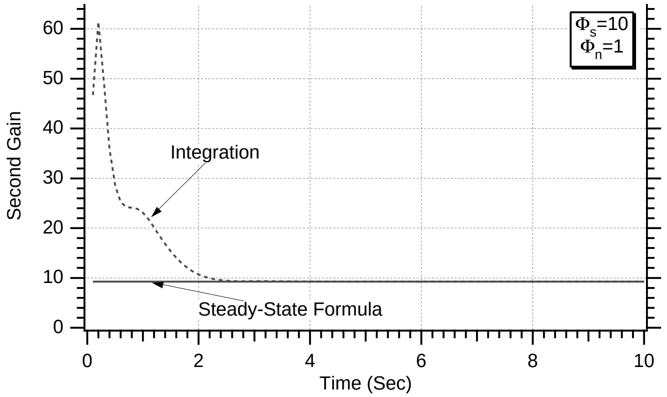

Fig. 6.22 Steady-state gain formula is accurate for second gain in continuous second-order polynomial Kalman filter.

Now that we have confidence in the steady-state gain calculation for the second-order continuous polynomial Kalman filter let us see if we can also develop a transfer function for the filter. For our problem, in which there is no control vector, the polynomial Kalman filter equation in matrix form is given by

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{F}\hat{\boldsymbol{x}} + \boldsymbol{K}(\boldsymbol{z} - \boldsymbol{H}\hat{\boldsymbol{x}})$$

Substitution of the appropriate matrices yields the three scalar differential equations

$$\dot{\hat{x}} = \hat{\dot{x}} + K_1(x^* - \hat{x})$$
$$\dot{\hat{\dot{x}}} = \hat{\ddot{x}} + K_2(x^* - \hat{x})$$
$$\dot{\hat{\ddot{x}}} = K_3(x^* - \hat{x})$$

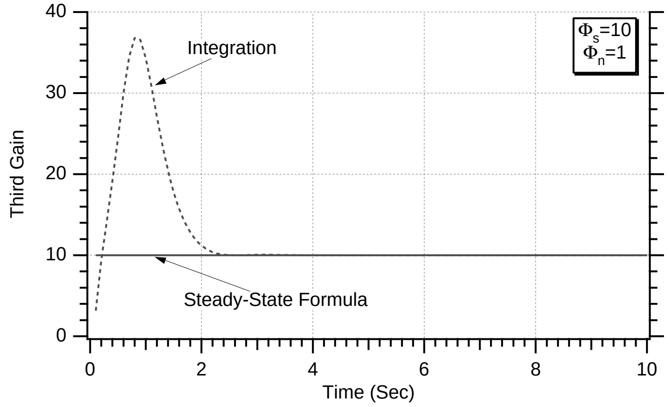

Fig. 6.23 Steady-state gain formula is accurate for third gain in continuous second-order polynomial Kalman filter.

where x^* is the measurement. We can convert the preceding three scalar differential equations to Laplace transform notation and get

$$\begin{aligned} s\hat{x} &= \hat{\dot{x}} + K_1(x^* - \hat{x}) \\ s\hat{\dot{x}} &= \hat{\ddot{x}} + K_2(x^* - \hat{x}) \\ s\hat{\ddot{x}} &= K_3(x^* - \hat{x}) \end{aligned}$$

After some algebraic manipulation we obtain the Kalman filter transfer function from the measurement to the state estimate as

$$\frac{\hat{x}}{x^*} = \frac{K_3 + sK_2 + s^2K_1}{K_3 + sK_2 + s^2K_1 + s^3}$$

If we define a natural frequency for this second-order filter (i.e., different from zeroth-order and first-order natural frequencies) to be

$$\omega_0 = \left(\frac{\Phi_s}{\Phi_n}\right)^{1/6}$$

we can express the Kalman gains in terms of the natural frequency as

$$\begin{aligned} K_1 &= \frac{P_{11}}{\Phi_n} = 2\left(\frac{\Phi_s}{\Phi_n}\right)^{1/6} = 2\omega_0 \\ K_2 &= \frac{P_{12}}{\Phi_n} = 2\left(\frac{\Phi_s}{\Phi_n}\right)^{1/3} = 2\omega_0^2 \\ K_3 &= \frac{P_{13}}{\Phi_n} = \left(\frac{\Phi_s}{\Phi_n}\right)^{1/2} = \omega_0^3 \end{aligned}$$

Substituting the preceding three expressions into the transfer function yields

$$\frac{\hat{x}}{x^*} = \frac{1 + 2s/\omega_0 + 2s^2/\omega_0^2}{1 + 2s/\omega_0 + 2s^2/\omega_0^2 + s^3/\omega_0^3}$$

In other words, in the steady state the continuous second-order polynomial Kalman filter is simply a third-order transfer function whose poles follow a Butterworth[4] *distribution with a natural frequency that depends on the ratio of the spectral densities of the process noise to the measurement noise. Having more process noise or less measurement noise will tend to increase the bandwidth of the Kalman filter.* Figure 6.24 shows more precisely how the bandwidth of the filter varies with the spectral density ratio.

