Recursive Least Squares Filtering

Recursive Least Squares Filtering Overview

- Making zeroth-order least squares filter recursive
- Deriving properties of recursive zeroth-order filter
- First and second-order recursive least squares filters
    - Structure and gains
    - Errors in estimates due to measurement noise and truncation error
- Comparison of various order recursive least squares filters

Review

- Method of least squares is a batch processing technique

  - All measurements must be taken before estimates can be made

- Matrix inverse required

  - Dimensions of matrix inverse proportional to order of polynomial fit (i.e. First-order fit requires two by two inverse)

Zeroth-Order Recursive Least Squares Filter

Making Zeroth-Order Filter Recursive - 1

Batch processing least squares filter formula

$$\hat{x}_k = a_0 = \frac{\sum_{i=1}^{k} x_i^*}{k}$$

Rewrite by changing subscripts

$$\hat{x}_{k+1} = \frac{\sum_{i=1}^{k+1} x_i^*}{k+1}$$

Expanding the numerator yields

$$\hat{x}_{k+1} = \frac{\sum_{i=1}^{k} x_i^* + x_{k+1}^*}{k+1}$$

Since

$$\sum_{i=1}^{k} x_i^* = k\hat{x}_k$$

By substitution we can say that

$$\hat{x}_{k+1} = \frac{k\hat{x}_k + x_{k+1}^*}{k+1}$$

Making Zeroth-Order Filter Recursive - 2

Can add and subtract the previous state estimate to the numerator

$$\hat{x}_{k+1} = \frac{k\hat{x}_k + \hat{x}_k + x^*_{k+1} - \hat{x}_k}{k+1} = \frac{(k+1)\hat{x}_k + x^*_{k+1} - \hat{x}_k}{k+1}$$

Rewrite the preceding equation as

$$\hat{x}_{k+1} = \hat{x}_k + \frac{1}{k+1}(x^*_{k+1} - \hat{x}_k)$$

Changing subscripts yields

$$\hat{x}_k = \hat{x}_{k-1} + \frac{1}{k}(x^*_k - \hat{x}_{k-1})$$

*This is recursive form we desire since the new estimate simply depends on the old estimate plus a gain (i.e., 1/k for the zeroth-order filter) times a residual (i.e., current measurement minus previous estimate)

Properties of the Zeroth-Order Recursive Filter

Another Form of the Zeroth-Order Recursive Filter

Recursive form of zeroth-order filter

$$\hat{x}_k = \hat{x}_{k-1} + K_{1_k} Res_k$$

Where filter gain is

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

And residual is given by

$$Res_k = x_k^* - \hat{x}_{k-1}$$

Numerical Example For the Zeroth-Order Filter-1

Previous measurement data

k	1	2	3	4
$(k-1)T_s$	0	1	2	3
x^*_k	1.2	.2	2.9	2.1

Gain for first measurement

$$K_{1_1} = \frac{1}{k} = \frac{1}{1} = 1$$

For lack of any a priori information assume

$$\hat{x}_0 = 0$$

Calculate residual as

$$Res_1 = x^*_1 - \hat{x}_0 = 1.2 - 0 = 1.2$$

New estimate becomes

$$\hat{x}_1 = \hat{x}_0 + K_{1_1}Res_1 = 0 + 1*1.2 = 1.2$$

***We are able to make estimates before all the data is collected**

Numerical Example For the Zeroth-Order Filter-2

For next cycle with k=2

$$K_{1_2} = \frac{1}{k} = \frac{1}{2} = .5$$

$$Res_2 = x_2^* - \hat{x}_1 = .2 - 1.2 = -1$$

$$\hat{x}_2 = \hat{x}_1 + K_{1_2} Res_2 = 1.2 + .5*(-1) = .7$$

Another estimate without collecting all the data

For next cycle with k=3

$$K_{1_3} = \frac{1}{k} = \frac{1}{3} = .333$$

$$Res_3 = x_3^* - \hat{x}_2 = 2.9 - .7 = 2.2$$

$$\hat{x}_3 = \hat{x}_2 + K_{1_3} Res_3 = .7 + .333*2.2 = 1.43$$

Another estimate

For last cycle with k=4

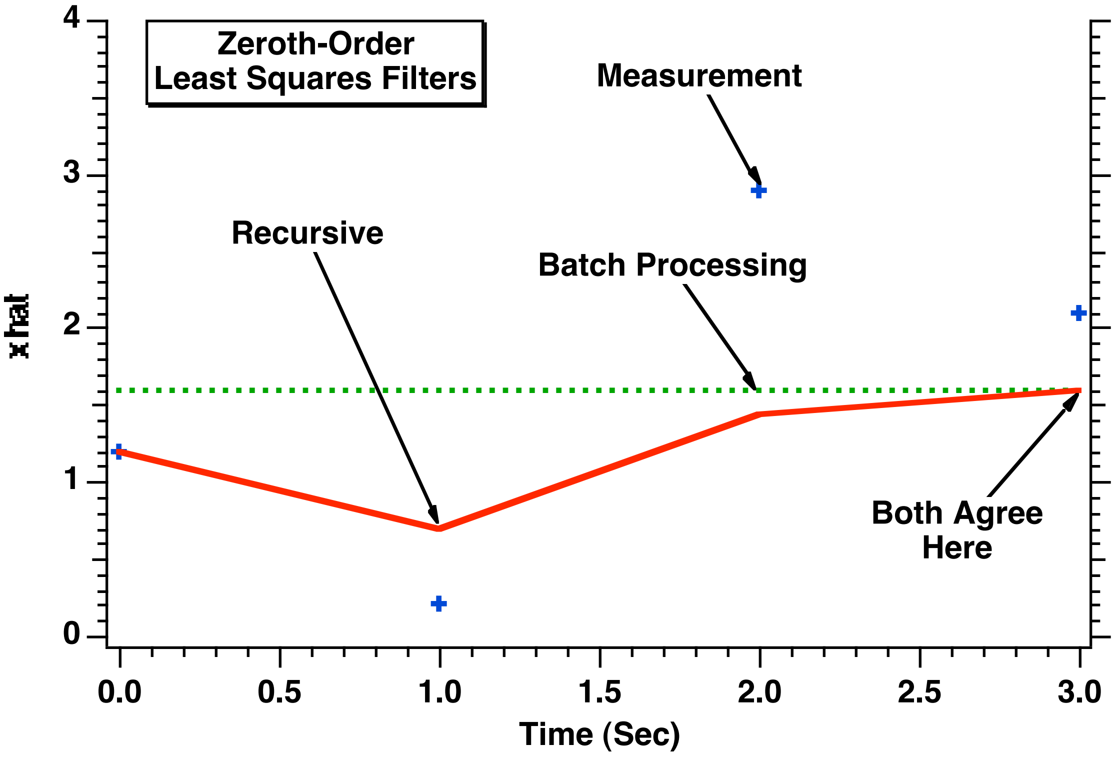
$$K_{1_4} = \frac{1}{k} = \frac{1}{4} = .25$$

$$Res_4 = x_4^* - \hat{x}_3 = 2.1 - 1.43 = .67$$

$$\boxed{\hat{x}_4 = \hat{x}_3 + K_{1_4} Res_4 = 1.43 + .25*.67 = 1.6}$$

Same answer obtained from batch processing method when all the data was collected

Batch Processing and Recursive Least Squares Methods Yield the Same Answers After All Measurements Are Taken



Initial Conditions For Recursive Least Squares Filter Are Not Important

Assume a different initial condition

$$\hat{x}_0 = 100$$

Start first cycle of recursive equations with k=1

$$K_{1_1} = \frac{1}{k} = \frac{1}{1} = 1$$

$$Res_1 = x_1^* - \hat{x}_0 = 1.2 - 100 = -98.8$$

$$\boxed{\hat{x}_1 = \hat{x}_0 + K_{1_1} Res_1 = 100 + 1*(-98.8) = 1.2}$$

This is same answer as when the initial condition was zero

Deriving a Formula For Variance in Filter's Estimate - 1

Recursive filter form is given by

$$\hat{x}_k = \hat{x}_{k-1} + \frac{1}{k}(x_k^* - \hat{x}_{k-1})$$

The error in the estimate is

$$x_k - \hat{x}_k = x_k - \hat{x}_{k-1} - \frac{1}{k}(x_k^* - \hat{x}_{k-1})$$

Signal minus estimate and not measurement minus estimate

Measurement is simply the signal plus noise

$$x_k^* = x_k + v_k$$

Substitution yields

$$x_k - \hat{x}_k = x_k - \hat{x}_{k-1} - \frac{1}{k}(x_k + v_k - \hat{x}_{k-1})$$

Since signal is constant for zeroth-order system

$$x_k = x_{k-1}$$

Deriving a Formula For Variance in Filter’s Estimate - 2

Substitution yields

$$x_k - \hat{x}_k = (x_{k-1} - \hat{x}_{k-1})\,(1 - \frac{1}{k}) - \frac{1}{k}\,v_k$$

Square both sides of the preceding equation

$$(x_k - \hat{x}_k)^2 = (x_{k-1} - \hat{x}_{k-1})^2\,(1 - \frac{1}{k})^2 - 2(1 - \frac{1}{k})(x_{k-1} - \hat{x}_{k-1})\frac{v_k}{k} + (\frac{1}{k}\,v_k)^2$$

Take expectations of both sides of the equation

$$E[(x_k - \hat{x}_k)^2] = E[(x_{k-1} - \hat{x}_{k-1})^2]\,(1 - \frac{1}{k})^2 - 2(1 - \frac{1}{k})E[(x_{k-1} - \hat{x}_{k-1})v_k]\frac{1}{k} + E[(\frac{1}{k}\,v_k)^2]$$

If we define

$$E[(x_k - \hat{x}_k)^2] = P_k$$

$$E[v_k^2] = \sigma_n^2$$

Deriving a Formula For Variance in Filter’s Estimate - 3

And assume that the noise is not correlated with the error in the estimate

$$E[(x_{k-1} - \hat{x}_{k-1})v_k] = 0$$

We get

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

Using engineering induction to solve preceding difference equation

$$P_1 = P_0(1 - \frac{1}{1})^2 + \frac{\sigma_n^2}{1^2} = \sigma_n^2$$

$$P_2 = P_1(1 - \frac{1}{2})^2 + \frac{\sigma_n^2}{2^2} = \sigma_n^2\frac{1}{4} + \frac{\sigma_n^2}{4} = \frac{\sigma_n^2}{2}$$

$$P_3 = P_2(1 - \frac{1}{3})^2 + \frac{\sigma_n^2}{3^2} = \frac{\sigma_n^2}{2}\frac{4}{9} + \frac{\sigma_n^2}{9} = \frac{\sigma_n^2}{3}$$

$$P_4 = P_3(1 - \frac{1}{4})^2 + \frac{\sigma_n^2}{4^2} = \frac{\sigma_n^2}{3}\frac{9}{16} + \frac{\sigma_n^2}{16} = \frac{\sigma_n^2}{4}$$

Trend indicates that

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

Formula for variance of error In the estimate

Deriving a Formula For Filter Truncation Error - 1

Suppose signal is one degree higher than filter

$$x_k = a_0 + a_1 t = a_0 + a_1(k-1)T_s$$

Error in the estimate

$$\varepsilon_k = x_k - \hat{x}_k$$

Recall batch processing formula for zeroth-order filter

$$\hat{x}_k = \frac{\sum_{i=1}^{k} x_i^*}{k}$$

In the noise free case we obtain

$$\hat{x}_k = \frac{\sum_{i=1}^{k} x_i}{k} = \frac{\sum_{i=1}^{k} [a_0 + a_1(i-1)T_s]}{k} = \frac{a_0\sum_{i=1}^{k} + a_1 T_s \sum_{i=1}^{k} i - a_1 T_s \sum_{i=1}^{k}}{k}$$

Since math handbooks tell us that

$$\sum_{i=1}^{k} = k$$

$$\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$$

Deriving a Formula For Filter Truncation Error - 2

Substitution yields

$$\hat{x}_k = \frac{a_0 k + a_1 T_s \frac{k(k+1)}{2} - a_1 T_s k}{k} = a_0 + \frac{a_1 T_s}{2}(k-1)$$

Therefore error in the estimate given by

$$\varepsilon_k = x_k - \hat{x}_k = a_0 + a_1 T_s (k-1) - a_0 - \frac{a_1 T_s}{2}(k-1) = \boxed{\frac{a_1 T_s}{2}(k-1)}$$

← **Truncation error formula**

FORTRAN Simulation For Testing Zeroth-Order Filter

```
        GLOBAL DEFINE
                INCLUDE 'quickdraw.inc'
        END
        IMPLICIT REAL*8(A-H,O-Z)
        OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
        TS=.1
        SIGNOISE=1.
        A0=1.
        A1=0.
        XH=0.
        XN=0.
        DO 10 T=0.,10.,TS
        XN=XN+1.
        CALL GAUSS(XNOISE,SIGNOISE)
        ACT=A0+A1*T
        XS=ACT+XNOISE
        XK=1./XN
        RES=XS-XH
        XH=XH+XK*RES
        SP11=SIGNOISE/SQRT(XN)
        XHERR=ACT-XH
        EPS=.5*A1*TS*(XN-1)
        WRITE(9,*)T,ACT,XS,XH,XHERR,SP11,-SP11,EPS
        WRITE(1,*)T,ACT,XS,XH,XHERR,SP11,-SP11,EPS
10      CONTINUE
        CLOSE(1)
        PAUSE
        END
```

Standard deviation of noise

Polynomial coefficients of signal

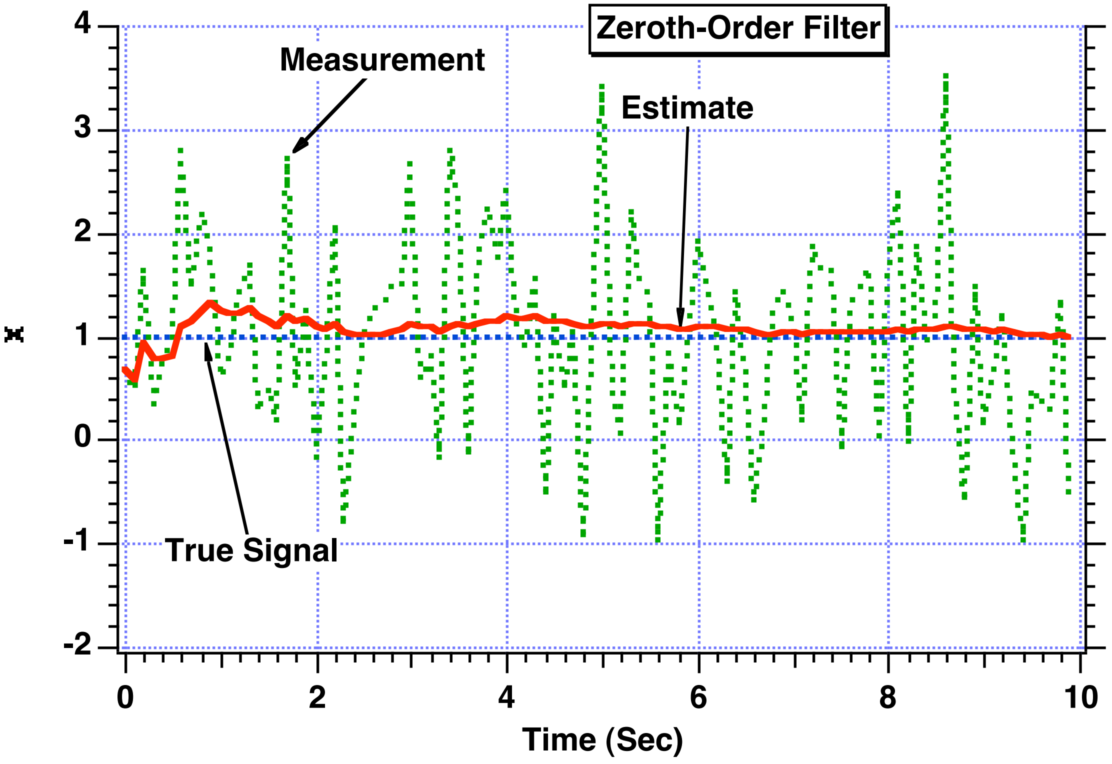
Signal

Measurement

Recursive filter

Actual error in estimate

Zeroth-Order Recursive Least Squares Filter is Able to Track Zero-Order Polynomial Plus Noise

