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

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

Rewrite the preceding equation as

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

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

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

Where filter gain is

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

And residual is given by

$$\text{Res}_k = x_k^* - \hat{x}_{k-1}$$
**Numerical Example For the Zeroth-Order Filter-1**

Previous measurement data

<table>
<thead>
<tr>
<th>k</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k-1)T_s</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>x_k*</td>
<td>1.22</td>
<td>.2</td>
<td>2.92</td>
<td>2.12</td>
</tr>
</tbody>
</table>

Gain for first measurement

\[ K_{11} = \frac{1}{k} = \frac{1}{1} = 1 \]

For lack of any a priori information assume

\[ \hat{x}_0 = 0 \]

Calculate residual as

\[ \text{Res}_{1} = x_1^* - \hat{x}_0 = 1.2 - 0 = 1.2 \]

New estimate becomes

\[ \hat{x}_1 = \hat{x}_0 + K_{11}\text{Res}_{1} = 0 + 1*1.2 = 1.2 \]

*We are able to make estimates before all the data is collected*
Numerical Example For the Zeroth-Order Filter-2

For next cycle with $k=2$

$$K_{12} = \frac{1}{k} = \frac{1}{2} = .5$$

$$Res_2 = \hat{x}^*_2 - \hat{x}_1 = .2 - 1.2 = -1$$

$$\hat{x}_2 = \hat{x}_1 + K_{12}Res_2 = 1.2 + .5*(-1) = .7$$

For next cycle with $k=3$

$$K_{13} = \frac{1}{k} = \frac{1}{3} = .333$$

$$Res_3 = \hat{x}^*_3 - \hat{x}_2 = 2.9 - .7 = 2.2$$

$$\hat{x}_3 = \hat{x}_2 + K_{13}Res_3 = .7 + .333*2.2 = 1.43$$

For last cycle with $k=4$

$$K_{14} = \frac{1}{k} = \frac{1}{4} = .25$$

$$Res_4 = \hat{x}^*_4 - \hat{x}_3 = 2.1 - 1.43 = .67$$

$$\hat{x}_4 = \hat{x}_3 + K_{14}Res_4 = 1.43 + .25*.67 = 1.6$$

Another estimate without collecting all the data

Another estimate

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 \( k=1 \)

\[ K_{11} = \frac{1}{k} = \frac{1}{1} = 1 \]

\[ \text{Res}_1 = x_1^* - \hat{x}_0 = 1.2 - 100 = -98.8 \]

\[ \hat{x}_1 = \hat{x}_0 + K_{11}\text{Res}_1 = 100 + 1*(-98.8) = 1.2 \]

This is same answer as when the initial condition was zero
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

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

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

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

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

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

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

Formula for variance of error in the estimate

Trend indicates that

\[ P_k = \frac{\sigma_n^2}{k} \]
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

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

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

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

Therefore error in the estimate given by

\[
\varepsilon_k = x_k - \hat{x}_k = a_0 + a_1 T_s (k-1) - \frac{a_1 T_s}{2}(k-1) = \frac{a_1 T_s}{2}(k-1)
\]
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*AI*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

Measurement

\[ x^* = 1 + \text{noise} \]
\[ \sigma_{\text{noise}} = 1 \]
Fundamentals of Kalman Filtering: A Practical Approach

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

\[ \varepsilon_k = \frac{a_1 T_s}{2} (k-1) = 0.5 \times 2 \times 0.1(k-1) = 0.1(k-1) \]
Summary of Results So Far For Zeroth-Order Recursive Least Squares Filter

Formulas for errors in estimates due to noise and truncation error

\[ \sqrt{P_{11k}} = \frac{\sigma_n}{\sqrt{k}} \]

\[ \varepsilon_k = .5 a_i T_s(k-1) \]

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
11 CONTINUE
CLOSE(K)
PAUSE
END
Monte Carlo Results Lie Within the Theoretical Bounds Approximately 68% of the Time

\[ \sqrt{P_k} = \frac{\sigma_n}{\sqrt{k}} \]
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

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

\[ K_{2k} = \frac{6}{k(k+1)T_s} \]

Filter

\[ \text{Res}_k = x_k^* - \hat{x}_{k-1} - \hat{x}_{k-1}T_s \]

\[ \hat{x}_k = \hat{x}_{k-1} + \hat{x}_{k-1}T_s + K_{1k}\text{Res}_k \]

\[ \hat{x}_k = \hat{x}_{k-1} + K_{2k}\text{Res}_k \]
Numerical Example For First-Order Filter-1

