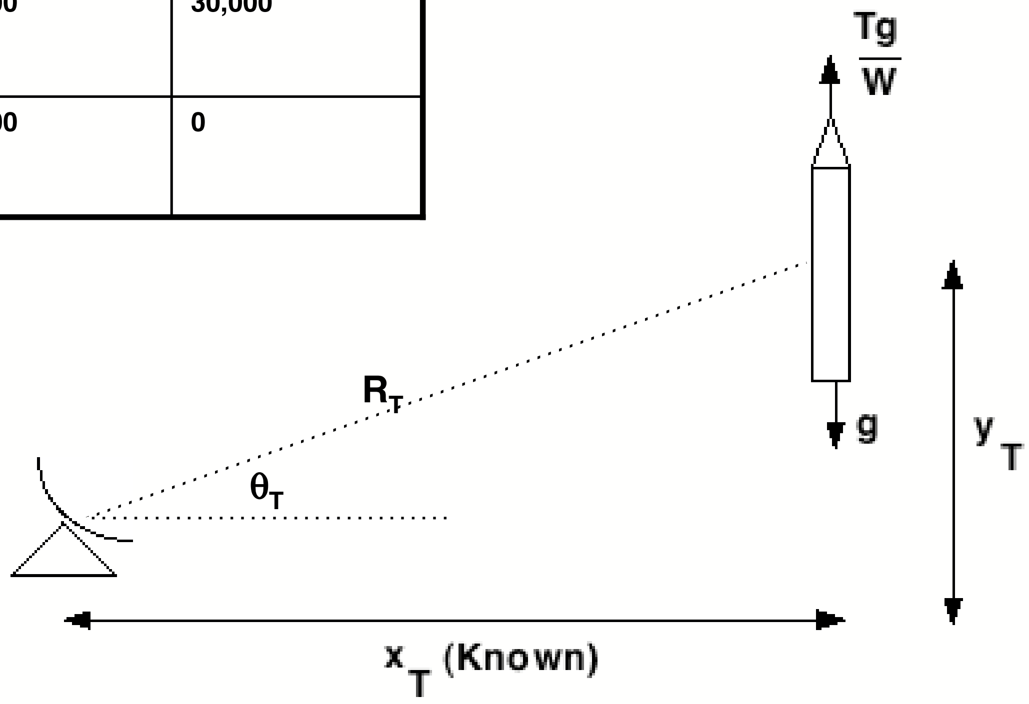
A Tracking Disaster

A Tracking Disaster
Overview

- Problem Setup
- Three-State Linear Polynomial Kalman Filter
- Six-State Linear Polynomial Kalman Filter
- Interesting Experiments
- Summary

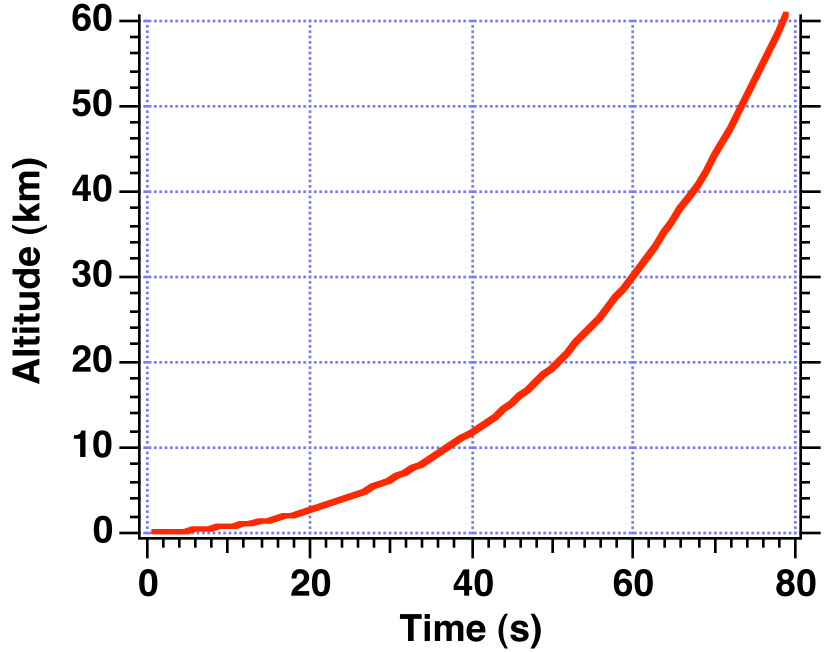
Problem Setup

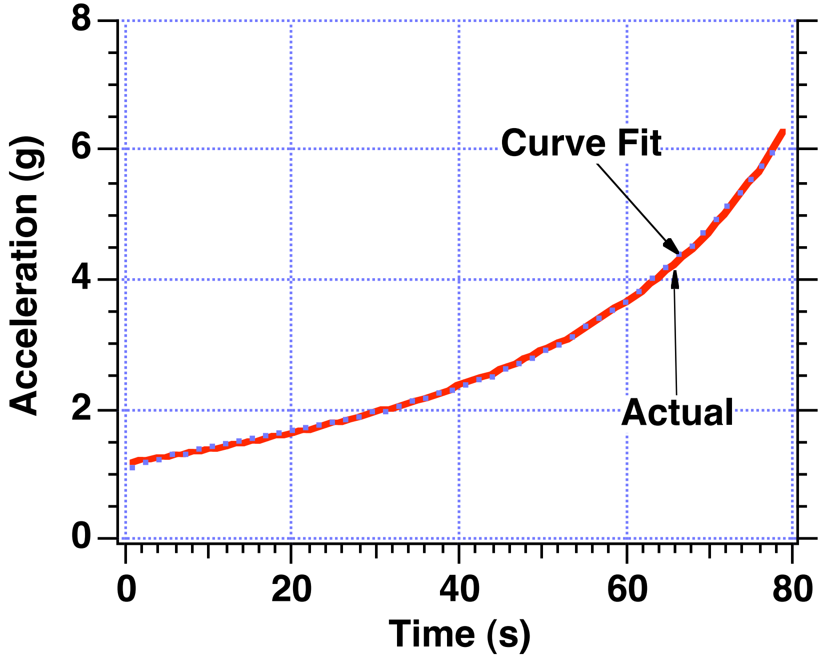
Time (s)	Weight (lbs)	Thrust (lbs)
0	14,000	30,000
80	4,000	30,000
80	4,000	0



If x_T is known, measuring R_T and θ_T is equivalent to measuring y_T

Target Altitude and Acceleration Profiles





Acceleration curve fit $n_T = \ddot{y}_T = \frac{33.8 + 1.24t - 0.0228t^2 + 0.000422t^3}{32.2} \rightarrow \dddot{n}_T = constant$

Potential perfect filter has six states

$$\begin{bmatrix} y_T \\ \dot{y}_T \\ \ddot{y}_T = n_T \\ \dot{n}_T \\ \ddot{n}_T \\ \dddot{n}_T = q_T \end{bmatrix}$$