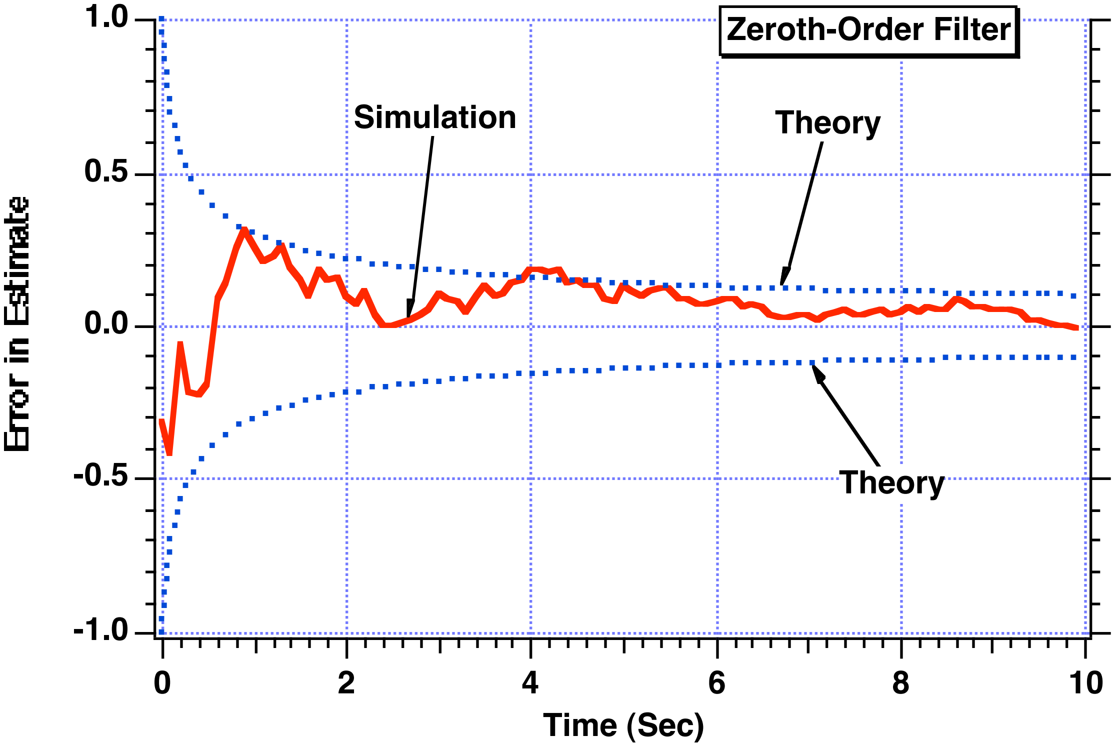


Measurement

$x^* = 1 + \text{noise}$

$\sigma_{\text{noise}} = 1$

Single Run Simulation Results Agree With Theoretical Formula



Theory

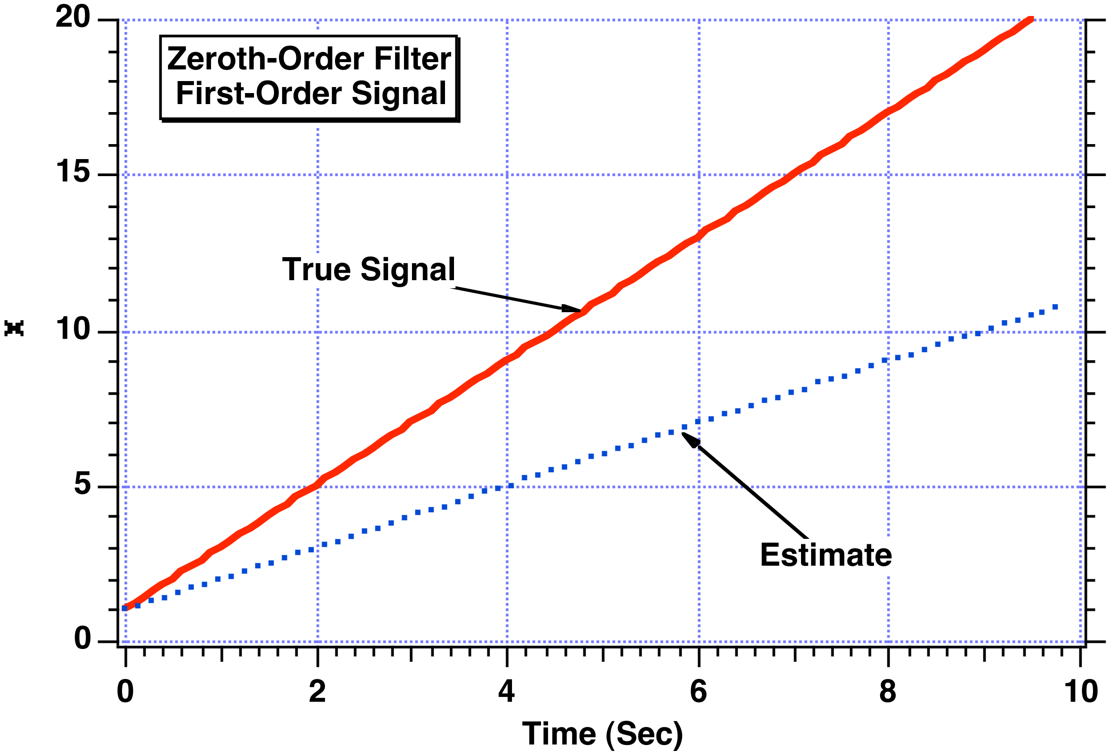
$$\sqrt{P_k} = \frac{\sigma_n}{\sqrt{k}}$$

Measurement

$$x^* = 1 + \text{noise}$$

$$\sigma_{noise} = 1$$

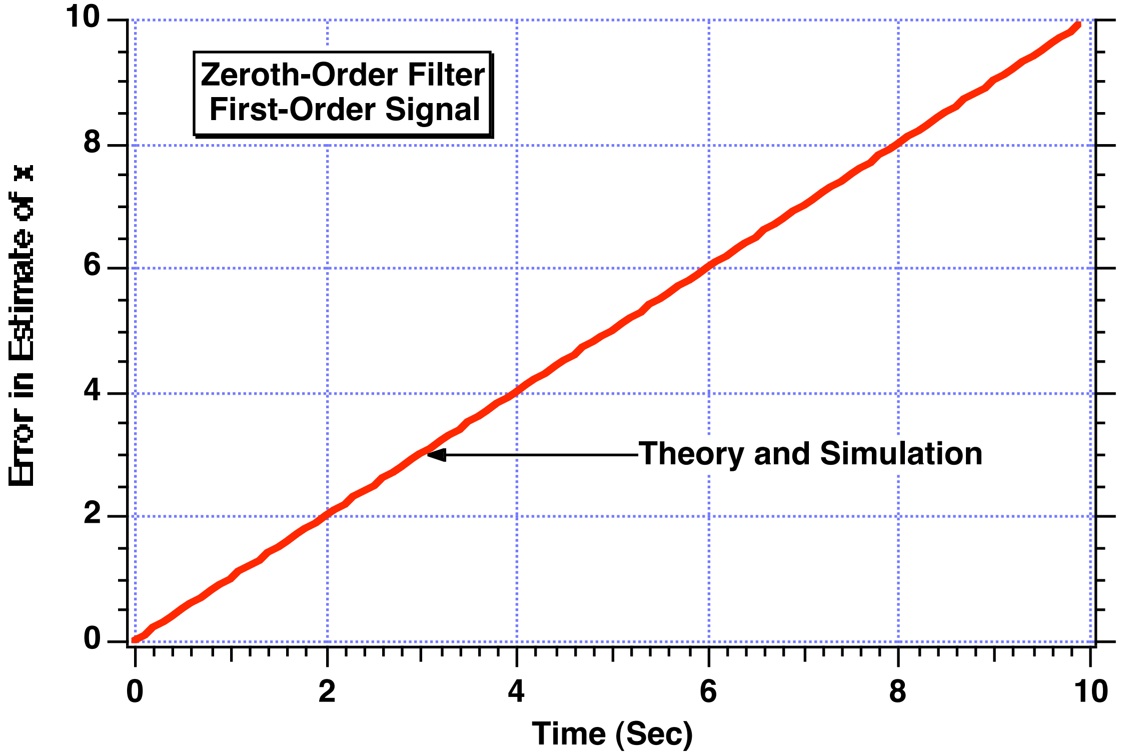
Zeroth-Order Recursive Least Squares Filter is Unable to Track First-Order Polynomial



Measurement

$x^* = 1 + 2t$

Simulation Results and Truncation Error Formula are in Excellent Agreement



Theory

$$\varepsilon_k = \frac{a_1 T_s}{2}(k-1) = .5*2*.1(k-1) = .1(k-1)$$

Summary of Results So Far For Zeroth-Order Recursive Least Squares Filter

Formulas for errors in estimates due to noise and truncation error

$$\sqrt{P_{11_k}} = \frac{\sigma_n}{\sqrt{k}}$$

$$\varepsilon_k = .5 a_1 T_s (k-1)$$

As more measurements are taken

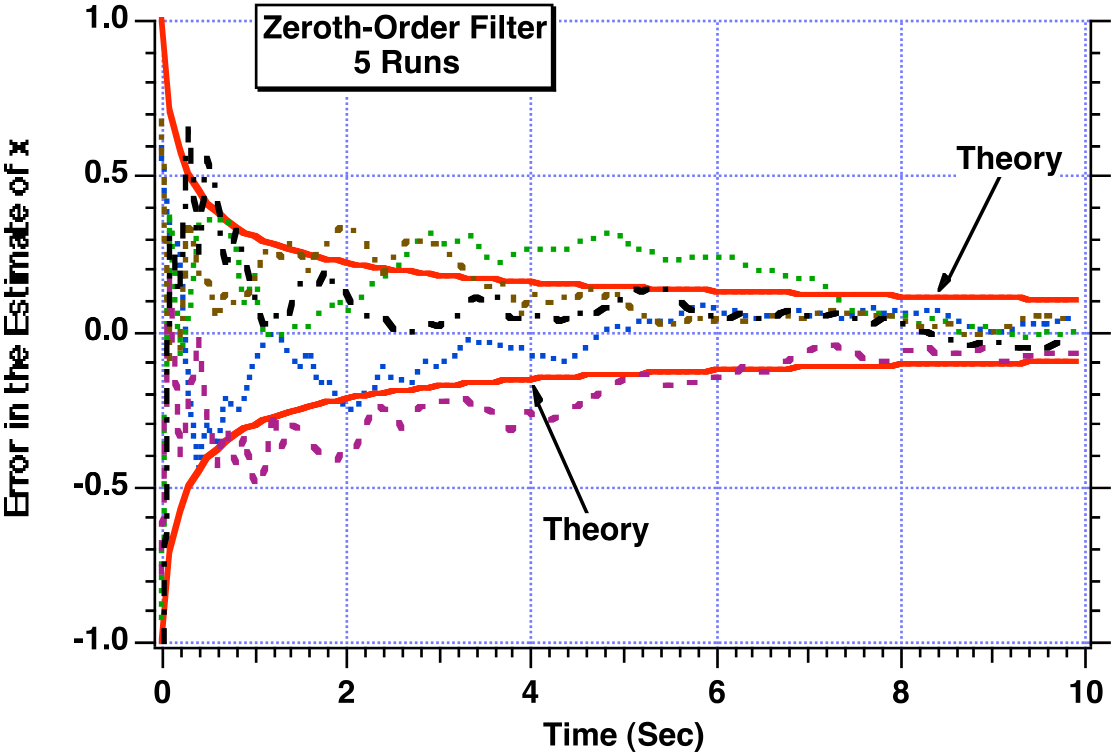
- **Less error in estimate due to measurement noise**
- **More error in estimate due to truncation error**

FORTRAN Monte Carlo Simulation for Testing Zeroth-Order Recursive Least Squares Filter

```
            GLOBAL DEFINE
                    INCLUDE 'quickdraw.inc'
            END
            IMPLICIT REAL*8(A-H,O-Z)
            OPEN(1,STATUS='UNKNOWN',FILE='DATFIL1')
            OPEN(2,STATUS='UNKNOWN',FILE='DATFIL2')
            OPEN(3,STATUS='UNKNOWN',FILE='DATFIL3')
            OPEN(4,STATUS='UNKNOWN',FILE='DATFIL4')
            OPEN(5,STATUS='UNKNOWN',FILE='DATFIL5')
            DO 11 K=1,5
            TS=.1
            SIGNOISE=1.
            A0=1.
            A1=0.
            XH=0.
            XN=0.
            DO 10 T=0.,10.,TS
            XN=XN+1.
            CALL GAUSS(XNOISE,SIGNOISE)
            ACT=A0+A1*T
            XS=ACT+XNOISE
            XK=1./XN
            RES=XS-XH
            XH=XH+XK*RES
            SP11=SIGNOISE/SQRT(XN)
            XHERR=ACT-XH
            EPS=.5*A1*TS*(XN-1)
            WRITE(9,*)T,XHERR,SP11,-SP11
            WRITE(K,*)T,XHERR,SP11,-SP11
10          CONTINUE
            CLOSE(K)
11          CONTINUE
            PAUSE
            END
```

Loop for making 5 runs

Monte Carlo Results Lie Within the Theoretical Bounds Approximately 68% of the Time



Theory

$$\sqrt{P_k} = \frac{\sigma_n}{\sqrt{k}}$$

First-Order Recursive Least Squares Filter

First-Order Recursive Filter Structure

Using techniques similar to those of the previous section, we can convert the batch processing first-order least squares filter to a recursive form. After much algebraic manipulation we obtain

Gains

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

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

Filter

$$Res_k = x_k^* - \hat{x}_{k-1} - \hat{\dot{x}}_{k-1}T_s$$

$$\hat{x}_k = \hat{x}_{k-1} + \hat{\dot{x}}_{k-1}T_s + K_{1_k}Res_k$$

$$\hat{\dot{x}}_k = \hat{\dot{x}}_{k-1} + K_{2_k}Res_k$$

Numerical Example For First-Order Filter-1

Recall from previous section measurement data given by

$$x_1^* = 1.2$$

$$x_2^* = .2$$

$$x_3^* = 2.9$$

$$x_4^* = 2.1$$

$$T_s = 1$$

Assume

$$\hat{x}_0 = 0$$

$$\hat{\dot{x}}_0 = 0$$

First iteration (k=1)

$$K_{1_1} = \frac{2(2k-1)}{k(k+1)} = \frac{2(2*1-1)}{1(1+1)} = 1$$

$$K_{2_1} = \frac{6}{k(k+1)T_s} = \frac{6}{1(1+1)*1} = 3$$

$$Res_1 = x_1^* - \hat{x}_0 - \hat{\dot{x}}_0 T_s = 1.2 - 0 - 0*1 = 1.2$$

$$\hat{x}_1 = \hat{x}_0 + \hat{\dot{x}}_0 T_s + K_{1_1} Res_1 = 0 + 0*1 + 1*1.2 = 1.2$$

$$\hat{\dot{x}}_1 = \hat{\dot{x}}_0 + K_{2_1} Res_1 = 0 + 3*1.2 = 3.6$$

Numerical Example For First-Order Filter-2

Second iteration (k=2)

$$K_{1_2} = \frac{2(2k-1)}{k(k+1)} = \frac{2(2*2-1)}{2(2+1)} = 1$$

$$K_{2_2} = \frac{6}{k(k+1)T_s} = \frac{6}{2(2+1)*1} = 1$$

$$Res_2 = x_2^* - \hat{x}_1 - \hat{\dot{x}}_1 T_s = .2 - 1.2 - 3.6*1 = -4.6$$

$$\hat{x}_2 = \hat{x}_1 + \hat{\dot{x}}_1 T_s + K_{1_2} Res_2 = 1.2 + 3.6*1 + 1*(-4.6) = .2$$

$$\hat{\dot{x}}_2 = \hat{\dot{x}}_1 + K_{2_2} Res_2 = 3.6 + 1*(-4.6) = -1$$

Third iteration (k=3)

$$K_{1_3} = \frac{2(2k-1)}{k(k+1)} = \frac{2(2*3-1)}{3(3+1)} = \frac{5}{6}$$

$$K_{2_3} = \frac{6}{k(k+1)T_s} = \frac{6}{3(3+1)*1} = .5$$

$$Res_3 = x_3^* - \hat{x}_2 - \hat{\dot{x}}_2 T_s = 2.9 - .2 - (-1)*1 = 3.7$$

$$\hat{x}_3 = \hat{x}_2 + \hat{\dot{x}}_2 T_s + K_{1_3} Res_3 = .2 + (-1)*1 + \frac{5}{6}*3.7 = 2.28$$

$$\hat{\dot{x}}_3 = \hat{\dot{x}}_2 + K_{2_3} Res_3 = -1 + .5*3.7 = .85$$

Numerical Example For First-Order Filter-3

Last iteration (k=4)

$$K_{1_4} = \frac{2(2k-1)}{k(k+1)} = \frac{2(2*4-1)}{4(4+1)} = .7$$

$$K_{2_4} = \frac{6}{k(k+1)T_s} = \frac{6}{4(4+1)*1} = .3$$

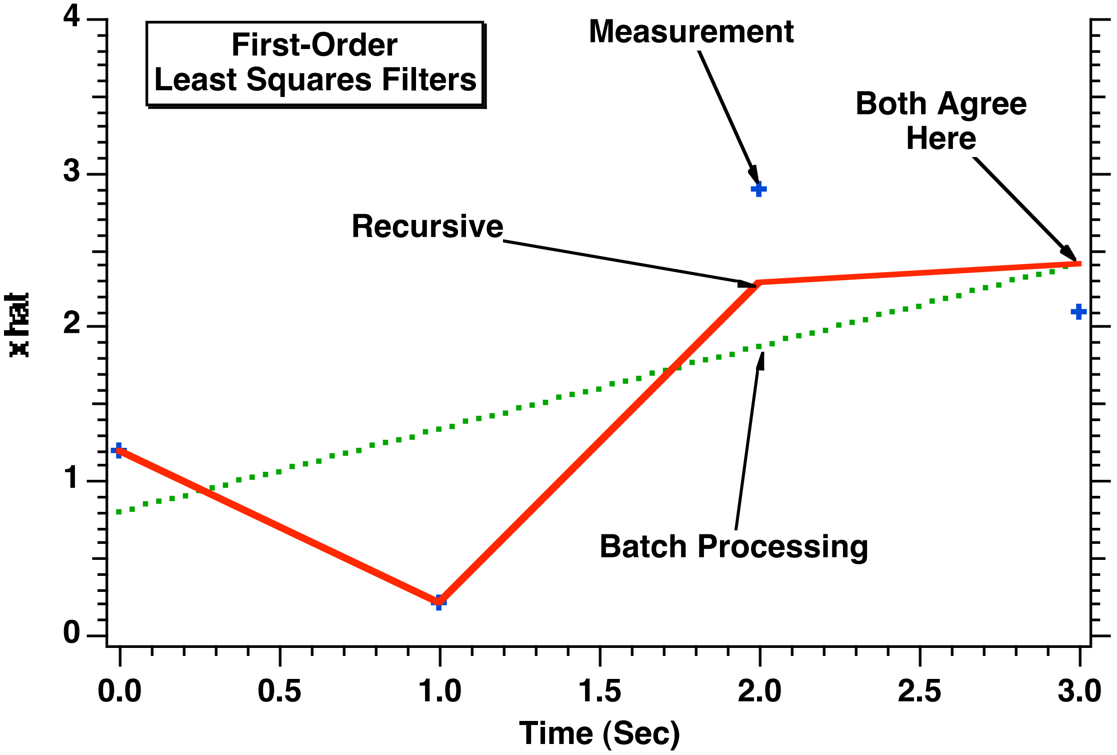
$$Res_4 = x_4^* - \hat{x}_3 - \hat{\dot{x}}_3 T_s = 2.1 - 2.28 - .85*1 = -1.03$$

$$\hat{x}_4 = \hat{x}_3 + \hat{\dot{x}}_3 T_s + K_{1_4} Res_4 = 2.28 + .85*1 +.7*(-1.03) = 2.41$$

