Continuous Polynomial Kalman Filter

Continuous Polynomial Kalman Filter Overview

- Theoretical equations
- Comparing continuous and discrete Kalman gains and covariances
    - Zeroth, first and second-order polynomial Kalman filters
- Steady-state approximations
    - Formulas for steady-state gains and covariances
    - Transfer functions for zeroth, first and second-order polynomial Kalman filters
- Filter comparisons

Theoretical Equations For Continuous Kalman Filter

Model of real world

$$\dot{x} = Fx + Gu + w$$

Process noise matrix

$$Q = E[ww^T]$$

Measurements are linearly related to states

$$z = Hx + v$$

Measurement noise matrix

$$R = E[vv^T]$$

Continuous Kalman filter

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

Gains obtained from continuous Riccati equations

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

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

Comparing Continuous and Discrete Kalman Gains and Covariances

Zeroth-Order Filter

Zeroth-Order Continuous Polynomial Kalman Filter

Model of real world

$$\dot{x} = u_s \longrightarrow \mathbf{F} = 0$$

Process noise matrix is scalar

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

Measurement equation

$$x^* = x + v_n \longrightarrow \mathbf{H} = 1$$

Measurement noise matrix is scalar

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

Riccati equation simplifies to

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

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

Kalman gain obtained from

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

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

Comparing Zeroth-Order Polynomial Kalman Filter Gain to Recursive Least Squares Filter Gain

Recall that zeroth-order recursive least squares filter gain is

$$K_k = \frac{1}{k} \quad k=1,2,...,n$$

While variance of error in estimate is

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

We have just shown that variance of error on estimate for Kalman filter is

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

The two filters should be equivalent if the Kalman filter has zero process noise

The spectral density of continuous noise is related to the variance of discrete noise according to

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

As the sampling time gets smaller continuous and discrete gains related

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

Integrating One-State Covariance Nonlinear Riccati Differential Equation With MATLAB-1

```
ORDER=1;
T=0.;
S=0.;
H=.001;
TS=.1;
TF=10.;
PHIS=0.;
XJ=1.;
F=[0];
P=[100];
Q=[PHIS];
HMAT=[1];
HT=HMAT';
SIGN2=1.^2;
PHIN=SIGN2*TS;
count=0;
while T<=TF
        S=S+H;
        POLD=P;
        FP=F*P;
        PFT=FP';
        PHT=P*HT;
        HP=HMAT*P;
        PHTHP=PHT*HP;
        PHTHPR=(1./PHIN)*PHTHP;
        PFTFP=PFT+FP;
        PFTFPQ=PFTFP+Q;
        PD=PFTFPQ-PHTHPR;
        K=(1./PHIN)*PHT;
        HPD=(H)*PD;
        P=P+HPD;
```

- H=.001; ← **Made small to get accurate answers**
- PHIS=0.; ← **Set to zero for comparison with least squares**
- P=[100]; ← **If made too large have numerical difficulties**
- SIGN2=1.^2; PHIN=SIGN2*TS; ← **Relationship between continuous and discrete noise**
- POLD=P; through K=(1./PHIN)*PHT; — **Matrix Riccati differential equation**
- S=S+H; through P=P+HPD; — **Second-order Runge-Kutta numerical integration**

Integrating One-State Covariance Nonlinear Riccati Differential Equation With MATLAB-2

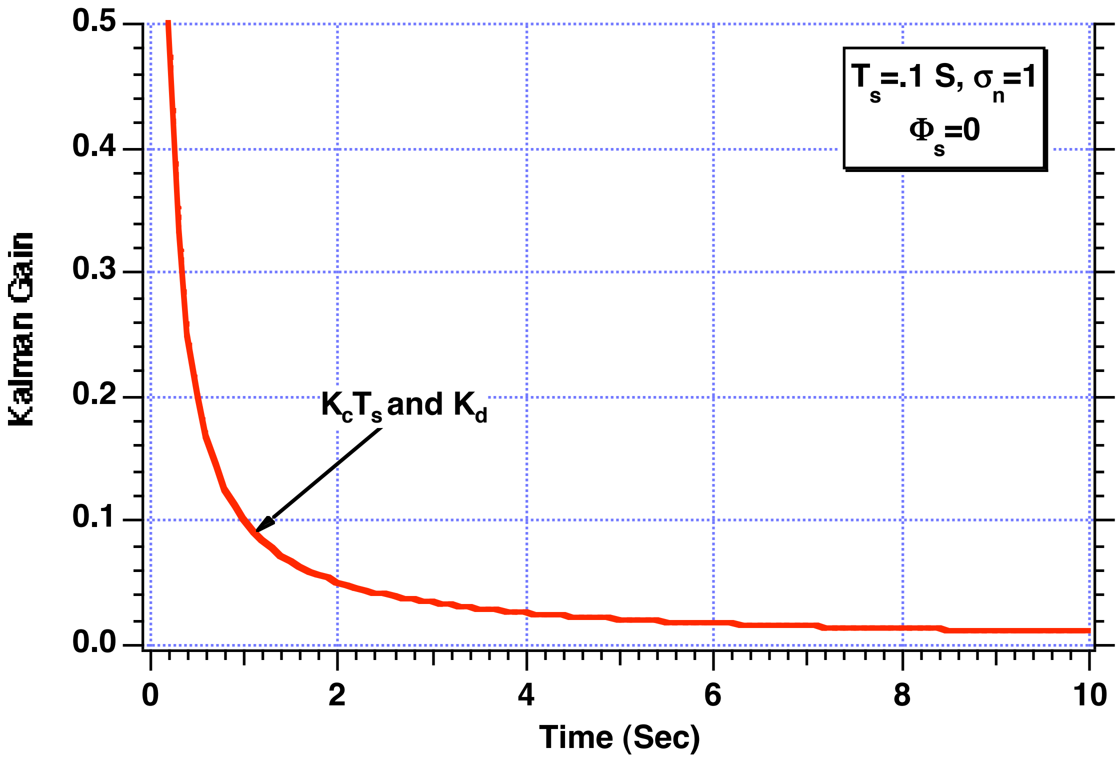
```
        T=T+H;
        FP=F*P;
        PFT=FP';
        PHT=P*HT;
        HP=HMAT*P;
        PHTHP=PHT*HP;
        PHTHPR=(1./PHIN)*PHTHP;
        PFTFP=PFT+FP;
        PFTFPQ=PFTFP+Q;
        PD=PFTFPQ-PHTHPR;
        K=(1./PHIN)*PHT;
        HPD=(H)*PD;
        PHPD=P+HPD;
        PPHPD=POLD+PHPD;
        P=(.5)*PPHPD;
        if S>=(TS-.00001)
                S=0.;
                XK1=1./XJ;
                PDISC=SIGN2/XJ;
                KTS=K(1,1)*TS;
                count=count+1;
                ArrayT(count)=T;
                ArrayKTS(count)=KTS;
                ArrayXK1(count)=XK1;
                ArrayPDISC(count)=PDISC;
                ArrayP(count)=P;
                XJ=XJ+1.;
        end
end
figure
plot(ArrayT,ArrayKTS,ArrayT,ArrayXK1),grid
xlabel('Time (Sec)')
ylabel('Continuous and Discrete Kalman Gain')
axis([0 10 0 .5])
```

Matrix Riccati differential equation (FP=F*P; through K=(1./PHIN)*PHT;)

Second-order Runge-Kutta numerical integration (T=T+H; through if S>=(TS-.00001))

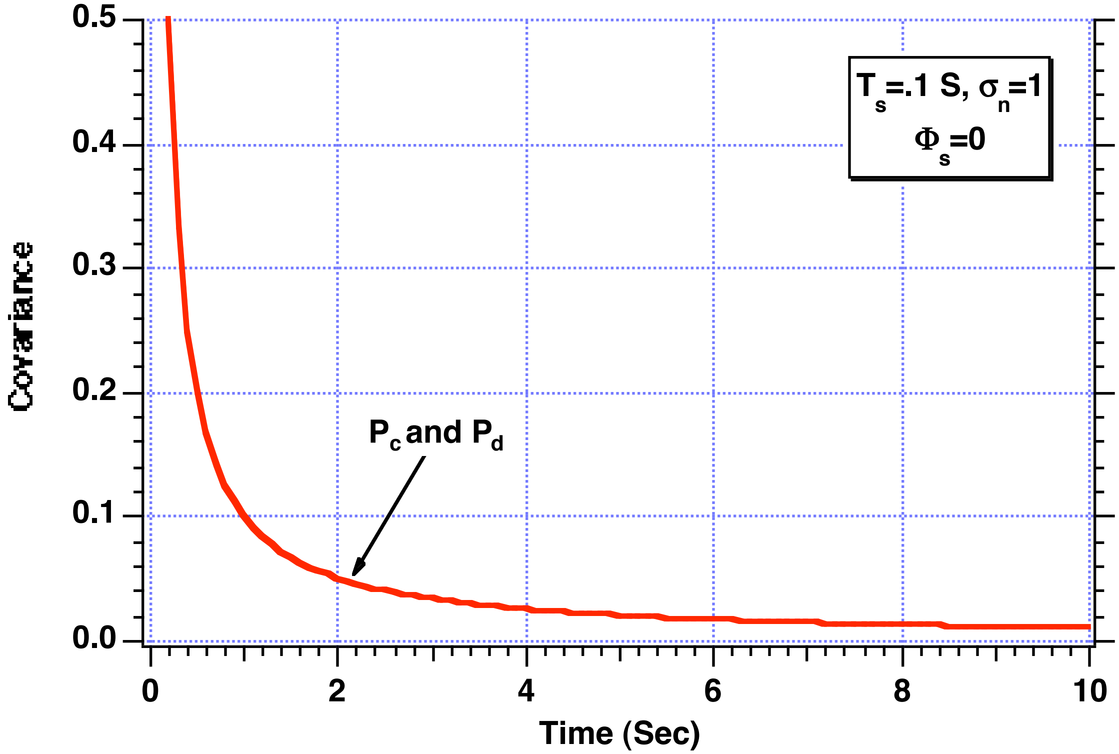
Save data in arrays for plotting and writing to file (count=count+1; through XJ=XJ+1.;)

Continuous and Discrete Kalman Gains are Identical for Zeroth-Order System



$$K_c = \frac{K_d}{T_s} \qquad K_d = \frac{1}{k} \quad k=1,2,...,n$$

Continuous and Discrete Covariances are Identical for Zeroth-Order System



$$P_c = P_d \qquad P_d = \frac{\sigma_n^2}{k}$$

First-Order Filter

First-Order Continuous Polynomial Kalman Filter

Model of real world

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

Process noise matrix

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

Measurement equation

$$x^* = x + v_n \longrightarrow x^* = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \end{bmatrix} + v_n \longrightarrow \mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Measurement noise matrix is scalar

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

Substitute matrices into Riccati equations

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

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

Comparing First-Order Polynomial Kalman Filter Gain to Recursive Least Squares Filter Gain

Recall that first-order recursive least squares filter gains are

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

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

While variance of error in the state estimates are

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

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

The two filters should be equivalent if the Kalman filter has zero process noise

The spectral density of continuous noise is related to the variance of discrete noise according to

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

As the sampling time gets smaller continuous and discrete gains related

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

Integrating Two-State Covariance Nonlinear Riccati Differential Equation With True BASIC-1

```
OPTION NOLET
REM UNSAVE "DATFIL"
OPEN #1:NAME "DATFIL",ACCESS OUTPUT,CREATE NEW, ORGANIZATION TEXT
SET #1: MARGIN 1000
DIM F(2,2),P(2,2),Q(2,2),POLD(2,2),HP(1,2)
DIM PD(2,2)
DIM HMAT(1,2),HT(2,1),FP(2,2),PFT(2,2),PHT(2,1),K(2,1)
DIM PHTHP(2,2),PHTHPR(2,2),PFTFP(2,2),PFTFPQ(2,2),HPD(2,2),PHPD(2,2),PPHPD(2,2)
ORDER=2
T=0.
S=0.
H=.001
TS=.1
TF=10.
PHIS=0.
XJ=1.
MAT F=ZER(ORDER,ORDER)
MAT P=ZER(ORDER,ORDER)
MAT Q=ZER(ORDER,ORDER)
MAT HMAT=ZER(1,ORDER)
MAT HT=ZER(ORDER,1)
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.
DO WHILE T<=TF
        S=S+H
        MAT POLD=P
        MAT FP=F*P
        MAT PFT=TRN(FP)
        MAT PHT=P*HT
        MAT HP=HMAT*P
        MAT PHTHP=PHT*HP
        MAT PHTHPR=(1./PHIN)*PHTHP
        MAT PFTFP=PFT+FP
        MAT PFTFPQ=PFTFP+Q
        MAT PD=PFTFPQ-PHTHPR
        MAT K=(1./PHIN)*PHT
```

Made small to get accurate answers (H=.001)

Set to zero for comparison with least squares (PHIS=0.)