Three-State Linear Polynomial Kalman Filter

Recall Important Matrices for Different Order Polynomial Kalman Filters

Important matrices for 3-state Kalman filter →

Order	Systmes Dynamics	Fundamental	Measurement	Noise
0	$\mathbf{F}=1$	$\Phi_k=1$	$\mathbf{H}=1$	$\mathbf{R_k}=\sigma_n^2$
1	$\mathbf{F}=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}$	$\Phi_k=\begin{bmatrix}1 & \mathbf{T_s}\\ 0 & 1\end{bmatrix}$	$\mathbf{H}=[1 \quad 0]$	$\mathbf{R_k}=\sigma_n^2$
2	$\mathbf{F}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{bmatrix}$	$\Phi_k=\begin{bmatrix}1 & T_s & .5T_s^2\\ 0 & 1 & T_s\\ 0 & 0 & 1\end{bmatrix}$	$\mathbf{H}=[1 \quad 0 \quad 0]$	$\mathbf{R_k}=\sigma_n^2$

Recall That The Discrete Process Noise Matrix Varies With System Order

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

Process noise matrix for 3-state Kalman filter →

Filter gains are obtained from Riccati equations

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

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

$$P_k = (I - K_k H) M_k$$

Three-State Linear Polynomial 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(3,3),IKH(3,3)
        INTEGER ORDER,STEP
        OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
        OPEN(2,STATUS='UNKNOWN',FILE='COVFIL')
        G=32.2
        XNT=193.2
        YT=0.
        YTD=0.
        ORDER=3
        PHIS=XNT*XNT/120.
        TS=1.
        YH=0.
        YDH=0.
        YDDH=0
        SIGNOISE=100.
        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
        RMAT(1,1)=SIGNOISE**2
        IDN(1,1)=1.
        IDN(2,2)=1.
        IDN(3,3)=1.
        P(1,1)=100.**2
        P(2,2)=100.**2
        P(3,3)=64.4**2
```

Process noise

Sampling time, initial state estimates and measurement noise

Measurement noise matrix

Initial covariance matrix

Three-State Linear Polynomial Kalman Filter-2

```
               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
```

Non zero elements of fundamental matrix

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

Measurement matrix

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

Process noise matrix

```
               S=0.
               H=.01
               T=0.
10             IF(T>79.)GOTO 999
               S=S+H
               YTOLD=YT
               YTDOLD=YTD
               STEP=1
               GOTO 200
66             STEP=2
               YT=YT+H*YTD
               YTD=YTD+H*YTDD
               T=T+H
               GOTO 200
55             CONTINUE
               YT=.5*(YTOLD+YT+H*YTD)
               YTD=.5*(YTDOLD+YTD+H*YTDD)
```

Second-order Runge-Kutta Integration for booster

Three-State Linear Polynomial Kalman Filter-3

```
IF(S<(TS-.00001))GOTO 10
S=0.
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,ORDER,ORDER,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)
```

Riccati equations

```
CALL GAUSS(YNOISE,SIGNOISE)
YS=YT+YNOISE
```

Measurement noise

```
XK1=K(1,1)
XK2=K(2,1)
XK3=K(3,1)
RES=YS-YH-TS*YDH-.5*TS*TS*YDDH
YH=YH+YDH*TS+.5*TS*TS*YDDH+XK1*RES
YDH=YDH+YDDH*TS+XK2*RES
YDDH=YDDH+XK3*RES
```

Three-state filter