Same answer as obtained with first-order batch processing filter

$$\hat{\dot{x}}_4 = \hat{\dot{x}}_3 + K_{2_4} Res_4 = .85 + .3*(-1.03) = .54$$

First-Order Recursive and Batch Processing Least Squares Filters Yield the Same Answers After All Measurements are Taken



Important Performance Formulas For First-Order Filter

The following formulas are stated but are not derived

Variance of error in estimate due to measurement noise

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

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

Error in estimate due to truncation error

$$x_k^* = a_0 + a_1 t + a_2 t^2 = a_0 + a_1(k-1)T_s + a_2(k-1)^2 T_s{}^2$$ ← **Given second-order signal**

$$\varepsilon_k = \frac{1}{6}a_2 T_s{}^2(k-1)(k-2)$$

$$\dot{\varepsilon}_k = a_2 T_s(k-1)$$

FORTRAN Simulation For Testing First-Order Recursive Least Squares Filter

```
   GLOBAL DEFINE
                  INCLUDE 'quickdraw.inc'
   END
   IMPLICIT REAL*8(A-H,O-Z)
   TS=.1
   SIGNOISE=5.
   OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
   OPEN(2,STATUS='UNKNOWN',FILE='COVFIL')
   A0=3.
   A1=1.
   A2=0.
   XH=0.
   XDH=0.
   XN=0
   DO 10 T=0.,10.,TS
   XN=XN+1.
   CALL GAUSS(XNOISE,SIGNOISE)
   X=A0+A1*T+A2*T*T
   XD=A1+2*A2*T
   XS=X+XNOISE
   XK1=2*(2*XN-1)/(XN*(XN+1))
   XK2=6/(XN*(XN+1)*TS)
   RES=XS-XH-TS*XDH
   XH=XH+XDH*TS+XK1*RES
   XDH=XDH+XK2*RES
   IF(XN.EQ.1)THEN
                  LET SP11=0
                  LET SP22=0
   ELSE
                  SP11=SIGNOISE*SQRT(2.*(2*XN-1)/(XN*(XN+1)))
                  SP22=SIGNOISE*SQRT(12/(XN*(XN*XN-1)*TS*TS))
   ENDIF
   XHERR=X-XH
   XDHERR=XD-XDH
   EPS=A2*TS*TS*(XN-1)*(XN-2)/6
   EPSD=A2*TS*(XN-1)
   WRITE(9,*)T,X,XS,XH,XD,XDH
   WRITE(1,*)T,X,XS,XH,XD,XDH
   WRITE(2,*)T,XHERR,SP11,-SP11,EPS,XDHERR,SP22,-SP22,EPSD
10 CONTINUE
   CLOSE(1)
   CLOSE(2)
   PAUSE
   END
```

Standard deviation of noise

Polynomial coefficients of signal

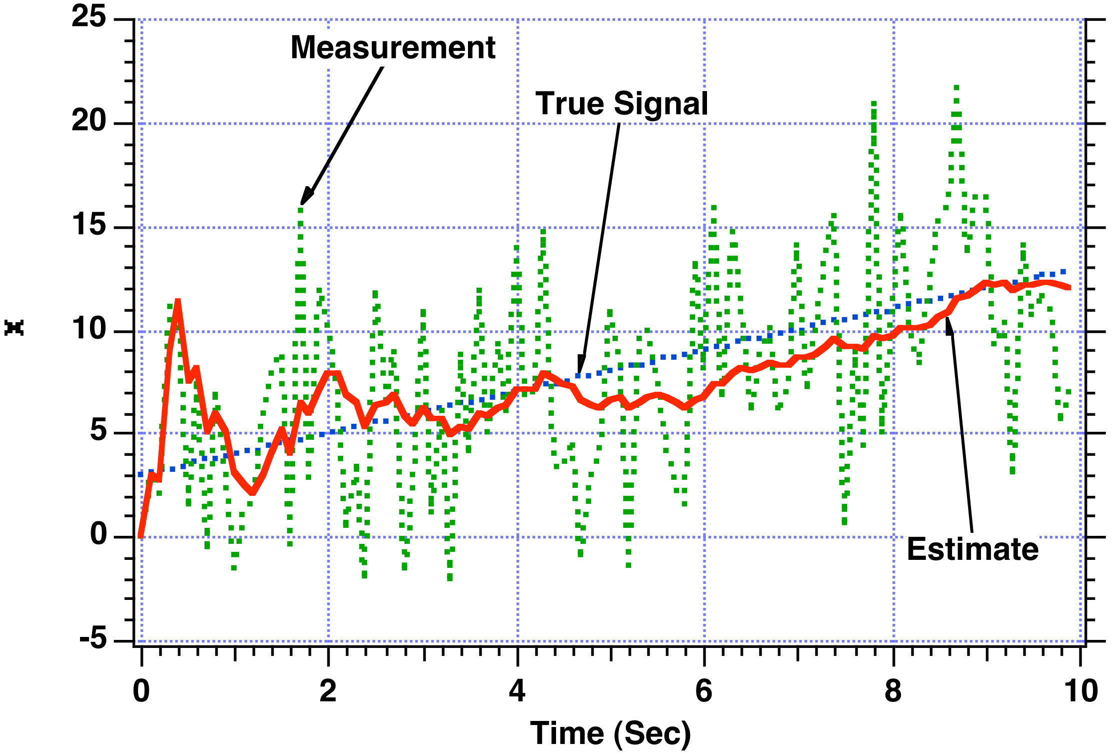
Signal and derivative

Measurement

Recursive filter

Actual errors in estimate

First-Order Recursive Least Squares Filter is Able to Track First-Order Signal Plus Noise

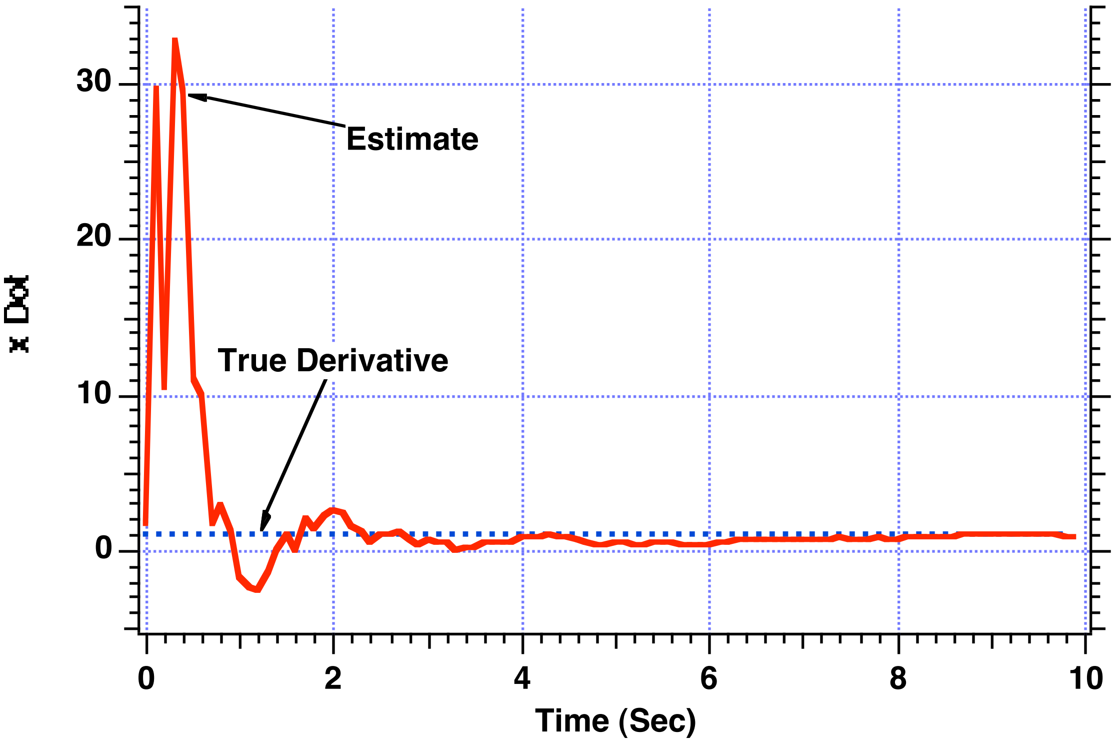


Measurement

$x^* = 3 + t + \text{noise}$

$\sigma_{\text{noise}} = 5$

First-Order Recursive Least Squares Filter is Able to Estimate Derivative of Signal

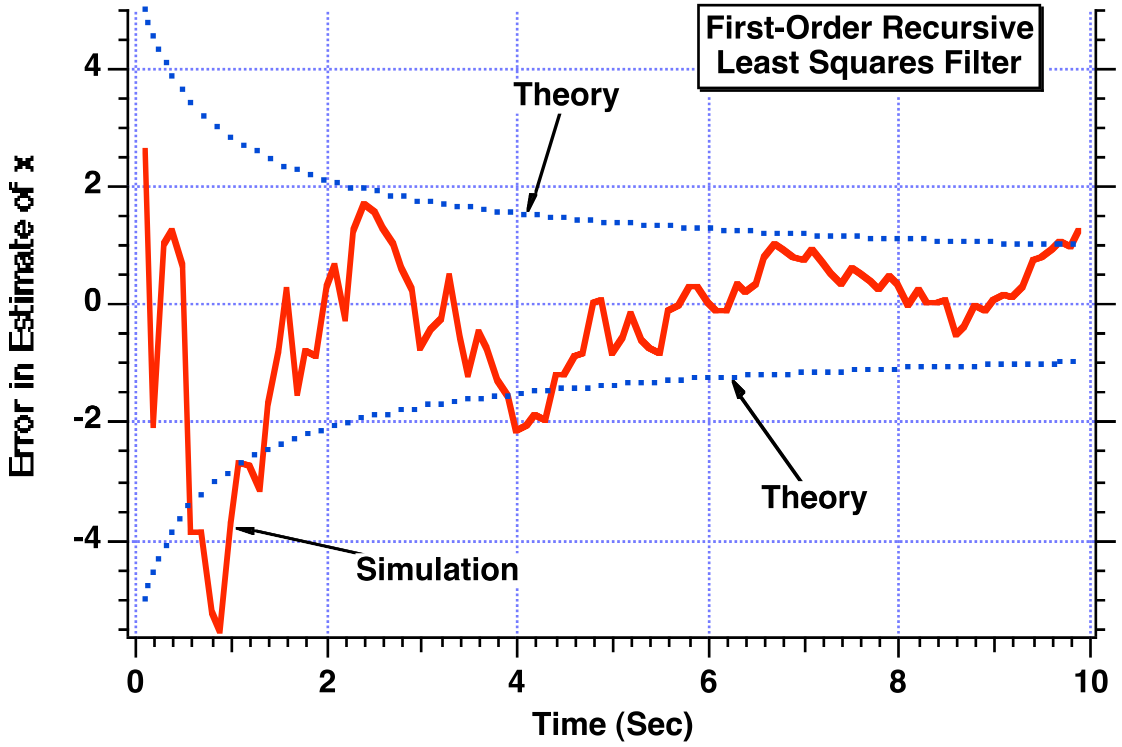


Measurement

$x^* = 3 + t + \text{noise}$

$\sigma_{\text{noise}} = 5$

Single Run Simulation Results For First State Agree With Theoretical Formula



Theory

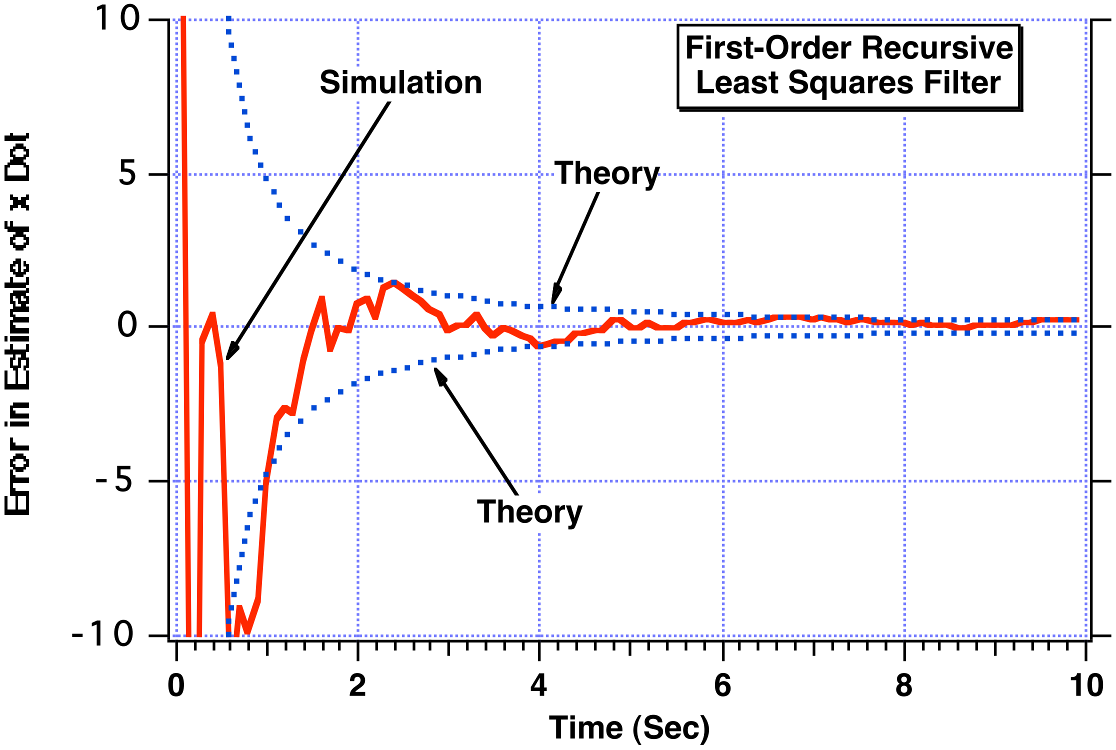
$$\sqrt{P_{11_k}} = \sqrt{\frac{2(2k-1)\sigma_n^2}{k(k+1)}}$$

Measurement

$$x^* = 3 + t + \text{noise}$$

$$\sigma_{\text{noise}} = 5$$

Single Run Simulation Results For Second State Also Agrees With Theoretical Formula



Theory

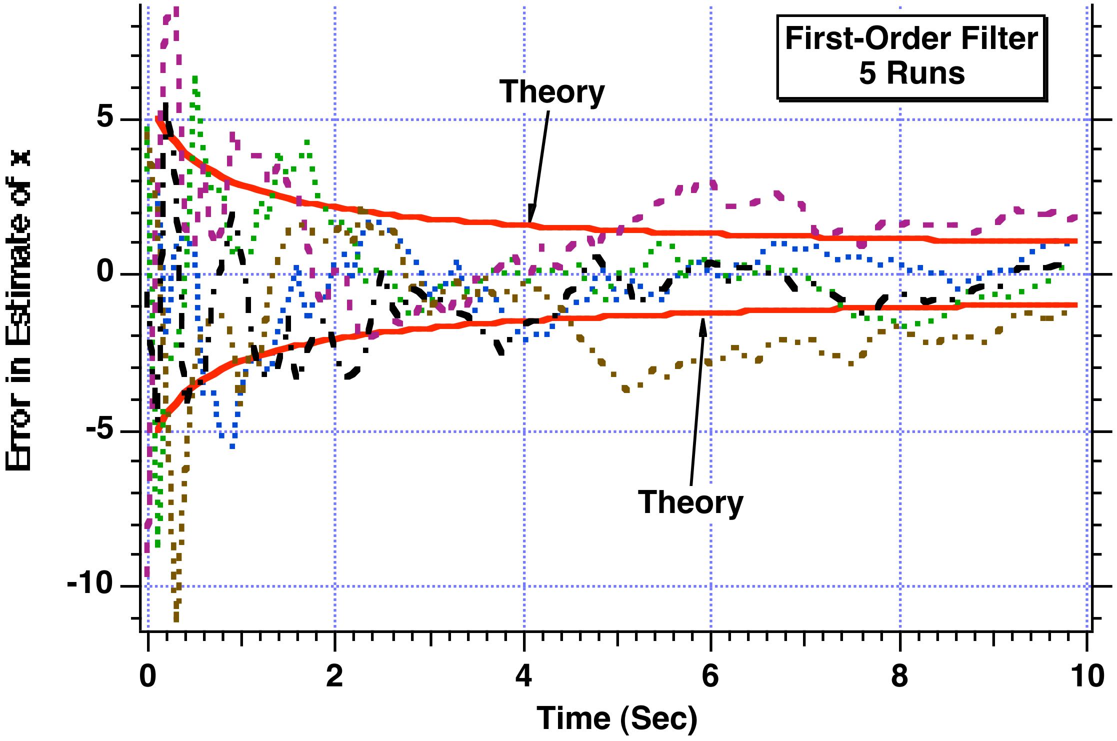
$$\sqrt{P_{22_k}} = \sqrt{\frac{12\sigma_n^2}{k(k^2-1)T_s^2}}$$