If made too large have numerical difficulties (P(1,1)=100.)

Matrix Riccati differential equation (MAT FP=F*P through MAT K=(1./PHIN)*PHT)

Second-order Runge-Kutta numerical integration (from S=S+H onward)

Integrating Two-State Covariance Nonlinear Riccati Differential Equation With True BASIC-2

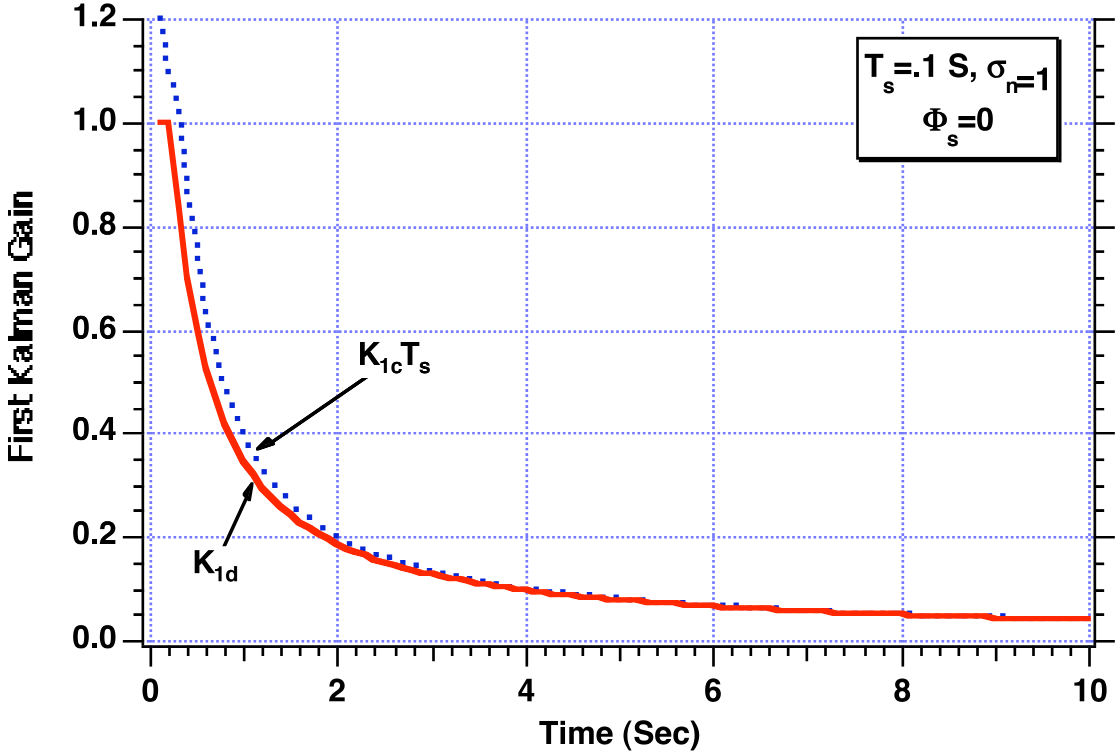
```
            MAT HPD=(H)*PD
            MAT P=P+HPD
            T=T+H
            MAT FP=F*P
            MAT PFT=TRN(FP)
            MAT PHT=P*HT
            MAT HP=HMAT*P
            MAT PHTHP=PHT*HP
            MAT PHTHPR=(1./PHIN)*PHTHP
            MAT PFTFP=PFT+FP
            MAT PFTFPQ=PFTFP+Q
            MAT PD=PFTFPQ-PHTHPR
            MAT K=(1./PHIN)*PHT
            MAT HPD=(H)*PD
            MAT PHPD=P+HPD
            MAT PPHPD=POLD+PHPD
            MAT P=(.5)*PPHPD
            IF S>(TS-.0001) 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=1 THEN
                                    P22DISC=0.
                        ELSE
                                    P22DISC=12*SIGN2/(XJ*(XJ*XJ-1)*TS*TS)
                        END IF
                        PRINT T,K(1,1)*TS,XK1,K(2,1)*TS,XK2
                        PRINT #1:T,K(1,1)*TS,XK1,K(2,1)*TS,XK2,P(1,1),P11DISC,P(2,2),P22DISC
                        XJ=XJ+1.
            END IF
LOOP
CLOSE #1
END
```

Matrix Riccati differential equation (bracketing lines T=T+H through MAT K=(1./PHIN)*PHT)

Second-order Runge-Kutta numerical integration (bracketing lines MAT HPD=(H)*PD through MAT P=(.5)*PPHPD)

Write data to screen and file (bracketing the two PRINT lines)

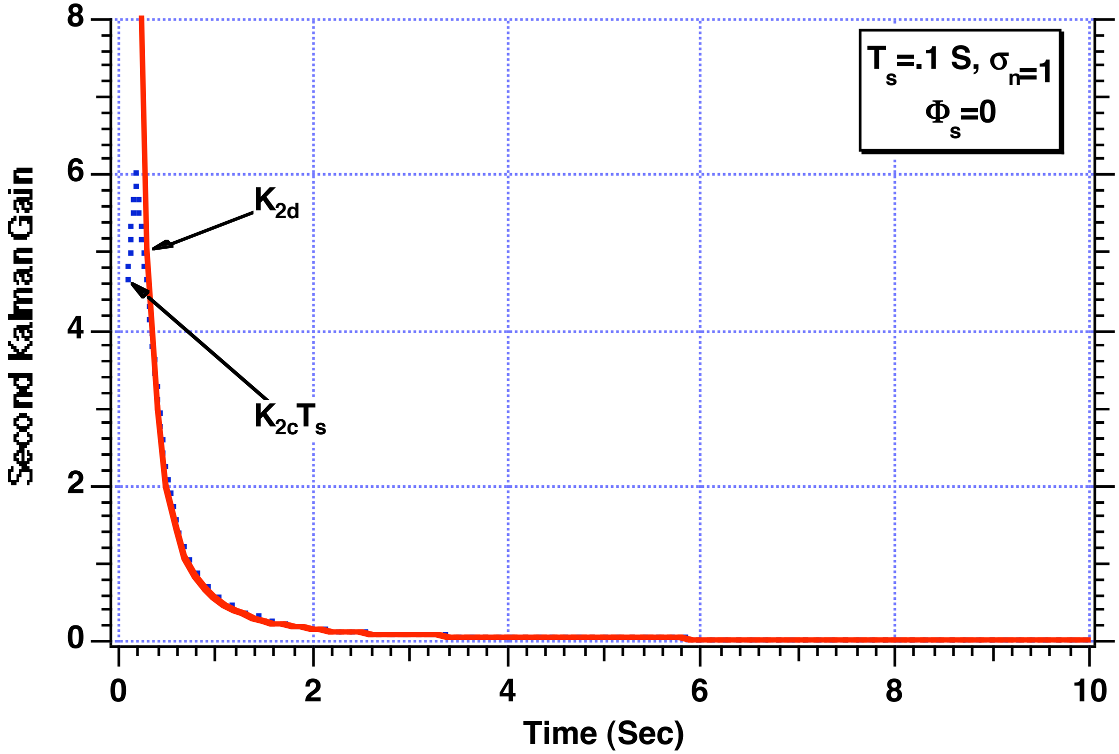
Integrating Two-State Nonlinear Matrix Riccati Differential Equation Yields Good Match With Formula for First Gain



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

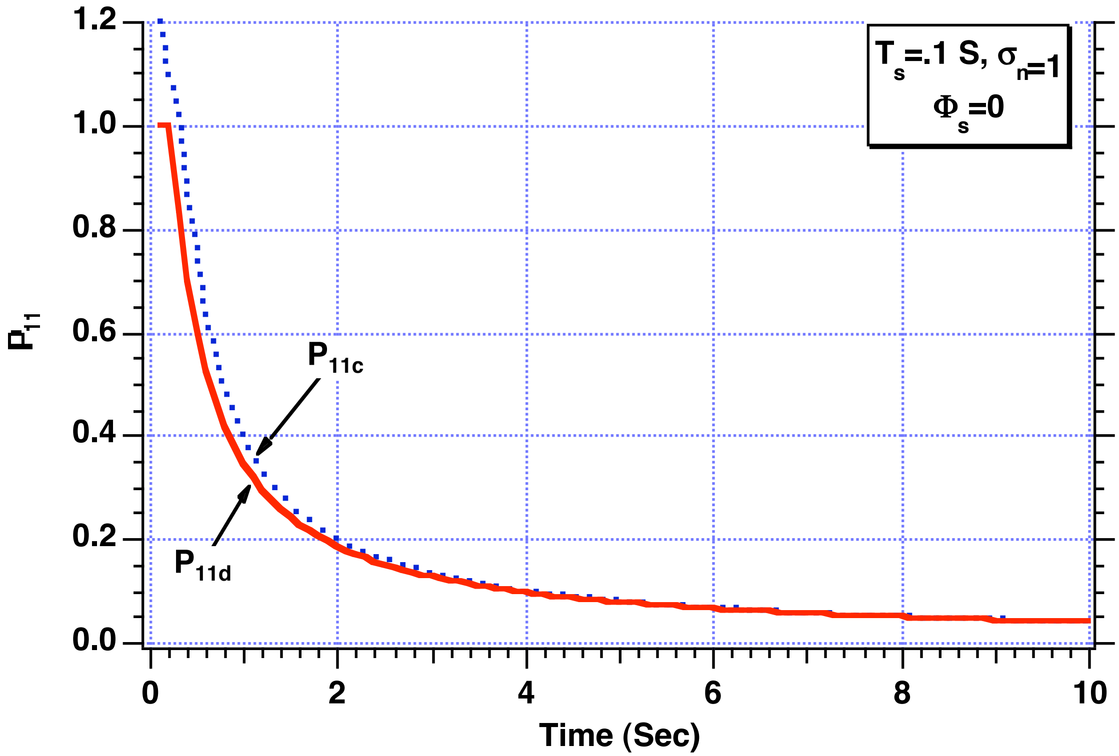
$$K_{1_d} = \frac{2(2k-1)}{k(k+1)} \quad k=1,2,...,n$$