Filter Comparison

It is sometimes easier to understand the properties of the continuous polynomial Kalman filters from a frequency domain point of view. Table 6.1 presents

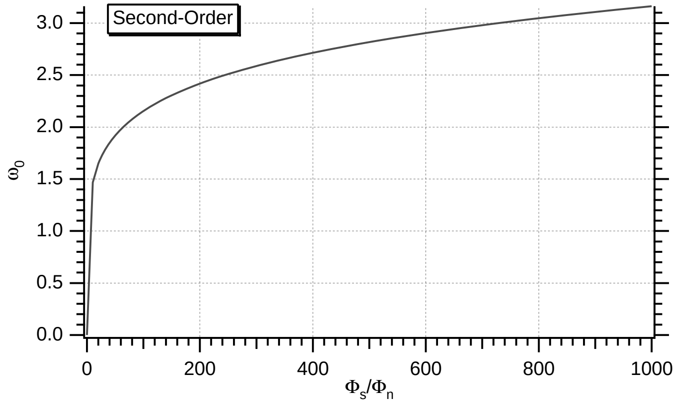

Fig. 6.24 Second-order Kalman filter natural frequency increases with increasing ratio of process to measurement noise spectral density.

the zeroth-, first- and second-order polynomial Kalman filter transfer functions, in the Laplace transform domain, that we just derived in this chapter. The other column of Table 6.1 presents the magnitude of the various filter transfer functions in the complex frequency domain. Recall that the magnitude of a transfer function can be found by replacing s with $j\omega$ and then finding the square root of the real part squared plus the imaginary part squared.

Listing 6.4 presents a program that was written to evaluate the polynomial Kalman filter transfer function magnitudes that are displayed in Table 6.1. We can see that the listing consists of one loop in which the frequency is varied from 1 rad/s to 100 rad/s in steps of 1 rad/s. The magnitudes of each of the filter transfer functions are computed and printed out for each frequency. Nominally the program is set up with a natural frequency of 10 rad/s.

The nominal case of Listing 6.4 was run, and the results for each of the three different filters are displayed in Fig. 6.25. We can see that each of the filters has a magnitude of approximately unity until 10 rad/s. This means that each filter simply passes the measurement, without amplification or attenuation, until the natural frequency (i.e., 10 rad/s in this example). For frequencies higher than the natural frequency, the output is attenuated (i.e., magnitude less than unity). This means that the filter is not passing those frequencies. In other words the filter is behaving as a low-pass filter. Everything gets passed until the natural frequency is reached, and afterwards everything is attenuated. We also can tell from Fig. 6.25 that the higher-order filters have less attenuation after the natural frequency than the lower-order filters. This simply indicates that there is more noise transmission with the higher-order filters (i.e., higher frequencies are less attenuated). We have already seen in the preceding chapters that higher-order filters result in more noise transmission.

Next, the magnitude of the second-order polynomial Kalman filter transfer function was computed for natural frequencies of 5, 10, and 25 rad/s. We can see from Fig. 6.26 that, as expected, the magnitude of the transfer function peaks at

Table 6.1 Transfer functions and magnitudes for different-order polynomial Kalman filters

Name	Laplace transform	Magnitude
Zeroth order	$\dfrac{\hat{x}}{x^*} = \dfrac{1}{1 + s/\omega_0}$	$\left\lvert \dfrac{\hat{x}}{x^*} \right\rvert = \dfrac{1}{\sqrt{1 + (\omega/\omega_0)^2}}$
First order	$\dfrac{\hat{x}}{x^*} = \dfrac{1 + \sqrt{2}s/\omega_0}{1 + \sqrt{2}s/\omega_0 + s^2/\omega_0^2}$	$\left\lvert \dfrac{\hat{x}}{x^*} \right\rvert = \sqrt{\dfrac{1 + (\sqrt{2}\omega/\omega_0)^2}{(1 - \omega^2/\omega_0^2)^2 + (\sqrt{2}\omega/\omega_0)^2}}$
Second order	$\dfrac{\hat{x}}{x^*} = \dfrac{1 + 2s/\omega_0 + 2s^2/\omega_0^2}{1 + 2s/\omega_0 + 2s^2/\omega_0^2 + s^3/\omega_0^3}$	$\left\lvert \dfrac{\hat{x}}{x^*} \right\rvert = \sqrt{\dfrac{(1 - 2\omega^2/\omega_0^2)^2 + (2\omega/\omega_0)^2}{(1 - 2\omega^2/\omega_0^2)^2 + (2\omega/\omega_0 - \omega^3/\omega_0^3)^2}}$

