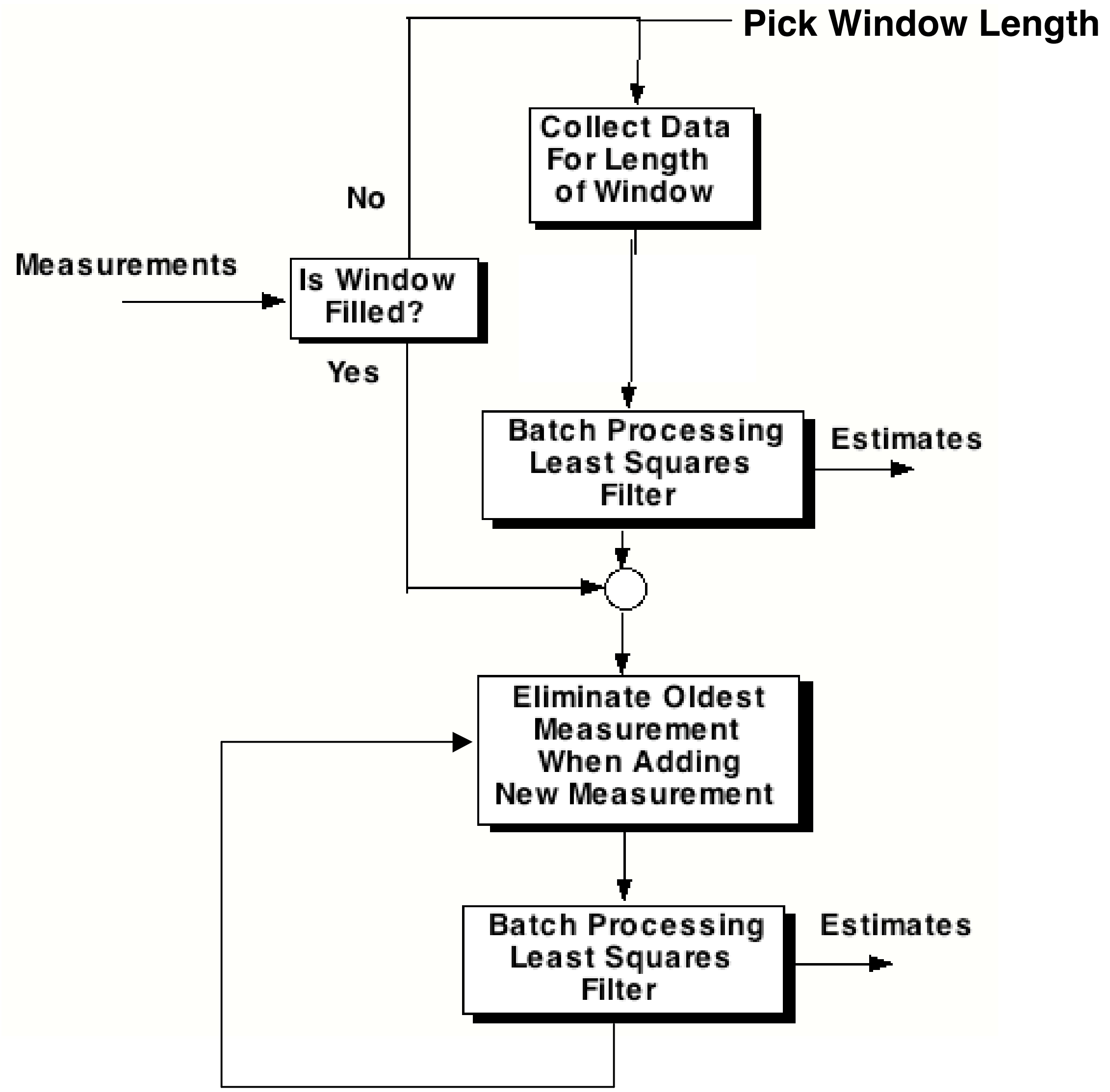
Comparison of Finite Memory and Kalman Filters

Recall

- A batch processing least squares filter and a Kalman filter are equivalent when Kalman filter has zero process noise and infinite initial covariance matrix
- A Kalman filter with zero process noise will have problems in the real world because the Kalman gains eventually go to zero. This means that the Kalman filter will no longer pay attention to measurements
- If the filter only has to work for a short period of time (window), having zero process noise might be ok

Main Idea Behind Finite Memory Filter



Review of Least Squares Method For Second-Order System-1

Fit measurement data with “best” parabola

$$\hat{x} = a_0 + a_1 t + a_2 t^2$$

Or in discrete form

$$\hat{x}_k = a_0 + a_1(k-1)T_s + a_2[(k-1)T_s]^2$$

We still want to minimize residual R

$$R = \sum_{k=1}^{n} (\hat{x}_k - x_k^*)^2 = \sum_{k=1}^{n} [a_0 + a_1(k-1)T_s + a_2(k-1)^2 T_s^2 - x_k^*]^2$$

We can expand R

$$R = (a_0 - x_1^*)^2 + [a_0 + a_1 T_s + a_2 T_s^2 - x_2^*)]^2 + \dots + [a_0 + a_1(n-1)T_s + a_2(n-1)^2 T_s^2 - x_n^*]^2$$

Minimize R by setting derivatives to zero

$$\frac{\partial R}{\partial a_0} = 0 = 2(a_0 - x_1^*) + 2[a_0 + a_1 T_s + a_2 T_s^2 - x_2^*)] + \dots + 2[a_0 + a_1(n-1)T_s + a_2(n-1)^2 T_s^2 - x_n^*]$$

$$\frac{\partial R}{\partial a_1} = 0 = 2[a_0 + a_1 T_s + a_2 T_s^2 - x_2^*)]T_s + \dots + 2[a_0 + a_1(n-1)T_s + a_2(n-1)^2 T_s^2 - x_n^*](n-1)T_s$$

$$\frac{\partial R}{\partial a_2} = 0 = 2[a_0 + a_1 T_s + a_2 T_s^2 - x_2^*)]T_s^2 + \dots + 2[a_0 + a_1(n-1)T_s + a_2(n-1)^2 T_s^2 - x_n^*](n-1)^2 T_s^2$$

Review of Least Squares Method For Second-Order System-2

We can simplify preceding three equations

$$na_0 + a_1\sum_{k=1}^{n}(k-1)T_s + a_2\sum_{k=1}^{n}[(k-1)T_s]^2 = \sum_{k=1}^{n}x_k^*$$

$$a_0\sum_{k=1}^{n}(k-1)T_s + a_1\sum_{k=1}^{n}[(k-1)T_s]^2 + a_2\sum_{k=1}^{n}[(k-1)T_s]^3 = \sum_{k=1}^{n}(k-1)T_s x_k^*$$

$$a_0\sum_{k=1}^{n}[(k-1)T_s]^2 + a_1\sum_{k=1}^{n}[(k-1)T_s]^3 + a_2\sum_{k=1}^{n}[(k-1)T_s]^4 = \sum_{k=1}^{n}[(k-1)T_s]^2 x_k^*$$

These equations can also be expressed in matrix form as

$$\begin{bmatrix} n & \sum_{k=1}^{n}(k-1)T_s & \sum_{k=1}^{n}[(k-1)T_s]^2 \\ \sum_{k=1}^{n}(k-1)T_s & \sum_{k=1}^{n}[(k-1)T_s]^2 & \sum_{k=1}^{n}[(k-1)T_s]^3 \\ \sum_{k=1}^{n}[(k-1)T_s]^2 & \sum_{k=1}^{n}[(k-1)T_s]^3 & \sum_{k=1}^{n}[(k-1)T_s]^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{n}x_k^* \\ \sum_{k=1}^{n}(k-1)T_s x_k^* \\ \sum_{k=1}^{n}[(k-1)T_s]^2 x_k^* \end{bmatrix}$$

Review of Least Squares Method For Second-Order System-3

We can solve for the coefficients by matrix inversion

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} n & \sum_{k=1}^{n} (k-1)T_s & \sum_{k=1}^{n} [(k-1)T_s]^2 \\ \sum_{k=1}^{n} (k-1)T_s & \sum_{k=1}^{n} [(k-1)T_s]^2 & \sum_{k=1}^{n} [(k-1)T_s]^3 \\ \sum_{k=1}^{n} [(k-1)T_s]^2 & \sum_{k=1}^{n} [(k-1)T_s]^3 & \sum_{k=1}^{n} [(k-1)T_s]^4 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{k=1}^{n} x_k^* \\ \sum_{k=1}^{n} (k-1)T_s x_k^* \\ \sum_{k=1}^{n} [(k-1)T_s]^2 x_k^* \end{bmatrix}$$