```
ERRY=YT-YH
ERRYD=YTD-YDH
ERRYDD=YTDD-YDDH
```

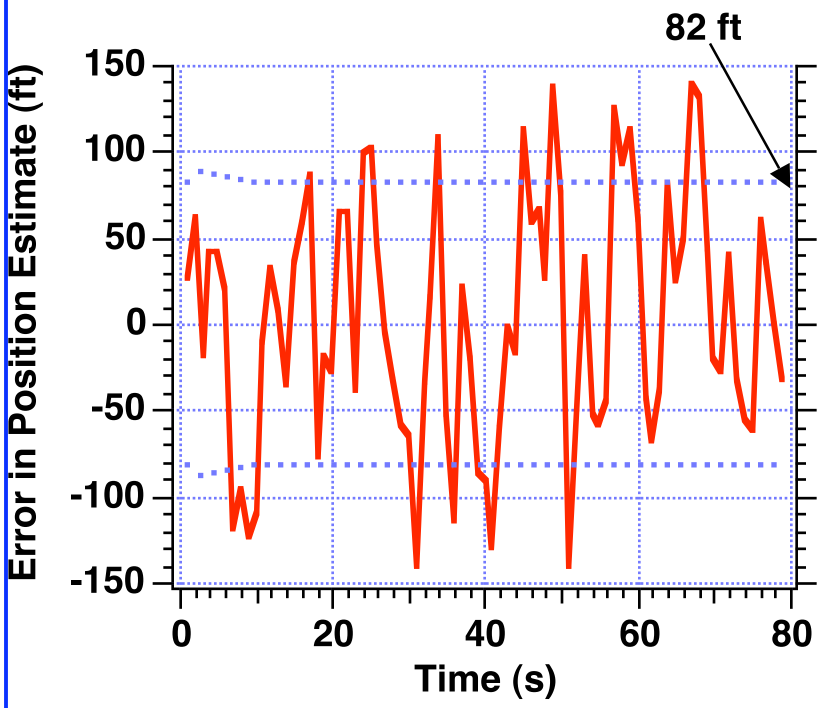
Three-State Linear Polynomial Kalman Filter-4

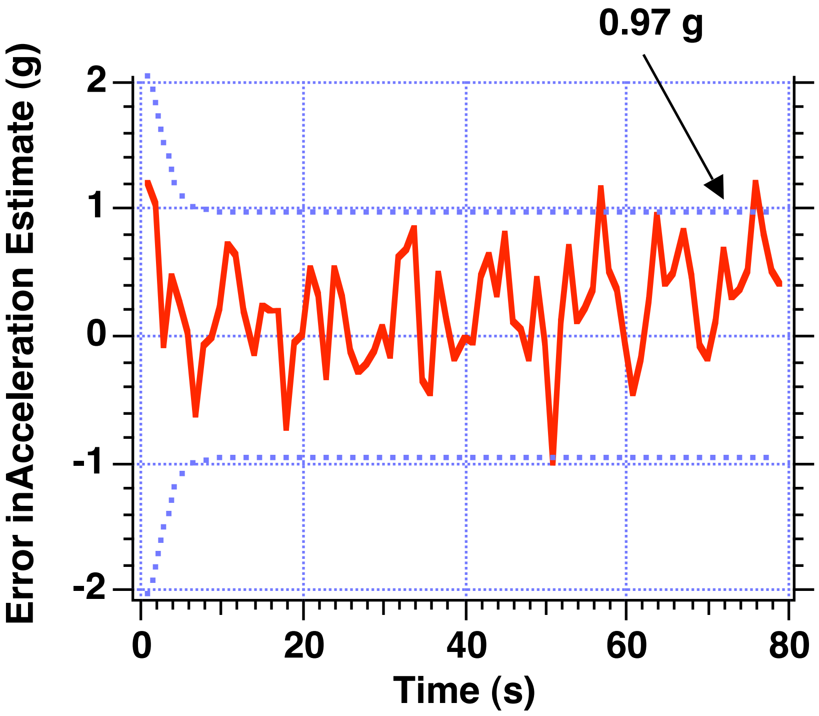
```
              SP11=SQRT(P(1,1))
              SP22=SQRT(P(2,2))
              SP33=SQRT(P(3,3))
              WRITE(9,*)T,YTDD/32.2,YDDH/32.2
              WRITE(1,*)T,YTDD/32.2,YDDH/32.2,YTD,YDH,ZDH
              WRITE(2,*)T,ERRY,SP11,-SP11,ERRYD,SP22,-SP22,ERRYDD/32.2
     1                                 ,SP33/32.2,-SP33/32.2
              GOTO 10
 200          CONTINUE
              IF(T<80.)THEN
                        WGT=-125.*T+14000.
                        TRST=30000.
              ELSE
                        WGT=4000.
                        TRST=0.
              ENDIF
              AT=G*TRST/WGT
              YTDD=AT-G
              IF(STEP-1)66,66,55
 999          CONTINUE
              PAUSE
              CLOSE(1)
              CLOSE(2)
              END
```

Booster differential equation

Three-State Kalman Filter is Able To Track Boosting Target

$$\Phi_s = \frac{193.2^2}{120}$$

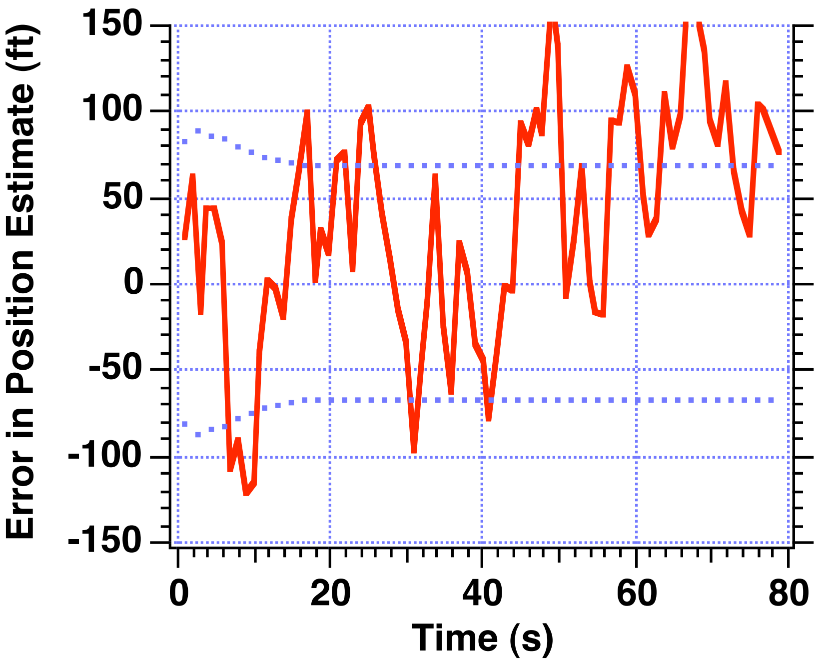


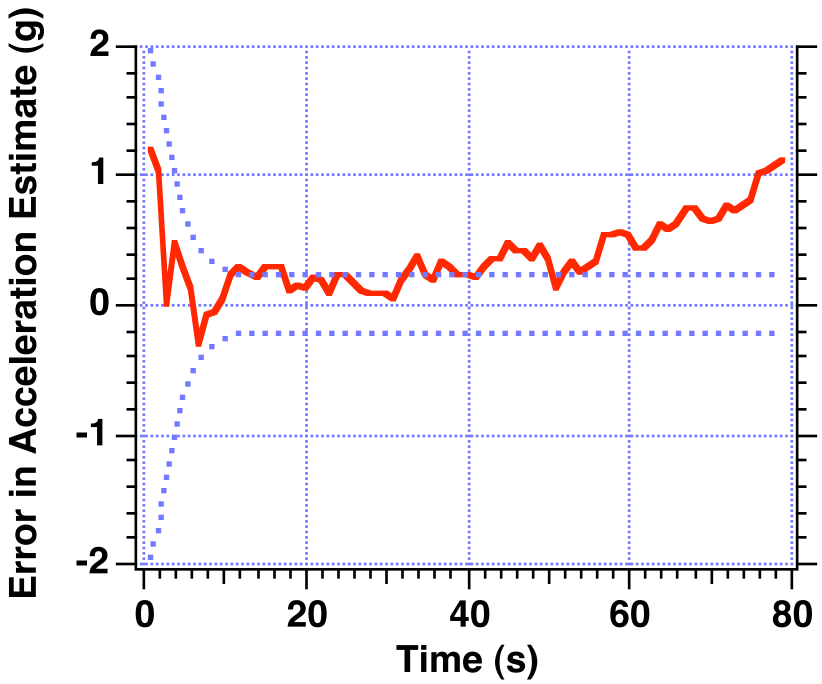


*100 ft measurement noise and 1 s sampling time

Reducing Process Noise With Three-State Kalman Filter Causes Divergence

$$\Phi_s = \frac{32.2^2}{120}$$





*100 ft measurement noise and 1 s sampling time

Six-State Linear Polynomial Kalman Filter

Measurement and Real World Plant Models

Model of Real World (6 states required to fit booster acceleration profile)

$$\begin{bmatrix} \dot{y}_T \\ \ddot{y}_T \\ \dot{n}_T \\ \ddot{n}_T \\ \dddot{n}_T \\ \dot{q}_T \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_T \\ \dot{y}_T \\ n_T \\ \dot{n}_T \\ \ddot{n}_T \\ \dddot{n}_T \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ u_s \end{bmatrix} \longrightarrow \mathbf{F} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Phi_s \end{bmatrix}$$

Measurement Equation

$$y_T^* = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_T \\ \dot{y}_T \\ n_T \\ \dot{n}_T \\ \ddot{n}_T \\ \dddot{n}_T \end{bmatrix} + u_n \longrightarrow \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad R = \Phi_s$$

Fundamental and Process Noise Matrices

Fundamental Matrix

$$\Phi_k = \begin{bmatrix} 1 & T_s & 0.5T_s^2 & T_s^3/6 & T_s^4/24 & T_s^5/120 \\ 0 & 1 & T_s & 0.5T_s^2 & T_s^3/6 & T_s^4/24 \\ 0 & 0 & 1 & T_s & 0.5T_s^2 & T_s^3/6 \\ 0 & 0 & 0 & 1 & T_s & 0.5T_s^2 \\ 0 & 0 & 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Discrete Process Noise Matrix

$$\mathbf{Q}_k = \int_0^{T_s} \Phi(\tau)\mathbf{Q}\Phi^T(\tau)d\tau = \Phi_s \begin{bmatrix} \frac{T_s^{11}}{158400} & \frac{T_s^{10}}{28800} & \frac{T_s^9}{6480} & \frac{T_s^8}{1920} & \frac{T_s^7}{840} & \frac{T_s^6}{720} \\ \frac{T_s^{10}}{28800} & \frac{T_s^9}{5184} & \frac{T_s^8}{1152} & \frac{T_s^7}{336} & \frac{T_s^6}{144} & \frac{T_s^5}{120} \\ \frac{T_s^9}{6480} & \frac{T_s^8}{1152} & \frac{T_s^7}{252} & \frac{T_s^6}{72} & \frac{T_s^5}{30} & \frac{T_s^4}{24} \\ \frac{T_s^8}{1920} & \frac{T_s^7}{336} & \frac{T_s^6}{72} & \frac{T_s^5}{20} & \frac{T_s^4}{8} & \frac{T_s^3}{6} \\ \frac{T_s^7}{840} & \frac{T_s^6}{144} & \frac{T_s^5}{30} & \frac{T_s^4}{8} & \frac{T_s^3}{3} & \frac{T_s^2}{2} \\ \frac{T_s^6}{720} & \frac{T_s^5}{120} & \frac{T_s^4}{24} & \frac{T_s^3}{6} & \frac{T_s^2}{2} & T_s \end{bmatrix}$$

Six-State Linear Polynomial Kalman Filter-1