Measurement

$x^* = 3 + t + \text{noise}$

$\sigma_{\text{noise}} = 5$

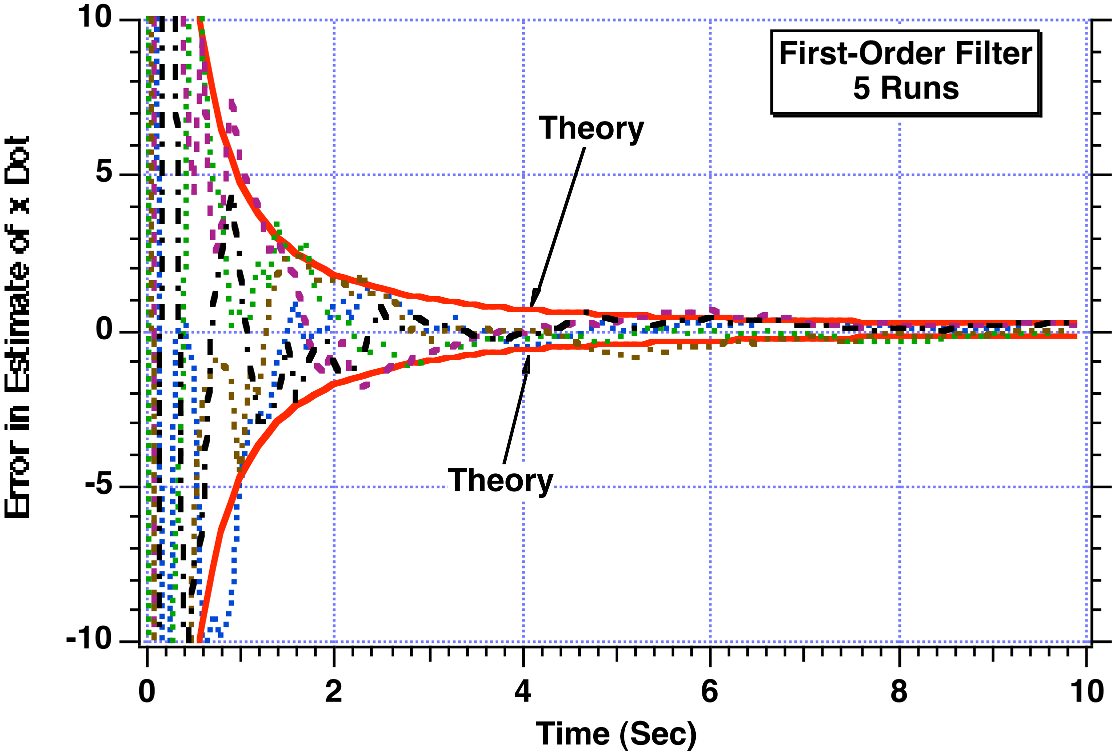
Simulated Error in the Estimates of First State Appear to Lie Within Theoretical Error Bounds 68% of the Time



Theory

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

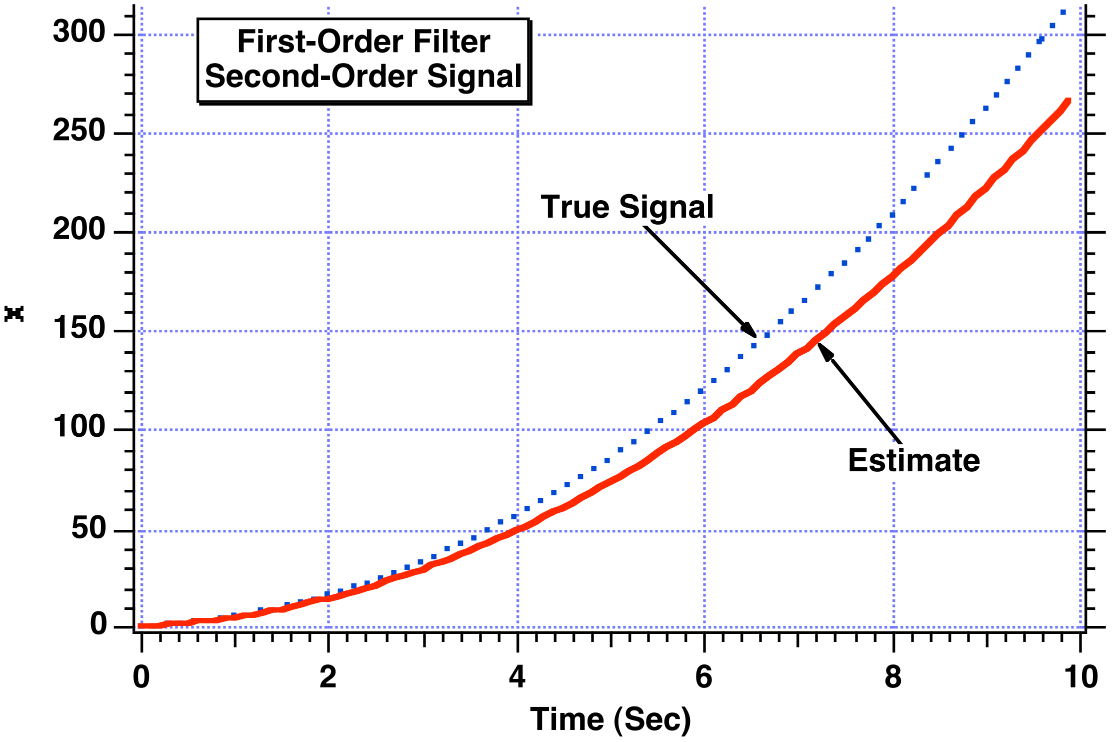
Simulated Error in the Estimates of Second State Appear to Lie Within Theoretical Error Bounds 68% of the Time



Theory

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

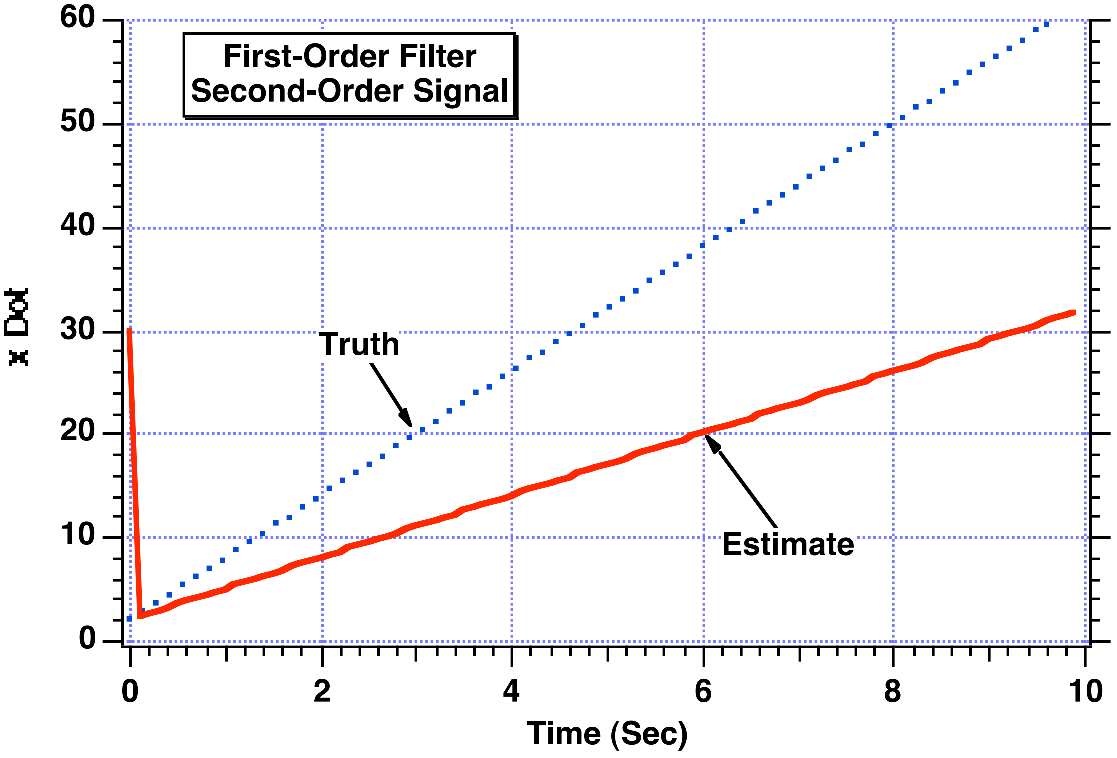
First-Order Recursive Least Squares Filter is Unable to Track the First State of a Second-Order Polynomial



Measurement

$x^* = 1 + 2t + 3t^2$

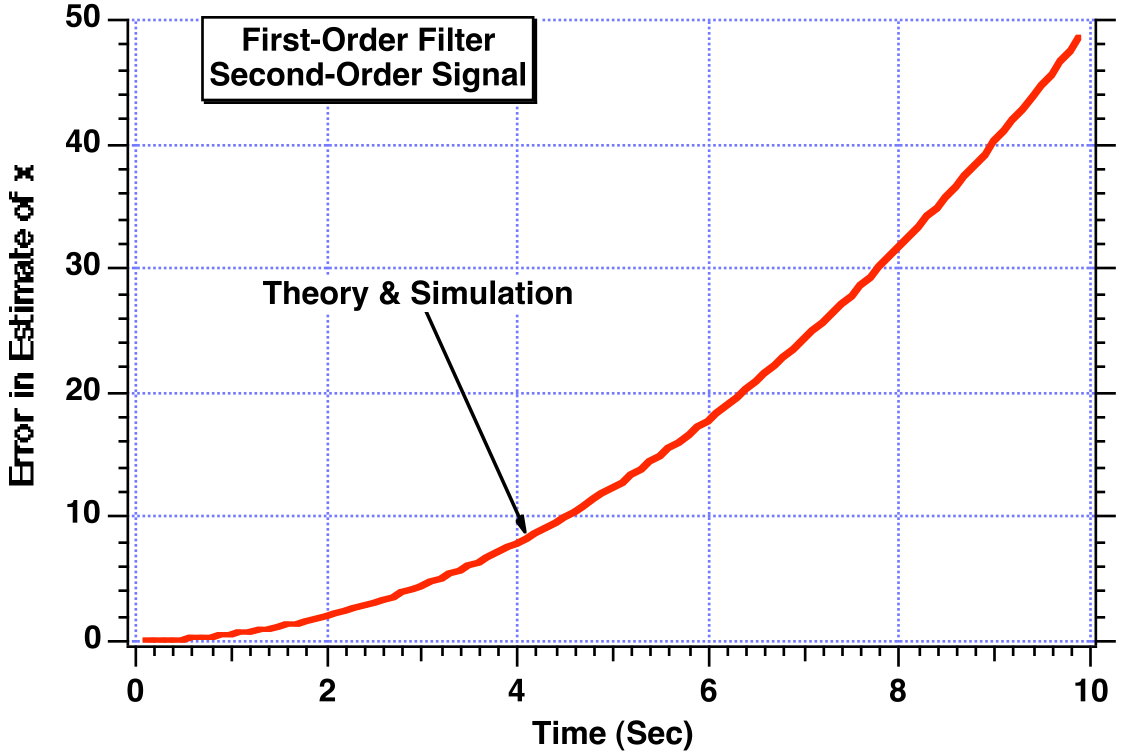
First-Order Recursive Least Squares Filter is Unable to Track the Second State of a Second-Order Polynomial



Measurement

$x^* = 1 + 2t + 3t^2$

Simulation Results and Truncation Error for First State are in Excellent Agreement



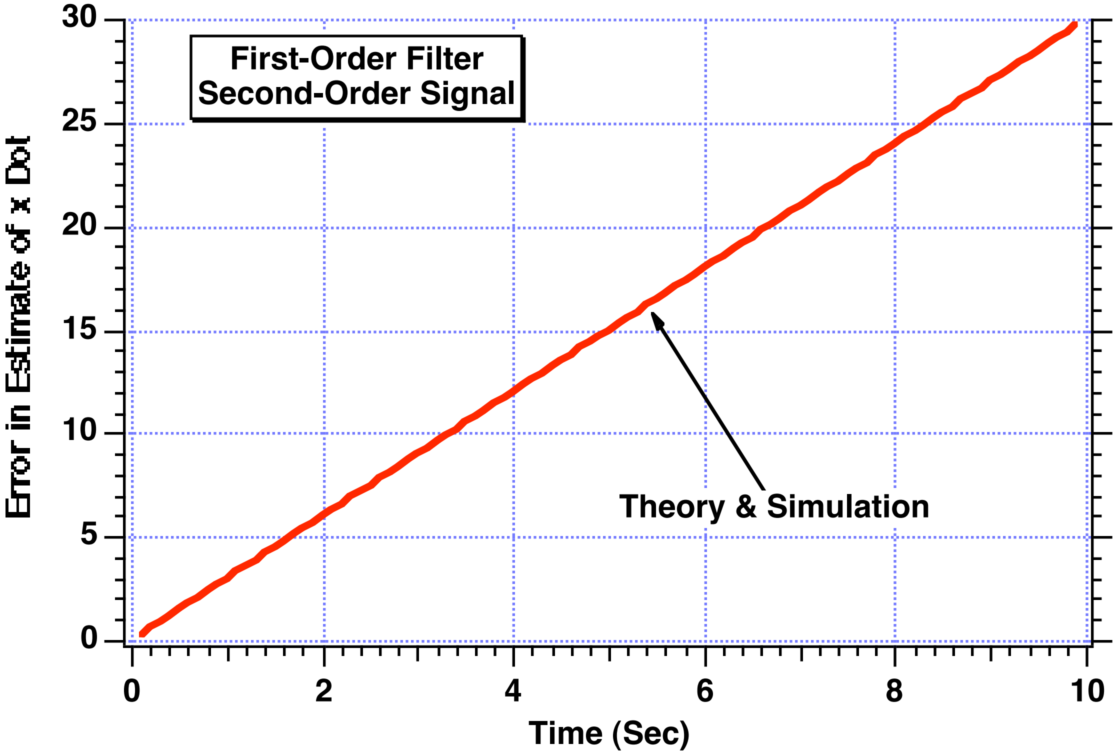
Theory

$$\epsilon_k = \frac{1}{6}a_2 T_s^2(k-1)(k-2)$$

Measurement

$$x^* = 1 + 2t + 3t^2$$

Simulation Results and Truncation Error for Second State are in Excellent Agreement



Theory

$\dot{\varepsilon}_k = a_2 T_s(k-1)$

Measurement

$x^* = 1 + 2t + 3t^2$

Second-Order Recursive Least Squares Filter

Second-Order Recursive Filter Structure

Using techniques similar to those of the first section, we can convert the batch processing second-order least squares filter to a recursive form. After much algebraic manipulation we obtain