Recall from previous section measurement data given by

\[ x_1^* = 1.2 \]
\[ x_2^* = 0.2 \]
\[ x_3^* = 2.9 \]
\[ x_4^* = 2.1 \]

Assume

\[ \hat{x}_0 = 0 \]
\[ \hat{x}_0 = 0 \]

First iteration (k=1)

\[ K_{11} = \frac{2(2k-1)}{k(k+1)} = \frac{2(2*1-1)}{1(1+1)} = 1 \]
\[ K_{21} = \frac{6}{k(k+1)T_s} = \frac{6}{1(1+1)*1} = 3 \]
\[ Res_1 = x_1^* - \hat{x}_0 - \hat{x}_0 T_s = 1.2 - 0 - 0*1 = 1.2 \]
\[ \hat{x}_1 = \hat{x}_0 + \hat{x}_0 T_s + K_{11} Res_1 = 0 + 0*1 + 1*1.2 = 1.2 \]
\[ \hat{x}_1 = \hat{x}_0 + K_{21} Res_1 = 0 + 3*1.2 = 3.6 \]
Numerical Example For First-Order Filter-2

Second iteration (k=2)

\[ K_{12} = \frac{2(2k-1)}{k(k+1)} = \frac{2(2*2-1)}{2(2+1)} = 1 \]
\[ K_{22} = \frac{6}{k(k+1)T_s} = \frac{6}{2(2+1)*1} = 1 \]

\[ \text{Res}_2 = x_2^* - \hat{x}_1 - \hat{x}_1T_s = .2 - 1.2 - 3.6*1 = -4.6 \]

\[ \hat{x}_2 = \hat{x}_1 + \hat{x}_1T_s + K_{12}\text{Res}_2 = 1.2 + 3.6*1 + 1*(-4.6) = .2 \]

\[ \hat{x}_2 = \hat{x}_1 + K_{22}\text{Res}_2 = 3.6 + 1*(-4.6) = -1 \]

Third iteration (k=3)

\[ K_{13} = \frac{2(2k-1)}{k(k+1)} = \frac{2(2*3-1)}{3(3+1)} = \frac{5}{6} \]
\[ K_{23} = \frac{6}{k(k+1)T_s} = \frac{6}{3(3+1)*1} = .5 \]

\[ \text{Res}_3 = x_3^* - \hat{x}_2 - \hat{x}_2T_s = 2.9 - .2 - (-1)*1 = 3.7 \]

\[ \hat{x}_3 = \hat{x}_2 + \hat{x}_2T_s + K_{13}\text{Res}_3 = .2 + (-1)*1 + \frac{5}{6}*3.7 = 2.28 \]

\[ \hat{x}_3 = \hat{x}_2 + K_{23}\text{Res}_3 = -1 + .5*3.7 = .85 \]
Numerical Example For First-Order Filter-3

Last iteration (k=4)

\[ K_{14} = \frac{2(2k-1)}{k(k+1)} = \frac{2(2*4-1)}{4(4+1)} = 0.7 \]

\[ K_{24} = \frac{6}{k(k+1)T_s} = \frac{6}{4(4+1)*1} = 0.3 \]

Res\_4 = x^*_4 - \hat{x}_3 - \hat{x}_3 T_s = 2.1 - 2.28 - 0.85*1 = -1.03

\[ \hat{x}_4 = \hat{x}_3 + \hat{x}_3 T_s + K_{14} \text{Res\_4} = 2.28 + 0.85*1 + 0.7*(-1.03) = 2.41 \]

\[ \hat{x}_4 = \hat{x}_3 + K_{24} \text{Res\_4} = 0.85 + 0.3*(-1.03) = 0.54 \]

Same answer as obtained with first-order batch processing filter.
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

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

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

Error in estimate due to truncation error

\[ x_k^* = a_0 + a_1t + a_2t^2 = a_0 + a_1(k-1)T_s + a_2(k-1)^2T_s^2 \]

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

\[ \dot{\epsilon}_k = a_2T_s(k-1) \]
FORTRAN Simulation For Testing First-Order Recursive Least Squares Filter