```
GLOBAL DEFINE
        INCLUDE 'quickdraw.inc'
END
IMPLICIT REAL*8(A-H,O-Z)
REAL*8 P(6,6),Q(6,6),M(6,6),PHI(6,6),HMAT(1,6),HT(6,1),PHIT(6,6)
REAL*8 RMAT(1,1),IDN(6,6),PHIP(6,6),PHIPPHIT(6,6),HM(1,6)
REAL*8 HMHT(1,1),HMHTR(1,1),HMHTRINV(1,1),MHT(6,1),K(6,1)
REAL*8 KH(6,6),IKH(6,6)
INTEGER ORDER,STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
OPEN(2,STATUS='UNKNOWN',FILE='COVFIL')
G=32.2
XNTDDD=.03
YT=0.
YTD=0.
ORDER=6
PHIS=XNTDDD*XNTDDD/120.
TS=1.
TS2=TS*TS
TS3=TS2*TS
TS4=TS3*TS
TS5=TS4*TS
TS6=TS5*TS
TS7=TS6*TS
TS8=TS7*TS
TS9=TS8*TS
TS10=TS9*TS
TS11=TS10*TS
YH=0.
YDH=0.
YDDH=0
YDDDH=0
YDDDDH=0.
YDDDDDH=0.
SIGNOISE=100.
```

Process noise and sampling time

Initial state estimates and measurement noise

Six-State Linear Polynomial Kalman Filter-2

```
        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
        RMAT(1,1)=SIGNOISE**2
        IDN(1,1)=1.
        IDN(2,2)=1.
        IDN(3,3)=1.
        IDN(4,4)=1.
        IDN(5,5)=1.
        IDN(6,6)=1.
        P(1,1)=100.**2
        P(2,2)=100.**2
        P(3,3)=64.4**2
        P(4,4)=2.5**2
        P(5,5)=.25**2
        P(6,6)=.025**2
        PHI(1,1)=1.
        PHI(1,2)=TS
        PHI(1,3)=.5*TS2
        PHI(1,4)=TS3/6.
        PHI(1,5)=TS4/24.
        PHI(1,6)=TS5/120.
        PHI(2,2)=1.
        PHI(2,3)=TS
        PHI(2,4)=.5*TS2
        PHI(2,5)=TS3/6.
        PHI(2,6)=TS4/24.
        PHI(3,3)=1.
        PHI(3,4)=TS
        PHI(3,5)=TS2/2.
```

Measurement noise matrix

Initial covariance matrix

Non zero elements of fundamental matrix

Six-State Linear Polynomial Kalman Filter-3

```
PHI(3,6)=TS3/6.
PHI(4,4)=1.
PHI(4,5)=TS
PHI(4,6)=TS2/2.
PHI(5,5)=1.
PHI(5,6)=TS
PHI(6,6)=1.
HMAT(1,1)=1.
HMAT(1,2)=0.
HMAT(1,3)=0.
HMAT(1,4)=0.
HMAT(1,5)=0.
HMAT(1,6)=0.
```

Measurement matrix

```
CALL MATTRN(PHI,ORDER,ORDER,PHIT)
CALL MATTRN(HMAT,1,ORDER,HT)
Q(1,1)=PHIS*TS11/158400.
Q(1,2)=PHIS*TS10/28800.
Q(1,3)=PHIS*TS9/6480.
Q(1,4)=PHIS*TS8/1920.
Q(1,5)=PHIS*TS7/840.
Q(1,6)=PHIS*TS6/720.
Q(2,1)=Q(1,2)
Q(2,2)=PHIS*TS9/5184.
Q(2,3)=PHIS*TS8/1152.
Q(2,4)=PHIS*TS7/336.
Q(2,5)=PHIS*TS6/144.
Q(2,6)=PHIS*TS5/120.
Q(3,1)=Q(1,3)
Q(3,2)=Q(2,3)
Q(3,3)=PHIS*TS7/252.
Q(3,4)=PHIS*TS6/72.
Q(3,5)=PHIS*TS5/30.
Q(3,6)=PHIS*TS4/24.
Q(4,1)=Q(1,4)
Q(4,2)=Q(2,4)
Q(4,3)=Q(3,4)
```

Process noise matrix

Six-State Linear Polynomial Kalman Filter-4

```
        Q(4,4)=PHIS*TS5/20.
        Q(4,5)=PHIS*TS4/8.
        Q(4,6)=PHIS*TS3/6.
        Q(5,1)=Q(1,5)
        Q(5,2)=Q(2,5)
        Q(5,3)=Q(3,5)
        Q(5,4)=Q(4,5)
        Q(5,5)=PHIS*TS3/3.
        Q(5,6)=PHIS*TS2/2.
        Q(6,1)=Q(1,6)
        Q(6,2)=Q(2,6)
        Q(6,3)=Q(3,6)
        Q(6,4)=Q(4,6)
        Q(6,5)=Q(5,6)
        Q(6,6)=PHIS*TS
        S=0.
        H=.01
        T=0.
10      IF(T>79.)GOTO 999
        S=S+H
        YTOLD=YT
        YTDOLD=YTD
        STEP=1
        GOTO 200
66      STEP=2
        YT=YT+H*YTD
        YTD=YTD+H*YTDD
        T=T+H
        GOTO 200
55      CONTINUE
        YT=.5*(YTOLD+YT+H*YTD)
        YTD=.5*(YTDOLD+YTD+H*YTDD)
```

Second-order Runge-Kutta Integration for booster

Six-State Linear Polynomial Kalman Filter-5

```
      IF(S<(TS-.00001))GOTO 10
      S=0.
```

```
      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,ORDER,ORDER,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)
```

Riccati equations