Gains

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

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

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

Filter

$$Res_k = x_k^* - \hat{x}_{k-1} - \hat{\dot{x}}_{k-1}T_s - .5\hat{\ddot{x}}_{k-1}T_s^2$$

$$\hat{x}_k = \hat{x}_{k-1} + \hat{\dot{x}}_{k-1}T_s + .5\hat{\ddot{x}}_{k-1}T_s^2 + K_{1_k}Res_k$$

$$\hat{\dot{x}}_k = \hat{\dot{x}}_{k-1} + \hat{\ddot{x}}_{k-1}T_s^2 + K_{2_k}Res_k$$

$$\hat{\ddot{x}}_k = \hat{\ddot{x}}_{k-1} + K_{3_k}Res_k$$

Numerical Example For Second-Order Filter-1

Recall from previous Lecture measurement data given by

$$x_1^* = 1.2$$

$$x_2^* = .2$$

$$x_3^* = 2.9$$

$$x_4^* = 2.1$$

$$T_s = 1$$

Assume

$$\hat{x}_0 = 0$$

$$\hat{\dot{x}}_0 = 0$$

$$\hat{\ddot{x}}_0 = 0$$

First iteration (k=1)

$$K_{1_1} = \frac{3(3k^2\text{-}3k+2)}{k(k+1)(k+2)} = \frac{3(3*1\text{-}3*1+2)}{1(2)(3)} = 1$$

$$K_{2_1} = \frac{18(2k\text{-}1)}{k(k+1)(k+2)T_s} = \frac{18(2\text{-}1)}{1(2)(3)(1)} = 3$$

$$K_{3_1} = \frac{60}{k(k+1)(k+2)T_s^2} = \frac{60}{1(2)(3)(1)} = 10$$

$$Res_1 = x_1^* - \hat{x}_0 - \hat{\dot{x}}_0 T_s - .5\hat{\ddot{x}}_0 T_s^2 = 1.2 - 0 - 0 - 0 = 1.2$$

$$\hat{x}_1 = \hat{x}_0 + \hat{\dot{x}}_0 T_s + .5\hat{\ddot{x}}_0 T_s^2 + K_{1_1} Res_1 = 0 + 0 + 0 + 1*1.2 = 1.2$$

$$\hat{\dot{x}}_1 = \hat{\dot{x}}_0 + \hat{\ddot{x}}_0 T_s + K_{2_1} Res_1 = 0 + 0 + 3*1.2 = 3.6$$

$$\hat{\ddot{x}}_1 = \hat{\ddot{x}}_0 + K_{3_1} Res_1 = 0 + 10*1.2 = 12$$

Numerical Example For Second-Order Filter-2

Second iteration (k=2)

$$K_{1_2} = \frac{3(3k^2-3k+2)}{k(k+1)(k+2)} = \frac{3(3*4-3*2+2)}{2(3)(4)} = 1$$

$$K_{2_2} = \frac{18(2k-1)}{k(k+1)(k+2)T_s} = \frac{18(2*2-1)}{2(3)(4)(1)} = 2.25$$

$$K_{3_2} = \frac{60}{k(k+1)(k+2)T_s^2} = \frac{60}{2(3)(4)(1)} = 2.5$$

$$Res_2 = x_2^* - \hat{x}_1 - \hat{\dot{x}}_1 T_s - .5\hat{\ddot{x}}_1 T_s^2 = .2 - 1.2 - 3.6 - .5*12 = -10.6$$

$$\hat{x}_2 = \hat{x}_1 + \hat{\dot{x}}_1 T_s + .5\hat{\ddot{x}}_1 T_s^2 + K_{1_2} Res_2 = 1.2 + 3.6 + .5*12 + 1*(-10.6) = .2$$

$$\hat{\dot{x}}_2 = \hat{\dot{x}}_1 + \hat{\ddot{x}}_1 T_s + K_{2_2} Res_2 = 3.6 + 12 + 2.25*(-10.6) = -8.25$$

$$\hat{\ddot{x}}_2 = \hat{\ddot{x}}_1 + K_{3_2} Res_2 = 12 + 2.5*(-10.6) = -14.5$$

Numerical Example For Second-Order Filter-3

Third iteration (k=3)

$$K_{1_3} = \frac{3(3k^2-3k+2)}{k(k+1)(k+2)} = \frac{3(3*9-3*3+2)}{3(4)(5)} = 1$$

$$K_{2_3} = \frac{18(2k-1)}{k(k+1)(k+2)T_s} = \frac{18(2*3-1)}{3(4)(5)(1)} = 1.5$$

$$K_{3_3} = \frac{60}{k(k+1)(k+2)T_s^2} = \frac{60}{3(4)(5)(1)} = 1$$

$$Res_3 = x_3^* - \hat{x}_2 - \hat{\dot{x}}_2 T_s - .5\hat{\ddot{x}}_2 T_s^2 = 2.9 - .2 - (-8.25) - .5*(-14.5) = 18.2$$

$$\hat{x}_3 = \hat{x}_2 + \hat{\dot{x}}_2 T_s + .5\hat{\ddot{x}}_2 T_s^2 + K_{1_3} Res_3 = .2 - 8.25 + .5*(-14.5) +1*18.2 = 2.9$$

$$\hat{\dot{x}}_3 = \hat{\dot{x}}_2 + \hat{\ddot{x}}_2 T_s + K_{2_3} Res_3 = -8.25 - 14.5 + 1.5*18.2 = 4.55$$

$$\hat{\ddot{x}}_3 = \hat{\ddot{x}}_2 + K_{3_3} Res_3 = -14.5 + 1*18.2 = 3.7$$

Numerical Example For Second-Order Filter-4

Last iteration (k=4)

$$K_{1_4} = \frac{3(3k^2-3k+2)}{k(k+1)(k+2)} = \frac{3(3*16-3*4+2)}{4(5)(6)} = \frac{19}{20}$$

$$K_{2_4} = \frac{18(2k-1)}{k(k+1)(k+2)T_s} = \frac{18(2*4-1)}{4(5)(6)(1)} = \frac{21}{20}$$

$$K_{3_4} = \frac{60}{k(k+1)(k+2)T_s^2} = \frac{60}{4(5)(6)(1)} = .5$$

$$Res_4 = x_4^* - \hat{x}_3 - \hat{\dot{x}}_3 T_s - .5\hat{\ddot{x}}_3 T_s^2 = 2.1 - 2.9 - 4.55 - .5*3.7 = -7.2$$

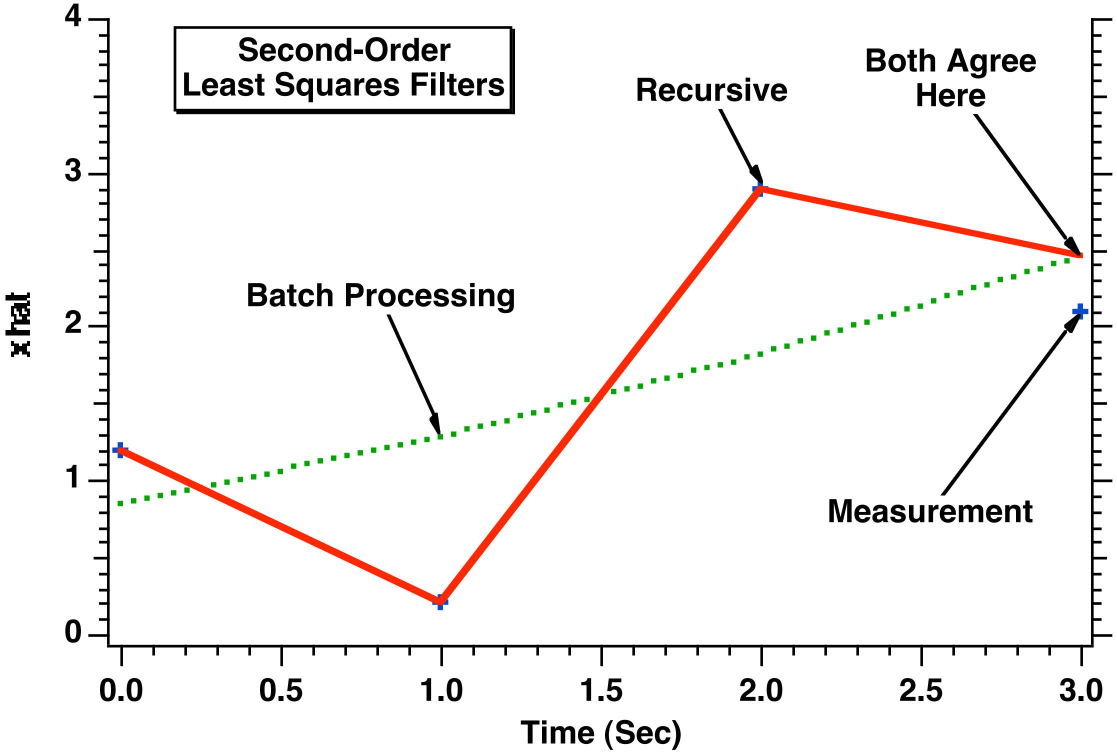
$$\hat{x}_4 = \hat{x}_3 + \hat{\dot{x}}_3 T_s + .5\hat{\ddot{x}}_3 T_s^2 + K_{1_4} Res_4 = 2.9 + 4.55 + .5*3.7 + \frac{19}{20}*(-7.2) = 2.46$$

Same answer as obtained with second-order batch processing filter

$$\hat{\dot{x}}_4 = \hat{\dot{x}}_3 + \hat{\ddot{x}}_3 T_s + K_{2_4} Res_4 = 4.55 + 3.7*1 + \frac{21}{20}*(-7.2) = .69$$

$$\hat{\ddot{x}}_4 = \hat{\ddot{x}}_3 + K_{3_4} Res_4 = 3.7 + .5*(-7.2) = .1$$

Recursive and Batch Processing Second-Order Least Squares Filters Yield the Same Answers After all the Measurements are Taken



Important Performance Formulas For Second-Order Filter

The following formulas are stated but are not derived

Variance of error in estimate due to measurement noise

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

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

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

Error in estimate due to truncation error

$$x_k^* = a_0 + a_1 t + a_2 t^2 + a_3 t^3 = a_0 + a_1(k-1)T_s + a_2(k-1)^2 T_s{}^2 + a_3(k-1)^3 T_s{}^3$$

Given third-order signal

$$\varepsilon_k = \frac{1}{20} a_3 T_s{}^3 (k-1)(k-2)(k-3)$$

$$\dot{\varepsilon}_k = \frac{1}{10} a_3 T_s^2 (6k^2-15k+11)$$

$$\ddot{\varepsilon}_k = 3a_3 T_s (k-1)$$

FORTRAN Simulation for Testing Second-Order Recursive Least Squares Filter - 1

```
GLOBAL DEFINE
            INCLUDE 'quickdraw.inc'
END
IMPLICIT REAL*8(A-H,O-Z)
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
OPEN(2,STATUS='UNKNOWN',FILE='COVFIL')
TS=.1
SIGNOISE=50.
A0=2.
A1=-2.
A2=5.
A3=0.
XH=0.
XDH=0.
XDDH=0.
XN=0.
DO 10 T=0,10.,TS
XN=XN+1.
CALL GAUSS(XNOISE,SIGNOISE)
X=A0+A1*T+A2*T*T+A3*T*T*T
XD=A1+2*A2*T+3.*A3*T*T
XDD=2*A2+6*A3*T
XS=X+XNOISE
XK1=3*(3*XN*XN-3*XN+2)/(XN*(XN+1)*(XN+2))
XK2=18*(2*XN-1)/(XN*(XN+1)*(XN+2)*TS)
XK3=60/(XN*(XN+1)*(XN+2)*TS*TS)
RES=XS-XH-TS*XDH-.5*TS*TS*XDDH
XH=XH+XDH*TS+.5*XDDH*TS*TS+XK1*RES
XDH=XDH+XDDH*TS+XK2*RES
XDDH=XDDH+XK3*RES
```

Standard deviation of noise

Polynomial coefficients of signal

Signal and it’s derivatives

Measurement

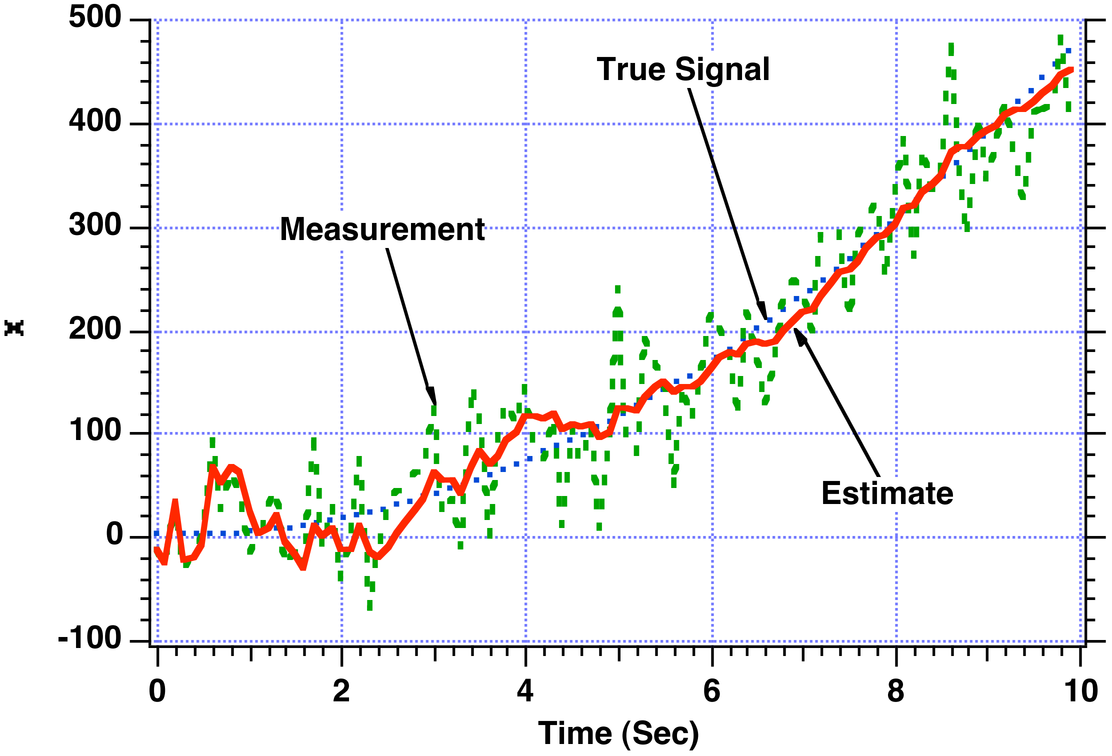
Recursive filter

FORTRAN Simulation for Testing Second-Order Recursive Least Squares Filter - 2

```
        IF(XN.EQ.1.OR.XN.EQ.2)THEN
                SP11=0
                SP22=0
                SP33=0
        ELSE
                SP11=SIGNOISE*SQRT(3*(3*XN*XN-3*XN+2)/(XN*(XN+1)*
     1                  (XN+2)))
                SP22=SIGNOISE*SQRT(12*(16*XN*XN-30*XN+11)/
     1                  (XN*(XN*XN-1)*(XN*XN-4)*TS*TS))
                SP33=SIGNOISE*SQRT(720/(XN*(XN*XN-1)*(XN*XN-4)
     1                  *TS*TS*TS*TS))
        ENDIF
        XHERR=X-XH
        XDHERR=XD-XDH
        XDDHERR=XDD-XDDH
        EPS=A3*TS*TS*TS*(XN-1)*(XN-2)*(XN-3)/20
        EPSD=A3*TS*TS*(6*XN*XN-15*XN+11)/10
        EPSDD=3*A3*TS*(XN-1)
        WRITE(9,*)T,X,XS,XH,XD,XDH,XDD,XDDH
        WRITE(1,*)T,X,XS,XH,XD,XDH,XDD,XDDH
        WRITE(2,*)T,XHERR,SP11,-SP11,EPS,XDHERR,SP22,-SP22,EPSD,
     1          XDDHERR,SP33,-SP33,EPSDD
 10     CONTINUE
        CLOSE(1)
        CLOSE(2)
        PAUSE
        END
```