Simplify Notation

Let

$$i = k-1$$

$$L = n-1$$

Therefore each element in the matrix to be Inverted can be simplified

$$n = L+1 = S_0$$

$$\sum_{k=1}^{n}(k-1)T_s = T_s\sum_{i=0}^{L} i = T_s\sum_{i=1}^{L} i = 0.5T_s[L(L+1)] = S_1T_s$$

$$\sum_{k=1}^{n}\left[(k-1)T_s\right]^2 = T_s^2\sum_{i=0}^{L} i^2 = T_s^2\sum_{i=1}^{L} i^2 = T_s^2\left[\frac{1}{6}L(L+1)(2L+1)\right] = S_2T_s^2$$

$$\sum_{k=1}^{n}\left[(k-1)T_s\right]^3 = T_s^3\sum_{i=0}^{L} i^3 = T_s^3\sum_{i=1}^{L} i^3 = T_s^3\left[\frac{1}{4}L^2(L+1)^2\right] = S_3T_s^3$$

$$\sum_{k=1}^{n}\left[(k-1)T_s\right]^4 = T_s^4\sum_{i=0}^{L} i^4 = T_s^4\sum_{i=1}^{L} i^4 = T_s^4\left[\frac{1}{30}L(L+1)(2L+1)(3L^2+3L-1)\right] = S_4T_s^4$$

From Sums of Powers Of the First n Integers*

Therefore we can express coefficients in shorthand notation as

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} S_0 & S_1T_s & S_2T_s^2 \\ S_1T_s & S_2T_s^2 & S_3T_s^3 \\ S_2T_s^2 & S_3T_s^3 & S_4T_s^4 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{L} x_i^* \\ \sum_{i=1}^{L} iT_sx_i^* \\ \sum_{i=1}^{L} (iT_s)^2 x_i^* \end{bmatrix}$$

***Selby, S.M., "CRC Standard Mathematical Tables, 20th Edition," The Chemical Rubber Co., 1972, p. 37**

Evaluating Matrix Inverse-1

Recall that a 3x3 matrix inverse can be evaluated exactly

If

$$\mathbf{A} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Then exact inverse given by

$$\mathbf{A}^{-1} = \frac{1}{aei + bfg + cdh - ceg - bdi - afh} \begin{bmatrix} ei\text{-}fh & ch\text{-}bi & bf\text{-}ec \\ gf\text{-}di & ai\text{-}gc & dc\text{-}af \\ dh\text{-}ge & gb\text{-}ah & ae\text{-}bd \end{bmatrix}$$

Note that a matrix is a set of numbers that only has to be evaluated once

Numbers in matrix inverse do not depend on measurements

Evaluating Matrix Inverse-2

Recall

$$A=\begin{bmatrix} S_0 & S_1T_s & S_2T_s^2 \\ S_1T_s & S_2T_s^2 & S_3T_s^3 \\ S_2T_s^2 & S_3T_s^3 & S_4T_s^4 \end{bmatrix}=\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

and

$$A^{-1}=\frac{1}{den}\begin{bmatrix} ei-fh & ch-bi & bf-ec \\ gf-di & ai-gc & dc-af \\ dh-ge & gb-ah & ae-bd \end{bmatrix}$$

Therefore

$$den=T_s^6\left(S_0S_2S_4-S_0S_3^2-S_1^2S_4+2S_1S_2S_3-S_2^3\right)$$

$$\frac{ei-fh}{den}=\frac{T_s^6}{den}\left(S_2S_4-S_3^2\right)=C_1 \qquad \frac{ch-bi}{den}=\frac{T_s^5\left(S_2S_3-S_1S_4\right)}{den}=C_2 \qquad \frac{bf-ec}{den}=\frac{T_s^4\left(S_1S_3-S_2^2\right)}{den}=C_3$$

$$\frac{gf-di}{den}=\frac{T_s^5\left(S_2S_3-S_1S_4\right)}{den}=C_2 \qquad \frac{ai-gc}{den}=\frac{T_s^4\left(S_0S_4-S_2^2\right)}{den}=C_4 \qquad \frac{dc-af}{den}=\frac{T_s^3\left(S_1S_2-S_0S_3\right)}{den}=C_5$$

$$\frac{dh-ge}{den}=\frac{T_s^4\left(S_1S_3-S_2^2\right)}{den}=C_3 \qquad \frac{gb-ah}{den}=\frac{T_s^3\left(S_1S_2-S_0S_3\right)}{den}=C_5 \qquad \frac{ae-bd}{den}=\frac{T_s^2\left(S_0S_2-S_1^2\right)}{den}=C_6$$

Therefore Coefficients Become

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}=\begin{bmatrix} S_0 & S_1T_s & S_2T_s^2 \\ S_1T_s & S_2T_s^2 & S_3T_s^3 \\ S_2T_s^2 & S_3T_s^3 & S_4T_s^4 \end{bmatrix}^{-1}\begin{bmatrix} \sum_{i=1}^{L} x_i^* \\ \sum_{i=1}^{L} iT_sx_i^* \\ \sum_{i=1}^{L} (iT_s)^2\, x_i^* \end{bmatrix}$$

or

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}=\begin{bmatrix} C_1 & C_2 & C_3 \\ C_2 & C_4 & C_5 \\ C_3 & C_5 & C_6 \end{bmatrix}\begin{bmatrix} \sum_{i=1}^{L} x_i^* \\ \sum_{i=1}^{L} iT_sx_i^* \\ \sum_{i=1}^{L} (iT_s)^2\, x_i^* \end{bmatrix}$$

Quadratic Finite Memory Filter-1

Collect measurements for certain period of time initially (Specify window or L)

Evaluate quadratic coefficients

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} S_0 & S_1T_s & S_2T_s^2 \\ S_1T_s & S_2T_s^2 & S_3T_s^3 \\ S_2T_s^2 & S_3T_s^3 & S_4T_s^4 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{L} x_i^* \\ \sum_{i=1}^{L} iT_s x_i^* \\ \sum_{i=1}^{L} (iT_s)^2 x_i^* \end{bmatrix}$$

or

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} C_1 & C_2 & C_3 \\ C_2 & C_4 & C_5 \\ C_3 & C_5 & C_6 \end{bmatrix} \begin{bmatrix} \sum_{i=1}^{L} x_i^* \\ \sum_{i=1}^{L} iT_s x_i^* \\ \sum_{i=1}^{L} \left(iT_s\right)^2 x_i^* \end{bmatrix}$$

At end of data collection period estimate states using coefficients

$$\hat{x} = a_0 + LT_s a_1 + \left(LT_s\right)^2 a_2$$

$$\hat{\dot{x}} = a_1 + 2LT_s a_2$$

$$\hat{\ddot{x}} = 2a_2$$

Quadratic Finite Memory Filter-2

After initial data collection period, add new measurement and eliminate oldest measurement and reevaluate

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} S_0 & S_1T_s & S_2T_s^2 \\ S_1T_s & S_2T_s^2 & S_3T_s^3 \\ S_2T_s^2 & S_3T_s^3 & S_4T_s^4 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{L} x_i^* \\ \sum_{i=1}^{L} iT_s x_i^* \\ \sum_{i=1}^{L} (iT_s)^2 x_i^* \end{bmatrix}$$

or

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} C_1 & C_2 & C_3 \\ C_2 & C_4 & C_5 \\ C_3 & C_5 & C_6 \end{bmatrix} \begin{bmatrix} \sum_{i=1}^{L} x_i^* \\ \sum_{i=1}^{L} iT_s x_i^* \\ \sum_{i=1}^{L} \left(iT_s\right)^2 x_i^* \end{bmatrix}$$

Unchanged

Changes Slightly

And get new estimate

$$\hat{x} = a_0 + LT_s a_1 + \left(LT_s\right)^2 a_2$$

$$\hat{\dot{x}} = a_1 + 2LT_s a_2$$

$$\hat{\ddot{x}} = 2a_2$$

Therefore this method appears to be a batch processing method only for the initial data collection

Recall We Have Shown That a Quadratic Batch Processing Least Squares Filter Can Be Made Recursive