```
      CALL GAUSS(YNOISE,SIGNOISE)
      YS=YT+YNOISE
```

Measurement noise

```
      XK1=K(1,1)
      XK2=K(2,1)
      XK3=K(3,1)
      XK4=K(4,1)
      XK5=K(5,1)
      XK6=K(6,1)
```

```
      RES=YS-YH-TS*YDH-.5*TS*TS*YDDH-TS3*YDDDH/6.-TS4*YDDDDH/
1                         24.-TS5*YDDDDDH/120.
      YH=YH+YDH*TS+.5*TS*TS*YDDH+TS3*YDDDH/6.+TS4*YDDDDH/24.
1                         +TS5*YDDDDDH/120.+XK1*RES
      YDH=YDH+YDDH*TS+.5*TS2*YDDDH+TS3*YDDDDH/6.+
1                         TS4*YDDDDDH/24.+XK2*RES
      YDDH=YDDH+YDDDH*TS+.5*TS2*YDDDDH+TS3*YDDDDDH/6.+XK3*RES
      YDDDH=YDDDH+YDDDDH*TS+.5*TS2*YDDDDDH+XK4*RES
      YDDDDH=YDDDDH+TS*YDDDDDH+XK5*RES
      YDDDDDH=YDDDDDH+XK6*RES
```

Six-state filter

```
      ERRY=YT-YH
      ERRYD=YTD-YDH
      ERRYDD=YTDD-YDDH
      SP11=SQRT(P(1,1))
```

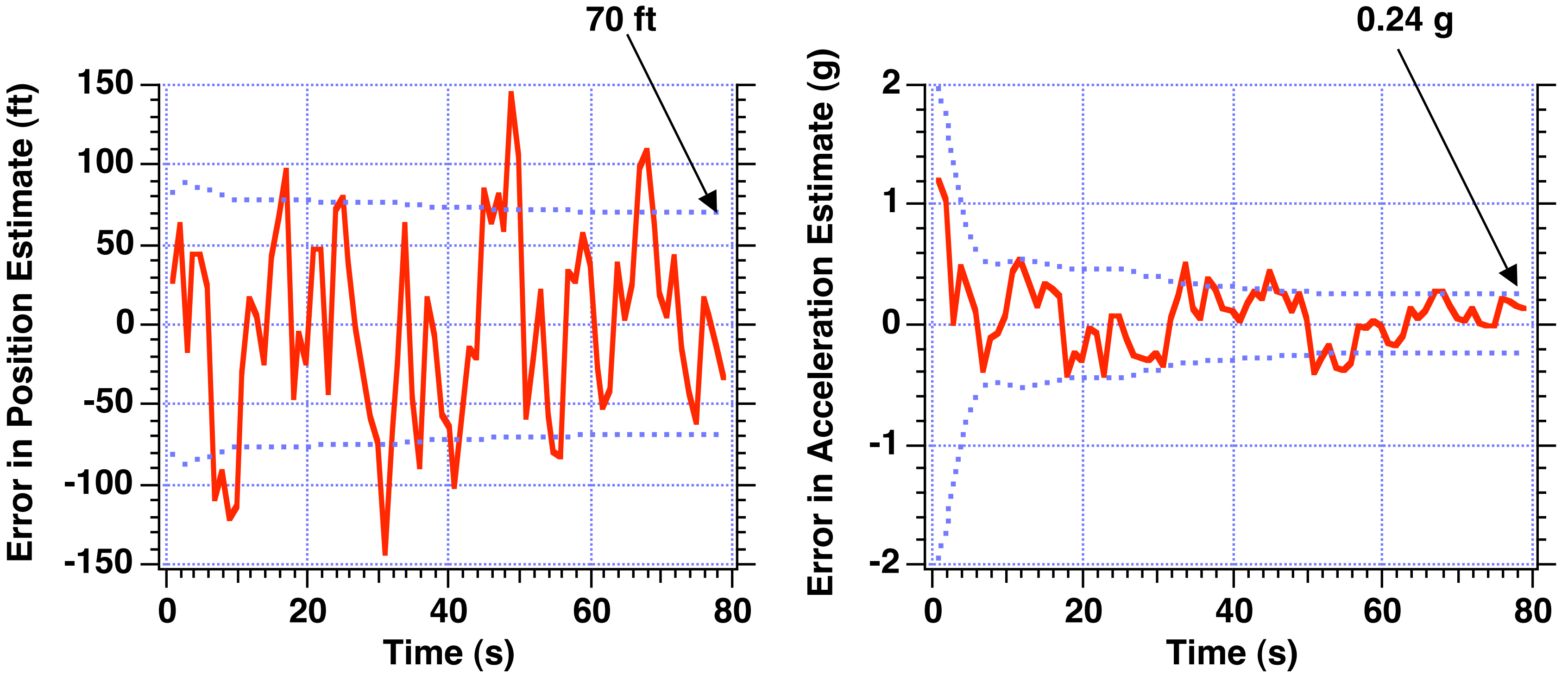
Six-State Linear Polynomial Kalman Filter-6