GLOBAL DEFINE
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*XN-1)*TS*TS)
RES=XS-XH-XD
XDHD=XD-XDHD
IF(XN.EQ.1)THEN
LET SP11=0
LET SP22=0
ELSE
SP11=SIGNOISE*SQR(2.*(2*XN-1)/(XN*(XN+1)))
SP22=SIGNOISE*SQR(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

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

\[ \sigma_{\text{noise}} = 5 \]
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

\[ P_{11k} = \frac{2(2k-1)! n^2}{k(k+1)} \]

Measurement
\[ x^* = 3 + t + \text{noise} \]
\[ \sigma_{\text{noise}} = 5 \]
Single Run Simulation Results For Second State
Also Agrees With Theoretical Formula

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

\[ \sqrt{P_{11_k}} = \sqrt{\frac{2(2k-1)\sigma^2_n}{k(k+1)}} \]

First-Order Filter
5 Runs
Simulated Error in the Estimates of Second State Appear to Lie Within Theoretical Error Bounds 68% of the Time
First-Order Recursive Least Squares Filter is Unable to Track the First State of a Second-Order Polynomial

\[ x^* = 1 + 2t + 3t^2 \]
First-Order Recursive Least Squares Filter is Unable to Track the Second State of a Second-Order Polynomial

Measurement

\[ x^* = 1 + 2t + 3t^2 \]
Fundamentals of Kalman Filtering: A Practical Approach

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

Theory

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

Measurement

$$x^* = 1 + 2t + 3t^2$$
Simulation Results and Truncation Error for Second State are in Excellent Agreement

**Measurement**

\[ x^* = 1 + 2t + 3t^2 \]

**Theory**

\[ \epsilon_k = a_2 T_s (k-1) \]

**Graph**

- **First-Order Filter**
- **Second-Order Signal**
- **Theory & Simulation**
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

\[ \text{Res}_k = x^*_k - \hat{x}_{k-1} - \hat{x}_{k-1}T_s - .5\hat{x}^2_{k-1}T_s^2 \]

\[ \hat{x}_k = \hat{x}_{k-1} + \hat{x}_{k-1}T_s + .5\hat{x}^2_{k-1}T_s^2 + K_{1k}\text{Res}_k \]

\[ \hat{x}_k = \hat{x}_{k-1} + \hat{x}^2_{k-1}T_s^2 + K_{2k}\text{Res}_k \]

\[ \hat{x}_k = \hat{x}_{k-1} + K_{3k}\text{Res}_k \]
Recall from previous Lecture measurement data given by

\[
\begin{align*}
x_1^* &= 1.2 \\
x_2^* &= 0.2 \\
x_3^* &= 2.9 \\
x_4^* &= 2.1 \\
\end{align*}
\]

Assume

\[
\begin{align*}
\hat{x}_0 &= 0 \\
\end{align*}
\]

First iteration (k=1)

\[
\begin{align*}
K_{1,1} &= \frac{3(3k^2-3k+2)}{k(k+1)(k+2)} = \frac{3(3*1^2-3*1+2)}{1(2)(3)} = 1 \\
K_{2,1} &= \frac{18(2k-1)}{k(k+1)(k+2)T_s} = \frac{18(2-1)}{1(2)(3)(1)} = 3 \\
K_{3,1} &= \frac{60}{k(k+1)(k+2)T_s^2} = \frac{60}{1(2)(3)(1)} = 10 \\
Res_{1} &= x_1^* - \hat{x}_0 - \hat{x}_0T_s - 0.5\hat{x}_0T_s^2 = 1.2 - 0 - 0 - 0 = 1.2 \\
\hat{x}_{1} &= \hat{x}_0 + \hat{x}_0T_s + 0.5\hat{x}_0T_s^2 + K_{1,1}Res_{1} = 0 + 0 + 0 + 1*1.2 = 1.2 \\
\hat{x}_{1} &= \hat{x}_0 + \hat{x}_0T_s + K_{2,1}Res_{1} = 0 + 0 + 3*1.2 = 3.6 \\
\hat{x}_{1} &= \hat{x}_0 + K_{3,1}Res_{1} = 0 + 10*1.2 = 12 \\
\end{align*}
\]
Numerical Example For Second-Order Filter-2