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

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

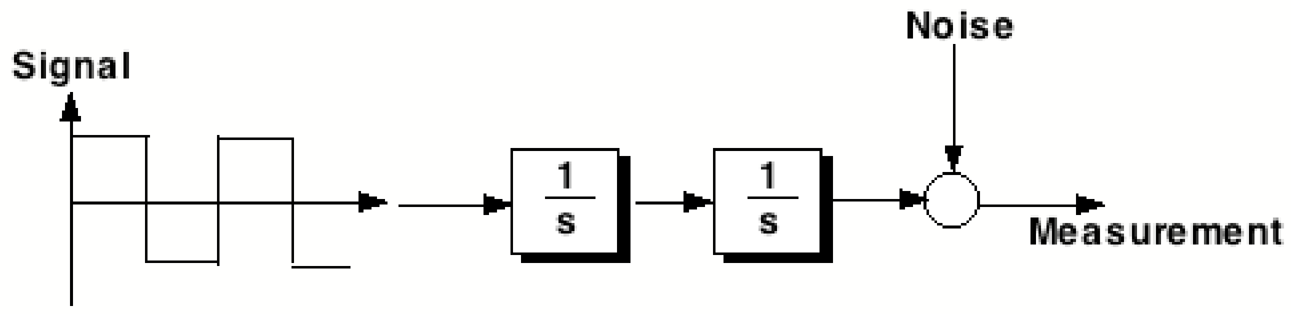
$$\hat{x}_k = \hat{x}_{k-1} + \hat{\dot{x}}_{k-1}T_s + .5\hat{\ddot{x}}_{k-1}T_s^2 + K_{1_k}Res_k$$

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

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

Therefore initial data collection period can be made recursive so that estimates are always available

Estimate Acceleration With Filter



We Measure Noisy Position and Want to Estimate Square Wave Acceleration

Quadratic Finite Memory Filter -1

```
GLOBAL DEFINE
        INCLUDE 'quickdraw.inc'
END
IMPLICIT REAL*8(A-H,O-Z)
REAL*8 Z(1000)
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
SIG=1.
XH=0.
XDH=0.
XDDH=0.
XL=140.
L=INT(XL)
TS=.1
S0=XL+1.
S1=.5*XL*(XL+1.)
S2=XL*(XL+1.)*(2.*XL+1.)/6.
S3=((XL*(XL+1.))**2)/4.
S4=XL*(XL+1.)*(2.*XL+1.)*(3.*XL*XL+3.*XL-1.)/30.
C1=S2*S4-S3*S3
C2=S2*S3-S1*S4
C3=S1*S3-S2*S2
DEN=S0*S2*S4-S0*S3*S3-S1*S1*S4+2.*S1*S2*S3-S2**3
C4=S0*S4-S2*S2
C5=S1*S2-S0*S3
C6=S0*S2-S1*S1
XN=0.
XTD=0.
XT=0.
```

Noise Standard Deviation

Window Length

Sampling Time

Short Hand Terms in Matrix To Be Inverted

Terms in Matrix Inverse

Quadratic Finite Memory Filter -2

```
DO 10 T=0.,100.,TS
XN=XN+1.
N=INT(XN)
IF(T<25.)THEN
        XTDD=1.
ELSEIF(T<50.)THEN
        XTDD=-1.
ELSEIF(T<75.)THEN
        XTDD=1.
ELSE
        XTDD=-1.
ENDIF
XTD=XTD+TS*XTDD
XT=XT+TS*XTD+.5*TS*TS*XTDD
CALL GAUSS(XNOISE,SIG)
XTMEAS1=XT+XNOISE
Z(N)=XTMEAS1
```

Actual Acceleration

Loop

Euler Integration to Get Velocity and Position

Noisy Measured Position

Save Measurements in Array

Quadratic Finite Memory Filter -3

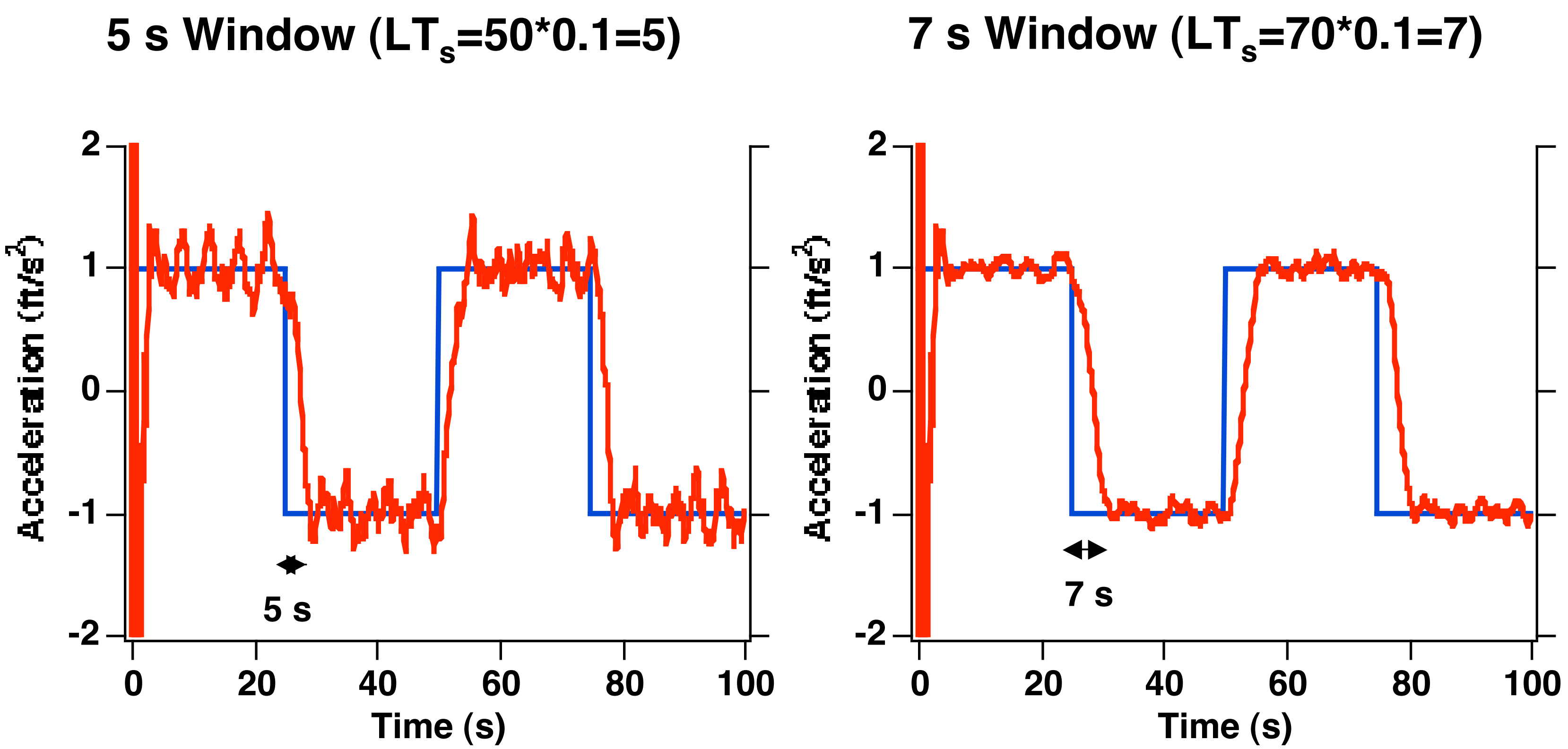
```
        IF(N<=(L+1))THEN
                XK1=3*(3*XN*XN-3*XN+2)/(XN*(XN+1)*(XN+2))
                XK2=18*(2*XN-1)/(XN*(XN+1)*(XN+2)*TS)
                XK3=60/(XN*(XN+1)*(XN+2)*TS*TS)
                RES=XTMEAS1-XH-TS*XDH-.5*TS*TS*XDDH
                XH=XH+XDH*TS+.5*TS*TS*XDDH+XK1*RES
                XDH=XDH+XDDH*TS+XK2*RES
                XDDH=XDDH+XK3*RES
        ELSE
                U=0.
                V=0.
                W=0.
                DO 20 I=0,L
                        U=U+Z(N-L+I)
                        XI=FLOAT(I)
                        V=V+XI*Z(N-L+I)
                        W=W+XI*XI*Z(N-L+I)
20              CONTINUE
                A0=(C1*U+C2*V+C3*W)/DEN
                A1=(C2*U+C4*V+C5*W)/(TS*DEN)
                A2=(C3*U+C5*V+C6*W)/(TS*TS*DEN)
                XH=A0+XL*TS*A1+XL*XL*TS*TS*A2
                XDH=A1+2.*A2*XL*TS
                XDDH=2.*A2
        ENDIF
        WRITE(9,*)T,XT,XH,XTD,XDH,XTDD,XDDH
        WRITE(1,*)T,XT,XH,XTD,XDH,XTDD,XDDH
10      CONTINUE
        CLOSE(1)
        PAUSE
        END
```