```
        SP22=SQRT(P(2,2))
        SP33=SQRT(P(3,3))
        WRITE(9,*)T,YTDD/32.2,YDDH/32.2
        WRITE(1,*)T,YTDD/32.2,YDDH/32.2,YTD,YDH,ZDH
        WRITE(2,*)T,ERRY,SP11,-SP11,ERRYD,SP22,-SP22,ERRYDD/32.2
  1                              ,SP33/32.2,-SP33/32.2
        GOTO 10
200     CONTINUE
        IF(T<80.)THEN
                WGT=-125.*T+14000.
                TRST=30000.
        ELSE
                WGT=4000.
                TRST=0.
        ENDIF
        AT=G*TRST/WGT
        YTDD=AT-G
        IF(STEP-1)66,66,55
999     CONTINUE
        PAUSE
        CLOSE(1)
        CLOSE(2)
        END
```

Booster differential equation

Six-State Kalman Filter is Able To Track Boosting Target Better

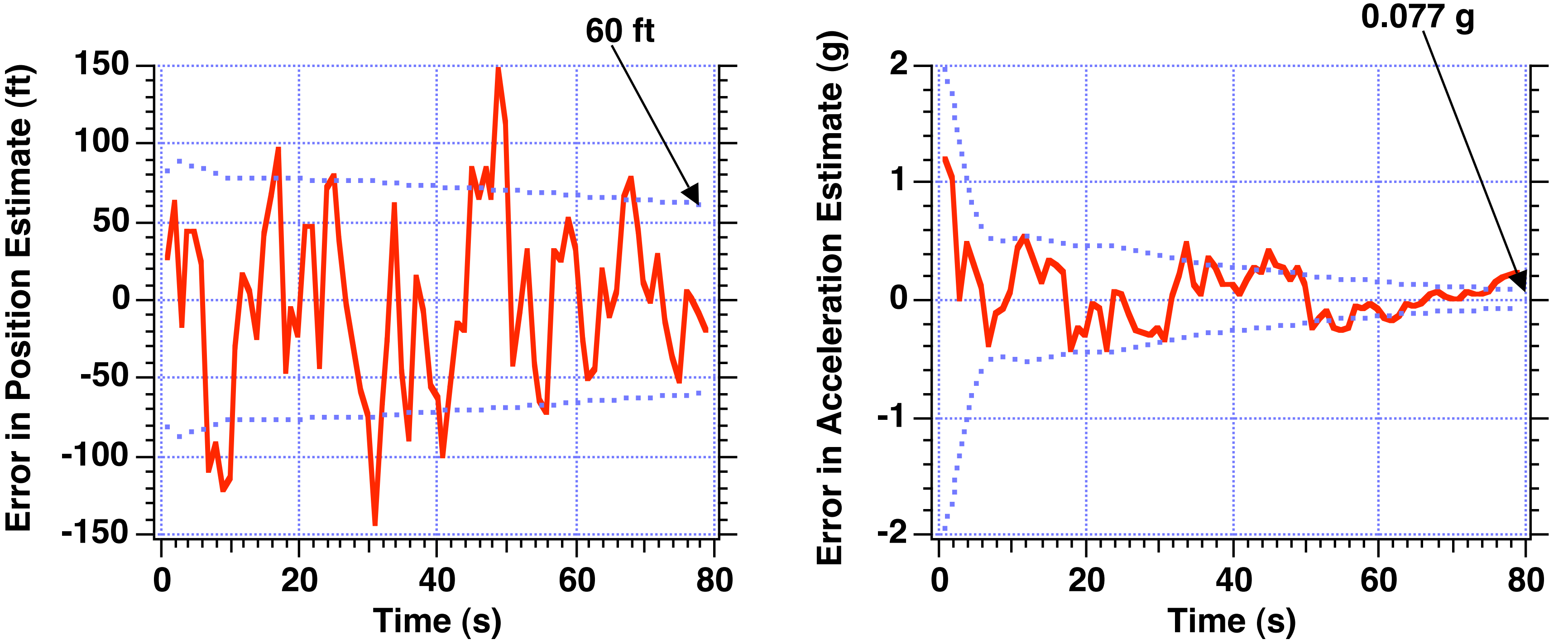
$$\Phi_s = \frac{0.03^2}{120}$$



*100 ft measurement noise and 1 s sampling time

Reducing Process Noise To Zero With Six-State Kalman Filter Does Not Cause Divergence

$$\Phi_s = \frac{0^2}{120}$$



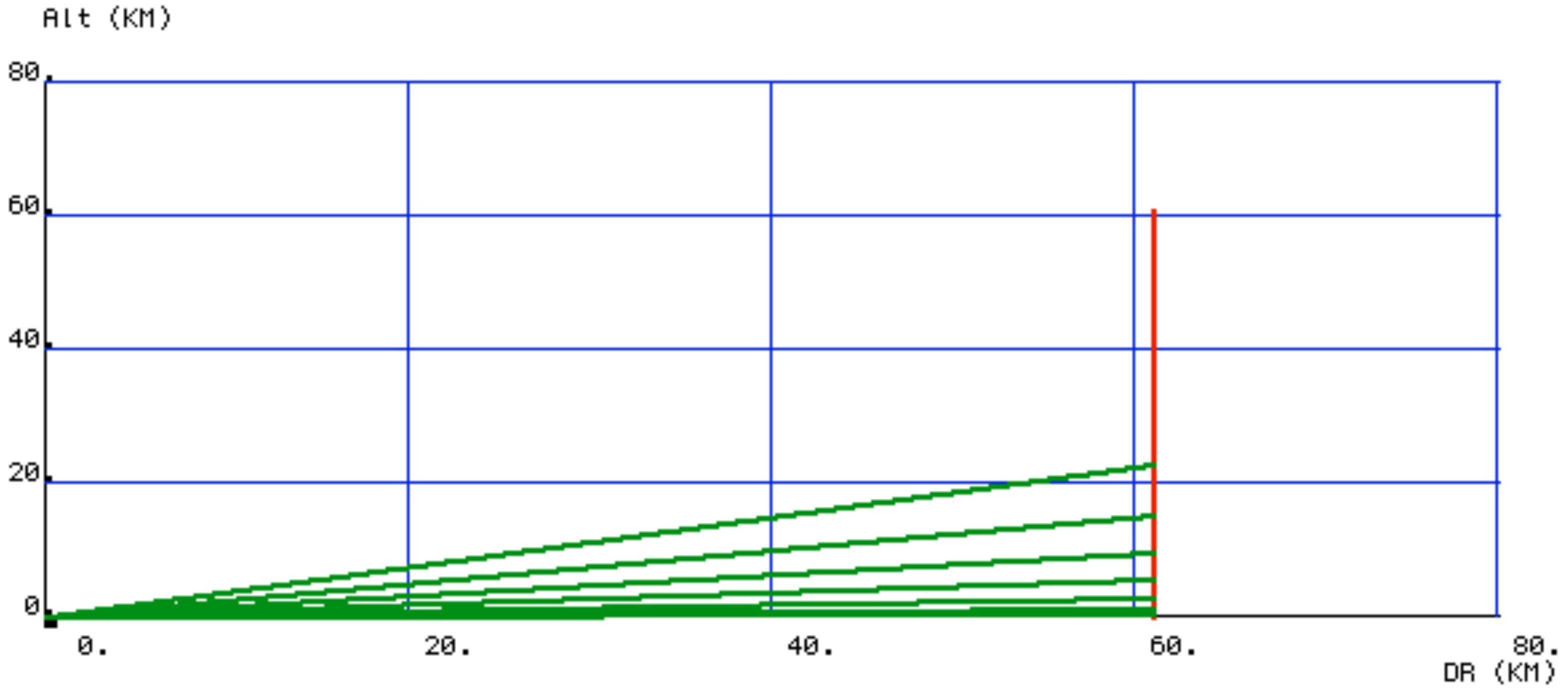
*100 ft measurement noise and 1 s sampling time

Tracking Accuracy Summary

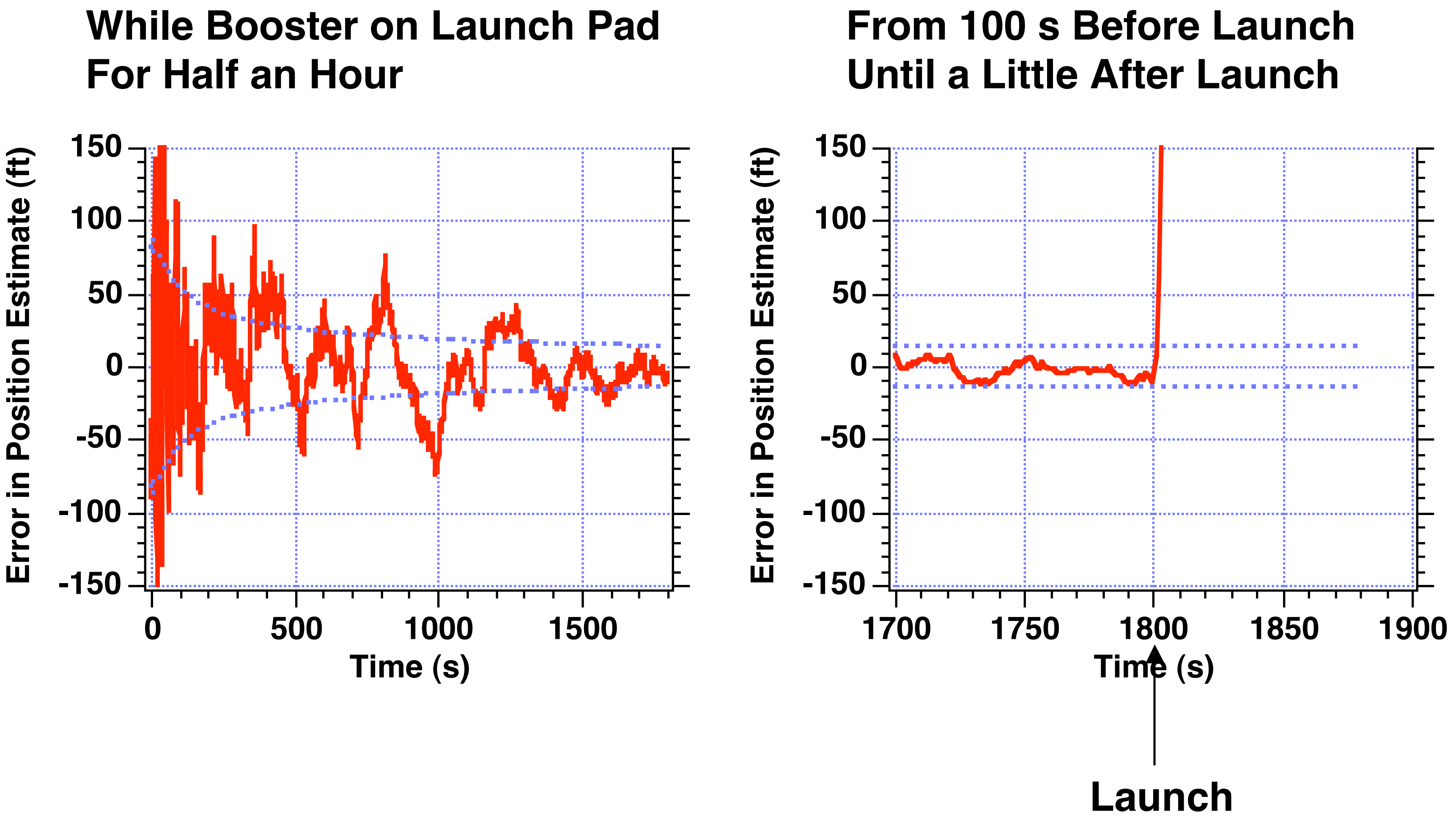
	Position Error	Acceleration Error
Nominal 3SKF	82 ft	0.97 g
Nominal 6SKF	70 ft	0.24 g
6SKF Zero Process Noise	60 ft	0.077 g