Second iteration (k=2)

\[ K_{12} = \frac{3(3k^2-3k+2)}{k(k+1)(k+2)} = \frac{3(3*4-3*2+2)}{2(3)(4)} = 1 \]

\[ K_{22} = \frac{18(2k-1)}{k(k+1)(k+2)T_s} = \frac{18(2*2-1)}{2(3)(4)(1)} = 2.25 \]

\[ K_{32} = \frac{60}{k(k+1)(k+2)T_s^2} = \frac{60}{2(3)(4)(1)} = 2.5 \]

\[ \text{Res}_2 = x_2^* - \hat{x}_1 - \hat{x}_1 T_s - .5 \hat{x}_1 T_s^2 = .2 - 1.2 - 3.6 - .5*12 = -10.6 \]

\[ \hat{x}_2 = \hat{x}_1 + \hat{x}_1 T_s + .5 \hat{x}_1 T_s^2 + K_{12}\text{Res}_2 = 1.2 + 3.6 + .5*12 +1*(-10.6) = .2 \]

\[ \hat{x}_2 = \hat{x}_1 + \hat{x}_1 T_s + K_{22}\text{Res}_2 = 3.6 + 12 + 2.25*(-10.6) = -8.25 \]

\[ \hat{x}_2 = \hat{x}_1 + K_{32}\text{Res}_2 = 12 + 2.5*(-10.6) = -14.5 \]
Numerical Example For Second-Order Filter-3

Third iteration (k=3)

\[ K_{13} = \frac{3(3k^2 - 3k + 2)}{k(k+1)(k+2)} = \frac{3(3*9 - 3*3 + 2)}{3(4)(5)} = 1 \]

\[ K_{23} = \frac{18(2k-1)}{k(k+1)(k+2)T_s} = \frac{18(2*3 - 1)}{3(4)(5)(1)} = 1.5 \]

\[ K_{33} = \frac{60}{k(k+1)(k+2)T_s^2} = \frac{60}{3(4)(5)(1)} = 1 \]

\[ \text{Res}_{3} = x_3^* - \hat{x}_2 - \hat{x}_2 T_s - .5\hat{x}_2 T_s^2 = 2.9 - .2 - (-8.25) - .5*(-14.5) = 18.2 \]

\[ \hat{x}_3 = \hat{x}_2 + \hat{x}_2 T_s + .5\hat{x}_2 T_s^2 + K_{13}\text{Res}_3 = .2 - 8.25 + .5*(-14.5) + 1*18.2 = 2.9 \]

\[ \hat{x}_3 = \hat{x}_2 + K_{23}\text{Res}_3 = -8.25 - 14.5 + 1.5*18.2 = 4.55 \]

\[ \hat{x}_3 = \hat{x}_2 + K_{33}\text{Res}_3 = -14.5 + 1*18.2 = 3.7 \]
Numerical Example For Second-Order Filter-4

Last iteration (k=4)

\[ K_{11} = \frac{3(3k^2 - 3k + 2)}{k(k+1)(k+2)} = \frac{3(3*16 - 3*4 + 2)}{4(5)(6)} = \frac{19}{20} \]

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

\[ K_{31} = \frac{60}{k(k+1)(k+2)T_s^2} = \frac{60}{4(5)(6)(1)} = .5 \]

\[ \text{Res}_{4} = x_4^\circ - \hat{x}_3 - \hat{x}_3T_s - .5\hat{x}_3T_s^2 = 2.1 - 2.9 - 4.55 - .5*3.7 = -7.2 \]

\[ \hat{x}_4 = \hat{x}_3 + \hat{x}_3T_s + .5\hat{x}_3T_s^2 + K_{11}\text{Res}_{4} = 2.9 + 4.55 + .5*3.7 + \frac{19}{20}*(-7.2) = 2.46 \]

\[ \hat{x}_4 = \hat{x}_3 + \hat{x}_3T_s + K_{21}\text{Res}_{4} = 4.55 + 3.7*1 + \frac{21}{20}*(-7.2) = .69 \]

\[ \hat{x}_4 = \hat{x}_3 + K_{31}\text{Res}_{4} = 3.7 + .5*(-7.2) = .1 \]
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

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

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

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

Error in estimate due to truncation error

\[ x_k^* = a_0 + a_1t + a_2t^2 + a_3t^3 = a_0 + a_1(k-1)T_s + a_2(k-1)^2T_s^2 + a_3(k-1)^3T_s^3 \]

\[ \varepsilon_k = \frac{1}{20}a_3T_s^3(k-1)(k-2)(k-3) \]

\[ \varepsilon_k' = \frac{1}{10}a_3T_s^2(6k^2-15k+11) \]

\[ \ddot{\varepsilon}_k = 3a_3T_s(k-1) \]