Integrating Two-State Nonlinear Matrix Riccati Differential Equation Yields Good Match With Formula for Second Gain



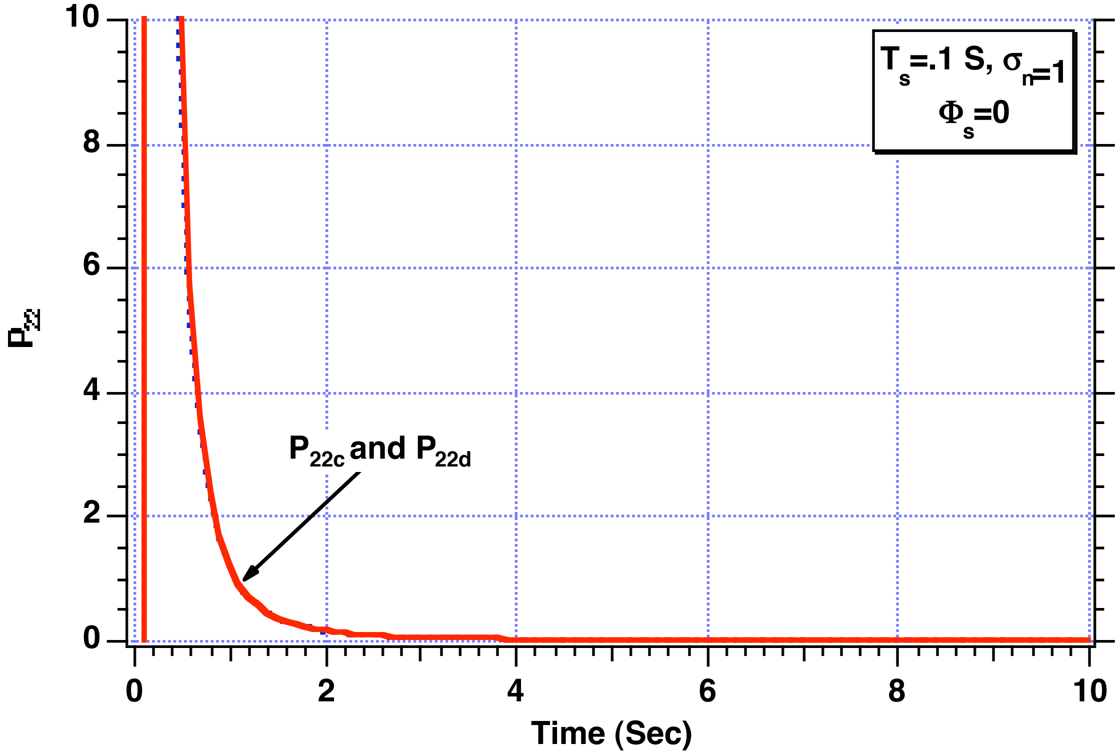
$$K_{2c} = \frac{K_{2d}}{T_s} \qquad K_{2_d} = \frac{6}{k(k+1)T_s}$$

Integrating Two-State Nonlinear Matrix Riccati Differential Equation Yields Good Match for First Diagonal Element of Covariance Matrix



$P_{11c} = P_{11d}$ $\qquad$ $P_{11_d} = \frac{2(2k-1)\sigma_n^2}{k(k+1)}$

Integrating Two-State Nonlinear Matrix Riccati Differential Equation Yields Good Match for Second Diagonal Element of Covariance Matrix



$$P_{22c} = P_{22d} \qquad P_{22_d} = \frac{12\sigma_n^2}{k(k^2-1)T_s^2}$$

Second-Order Filter

Second-Order Continuous Polynomial Kalman Filter

Model of real world

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

Process noise matrix

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

Measurement equation

$$x^* = x + v_n \longrightarrow \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

Measurement noise matrix is scalar

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

Substitute matrices into Riccati equations

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

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

Comparing Second-Order Polynomial Kalman Filter Gain to Recursive Least Squares Filter Gain

Recall that second-order recursive least squares filter gains are

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

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

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

While variance of error in the state estimates are

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

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

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

The two filters should be equivalent if the Kalman filter has zero process noise

The spectral density of continuous noise is related to the variance of discrete noise according to

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

As the sampling time gets smaller continuous and discrete gains related

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

Integrating Three-State Covariance Nonlinear Riccati Differential Equation With FORTRAN-1

```
      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.
```

Made small to get accurate answers (H=.001)

Set to zero for comparison with least squares (PHIS=0.)

If made too large have numerical difficulties (P(2,2)=100.)

Integrating Three-State Covariance Nonlinear Riccati Differential Equation With FORTRAN-2

```
        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)
            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
```

Matrix Riccati differential equation (first bracket, statement 20 CONTINUE through 13 CONTINUE)

Matrix Riccati differential equation (second bracket, T=T+H through 16 CONTINUE)

Integrating Three-State Covariance Nonlinear Riccati Differential Equation With FORTRAN-3

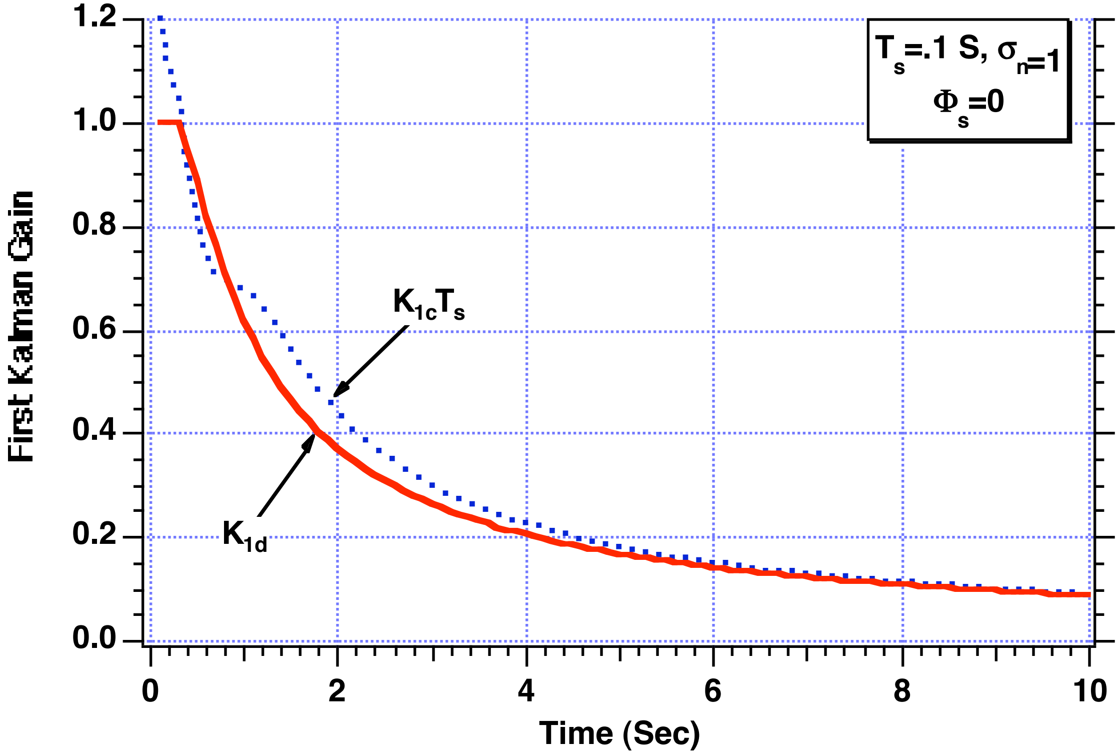
```
          DO 60 I=1,ORDER
          DO 60 J=1,ORDER
               P(I,J)=.5*(POLD(I,J)+P(I,J)+H*PD(I,J))
60        CONTINUE
          S=S+H
          IF(S>=(TS-.00001))THEN
               S=0.
               XK1=3.*(3*XJ*XJ-3.*XJ+2.)/(XJ*(XJ+1)*(XJ+2))
               XK2=18.*(2.*XJ-1.)/(XJ*(XJ+1)*(XJ+2)*TS)
               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
               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
```

Second-order Runge-Kutta numerical integration (annotation for the DO 60 loop)

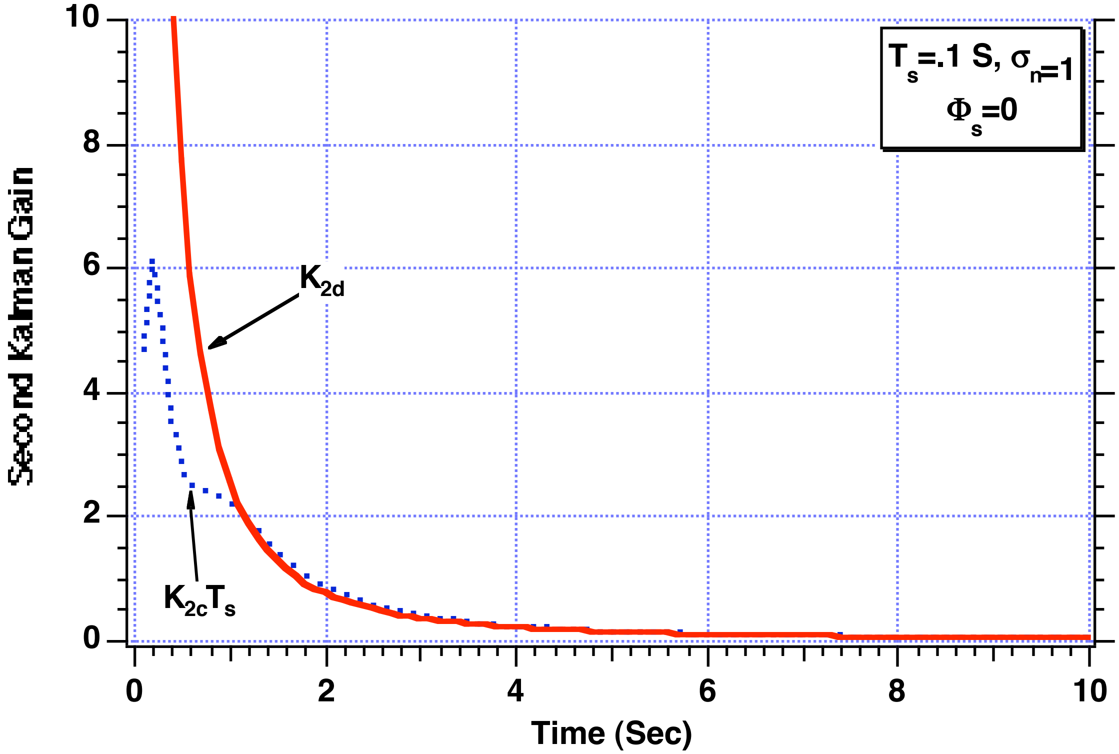
Write data to screen and file (annotation for the WRITE statements)