Initially Use Recursive Least Squares Filter Until Window Is Filled

Evaluate Summations By Adding New Measurement and Throwing Out Oldest

Calculate Coefficients And Obtain State Estimates

Acceleration Estimates For Different Window Size - 1

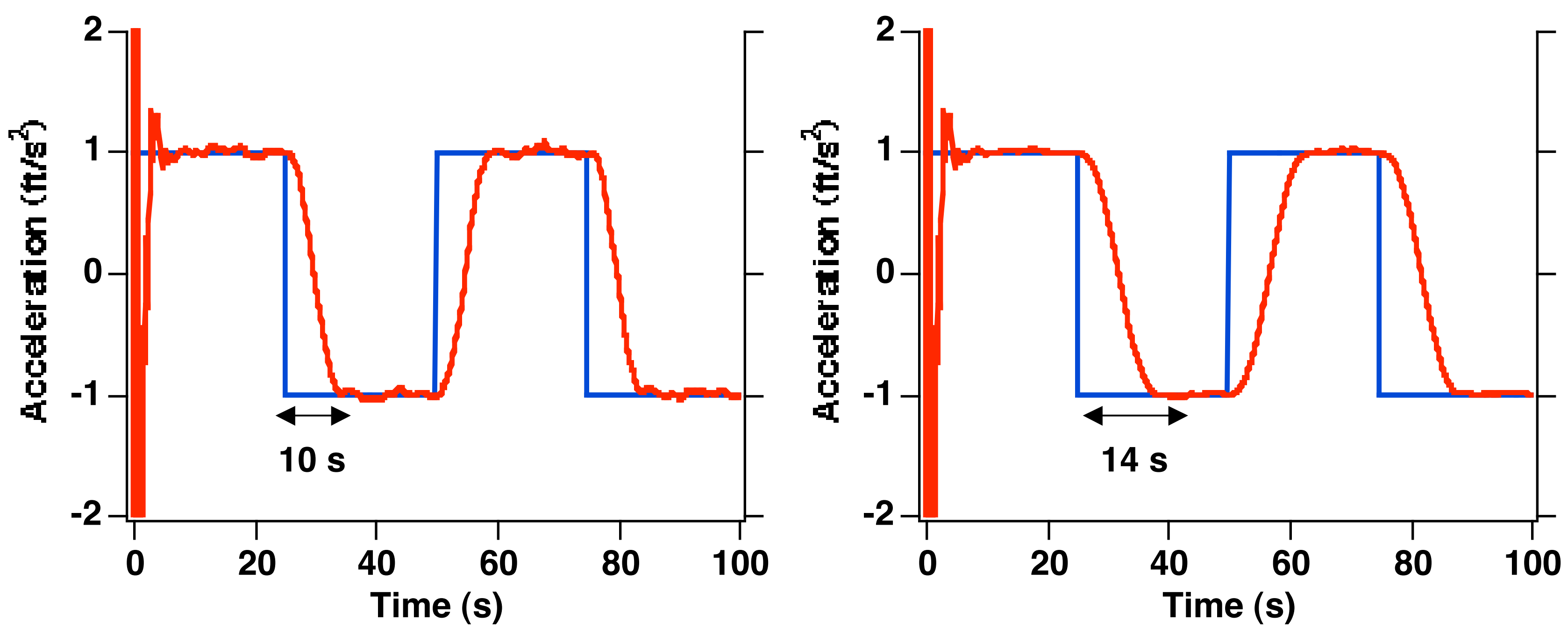


Increasing window length removes noise but increases amount of time required to estimate acceleration switching

Acceleration Estimates For Different Window Size - 2

10 s Window (LT_s=100*0.1=10)

14 s Window (LT_s=140*0.1=14)



Increasing window length removes noise but increases amount of time required to estimate acceleration switching

Important Matrices For 3-State Kalman Filter

Order	Continuous Q	Fundamental	Discrete Q
0	$Q = \Phi_s$	$\Phi_k = 1$	$Q_k = \Phi_s$
1	$Q = \Phi_s \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$	$\Phi_k = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}$	$Q_k = \Phi_s \begin{bmatrix} \frac{T_s^3}{3} & \frac{T_s^2}{2} \\ \frac{T_s^2}{2} & T_s \end{bmatrix}$
2	$Q = \Phi_s \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\Phi_k = \begin{bmatrix} 1 & T_s & .5T_s^2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}$	$Q_k = \Phi_s \begin{bmatrix} \frac{T_s^5}{20} & \frac{T_s^4}{8} & \frac{T_s^3}{6} \\ \frac{T_s^4}{8} & \frac{T_s^3}{3} & \frac{T_s^2}{2} \\ \frac{T_s^3}{6} & \frac{T_s^2}{2} & T_s \end{bmatrix}$

3-State Polynomial Kalman Filter - 1

For a 3-state polynomial Kalman filter we assume

$$\dddot{x} = u_s$$

In state space form our model of the real world is

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

Polynomial filter formulation

Measurement model

$$x^* = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + v$$

3-State Polynomial Kalman Filter - 2

Therefore the fundamental and measurement noise matrices are

$$\mathbf{\Phi_k} = \begin{bmatrix} 1 & T_s & .5T_s^2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{H} = [1 \quad 0 \quad 0]$$

Recall Kalman filtering equation is given by

$$\mathbf{\hat{x}_k} = \mathbf{\Phi_k \hat{x}_{k\text{-}1}} + \mathbf{K_k}(\mathbf{z_k} - \mathbf{H\Phi_k \hat{x}_{k\text{-}1}})$$

Substitution and simplification yields second-order polynomial Kalman filter

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

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

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

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

Gains obtained from Riccati equations

$$\mathbf{M_k} = \mathbf{\Phi_k P_{k\text{-}1}\Phi_k^T} + \mathbf{Q_k}$$

$$\mathbf{K_k} = \mathbf{M_k H^T}(\mathbf{HM_k H^T} + \mathbf{R_k})^{-1}$$

$$\mathbf{P_k} = (\mathbf{I} - \mathbf{K_k H})\mathbf{M_k}$$

$$\mathbf{R_k} = \sigma_n^2$$

$$\mathbf{Q_k} = \Phi_s \begin{bmatrix} \frac{T_s^5}{20} & \frac{T_s^4}{8} & \frac{T_s^3}{6} \\ \frac{T_s^4}{8} & \frac{T_s^3}{3} & \frac{T_s^2}{2} \\ \frac{T_s^3}{6} & \frac{T_s^2}{2} & T_s \end{bmatrix}$$

3-State Kalman Filter - 1

```
        GLOBAL DEFINE
                INCLUDE 'quickdraw.inc'
        END
        IMPLICIT REAL*8(A-H,O-Z)
        REAL*8 P(3,3),Q(3,3),M(3,3),PHI(3,3),HMAT(1,3),HT(3,1),PHIT(3,3)
        REAL*8 RMAT(1,1),IDN(3,3),PHIP(3,3),PHIPPHIT(3,3),HM(1,3)
        REAL*8 HMHT(1,1),HMHTR(1,1),HMHTRINV(1,1),MHT(3,1),K(3,1)
        REAL*8 KH(23,3),IKH(3,3)
        INTEGER ORDER
        OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
        ORDER =3
        PHIS=.00001
        TS=.1
        XH=0.
        XDH=0.
        XDDH=0
```

Initial State Estimates

```
        SIGNOISE=1.
        DO 14 I=1,ORDER
        DO 14 J=1,ORDER
        PHI(I,J)=0.
        P(I,J)=0.
        Q(I,J)=0.
        IDN(I,J)=0.
14      CONTINUE
```

Zero Out Matrices Initially

```
        RMAT(1,1)=SIGNOISE**2
```

R Matrix

```
        IDN(1,1)=1.
        IDN(2,2)=1.
        IDN(3,3)=1.
```

Identity Matrix

3-State Kalman Filter - 2

```
P(1,1)=99999999999.
P(2,2)=99999999999.
P(3,3)=99999999999.
```

Initial Covariance Matrix

```
PHI(1,1)=1
PHI(1,2)=TS
PHI(1,3)=.5*TS*TS
PHI(2,2)=1
PHI(2,3)=TS
PHI(3,3)=1
```

Fundamental Matrix, Φ

```
HMAT(1,1)=1.
HMAT(1,2)=0.
HMAT(1,3)=0.
```

H Matrix

```
CALL MATTRN(PHI,ORDER,ORDER,PHIT)
CALL MATTRN(HMAT,1,ORDER,HT)
```

Φ^T and H^T

```
Q(1,1)=PHIS*TS**5/20
Q(1,2)=PHIS*TS**4/8
Q(1,3)=PHIS*TS**3/6
Q(2,1)=Q(1,2)
Q(2,2)=PHIS*TS**3/3
Q(2,3)=PHIS*TS*TS/2
Q(3,1)=Q(1,3)
Q(3,2)=Q(2,3)
Q(3,3)=PHIS*TS
```