Actual errors in estimates

Second-Order Recursive Filter is Able to Track Second-Order Signal Plus Noise

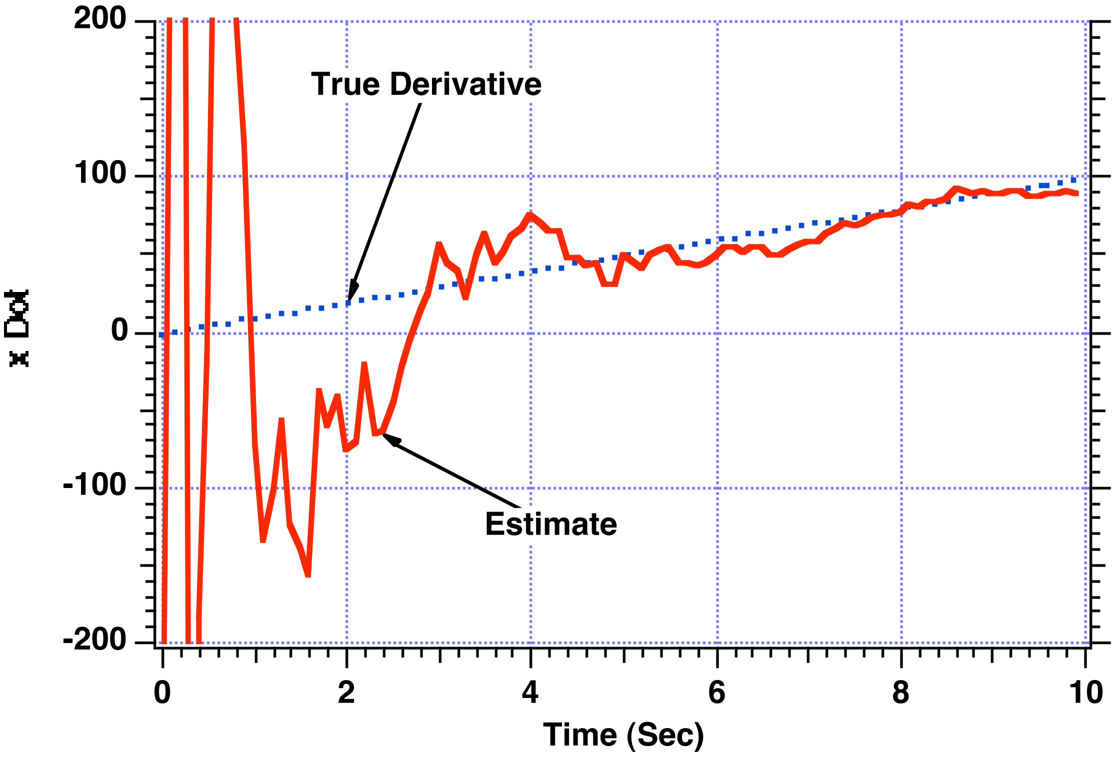


Measurement

$x^* = 2 - 2t + 5t^2 + \text{noise}$

$\sigma_{noise} = 50$

Estimate of Derivative is Excellent

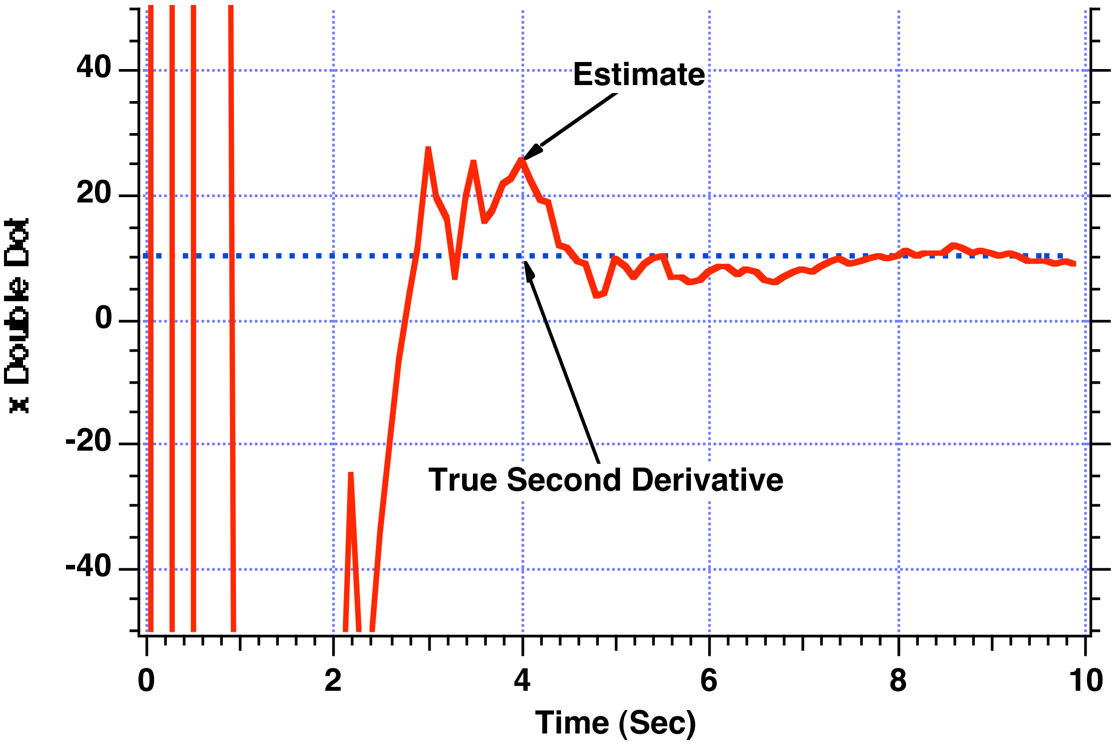


Measurement

$x^* = 2 - 2t + 5t^2 + \text{noise}$

$\sigma_{noise} = 50$

Estimate of Second Derivative is Also Excellent

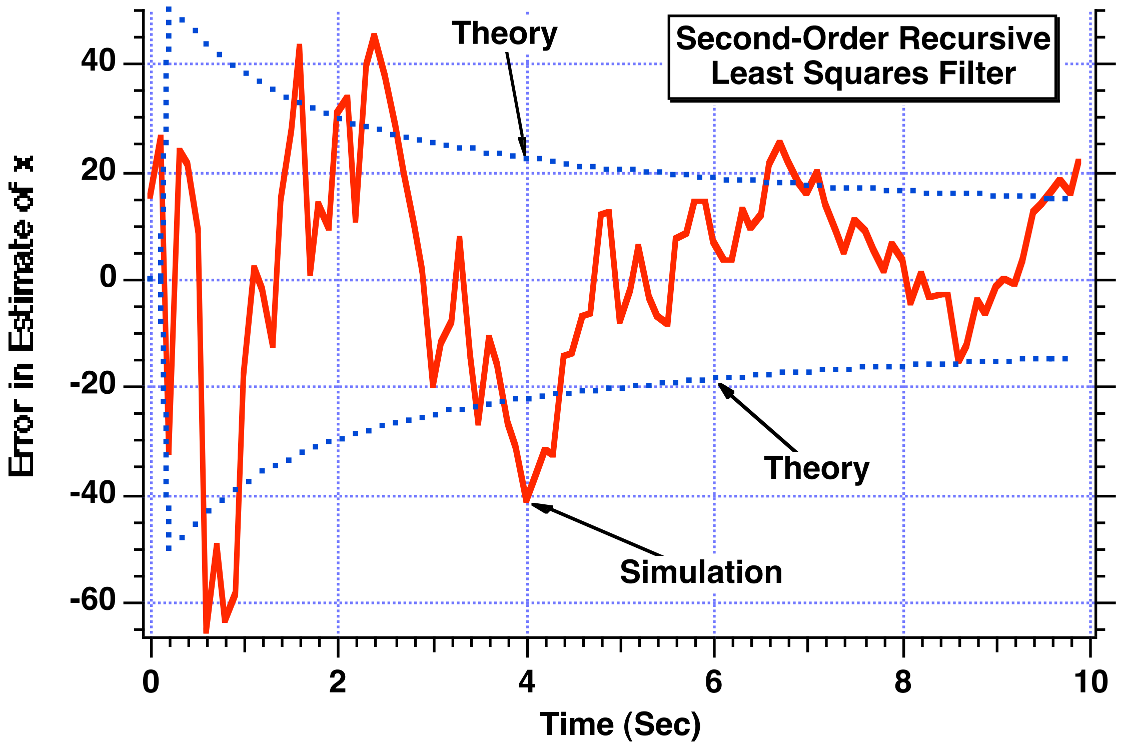


Measurement

$x^* = 2 - 2t + 5t^2 + \text{noise}$

$\sigma_{noise} = 50$

Error in Estimate of First State Appears to be Within Theoretical Error Bounds



Measurement

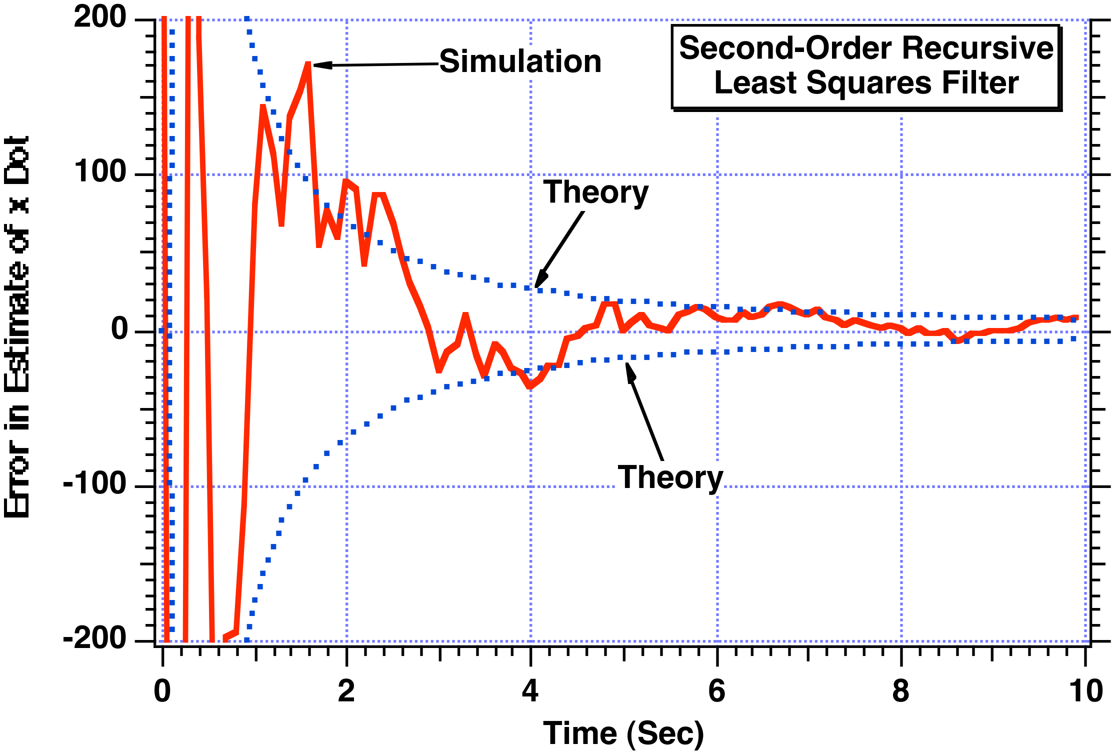
$$x^* = 2 - 2t + 5t^2 + \text{noise}$$

$$\sigma_{noise} = 50$$

Theory

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

Error in Estimate of Second State Appears to be Within Theoretical Error Bounds



Measurement

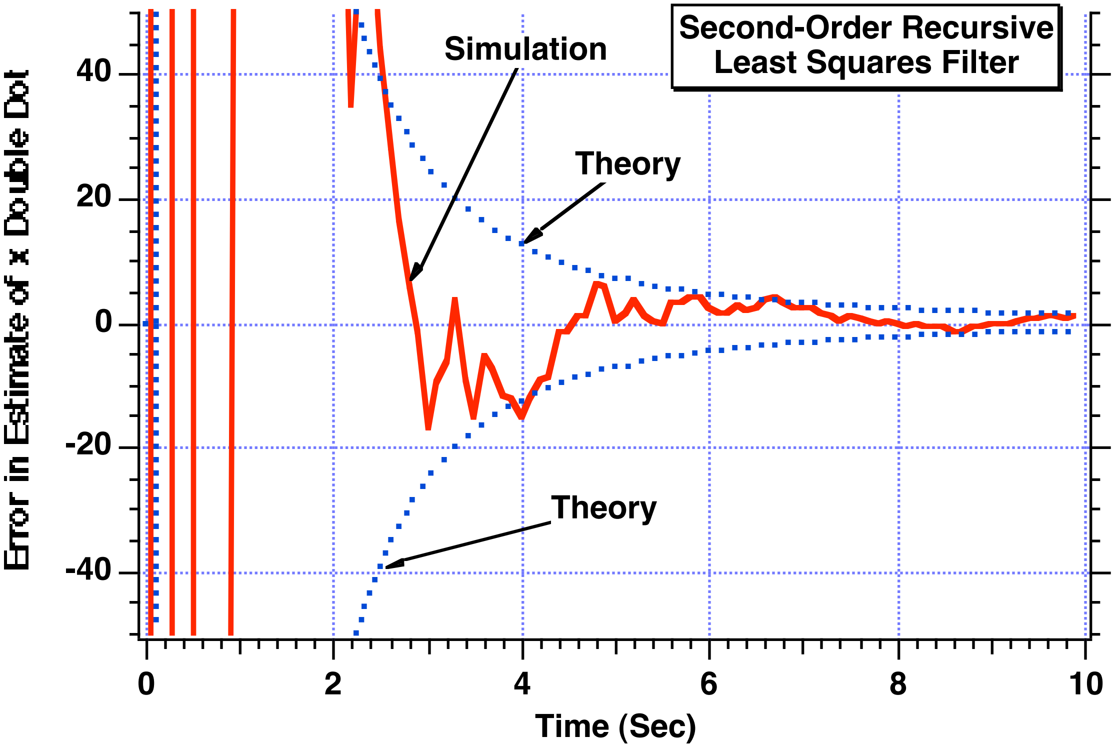
$x^* = 2 - 2t + 5t^2 + \text{noise}$

$\sigma_{\text{noise}} = 50$

Theory

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

Error in Estimate of Third State Appears to be Within Theoretical Error Bounds



Measurement

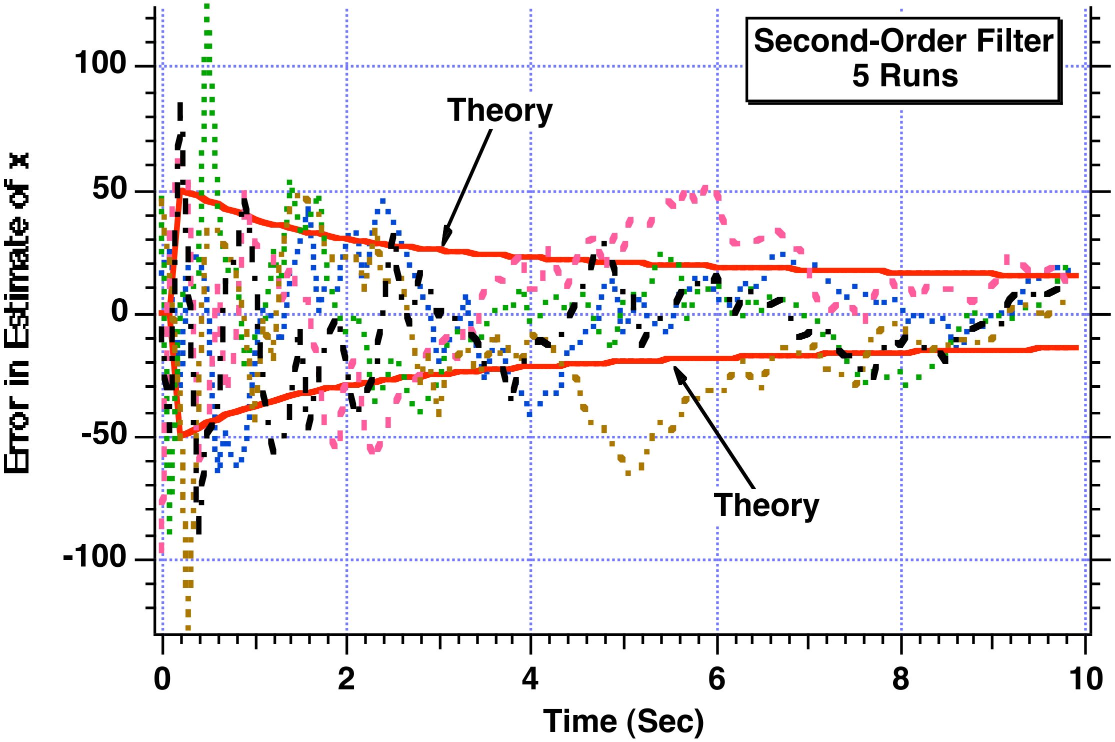
$$x^* = 2 - 2t + 5t^2 + \text{noise}$$

$$\sigma_{noise} = 50$$

Theory

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

Multiple Runs Indicate That on Average the Error in the Estimate of First State Appears to be Within Error Bounds 68% of the Time



Measurement

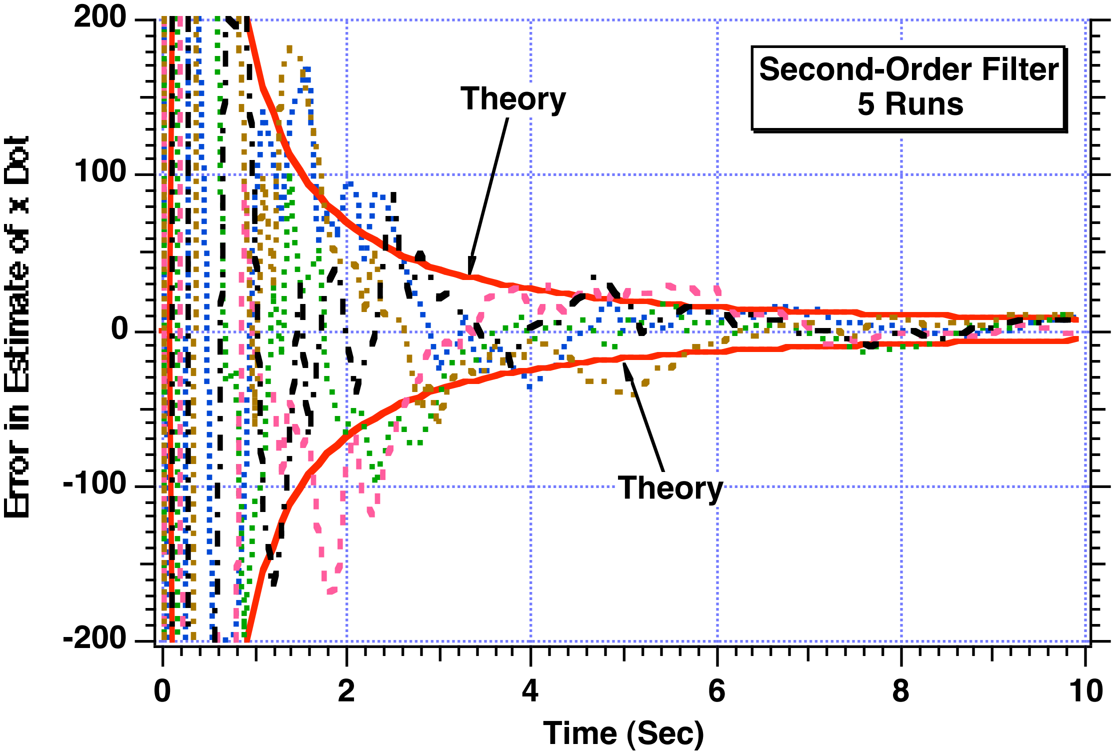
$$x^* = 2 - 2t + 5t^2 + \text{noise}$$

$$\sigma_{\text{noise}} = 50$$

Theory

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

Multiple Runs Indicate That on Average the Error in the Estimate of Second State Appears to be Within Error Bounds 68% of the Time