Integrating Three-State Nonlinear Matrix Riccati Differential Equation Yields Good Match With Formula for First Gain



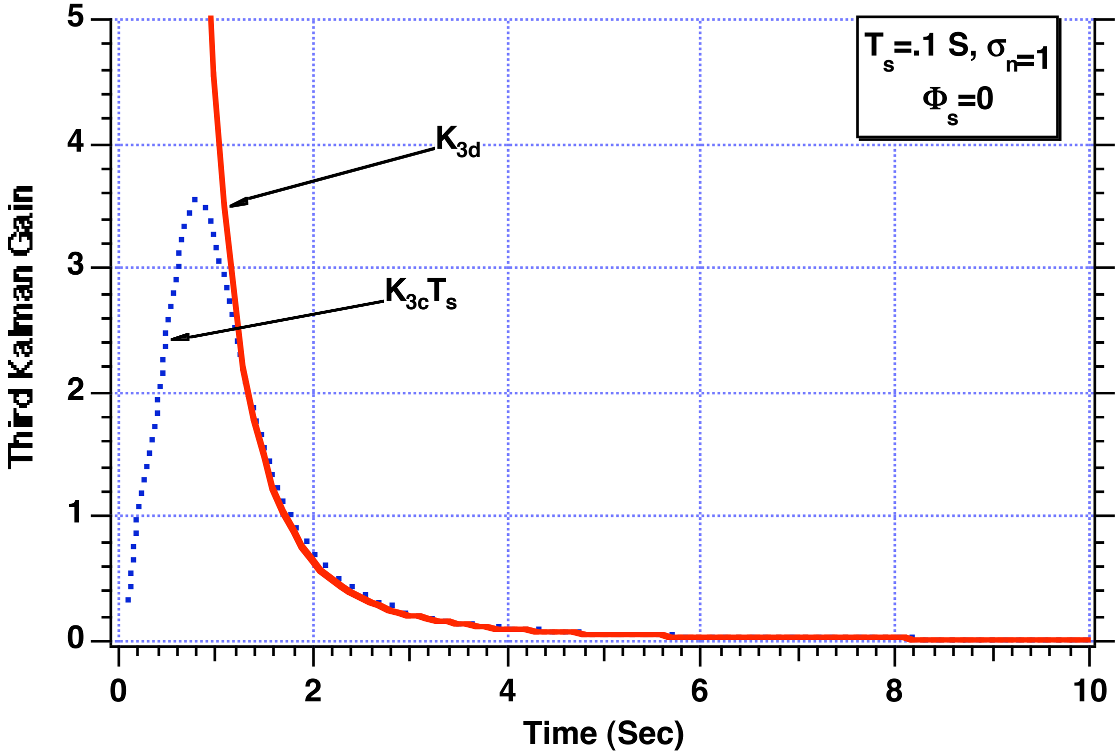
$$K_{1c} = \frac{K_{1d}}{T_s} \qquad K_{1_d} = \frac{3(3k^2-3k+2)}{k(k+1)(k+2)} \quad k=1,2,...,n$$

Integrating Three-State Nonlinear Matrix Riccati Differential Equation Yields Good Match With Formula for Second Gain



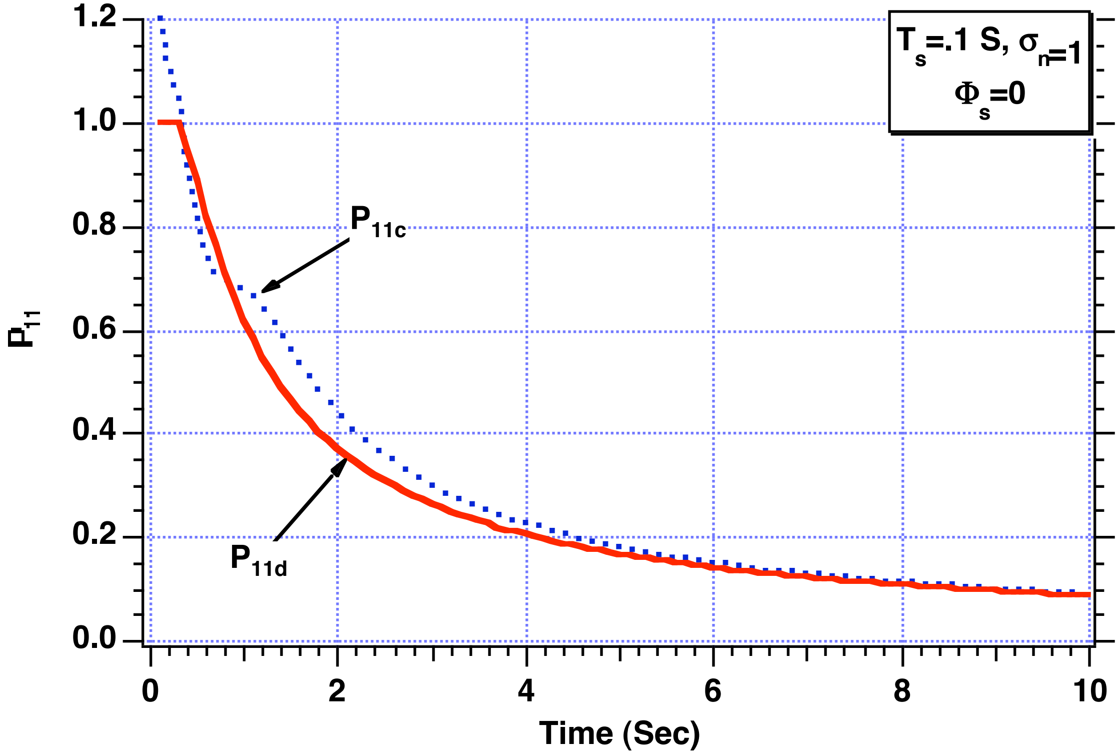
$$K_{2c} = \frac{K_{2d}}{T_s} \qquad K_{2_d} = \frac{18(2k-1)}{k(k+1)(k+2)T_s}$$

Integrating Three-State Nonlinear Matrix Riccati Differential Equation Yields Good Match With Formula for Third Gain



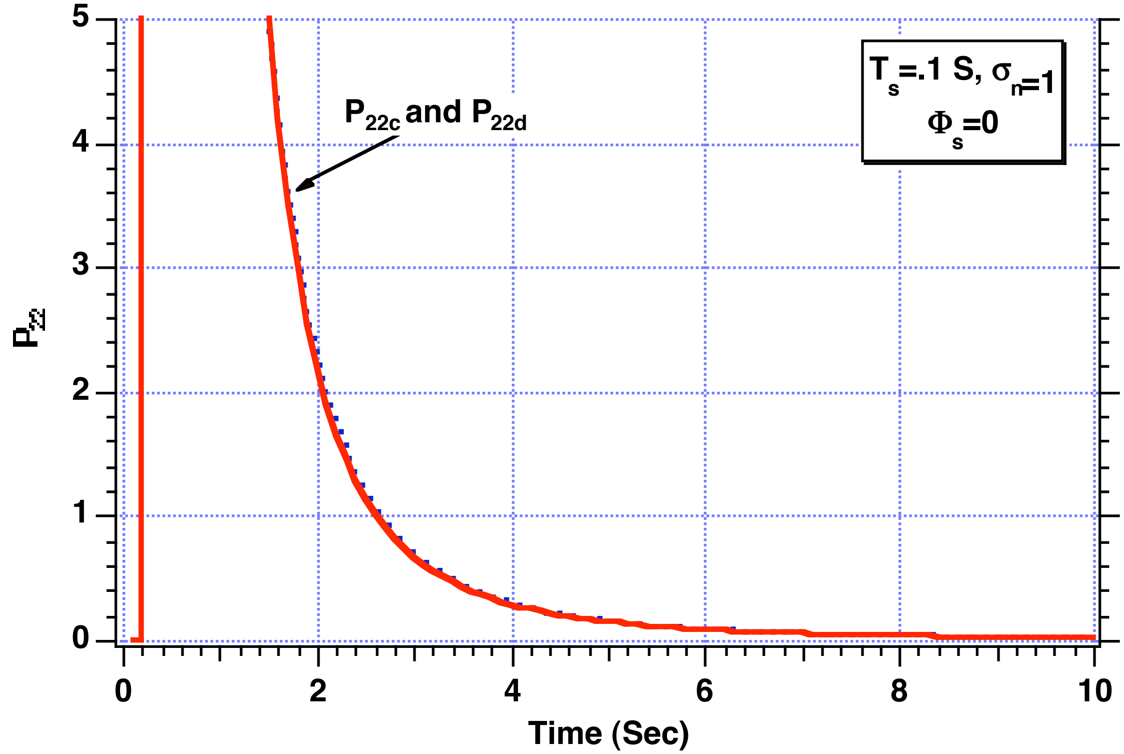
$$K_{3c} = \frac{K_{3d}}{T_s} \qquad K_{3_d} = \frac{60}{k(k+1)(k+2)T_s^2}$$

Integrating Three-State Nonlinear Matrix Riccati Differential Equation Yields Good Match for First Diagonal Element of Covariance Matrix



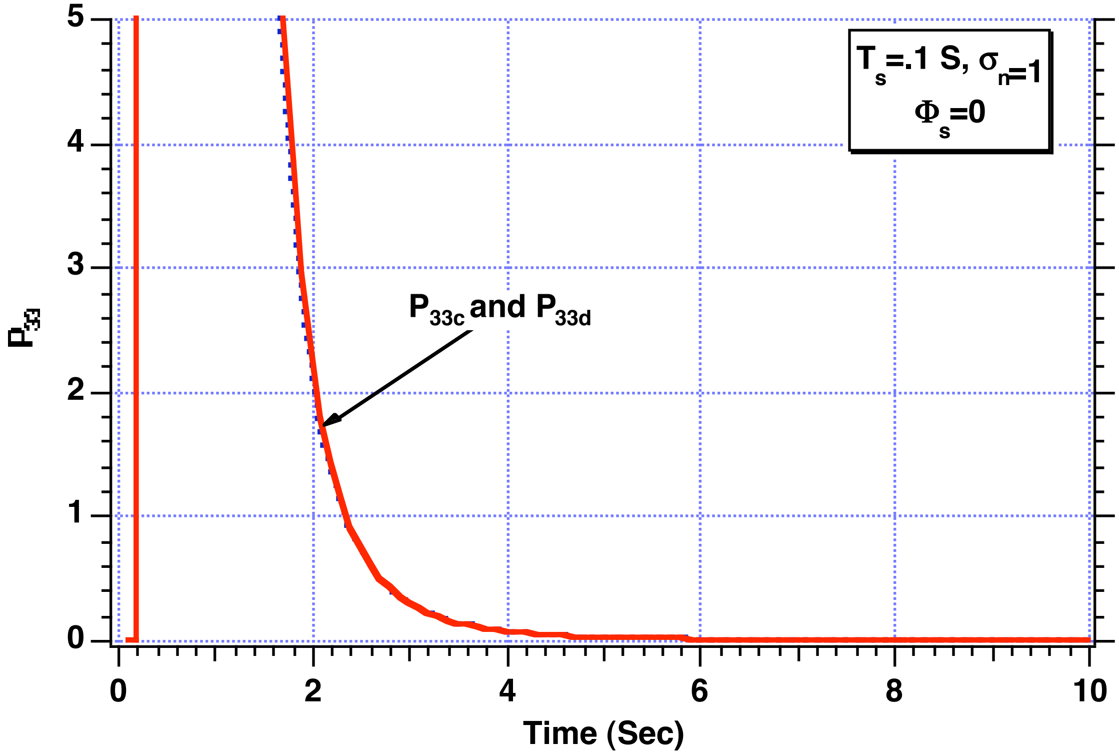
$$P_{11c} = P_{11d} \qquad P_{11_d} = \frac{3(3k^2-3k+2)\sigma_n^2}{k(k+1)(k+2)}$$