Given third-order signal
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.
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
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)*
   (XN+2)))
   SP22=SIGNOISE*SQRT(12*(16*XN*XN-30*XN+11)/
   (XN*(XN*XN-1)*(XN*XN-4)*TS*TS))
   SP33=SIGNOISE*SQRT(720/(XN*(XN*XN-1)*(XN*XN-4)*
   *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,XHERR,SP11,-SP11,EPS,XDHERR,SP22,-SP22,EPSD,
                      XDDHERR,SP33,-SP33,EPSDD
10 CONTINUE
CLOSE(1)
CLOSE(2)
PAUSE
END
Second-Order Recursive Filter is Able to Track Second-Order Signal Plus Noise

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

\[ \sigma_{\text{noise}} = 50 \]
Estimate of Derivative is Excellent

Measurement

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

\( \sigma_{\text{noise}} = 50 \)
Estimate of Second Derivative is Also Excellent

Measurement

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

\[ \sigma_{\text{noise}} = 50 \]
Error in Estimate of First State Appears to be Within Theoretical Error Bounds

\[ P_{11k} = \frac{3(3k^2 - 3k + 2)}{k(k+1)(k+2)} \]

Measurement
\[ x^* = 2 - 2t + 5t^2 + \text{noise} \]
\[ \sigma_{\text{noise}} = 50 \]

Theory
\[ \sqrt{P_{11k}} = \sqrt{\frac{3(3k^2 - 3k + 2)\sigma_n^2}{k(k+1)(k+2)}} \]
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

\[ \sqrt{P_{22k}} = \sqrt{\frac{12(16k^2 - 30k + 11)\sigma_{ii}^2}{k(k^2 - 1)(k^2 - 4)T_s^2}} \]
Error in Estimate of Third State Appears to be Within Theoretical Error Bounds

\[ P_{33k} = 720n^2k(k^2-1)(k^2-4)T_s^4 \]

Measurement

\[ x^* = 2 - 2t + 5t^2 + \text{noise} \]
\[ \sigma_{\text{noise}} = 50 \]

Theory

\[ \sqrt{\text{P}_{33k}} = \sqrt{\frac{720\sigma_n^2}{k(k^2-1)(k^2-4)T_s^4}} \]
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

\[ P_{11_k} = \frac{3(3k^2 - 3k + 2)\sigma_n^2}{k(k+1)(k+2)} \]
Multiple Runs Indicate That on Average the Error in the Estimate of Second State Appears to be Within Error Bounds 68% of the Time

\[ P_{22k} = \frac{12(16k^2 - 30k + 11)\sigma_n^2}{\left(k(k^2 - 1)(k^2 - 4)T_s^2\right)} \]

Measurement

\[ x* = 2 - 2t + 5t^2 + \text{noise} \]
\[ \sigma_{\text{noise}} = 50 \]

Theory

\[ \sqrt{P_{22k}} = \sqrt{\frac{12(16k^2 - 30k + 11)\sigma_n^2}{k(k^2 - 1)(k^2 - 4)T_s^2}} \]
Multiple Runs Indicate That on Average the Error in the Estimate of Third State Appears to be Within Error Bounds 68% of the Time

Theory

\[ P_{33k} = \frac{720\sigma^2_n}{k(k^2-1)(k^2-4)T_s^4} \]

Measurement

\[ x^* = 2 - 2t + 5t^2 + \text{noise} \]
\[ \sigma_{\text{noise}} = 50 \]
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

\[ x^* = 1 + 2t + 3t^2 + 4t^3 \]

**Theory**

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

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

Measurement

\[ x^* = 1 + 2t + 3t^2 + 4t^3 \]

Theory

\[ \varepsilon_k = \frac{1}{10} a T_s^2 (6k^2 - 15k + 11) \]
Simulation Results and Truncation Error Formula for the Third State are in Excellent Agreement

\[ x^* = 1 + 2t + 3t^2 + 4t^3 \]

Measurement

Theory

\[ \ddot{e}_k = 3a_3T_s(k-1) \]
Recursive Least Squares Filter Comparison and Summary
Recursive Least Squares Filter Comparison in Terms of Structure