Listing 6.4 Program to calculate magnitudes of Kalman-filter transfer functions

```
   IMPLICIT REAL*8 (A-H)
   IMPLICIT REAL*8 (O-Z)
   OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
   W0=10.
   DO 10 W=1.,100.
   XMAG1=1./SQRT(1.+(W/W0)**2)
   TOP1=1.+2.*(W/W0)**2
   BOT1=(1.-(W*W/(W0*W0)))**2+2.*(W/W0)**2
   XMAG2=SQRT(TOP1/(BOT1+.00001))
   TOP2=(1.-2.*W*W/(W0*W0))**2+(2.*W/W0)**2
   TEMP1=(1.-2.*W*W/(W0*W0))**2
   TEMP2=(2.*W/W0-(W/W0)**3)**2
   XMAG3=SQRT(TOP2/(TEMP1+TEMP2+.00001))
   WRITE(9,*)W,XMAG1,XMAG2,XMAG3
   WRITE(1,*)W,XMAG1,XMAG2,XMAG3
10 CONTINUE
   CLOSE(1)
   PAUSE
   END
```

the filter natural frequency. If the filter natural frequency is high, the filter will have a wider bandwidth. This means that the filter will be more responsive, but it also means that the filter will pass more noise. Again, we have seen in preceding chapters that when we increase or add process noise the filter is able to better track the signal (i.e., less lag or higher bandwidth) but has more noise transmission (i.e., noisier estimates).

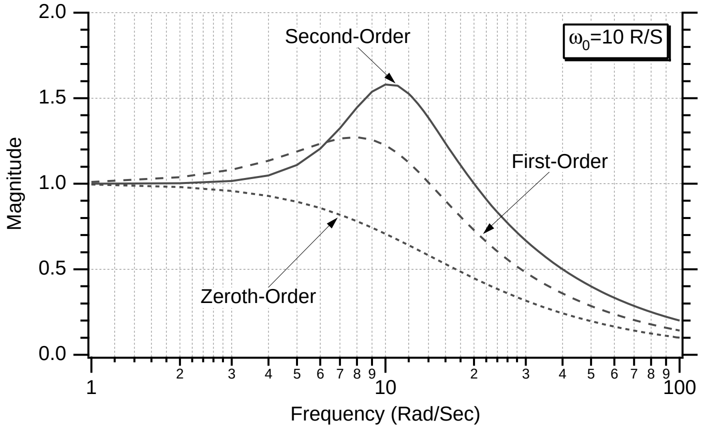

Fig. 6.25 Higher-order filters have less attenuation after filter natural frequency.

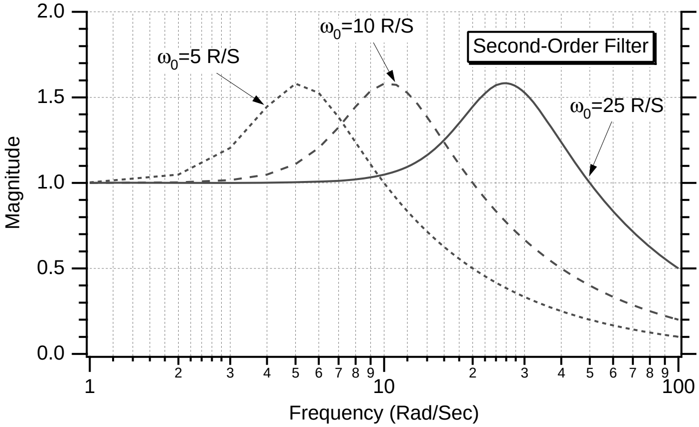

Fig. 6.26 Increasing filter natural frequency increases filter bandwidth.

Summary

In this chapter we have investigated different-order continuous polynomial Kalman filters. Although these filters are generally not implemented, they can be useful for helping us to understand the properties of the more popular discrete Kalman filters. Steady-state solutions for the gains and covariances of the various-order polynomial Kalman filters were obtained. From these solutions we also were able to derive transfer functions for the Kalman filter. We saw from the resultant transfer functions that the polynomial Kalman filter acted as a low-pass filter. The bandwidth of the various-order polynomial Kalman filters depended on the ratio of the spectral density of the process noise to the spectral density of the measurement noise. Increasing the spectral density ratio, by either increasing the process noise or decreasing the measurement noise, tended to increase the bandwidth of the Kalman filter. All results were derived analytically and confirmed by simulation.

References

[1] Kalman, R. E. and Bucy, R. S., "New Results in Linear Filtering and Prediction Theory," *Journal of Basic Engineering*, Vol. 83, No. 3, 1961, pp. 95–108.

[2] Gelb., A., *Applied Optimal Estimation*, Massachusetts Inst. of Technology Press, Cambridge, MA, 1974, pp. 119–127.

[3] Nesline, F. W. and Zarchan, P., "A New Look at Classical Versus Modern Homing Guidance," *Journal of Guidance and Control*, Vol. 4, No. 1, 1981, pp. 78–85.

[4] Karni, S., *Network Theory: Analysis and Synthesis*, Allyn and Bacon, London, 1966, pp. 344–349.