Integrating Three-State Nonlinear Matrix Riccati Differential Equation Yields Good Match for Second Diagonal Element of Covariance Matrix



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

Integrating Three-State Nonlinear Matrix Riccati Differential Equation Yields Good Match for Third Diagonal Element of Covariance Matrix



$$P_{33c} = P_{33d} \qquad P_{33_d} = \frac{720\sigma_n^2}{k(k^2-1)(k^2-4)T_s^4}$$

Steady-State Approximations

Zeroth-Order Filter

Gain Formula For Zeroth-Order Filter

In steady-state Riccati equation for zeroth-order filter is

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

We can solve equation algebraically

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

Kalman gain turns out to be

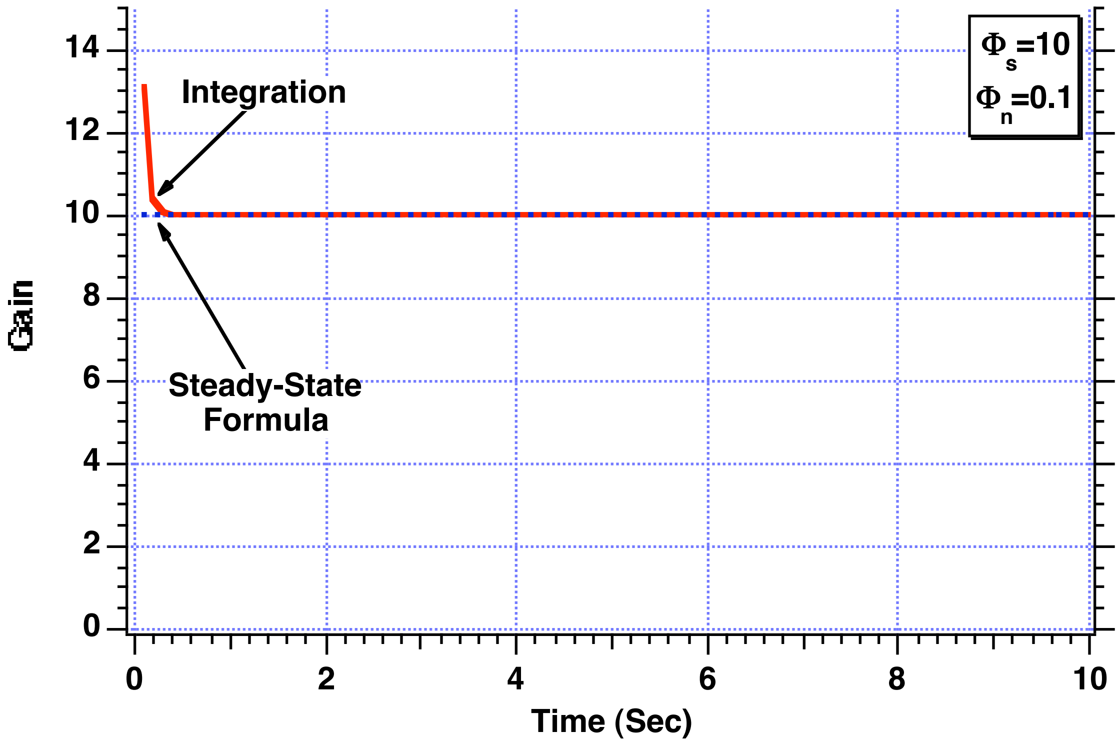
$$K = \frac{P}{\Phi_n} = \frac{(\Phi_s \Phi_n)^{1/2}}{\Phi_n}$$

Or

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

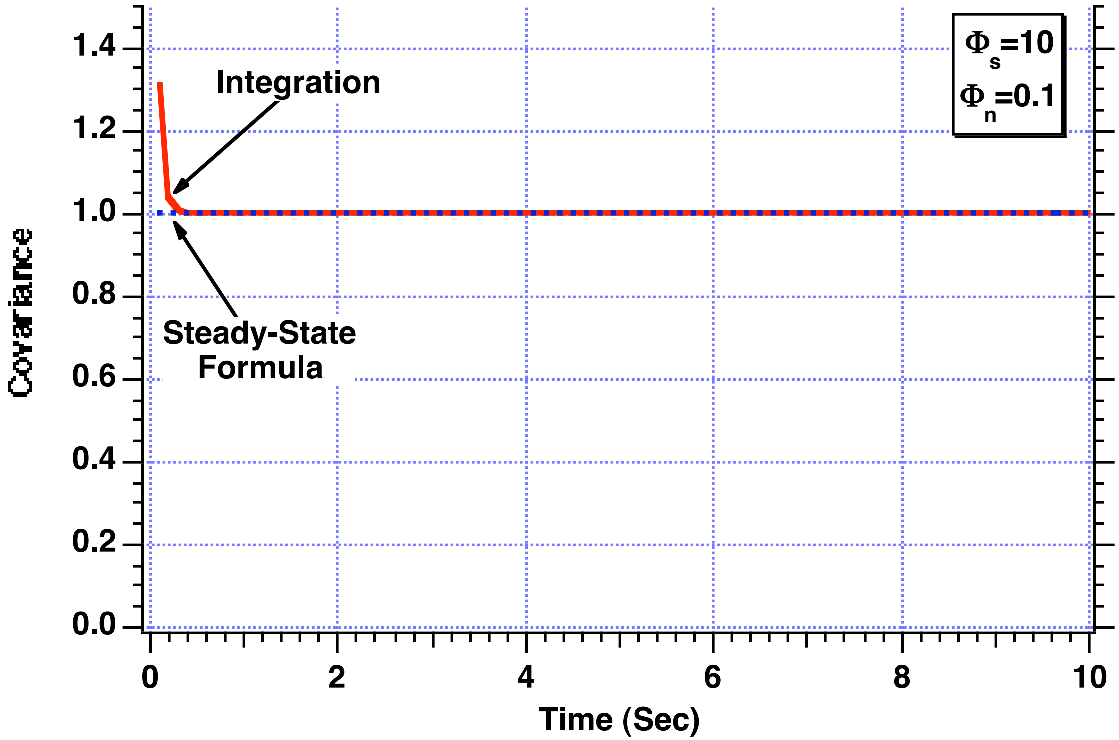
Thus the continuous steady-state Kalman gain only depends on the ratio of the process and measurement noise spectral densities

Steady-State Formula Accurately Predicts Kalman Gain for Zeroth-Order Continuous Polynomial Kalman Filter



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

Steady-State Formula Accurately Predicts Kalman Covariance for Zeroth-Order Continuous Polynomial Kalman Filter



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

Deriving Transfer Function For Zeroth-Order Polynomial Kalman Filter

Recall continuous Kalman filter formula

$$\dot{\hat{\mathbf{x}}} = \mathbf{F}\hat{\mathbf{x}} + \mathbf{K}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) \qquad \mathbf{F} = 0 \qquad \mathbf{H} = 1$$

Substitution yields

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

Convert to Laplace transform notation

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

After some manipulations we get

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

Defining a natural frequency

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

We can rewrite filter transfer function as

$$\frac{\hat{x}}{x^*} = \frac{1}{1 + \frac{s}{\omega_0}} \longleftarrow$$ **Low-pass filter**

Zeroth-Order Continuous Polynomial Kalman Filter's Natural Frequency Increases as the Ratio of Process to Measurement Noise Increases

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

First-Order Filter

Gain Formula For First-Order Filter-1

Recall regular Riccati equation

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

Matrix symmetric

From steady-state Riccati equation

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

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} P_{11} & P_{12} \\ P_{12} & P_{22} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \Phi_s \end{bmatrix} = 0$$

Using symmetry we get three scalar equations with three unknowns

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

$$0 = P_{22} - \frac{P_{11}P_{12}}{\Phi_n}$$

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

Gain Formula For First-Order Filter-2

Solving the algebraic equations yields

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

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

$$P_{12} = \Phi_s^{1/2}\Phi_n^{1/2}$$

Since

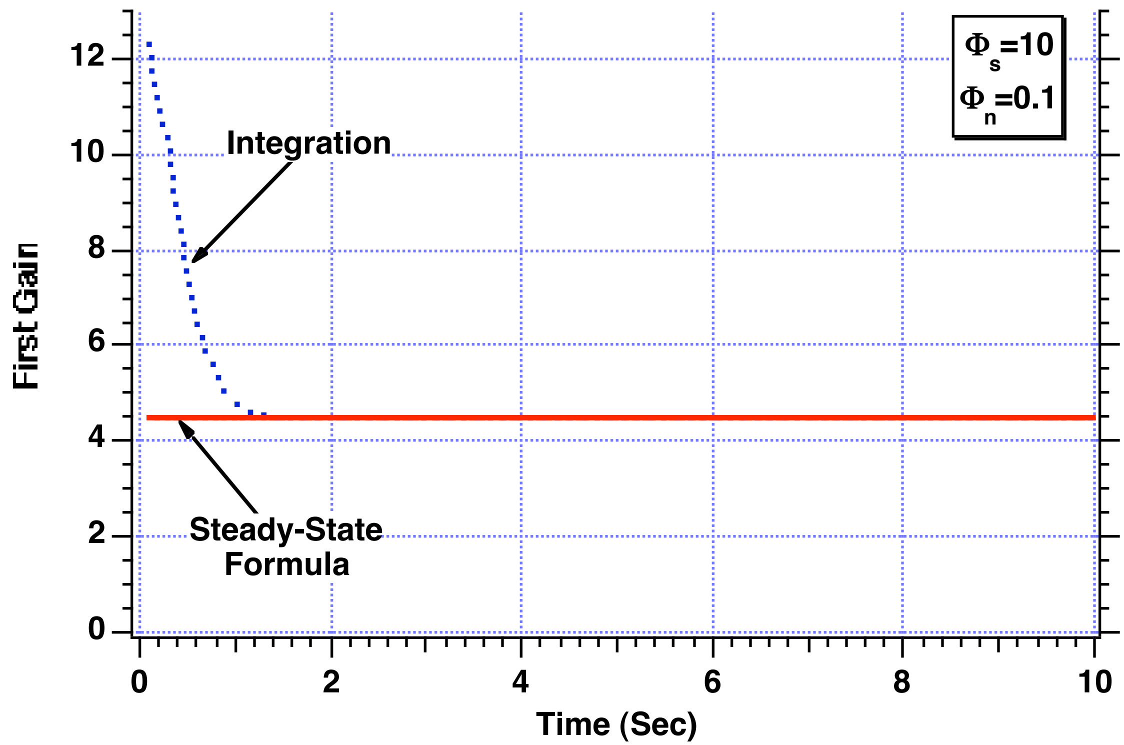
$$\mathbf{K} = \mathbf{P}\,\mathbf{H}^{\mathbf{T}}\mathbf{R}^{-1}$$