Q Matrix

```
XD=0.
X=0.
```

3-State Kalman Filter - 3

```
        DO 10 T=0.,100.,TS
                CALL MATMUL(PHI,ORDER,ORDER,P,ORDER,ORDER,PHIP)
                CALL MATMUL(PHIP,ORDER,ORDER,PHIT,ORDER,ORDER,PHIPPHIT)
                CALL MATADD(PHIPPHIT,ORDER,ORDER,Q,M)
                CALL MATMUL(HMAT,1,ORDER,M,ORDER,ORDER,HM)
                CALL MATMUL(HM,1,ORDER,HT,ORDER,1,HMHT)
                CALL MATADD(HMHT,1,1,RMAT,HMHTR)
                HMHTRINV(1,1)=1./HMHTR(1,1)
                CALL MATMUL(M,ORDER,ORDER,HT,ORDER,1,MHT)
                CALL MATMUL(MHT,ORDER,1,HMHTRINV,1,1,K)
                CALL MATMUL(K,ORDER,1,HMAT,1,ORDER,KH)
                CALL MATSUB(IDN,ORDER,ORDER,KH,IKH)
                CALL MATMUL(IKH,ORDER,ORDER,M,ORDER,ORDER,P)
                CALL GAUSS(XNOISE,SIGNOISE)
                IF(T<25.)THEN
                        XDD=1.
                ELSEIF(T<50.)THEN
                        XDD=-1.
                ELSEIF(T<75.)THEN
                        XDD=1.
                ELSE
                        XDD=-1.
                ENDIF
                XD=XD+TS*XDD
                X=X+TS*XD+.5*TS*TS*XDD
                XS=X+XNOISE
                RES=XS-XH-TS*XDH-.5*TS*TS*XDDH
                XH=XH+XDH*TS+.5*TS*TS*XDDH+K(1,1)*RES
                XDH=XDH+XDDH*TS+K(2,1)*RES
                XDDH=XDDH+K(3,1)*RES
                WRITE(9,*)T,X,XH,XD,XDH,XDD,XDDH
                WRITE(1,*)T,X,XH,XD,XDH,XDD,XDDH
10      CONTINUE
        PAUSE
        CLOSE(1)
        END
```

Ricatti Equations

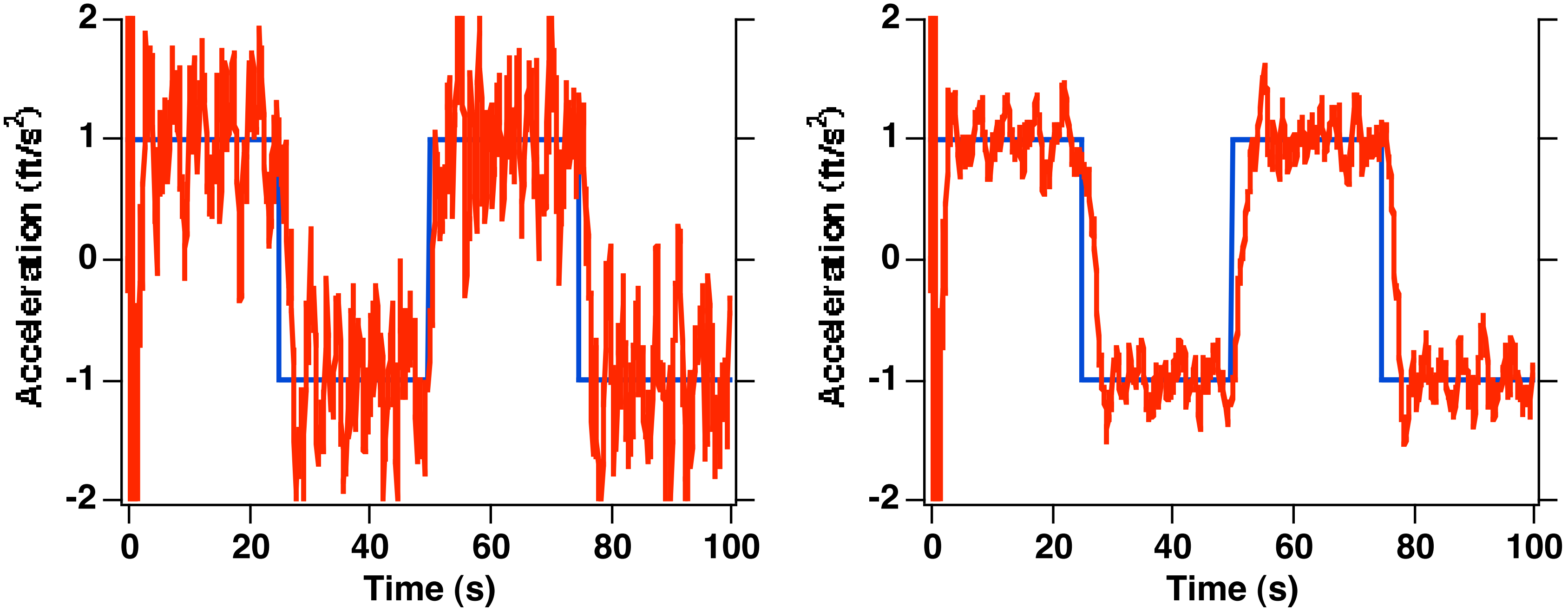
Actual Acceleration

Euler Integration to Get Velocity and Position

Measurement and Kalman Filter

Loop

Acceleration Estimates of 3-State Kalman Filter For Different Values of Process Noise - 1



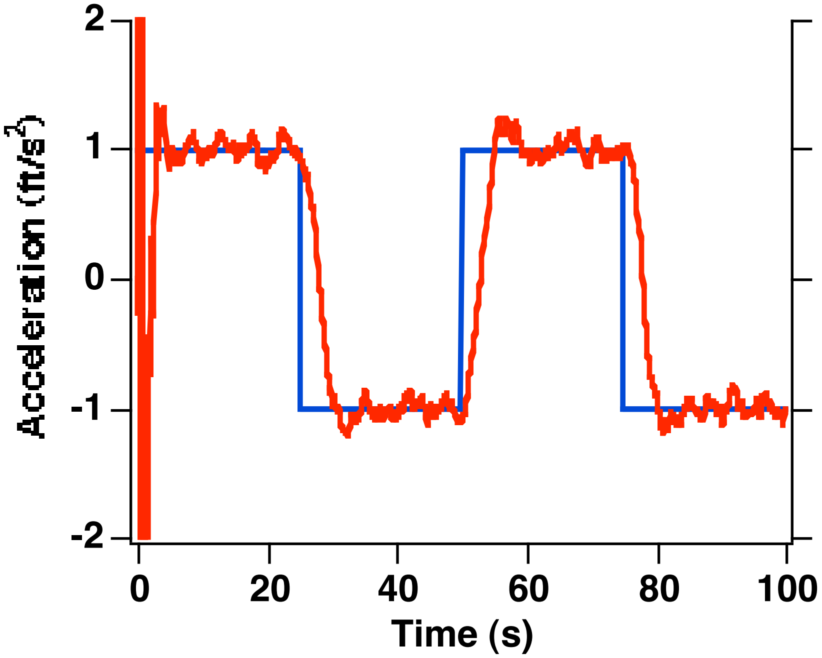
$\Phi_s=1$

$\Phi_s=0.1$

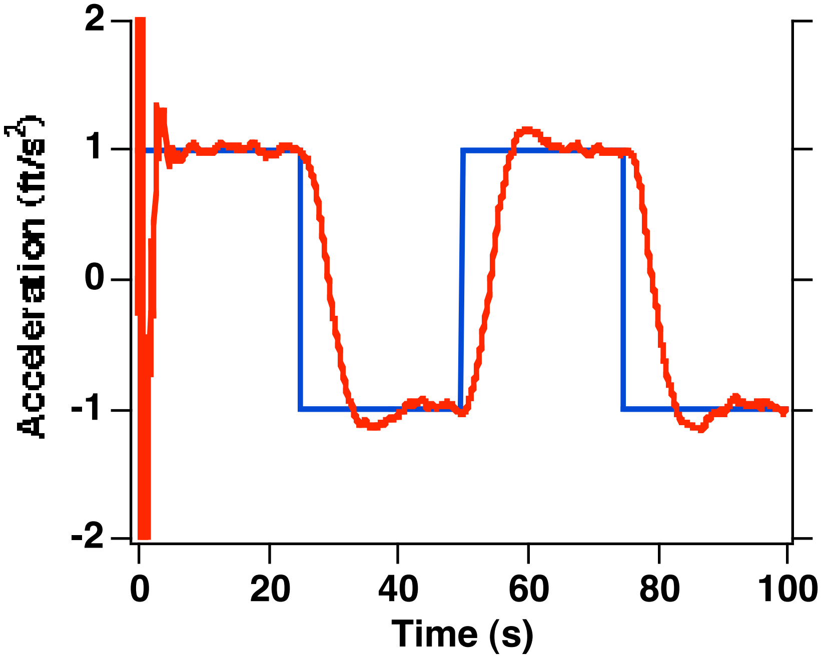
Decreasing process noise reduces noise on acceleration estimate