Measurement

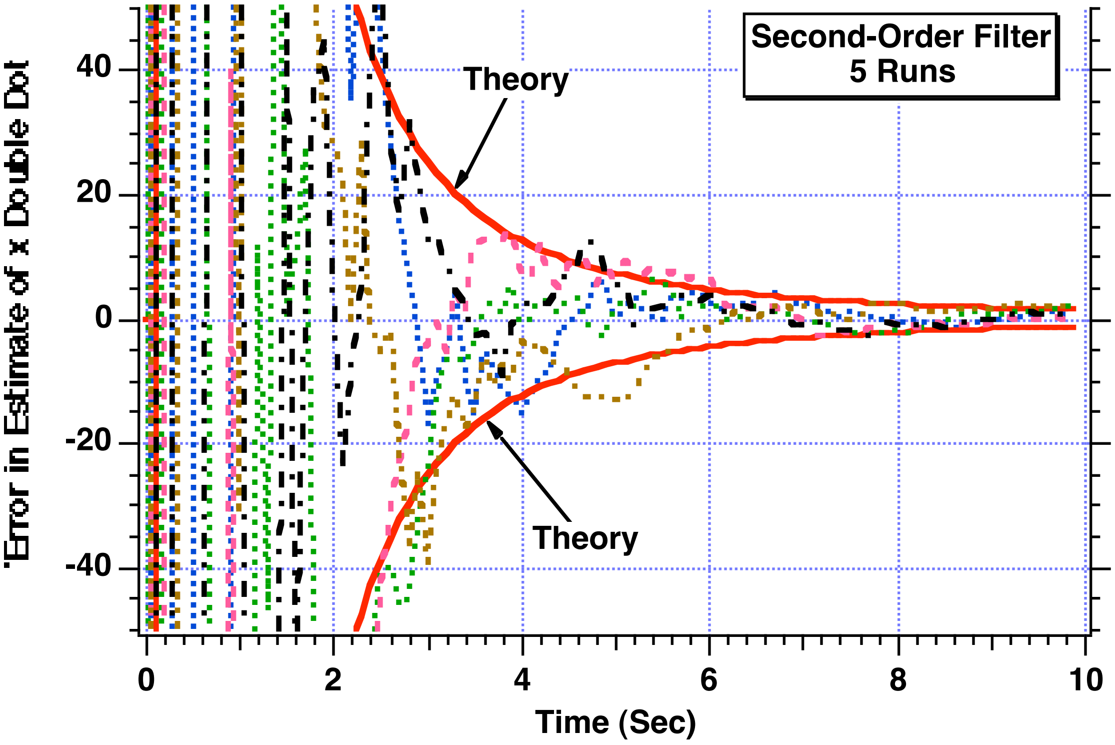
$x^* = 2 - 2t + 5t^2 + \text{noise}$

$\sigma_{noise} = 50$

Theory

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

Multiple Runs Indicate That on Average the Error in the Estimate of Third State Appears to be Within Error Bounds 68% of the Time



Measurement

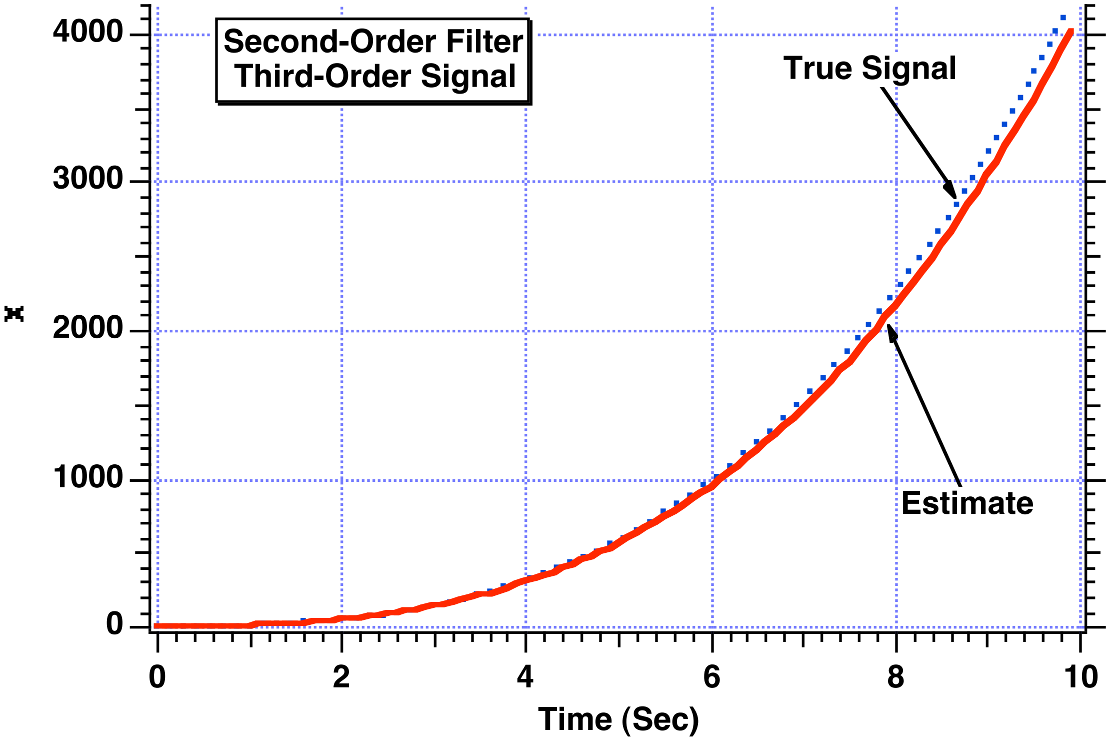
$x^* = 2 - 2t + 5t^2 + \text{noise}$

$\sigma_{\text{noise}} = 50$

Theory

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

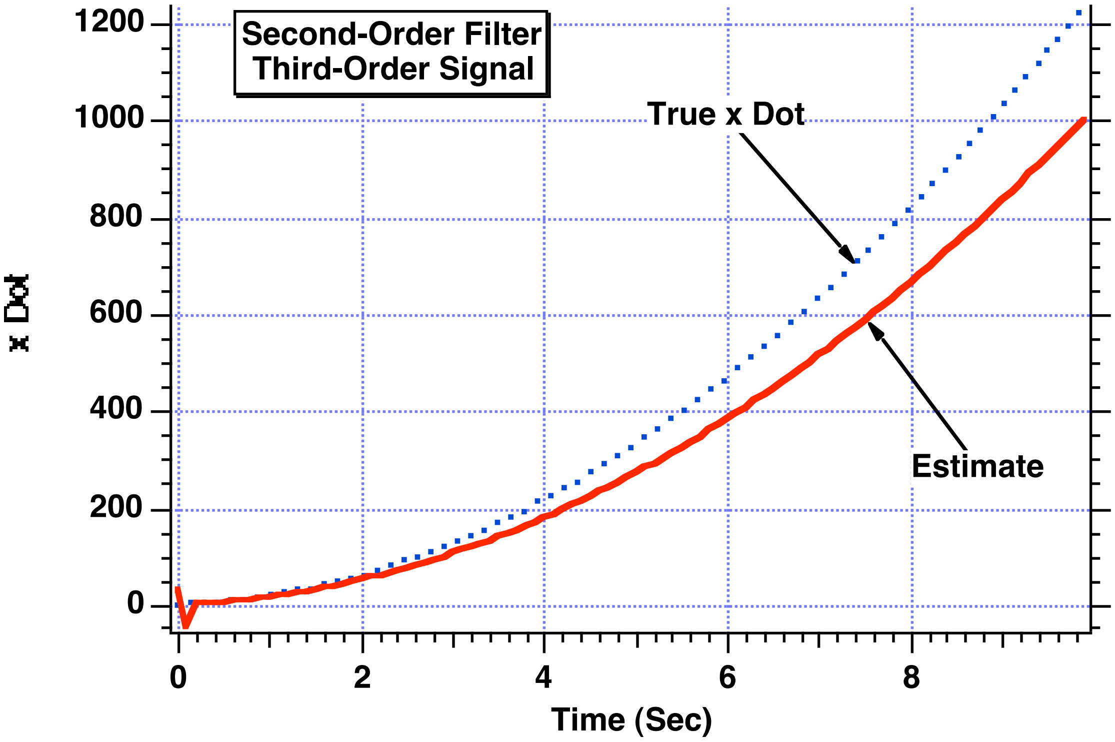
Second-Order Recursive Filter is Unable to Track the First State of a Third-Order Polynomial



Measurement

$x^* = 1 + 2t + 3t^2 + 4t^3$

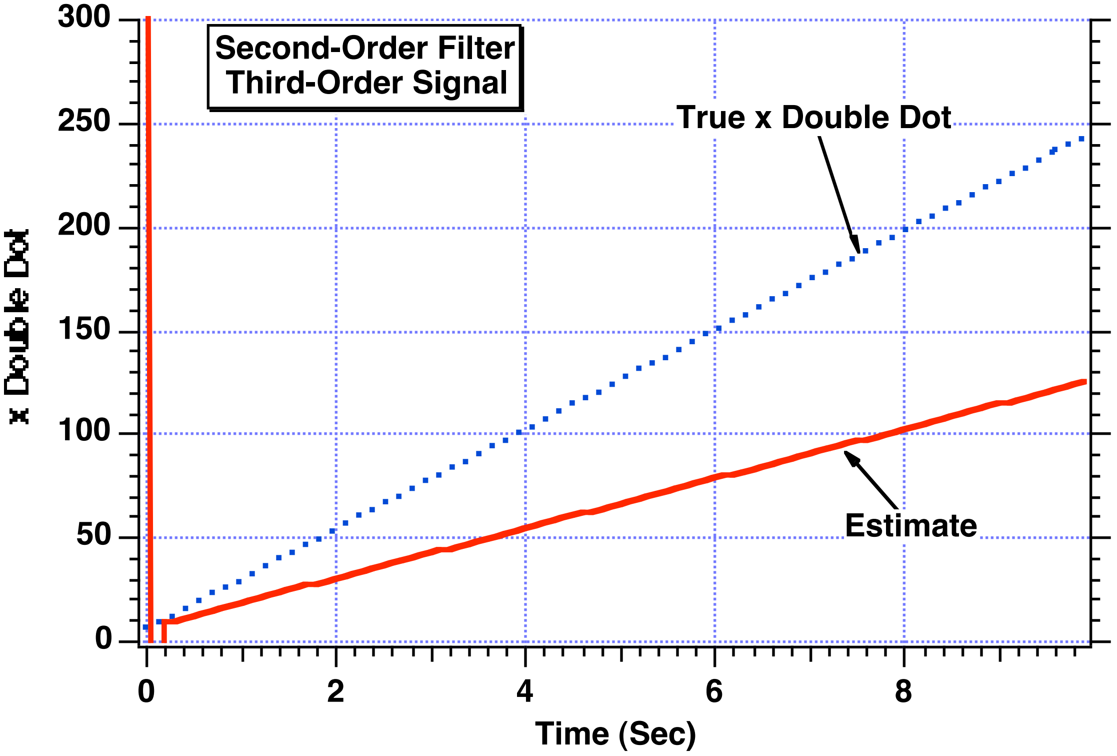
Second-Order Recursive Filter is Unable to Track the Second State of a Third-Order Polynomial



Measurement

$x^* = 1 + 2t + 3t^2 + 4t^3$

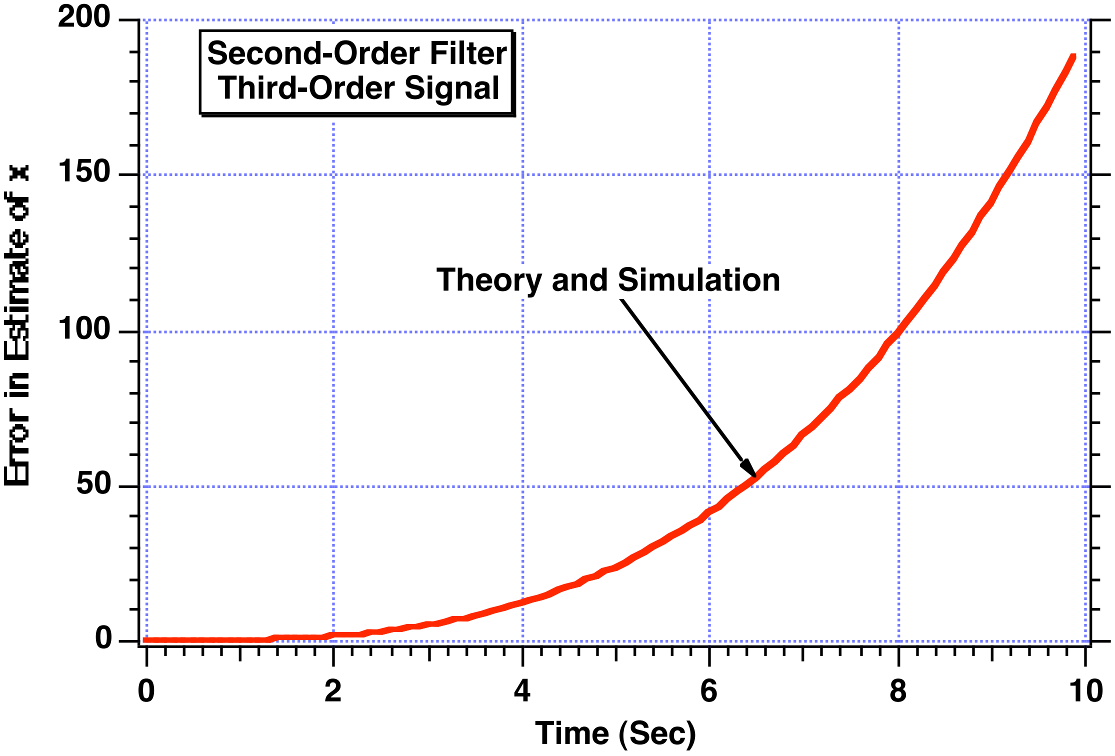
Second-Order Recursive Filter is Unable to Track the Third State of a Third-Order Polynomial



Measurement

$x^* = 1 + 2t + 3t^2 + 4t^3$

Simulation Results and Truncation Error Formula for the First State are in Excellent Agreement



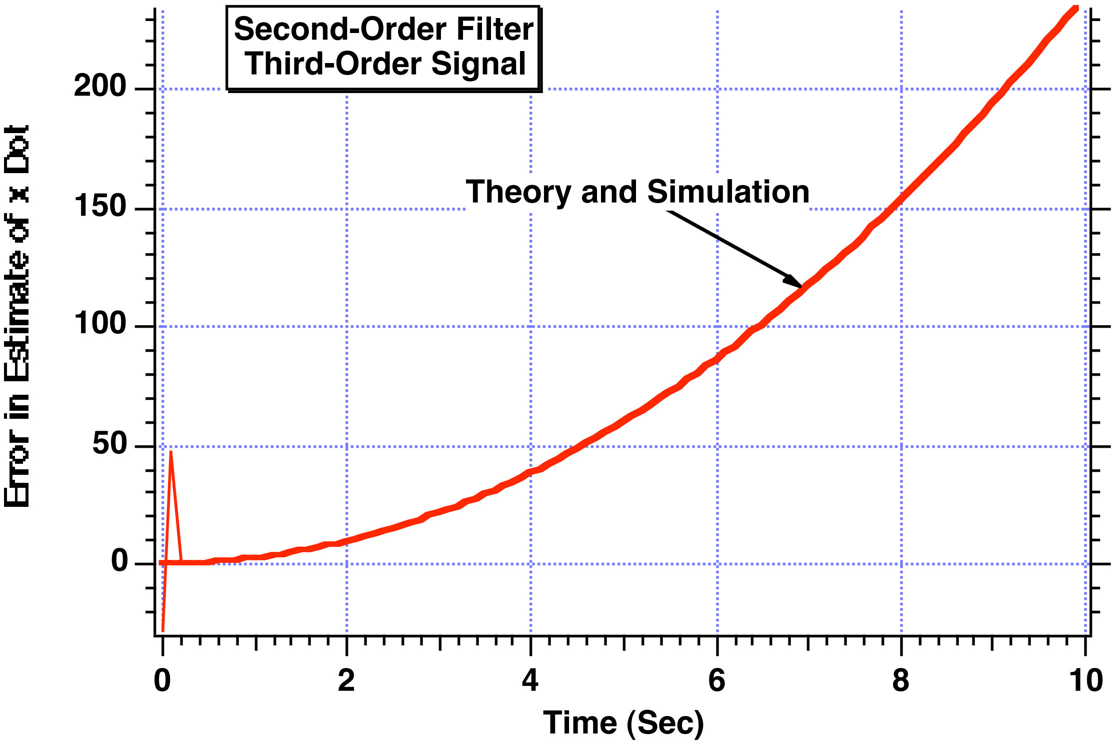
Measurement

$$x^* = 1 + 2t + 3t^2 + 4t^3$$

Theory

$$\varepsilon_k = \frac{1}{20} a_3 T_s^3 (k-1)(k-2)(k-3)$$

Simulation Results and Truncation Error Formula for the Second State are in Excellent Agreement