The gains become

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

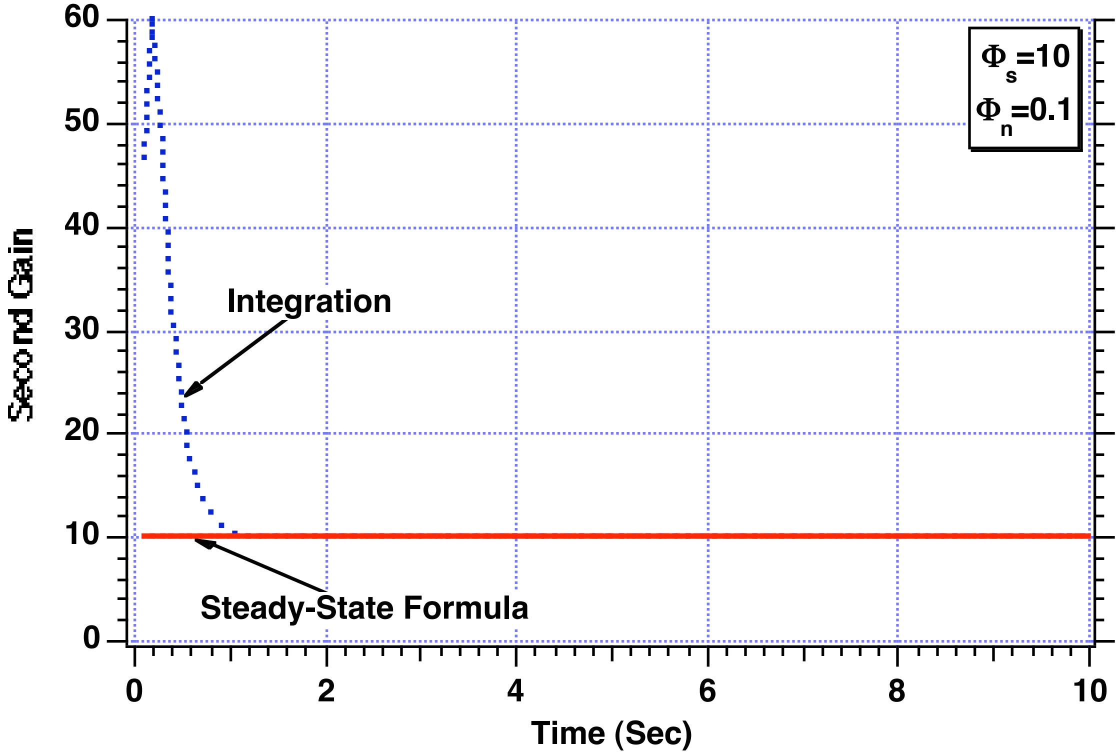
$$K_2 = \frac{P_{12}}{\Phi_n} = \left(\frac{\Phi_s}{\Phi_n}\right)^{1/2}$$

Steady-State Gain Formula is Accurate for First Gain in Continuous First-Order Polynomial Kalman Filter



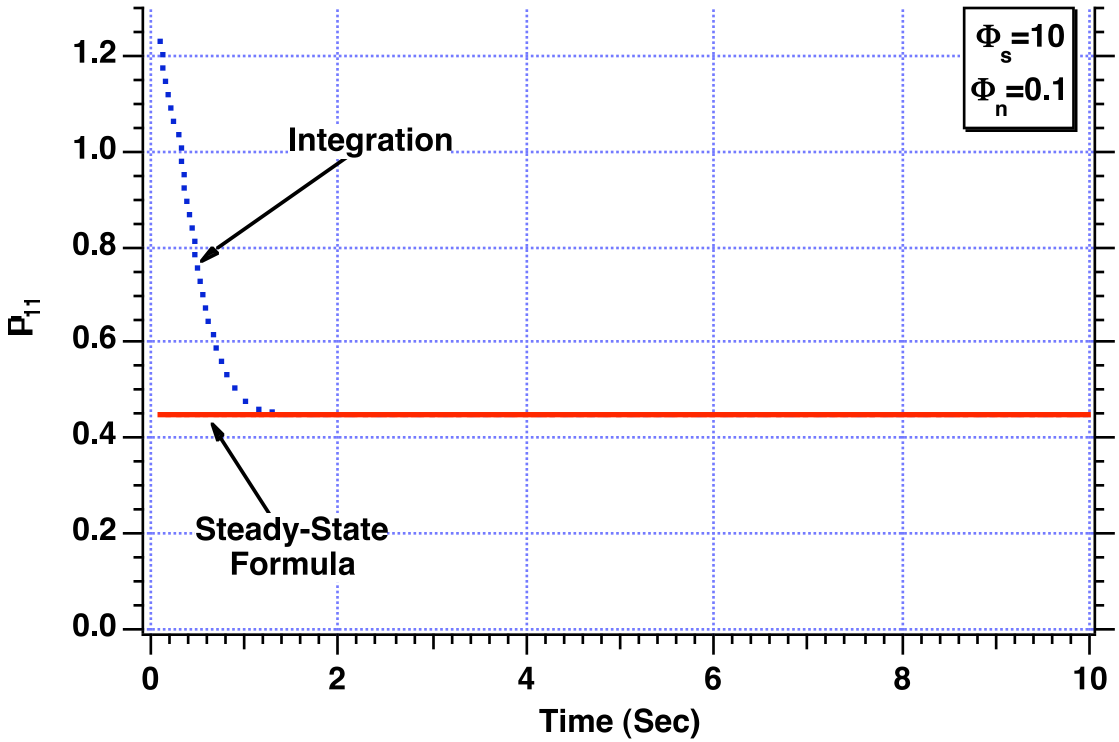
$$K_1 = \sqrt{2}\left(\frac{\Phi_s}{\Phi_n}\right)^{1/4}$$

Steady-State Gain Formula is Accurate for Second Gain in Continuous First-Order Polynomial Kalman Filter



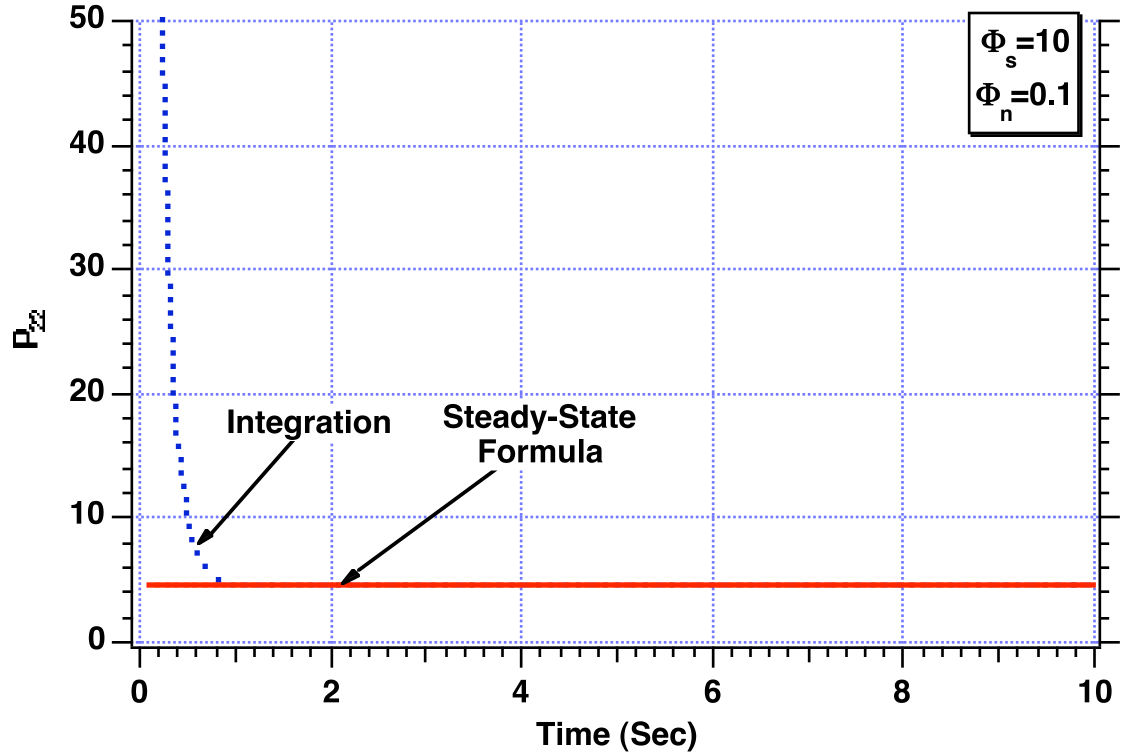
$$K_2 = \left(\frac{\Phi_s}{\Phi_n}\right)^{1/2}$$

Steady-State Formula for First Diagonal Element of Covariance Matrix is Accurate for Continuous First-Order Polynomial Kalman Filter



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

Steady-State Formula for Second Diagonal Element of Covariance Matrix is Accurate for Continuous First-Order Polynomial Kalman Filter



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

Deriving Transfer Function For First-Order Polynomial Kalman Filter

Recall continuous Kalman filter formula

$$\dot{\hat{\mathbf{x}}} = \mathbf{F}\hat{\mathbf{x}} + \mathbf{K}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) \qquad \mathbf{F} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Substitution yields

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

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

Convert to Laplace transform notation

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

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

After some manipulations we get

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

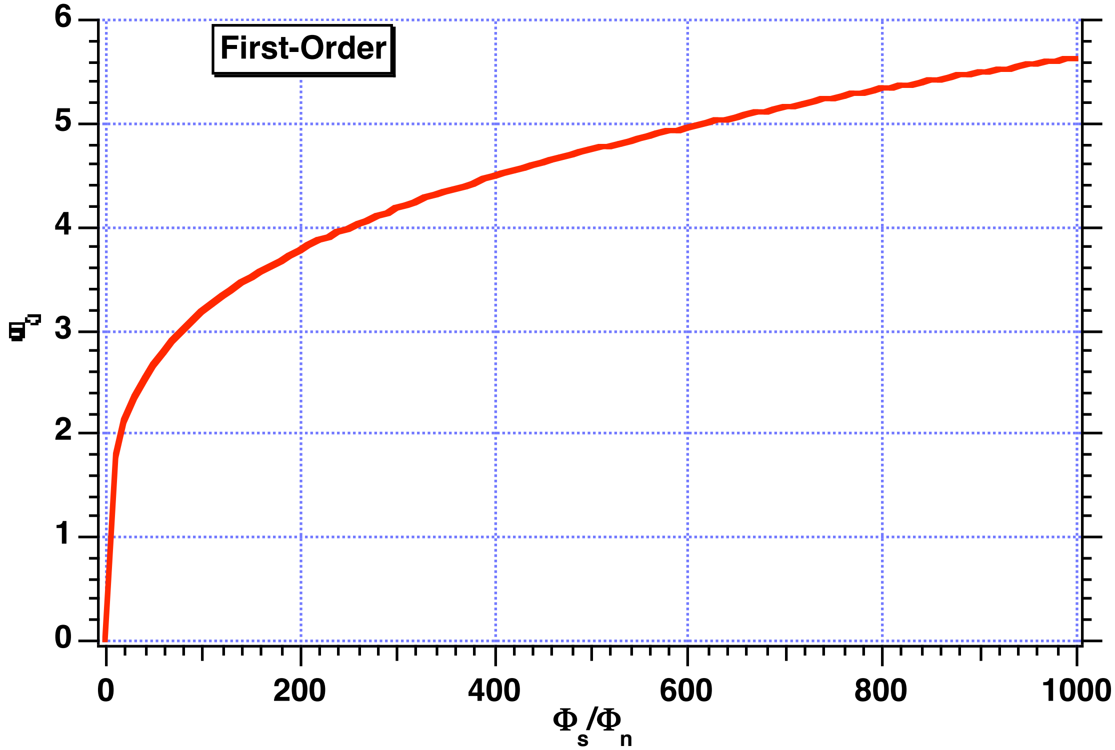
Defining a natural frequency

$$K_1 = \sqrt{2}\left(\frac{\Phi_s}{\Phi_n}\right)^{1/4} \qquad K_2 = \left(\frac{\Phi_s}{\Phi_n}\right)^{1/2} \longrightarrow \omega_0 = \left(\frac{\Phi_s}{\Phi_n}\right)^{1/4} \longrightarrow K_1 = \frac{P_{11}}{\Phi_n} = \sqrt{2}\left(\frac{\Phi_s}{\Phi_n}\right)^{1/4} = \sqrt{2}\,\omega_0 \qquad K_2 = \frac{P_{12}}{\Phi_n} = \left(\frac{\Phi_s}{\Phi_n}\right)^{1/2} = \omega_0^2$$