Acceleration Estimates of 3-State Kalman Filter For Different Values of Process Noise - 2

Φ_s=0.01

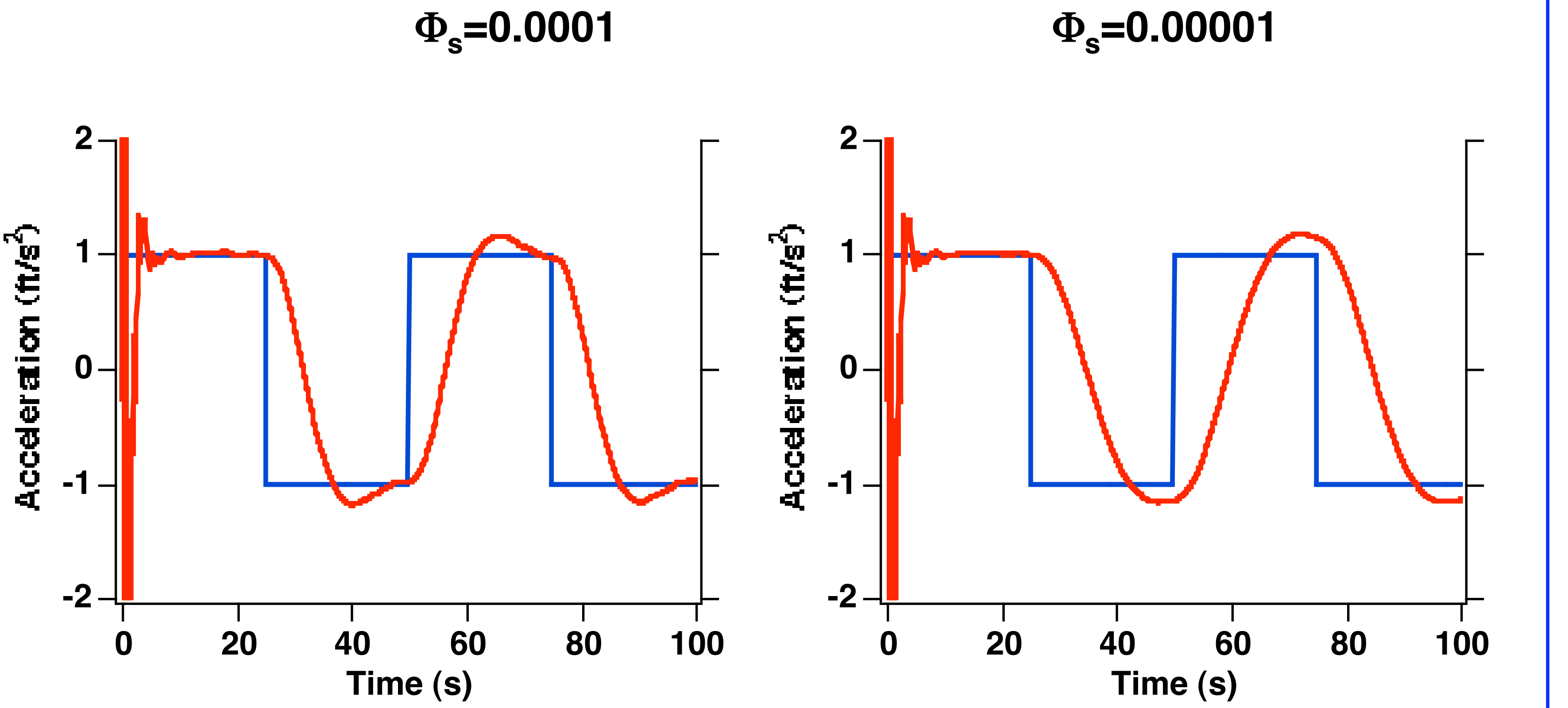


Φ_s=0.001



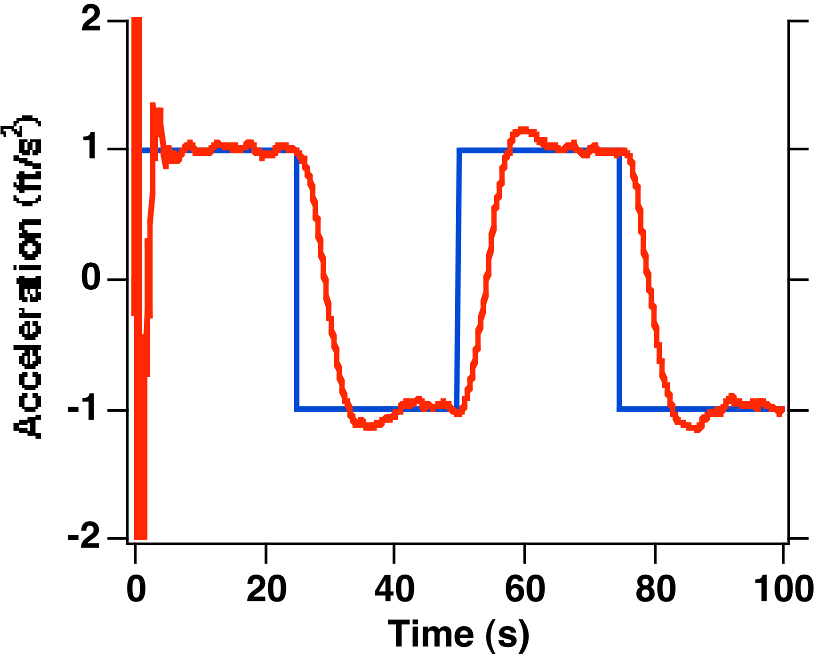
Decreasing process noise further reduces noise on acceleration estimate increases amount of time required to estimate acceleration switching

Acceleration Estimates of 3-State Kalman Filter For Different Values of Process Noise - 3

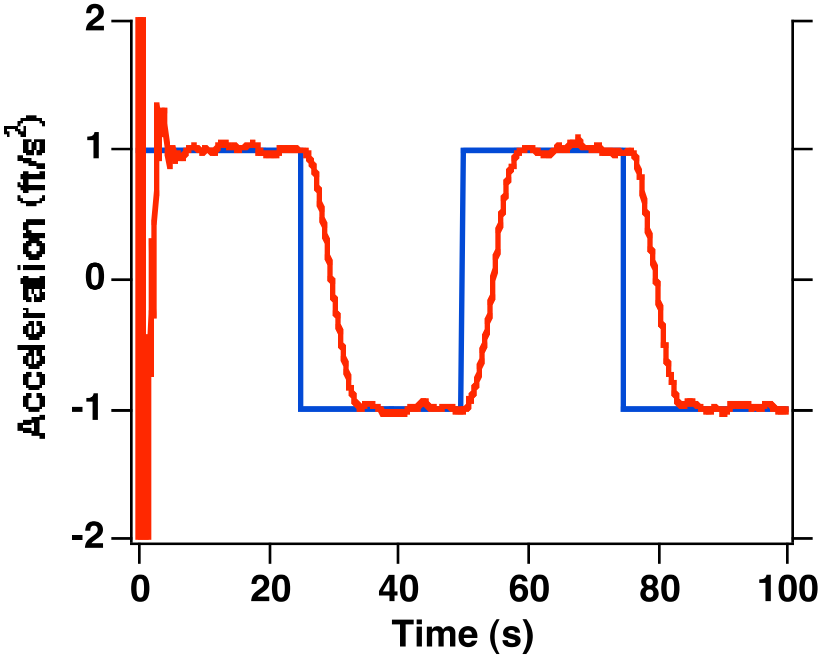


Decreasing process noise even further eliminates noise but acceleration estimate does not track actual acceleration