<table>
<thead>
<tr>
<th>Filter</th>
<th>Gains</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1 State</strong></td>
<td></td>
</tr>
<tr>
<td>( \text{Res}<em>k = x_k^* - \hat{x}</em>{k-1} )</td>
<td>( K_{1k} = \frac{1}{k} )</td>
</tr>
<tr>
<td>( \hat{x}<em>k = \hat{x}</em>{k-1} + K_{1k}\text{Res}_k )</td>
<td></td>
</tr>
<tr>
<td><strong>2 State</strong></td>
<td></td>
</tr>
<tr>
<td>( \text{Res}<em>k = x_k^* - \hat{x}</em>{k-1} - \hat{x}_{k-1}T_s )</td>
<td>( K_{1k} = \frac{2(2k-1)}{k(k+1)} )</td>
</tr>
<tr>
<td>( \hat{x}<em>k = \hat{x}</em>{k-1} + \hat{x}<em>{k-1}T_s + K</em>{1k}\text{Res}_k )</td>
<td>( K_{2k} = \frac{6}{(k+1)T_s} )</td>
</tr>
<tr>
<td>( \hat{x}<em>k = \hat{x}</em>{k-1} + K_{2k}\text{Res}_k )</td>
<td></td>
</tr>
<tr>
<td><strong>3 State</strong></td>
<td></td>
</tr>
<tr>
<td>( \text{Res}<em>k = x_k^* - \hat{x}</em>{k-1} - \hat{x}<em>{k-1}T_s - .5\hat{x}</em>{k-1}T_s^2 )</td>
<td>( K_{1k} = \frac{3(3k^2-3k+2)}{k(k+1)(k+2)} )</td>
</tr>
<tr>
<td>( \hat{x}<em>k = \hat{x}</em>{k-1} + \hat{x}<em>{k-1}T_s + .5\hat{x}</em>{k-1}T_s^2 + K_{1k}\text{Res}_k )</td>
<td>( K_{2k} = \frac{18(2k-1)}{k(k+1)(k+2)T_s} )</td>
</tr>
<tr>
<td>( \hat{x}<em>k = \hat{x}</em>{k-1} + \hat{x}<em>{k-1}T_s^2 + K</em>{2k}\text{Res}_k )</td>
<td>( K_{3k} = \frac{60}{k(k+1)(k+2)T_s^2} )</td>
</tr>
<tr>
<td>( \hat{x}<em>k = \hat{x}</em>{k-1} + K_{3k}\text{Res}_k )</td>
<td></td>
</tr>
</tbody>
</table>
### Standard Deviation of Errors in Estimates and Truncation Error Formulas for Various Order Recursive Least Squares Filters

<table>
<thead>
<tr>
<th>State</th>
<th>Standard Deviation</th>
<th>Truncation Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 State</td>
<td>$\sqrt{P_k} = \frac{\sigma_n}{\sqrt{k}}$</td>
<td>$\varepsilon_k = \frac{a_1 T_e}{2}(k-1)$</td>
</tr>
<tr>
<td>2 State</td>
<td>$\sqrt{P_{11k}} = \sigma_n \sqrt{\frac{2(2k-1)}{k(k+1)}}$</td>
<td>$\varepsilon_k = \frac{1}{6} a_2 T_e^2(k-1)(k-2)$</td>
</tr>
<tr>
<td></td>
<td>$\sqrt{P_{22k}} = \frac{\sigma_n}{T_e} \sqrt{\frac{12}{k(k^2-1)}}$</td>
<td>$\varepsilon_k = a_2 T_e(k-1)$</td>
</tr>
<tr>
<td>3 State</td>
<td>$\sqrt{P_{11k}} = \sigma_n \sqrt{\frac{3(3k^2-3k+2)}{k(k+1)(k+2)}}$</td>
<td>$\varepsilon_k = \frac{1}{20} a_3 T_e^3(k-1)(k-2)(k-3)$</td>
</tr>
<tr>
<td></td>
<td>$\sqrt{P_{22k}} = \frac{\sigma_n}{T_e} \sqrt{\frac{12(16k^2-30k+11)}{k(k^2-1)(k^2-4)}}$</td>
<td>$\varepsilon_k = \frac{1}{10} a_3 T_e^2(6k^2-15k+11)$</td>
</tr>
<tr>
<td></td>
<td>$\sqrt{P_{33k}} = \frac{\sigma_n}{T_e} \sqrt{\frac{720}{k(k^2-1)(k^2-4)}}$</td>
<td>$\varepsilon_k = 3 a_3 T_e(k-1)$</td>
</tr>
</tbody>
</table>
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.