We can rewrite filter transfer function as

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

Filter Natural Frequency Increases as the Ratio of the Process to Measurement Noise Spectral Densities Increases



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

Second-Order Filter

Gain Formula For Second-Order Filter-1

Recall regular Riccati equation

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

Matrix symmetric

From steady-state Riccati equation

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

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

Using symmetry we get six scalar equations with six unknowns

$$P_{11}^2 = 2P_{12}\Phi_n \qquad P_{11}P_{12} = \Phi_n(P_{22} + P_{13})$$

$$P_{12}^2 = 2P_{23}\Phi_n \qquad P_{11}P_{13} = P_{23}\Phi_n$$

$$P_{13}^2 = \Phi_s\Phi_n \qquad P_{12}P_{13} = P_{33}\Phi_n$$

Gain Formula For Second-Order Filter-2

Solving the algebraic equations yields

$$P_{11} = 2\Phi_s^{1/6}\Phi_n^{5/6} \qquad P_{22} = 3\Phi_s^{1/2}\Phi_n^{1/2}$$

$$P_{12} = 2\Phi_s^{1/3}\Phi_n^{2/3} \qquad P_{23} = 2\Phi_s^{2/3}\Phi_n^{1/3}$$

$$P_{13} = \Phi_s^{1/2}\Phi_n^{1/2} \qquad P_{33} = 2\Phi_s^{5/6}\Phi_n^{1/6}$$

Since

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

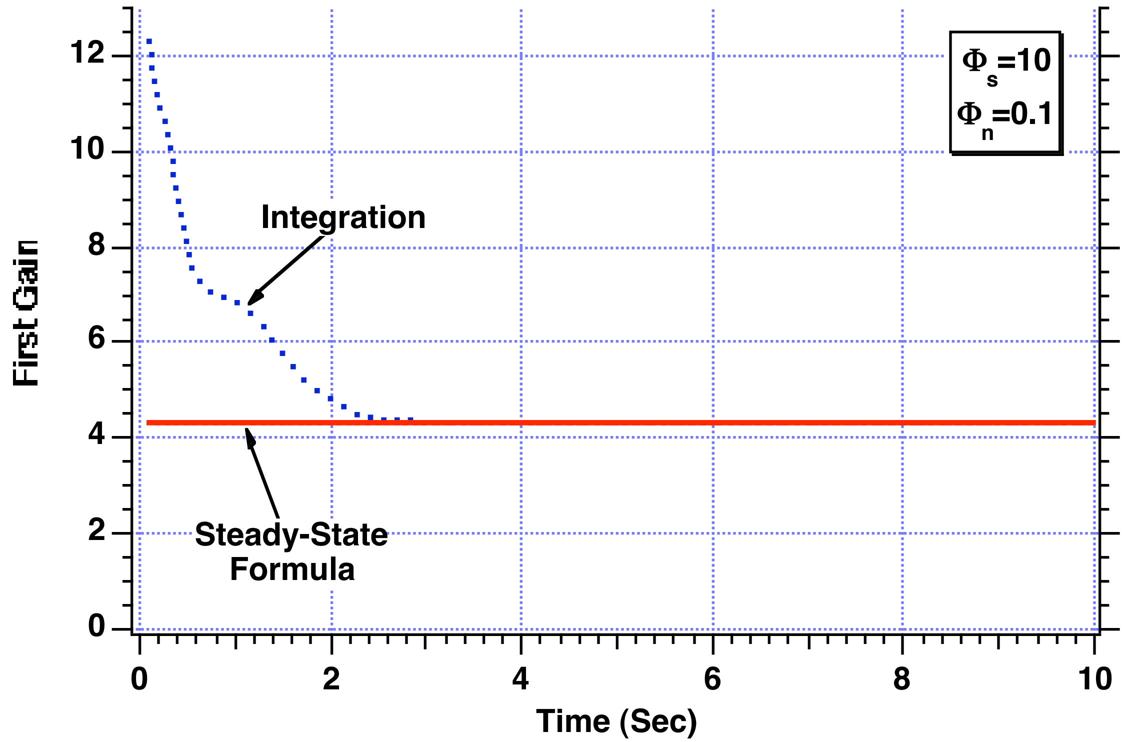
The gains become

$$K_1 = \frac{P_{11}}{\Phi_n} = 2\left(\frac{\Phi_s}{\Phi_n}\right)^{1/6}$$

$$K_2 = \frac{P_{12}}{\Phi_n} = 2\left(\frac{\Phi_s}{\Phi_n}\right)^{1/3}$$

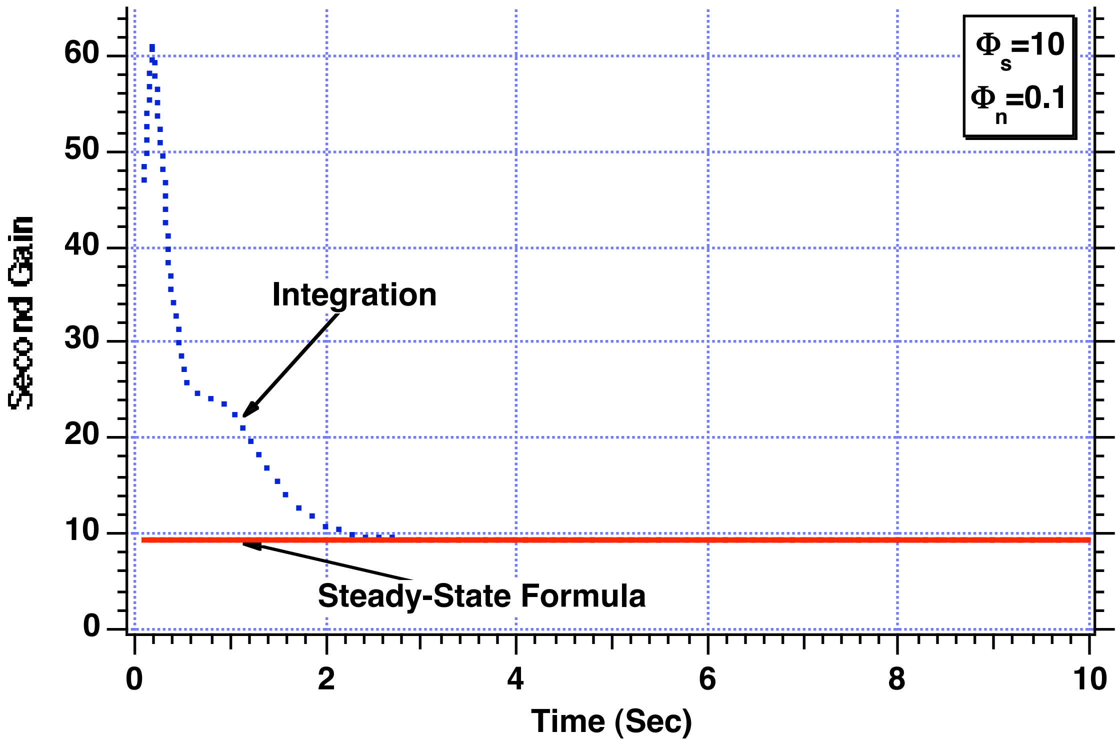
$$K_3 = \frac{P_{13}}{\Phi_n} = \left(\frac{\Phi_s}{\Phi_n}\right)^{1/2}$$

Steady-State Gain Formula is Accurate for First Gain in Continuous Second-Order Polynomial Kalman Filter



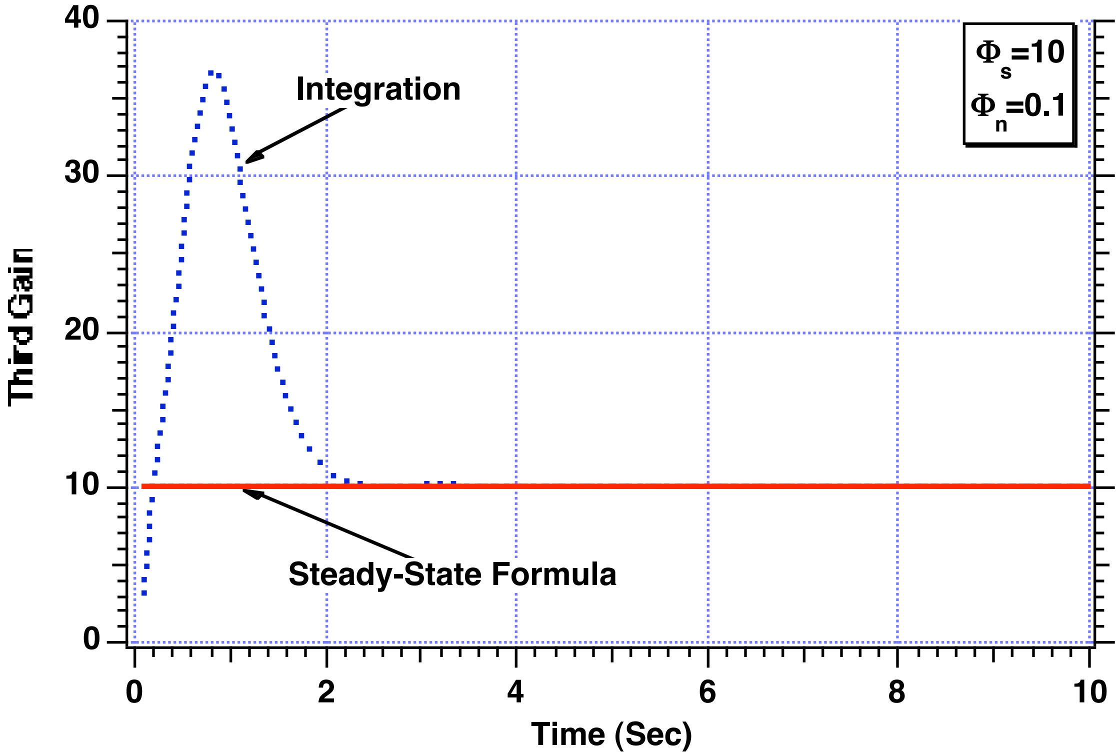
$$K_1 = 2\left(\frac{\Phi_s}{\Phi_n}\right)^{1/6}$$