“Best” Acceleration Estimate Comparison



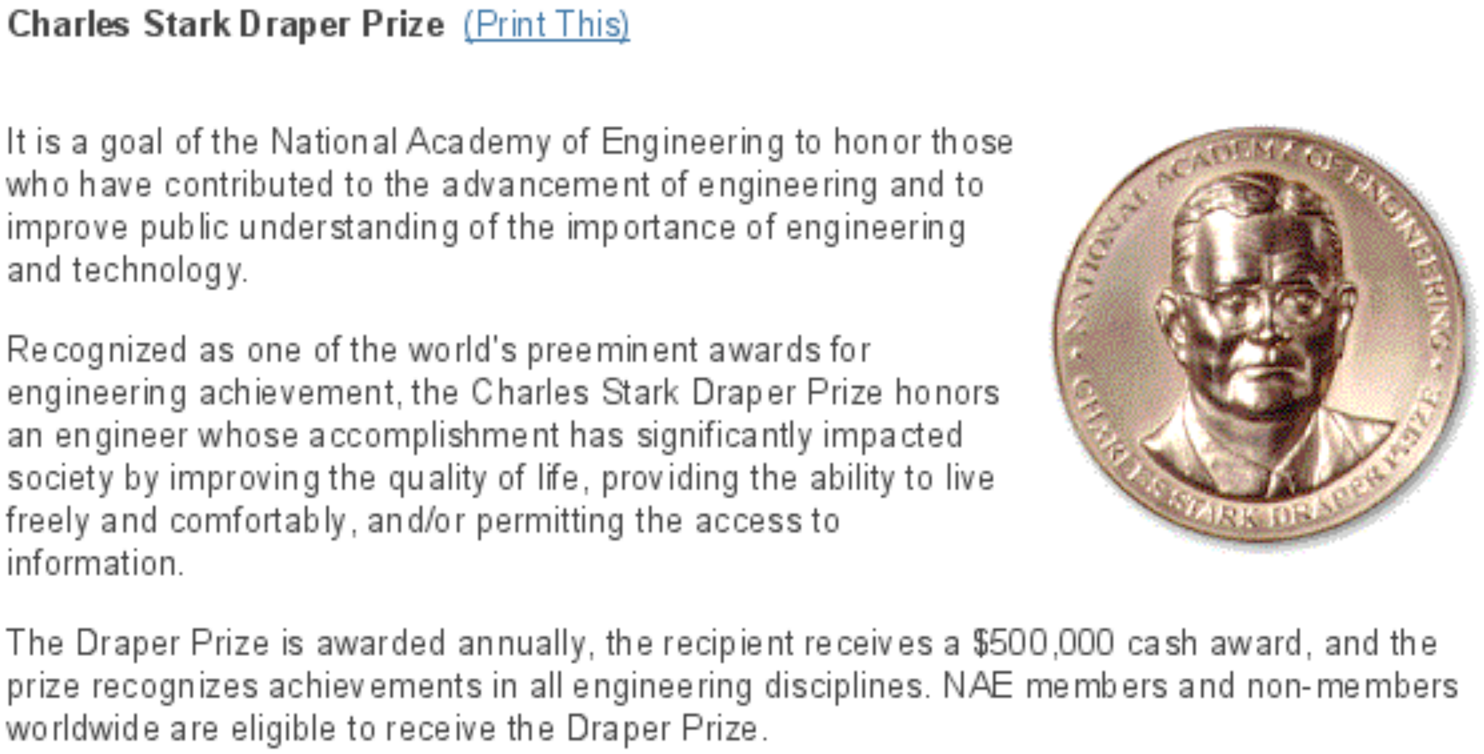
3 State Kalman Filter
Φ_s=0.001



Finite Memory
10 s Window (LT$_s$=100*0.1=10)

Finite memory filter yields better estimates when there are abrupt changes in acceleration

Draper Prize



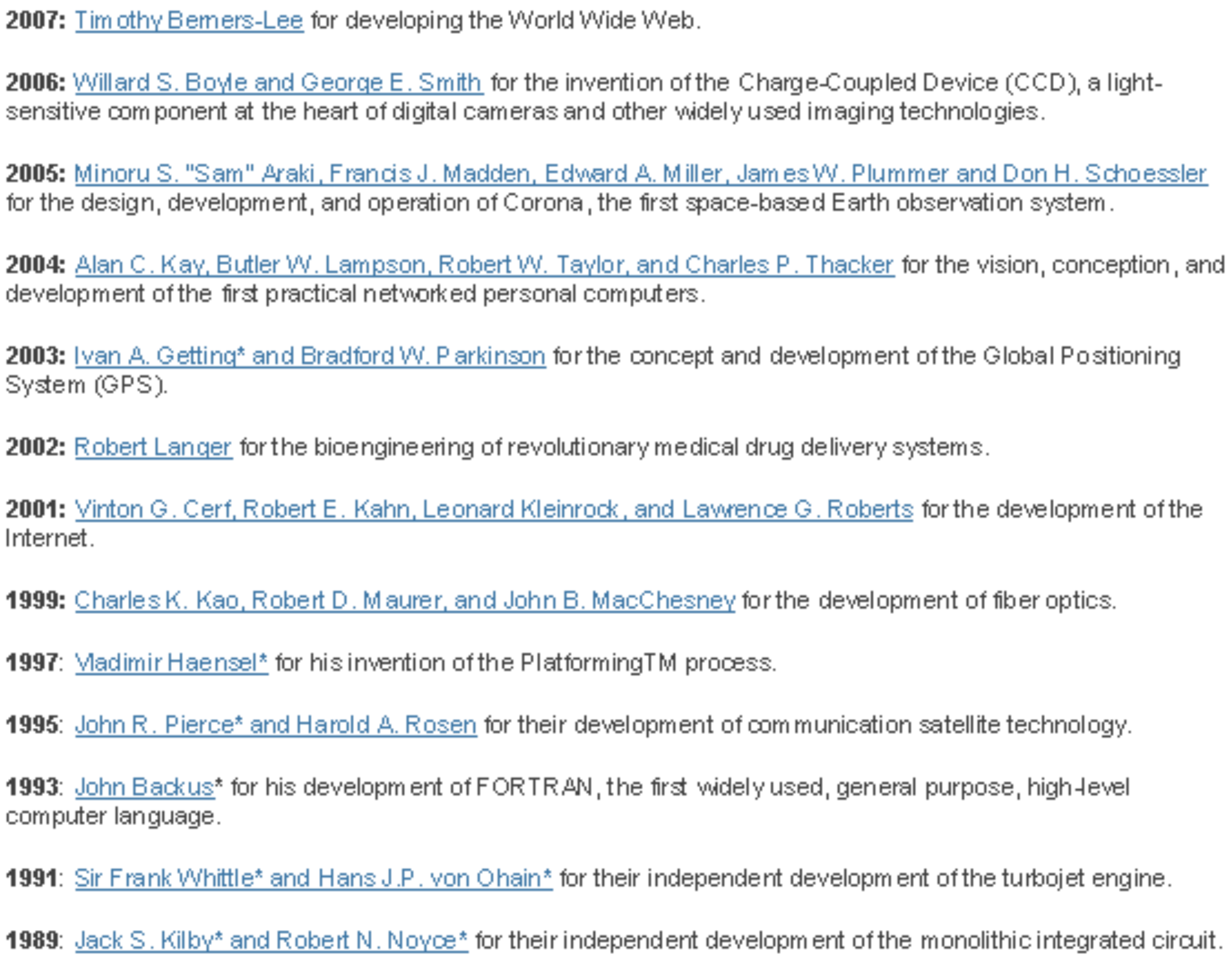
Charles Stark Draper Prize (Print This)

It is a goal of the National Academy of Engineering to honor those who have contributed to the advancement of engineering and to improve public understanding of the importance of engineering and technology.

Recognized as one of the world's preeminent awards for engineering achievement, the Charles Stark Draper Prize honors an engineer whose accomplishment has significantly impacted society by improving the quality of life, providing the ability to live freely and comfortably, and/or permitting the access to information.

The Draper Prize is awarded annually, the recipient receives a $500,000 cash award, and the prize recognizes achievements in all engineering disciplines. NAE members and non-members worldwide are eligible to receive the Draper Prize.

Previous Recipients of Draper Prize



2007: Timothy Berners-Lee for developing the World Wide Web.

2006: Willard S. Boyle and George E. Smith for the invention of the Charge-Coupled Device (CCD), a light-sensitive component at the heart of digital cameras and other widely used imaging technologies.

2005: Minoru S. "Sam" Araki, Francis J. Madden, Edward A. Miller, James W. Plummer and Don H. Schoessler for the design, development, and operation of Corona, the first space-based Earth observation system.

2004: Alan C. Kay, Butler W. Lampson, Robert W. Taylor, and Charles P. Thacker for the vision, conception, and development of the first practical networked personal computers.

2003: Ivan A. Getting* and Bradford W. Parkinson for the concept and development of the Global Positioning System (GPS).

2002: Robert Langer for the bioengineering of revolutionary medical drug delivery systems.