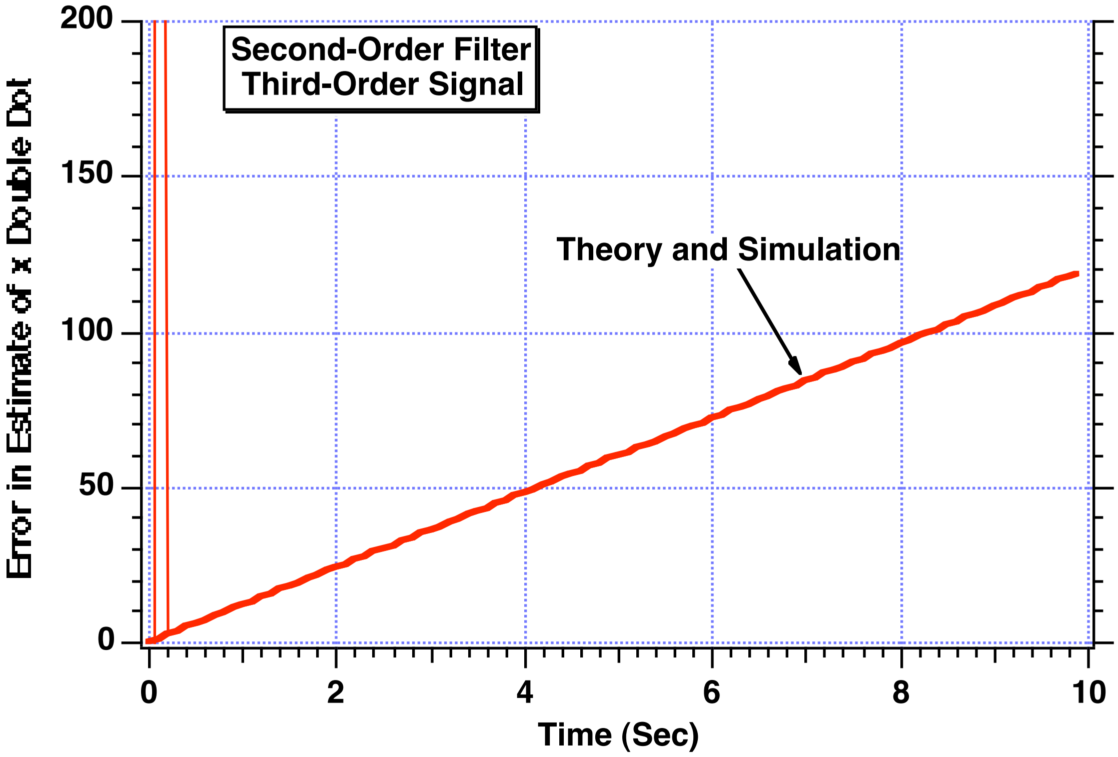
Measurement

$x^* = 1 + 2t + 3t^2 + 4t^3$

Theory

$$\dot{\varepsilon}_k = \frac{1}{10} a_3 T_s^2 (6k^2 - 15k + 11)$$

Simulation Results and Truncation Error Formula for the Third State are in Excellent Agreement



Measurement

$x^* = 1 + 2t + 3t^2 + 4t^3$

Theory

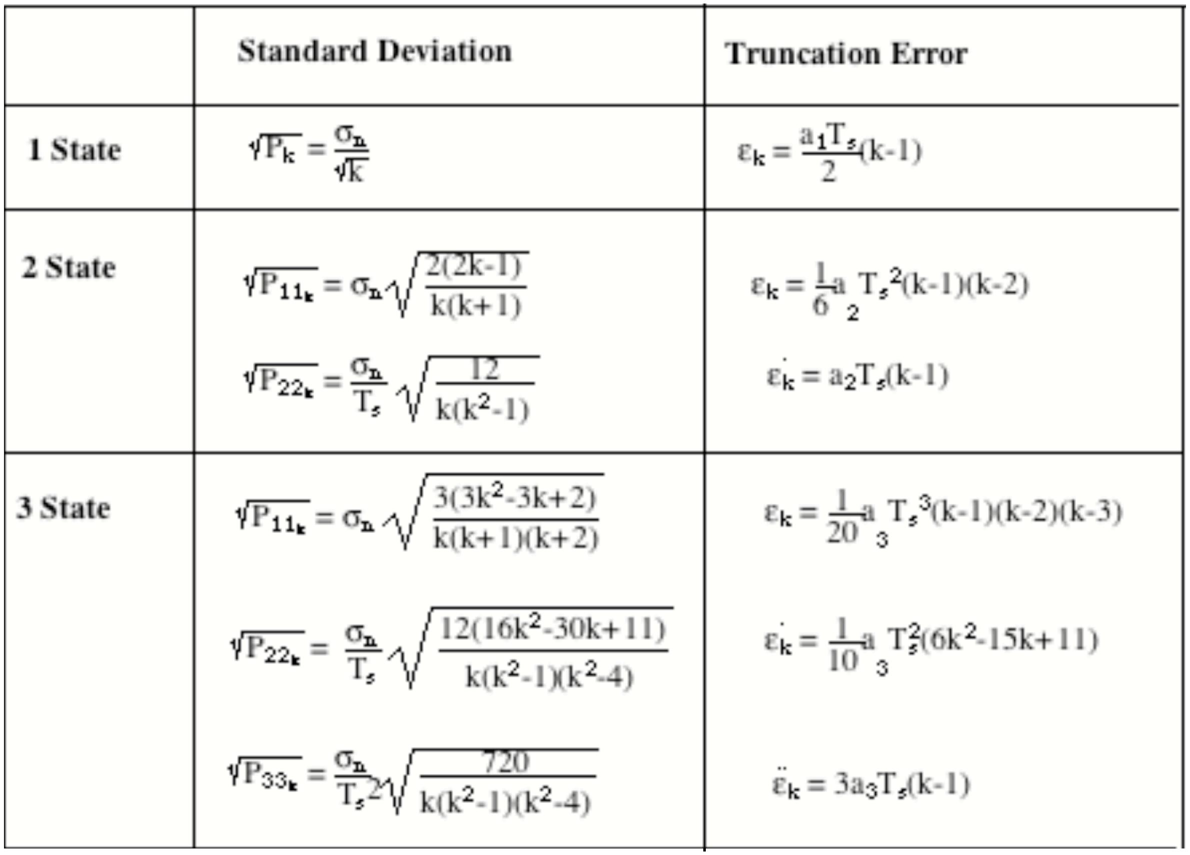
$\ddot{\epsilon}_k = 3a_3T_s(k-1)$

Recursive Least Squares Filter Comparison and Summary

Recursive Least Squares Filter Comparison in Terms of Structure

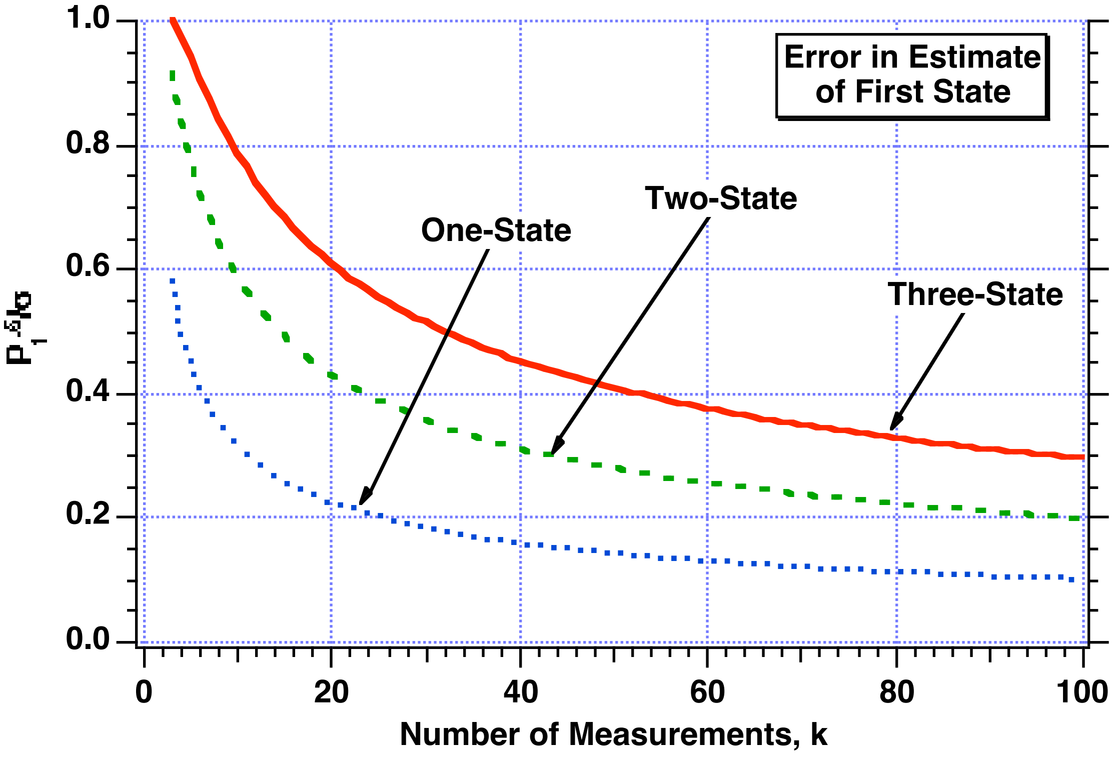
	Filter	**Gains**
1 State	$Res_k = x_k^* - \hat{x}_{k-1}$ $\hat{x}_k = \hat{x}_{k-1} + K_{1_k} Res_k$	$K_{1_k} = \frac{1}{k}$
2 State	$Res_k = x_k^* - \hat{x}_{k-1} - \hat{\dot{x}}_{k-1} T_s$ $\hat{x}_k = \hat{x}_{k-1} + \hat{\dot{x}}_{k-1} T_s + K_{1_k} Res_k$ $\hat{\dot{x}}_k = \hat{\dot{x}}_{k-1} + K_{2_k} Res_k$	$K_{1_k} = \frac{2(2k-1)}{k(k+1)}$ $K_{2_k} = \frac{6}{k(k+1)T_s}$
3 State	$Res_k = x_k^* - \hat{x}_{k-1} - \hat{\dot{x}}_{k-1} T_s - .5\hat{\ddot{x}}_{k-1} T_s^2$ $\hat{x}_k = \hat{x}_{k-1} + \hat{\dot{x}}_{k-1} T_s + .5\hat{\ddot{x}}_{k-1} T_s^2 + K_{1_k} Res_k$ $\hat{\dot{x}}_k = \hat{\dot{x}}_{k-1} + \hat{\ddot{x}}_{k-1} T_s^2 + K_{2_k} Res_k$ $\hat{\ddot{x}}_k = \hat{\ddot{x}}_{k-1} + K_{3_k} Res_k$	$K_{1_k} = \frac{3(3k^2-3k+2)}{k(k+1)(k+2)}$ $K_{2_k} = \frac{18(2k-1)}{k(k+1)(k+2)T_s}$ $K_{3_k} = \frac{60}{k(k+1)(k+2)T_s^2}$

Standard Deviation of Errors in Estimates and Truncation Error Formulas for Various Order Recursive Least Squares Filters

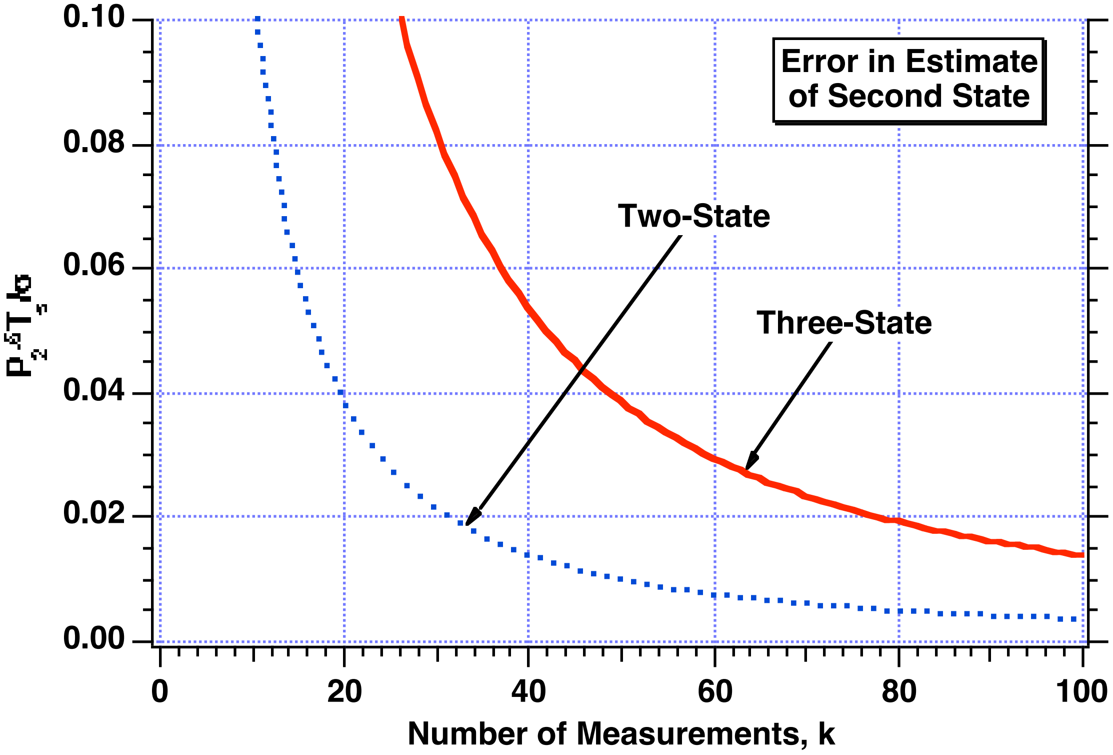


	Standard Deviation	Truncation Error
1 State	$\sqrt{P_k} = \frac{\sigma_n}{\sqrt{k}}$	$\varepsilon_k = \frac{a_1 T_s}{2}(k-1)$
2 State	$\sqrt{P_{11_k}} = \sigma_n \sqrt{\frac{2(2k-1)}{k(k+1)}}$	$\varepsilon_k = \frac{1}{6} a_2 T_s^2 (k-1)(k-2)$
	$\sqrt{P_{22_k}} = \frac{\sigma_n}{T_s}\sqrt{\frac{12}{k(k^2-1)}}$	$\dot{\varepsilon}_k = a_2 T_s (k-1)$
3 State	$\sqrt{P_{11_k}} = \sigma_n \sqrt{\frac{3(3k^2-3k+2)}{k(k+1)(k+2)}}$	$\varepsilon_k = \frac{1}{20} a_3 T_s^3 (k-1)(k-2)(k-3)$
	$\sqrt{P_{22_k}} = \frac{\sigma_n}{T_s}\sqrt{\frac{12(16k^2-30k+11)}{k(k^2-1)(k^2-4)}}$	$\dot{\varepsilon}_k = \frac{1}{10} a_3 T_s^2 (6k^2-15k+11)$
	$\sqrt{P_{33_k}} = \frac{\sigma_n}{T_s^2}\sqrt{\frac{720}{k(k^2-1)(k^2-4)}}$	$\ddot{\varepsilon}_k = 3 a_3 T_s (k-1)$

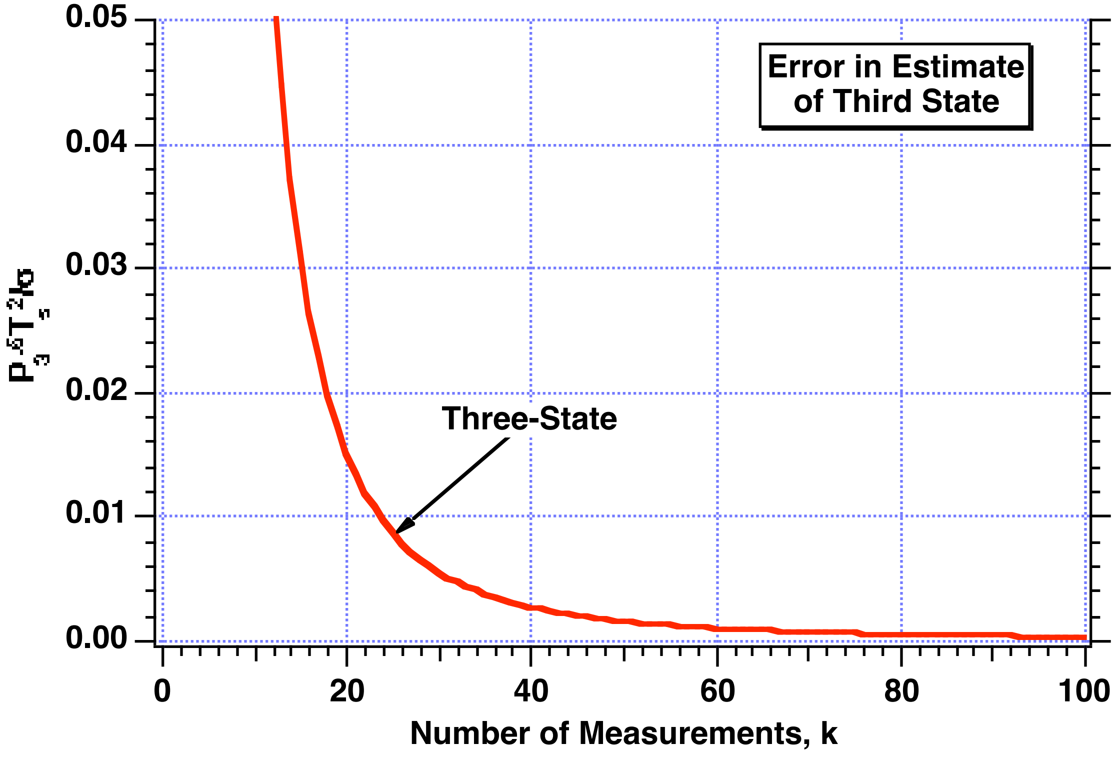
Error in the Estimate of the First State Decreases with Decreasing Filter Order and Increasing Number of Measurements Taken



Error in the Estimate of the Second State Decreases with Decreasing Filter Order and Increasing Number of Measurements Taken



Error in the Estimate of the Third State Decreases with Decreasing Filter Order and Increasing Number of Measurements Taken


Error in Estimate
of Third State
Three-State
$P_3^{.5}T_s^2/\sigma$
Number of Measurements, k