Steady-State Gain Formula is Accurate for Second Gain in Continuous Second-Order Polynomial Kalman Filter



$$K_2 = 2\left(\frac{\Phi_s}{\Phi_n}\right)^{1/3}$$

Steady-State Gain Formula is Accurate for Third Gain in Continuous Second-Order Polynomial Kalman Filter



$$K_3 = \left(\frac{\Phi_s}{\Phi_n}\right)^{1/2}$$

Deriving Transfer Function For Second-Order Polynomial Kalman Filter

Recall continuous Kalman filter formula

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

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

Substitution yields

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

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

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

Convert to Laplace transform notation

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

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

$$s\hat{\ddot{x}} = K_3(x^* - \hat{x})$$

After some manipulations we get

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

Defining a natural frequency

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

$$K_1 = \frac{P_{11}}{\Phi_n} = 2\left(\frac{\Phi_s}{\Phi_n}\right)^{1/6} = 2\omega_0$$

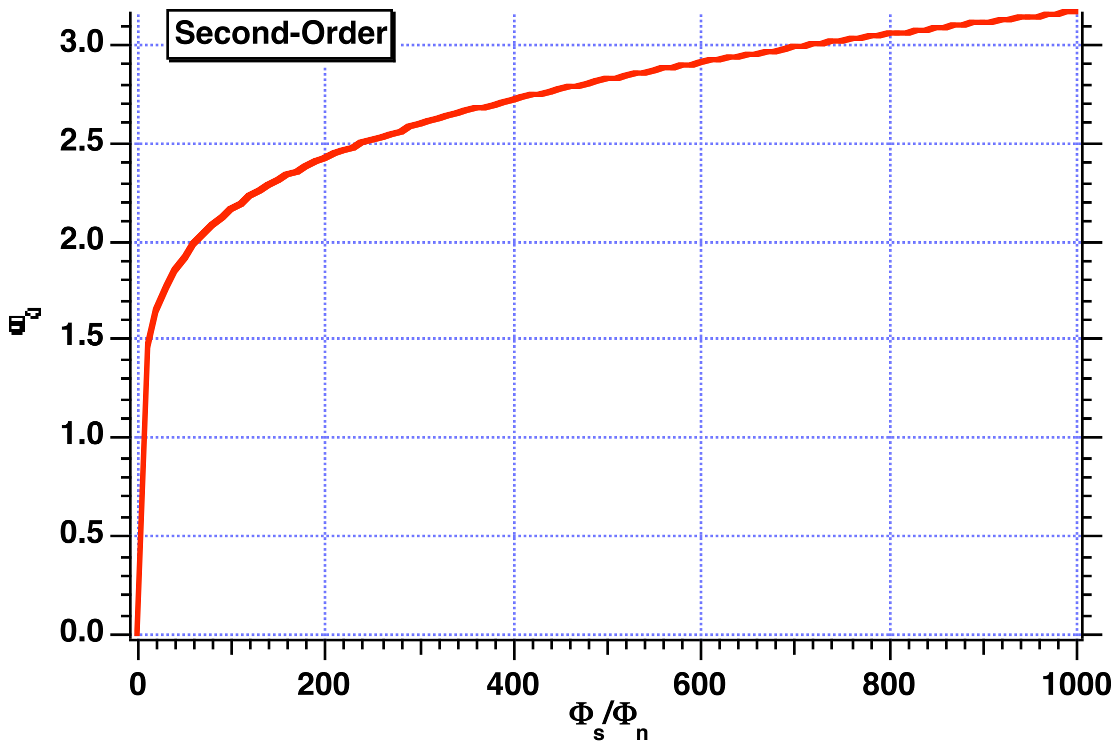
$$K_2 = \frac{P_{12}}{\Phi_n} = 2\left(\frac{\Phi_s}{\Phi_n}\right)^{1/3} = 2\omega_0^2$$

$$K_3 = \frac{P_{13}}{\Phi_n} = \left(\frac{\Phi_s}{\Phi_n}\right)^{1/2} = \omega_0^3$$

We can rewrite filter transfer function as

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

Second-Order Kalman Filter Natural Frequency Increases With Increasing Ratio of Process to Measurement Noise Spectral Density



Filter Comparison

Transfer Functions and Magnitudes for Different Order Polynomial Kalman Filters

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

FORTRAN Program to Calculate Magnitudes of Kalman Filter Transfer Functions

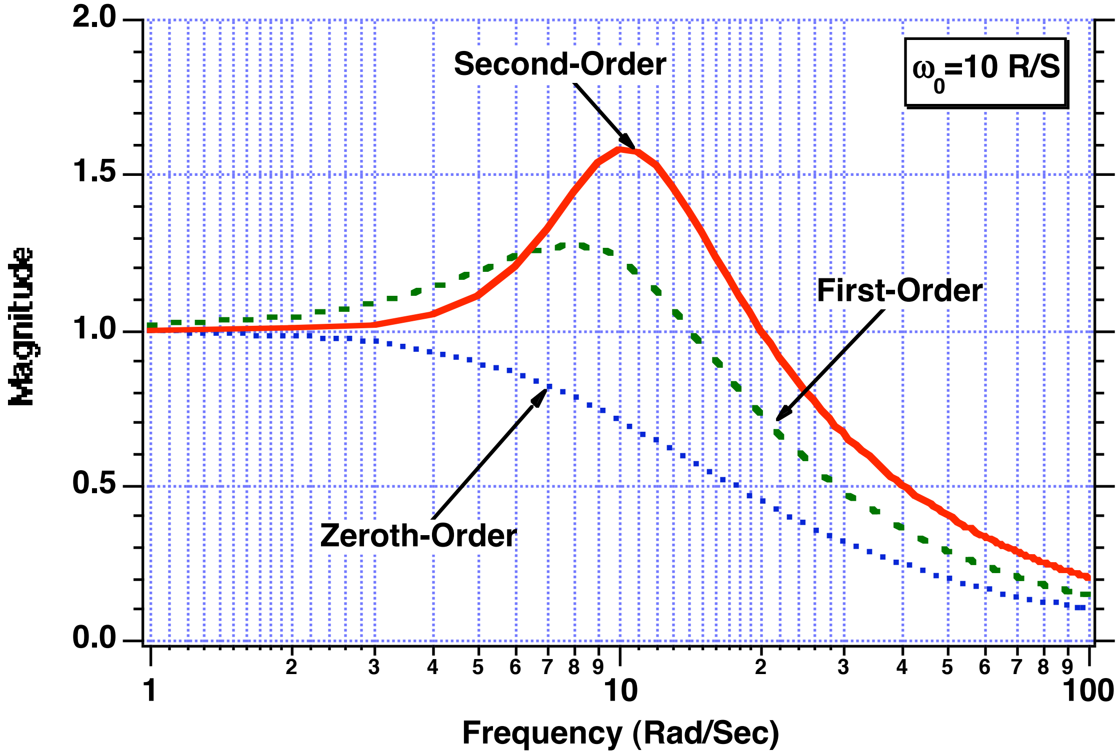
```
      IMPLICIT REAL*8 (A-H)
      IMPLICIT REAL*8 (O-Z)
      OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
      W0=10.
      DO 10 W=1.,100.
      XMAG1=1./SQRT(1.+(W/W0)**2)
      TOP1=1.+2.*(W/W0)**2
      BOT1=(1.-(W*W/(W0*W0)))**2+2.*(W/W0)**2
      XMAG2=SQRT(TOP1/(BOT1+.00001))
      TOP2=(1.-2.*W*W/(W0*W0))**2+(2.*W/W0)**2
      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
```

Zeroth-order filter (XMAG1)

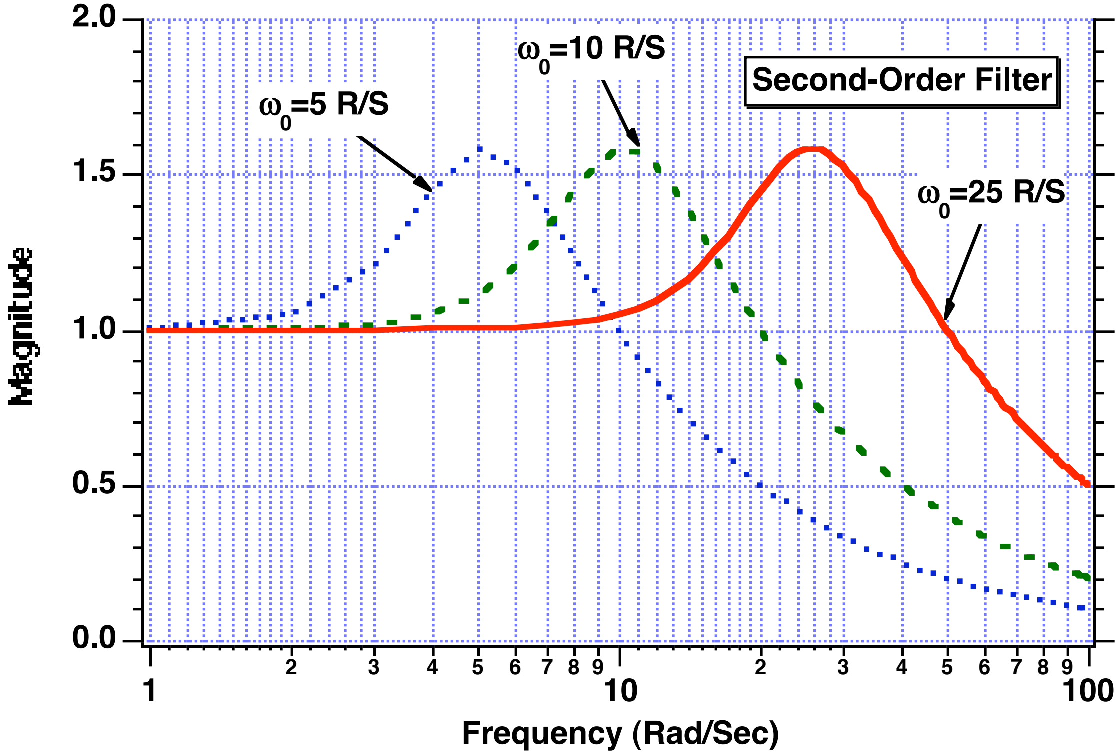
First-order filter (TOP1, BOT1, XMAG2)

Second-order filter (TOP2, TEMP1, TEMP2, XMAG3)

Higher-Order Filters Have Less Attenuation After Filter Natural Frequency



Increasing Filter Natural Frequency Increases Filter Bandwidth



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

Continuous Polynomial Kalman Filter Summary

- Continuous Kalman filtering equations useful for understanding the properties of the discrete filter
- Relationship between continuous and discrete Kalman gains and covariances established
- Formulas for steady-state Kalman gains and covariances derived
- Transfer functions for zeroth, first and second-order polynomial Kalman filters derived
- Bandwidth of polynomial Kalman filter shown to be proportional to ratio of process to measurement noise spectral densities