2001: Vinton G. Cerf, Robert E. Kahn, Leonard Kleinrock, and Lawrence G. Roberts for the development of the Internet.

1999: Charles K. Kao, Robert D. Maurer, and John B. MacChesney for the development of fiber optics.

1997: Vladimir Haensel* for his invention of the PlatformingTM process.

1995: John R. Pierce* and Harold A. Rosen for their development of communication satellite technology.

1993: John Backus* for his development of FORTRAN, the first widely used, general purpose, high-level computer language.

1991: Sir Frank Whittle* and Hans J.P. von Ohain* for their independent development of the turbojet engine.

1989: Jack S. Kilby* and Robert N. Noyce* for their independent development of the monolithic integrated circuit.

My Nominating Letter For Dr. Kalman - 1

Dr. Robert W. Bass
45960 Indian Way
#612
Lexington Park, MD 20653

Dear Dr. Bass

In the past the Draper Prize has mainly been awarded for significant hardware developments that have influenced the world. In 1993 John Backus was awarded the Draper Prize for his development of FORTRAN which was a revolutionary software development for the time. Although FORTRAN is still in use today it has taken the back seat to other computer languages. The Kalman filter can also be considered to be a software development, but unlike FORTRAN, it is even more popular today than it was 44 years ago when it was invented. There are still no serious software filtering rivals to the Kalman filter. In fact many of the software innovations which were required from the 1960's through the 1980's for the practical implementation of the Kalman filter are no longer required today because of the advances in computer technology. In other words, the Kalman filter is even easier to implement today than it was years ago.

I also believe that the development of the Kalman filter also satisfies each of the criteria developed by the Draper Prize Committee. For example

- Anybody who uses a GPS receiver or flies in a commercial aircraft benefits from the Kalman filter
- The idea that filtering could be done systematically was a breakthrough during the 1960's because most filtering schemes were ad hoc at the time. I believe the original paper on Kalman filtering was first rejected by the IEEE journals because they felt the idea behind the paper could not possibly be true.
- The Wiener filter was a popular theoretical filter before the Kalman filter. Its only disadvantage was that engineers could not apply it to practical problems because of the complex equations in the frequency domain that had to be solved. The Kalman filter was a straight forward time domain algorithm for solving the filtering problem. In fact today it can be shown that the Wiener filter is a subset of the Kalman filter.

My Nominating Letter For Dr. Kalman - 2

- The richness in the technical ramifications of the Kalman filter were enormous. Even if one forgets about all the filtering ramifications new insights were gained in the field of optimal control because it is actually related to the dual of the filtering problem. Kalman filtering also popularized the statistical analysis technique of covariance analysis. In fact the Kalman filter automatically provides internal estimates of how well it is doing by using covariance analysis.
- The follow-through in the Kalman filter after it was developed was also enormous. Practical innovations were made to get the filtering algorithm to work on the primitive computers of the time. In recent years computer technology has developed to the point where most, if not all, of the innovations are no longer necessary because often the original Kalman filter works quite well on modern flight computers.
- The economic impact of the Kalman filter is difficult to quantify but it must be measured at least in billions or possibly trillions of dollars.

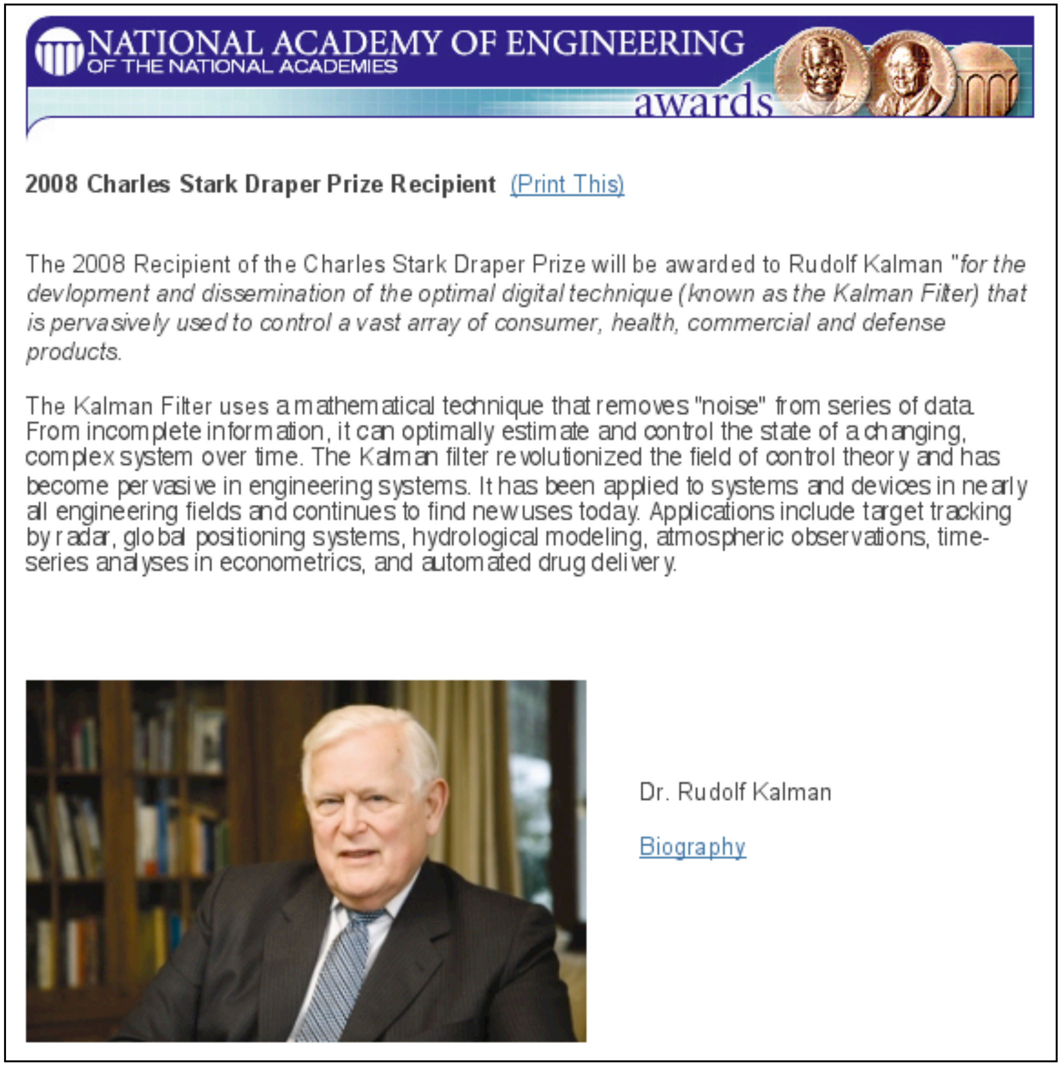
To further clarify why I think that Dr. Kalman should receive the Draper prize I have included some exhibits which go in to more explicit detail on the importance of Kalman filtering. two testimonials from Dr. Fred Daum and Dr. Howard Musoff. I totally agree with Dr. Fred Daum's testimonial in regard to the importance of the invention of the Kalman filter and to Dr. Howard Musoff's testimonial in regard to the importance of Kalman filtering to inertial navigation systems. In addition, I was personally involved with one of the first successful implementations of the Kalman filter to a homing missile guidance system during the 1960's. On a personal note, Howard died this week and I know he was honored in being asked to provide information that might enable Dr. Kalman to receive the Draper Prize.

I believe that the award of the Draper Prize to Dr. Kalman will be consistent with the excellent choices that have already been made by the Charles Stark Draper Prize selection committee since the award's inception.

Sincerely yours

Paul Zarchan

$500,000 Prize Announced on Web



NATIONAL ACADEMY OF ENGINEERING
OF THE NATIONAL ACADEMIES
awards

2008 Charles Stark Draper Prize Recipient (Print This)

The 2008 Recipient of the Charles Stark Draper Prize will be awarded to Rudolf Kalman "*for the devlopment and dissemination of the optimal digital technique (known as the Kalman Filter) that is pervasively used to control a vast array of consumer, health, commercial and defense products.*

The Kalman Filter uses a mathematical technique that removes "noise" from series of data. From incomplete information, it can optimally estimate and control the state of a changing, complex system over time. The Kalman filter revolutionized the field of control theory and has become pervasive in engineering systems. It has been applied to systems and devices in nearly all engineering fields and continues to find new uses today. Applications include target tracking by radar, global positioning systems, hydrological modeling, atmospheric observations, time-series analyses in econometrics, and automated drug delivery.

Dr. Rudolf Kalman

Biography

At Award Ceremony in DC