Interesting Experiments

If Booster Takes Off Half Hour Late Radar Loses Track With Six-State Filter and Zero Process Noise

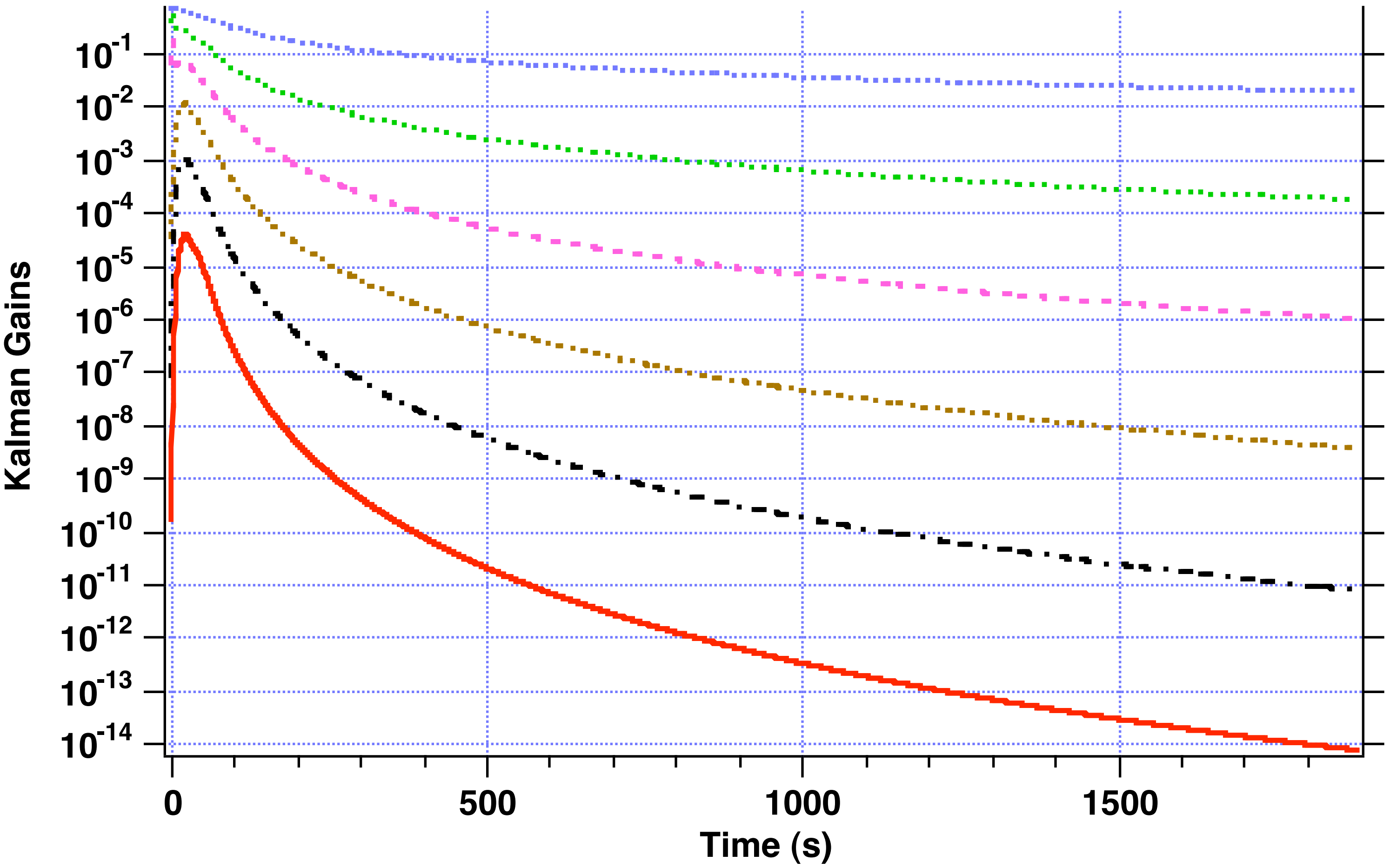


With Zero Process Noise 6-State Filter Diverges After Booster Launch



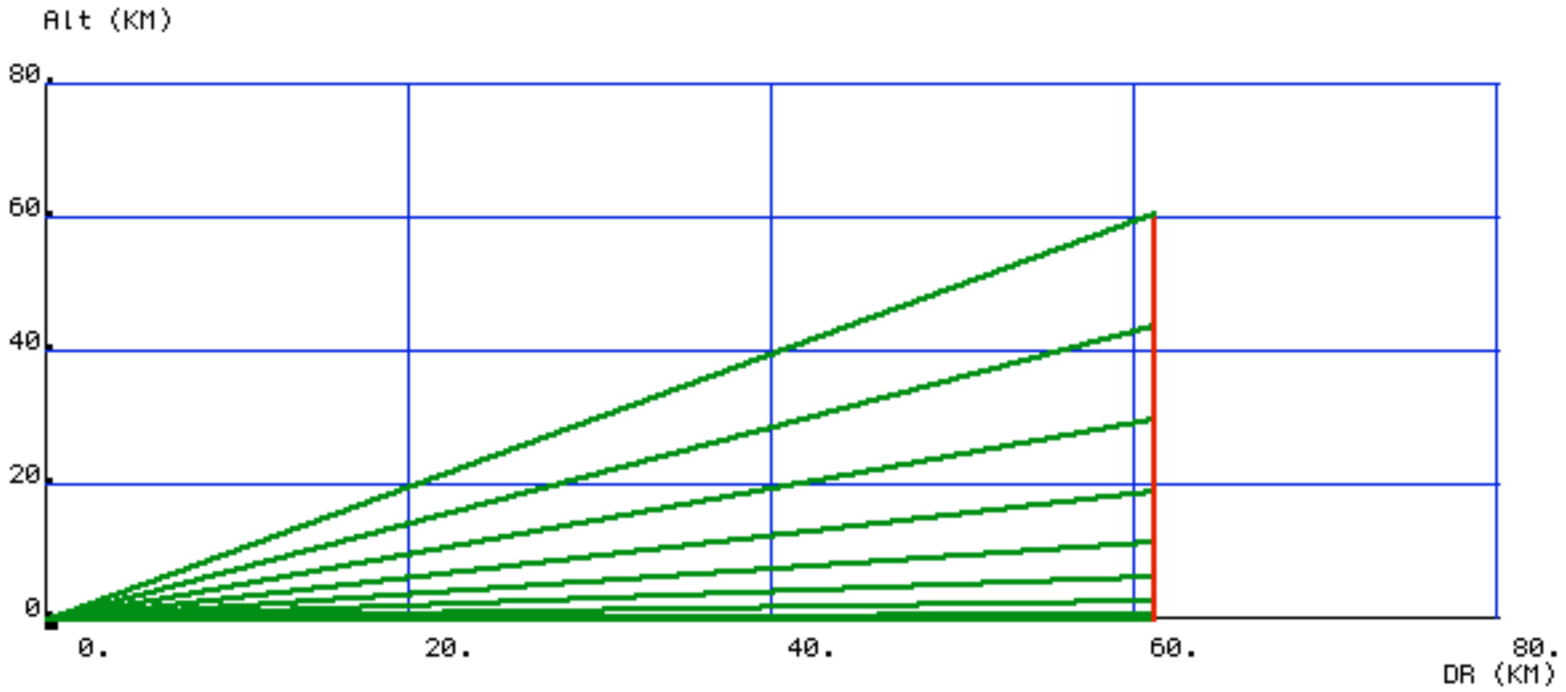
*100 ft measurement noise and 1 s sampling time

All Kalman Gains Go To Zero When There is No Process Noise

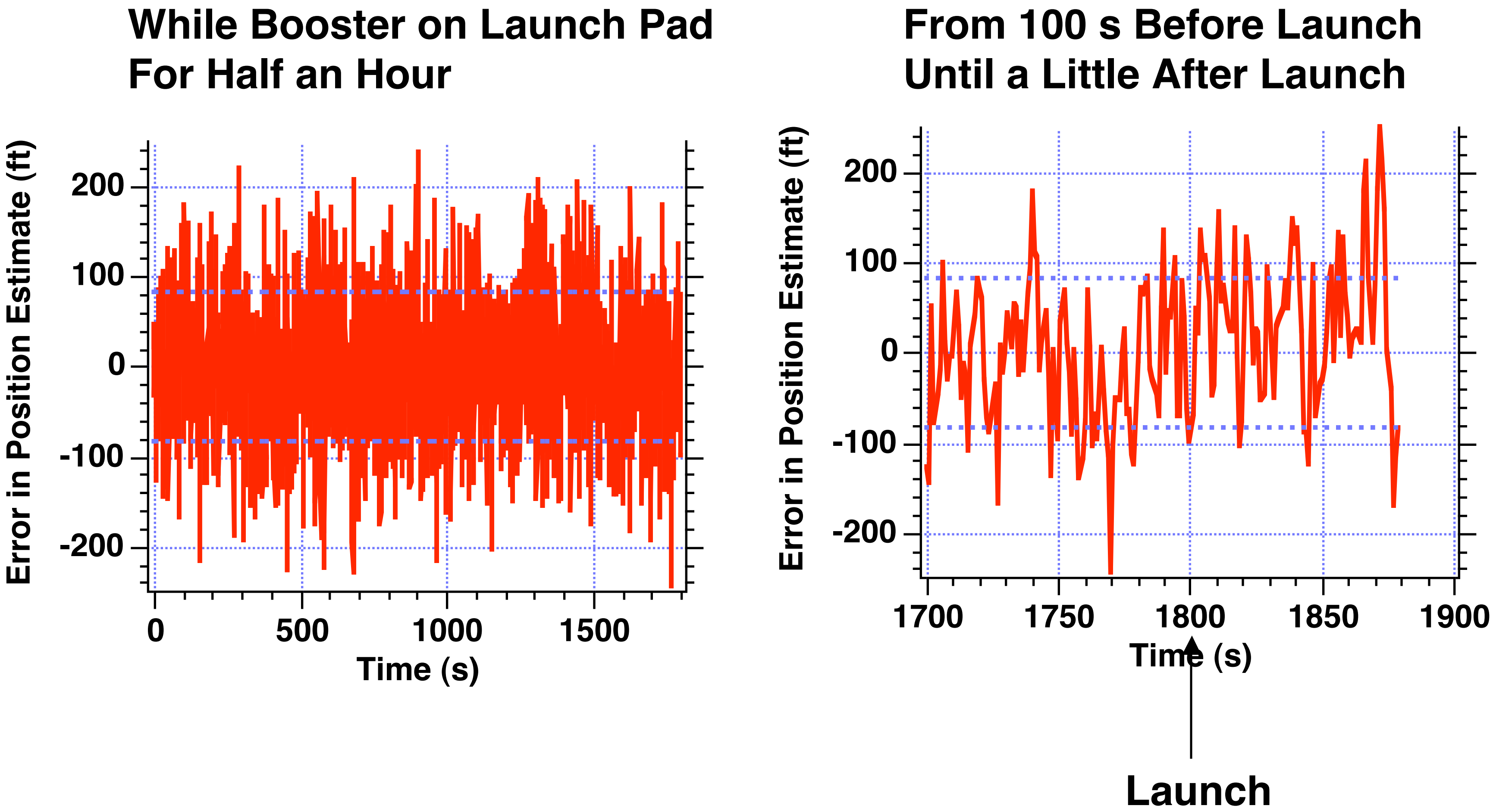


When Kalman gains are small, filter stops paying attention to measurements

If Booster Takes Off Half Hour Late Radar Does Not Lose Track With Three-State Filter With Process Noise



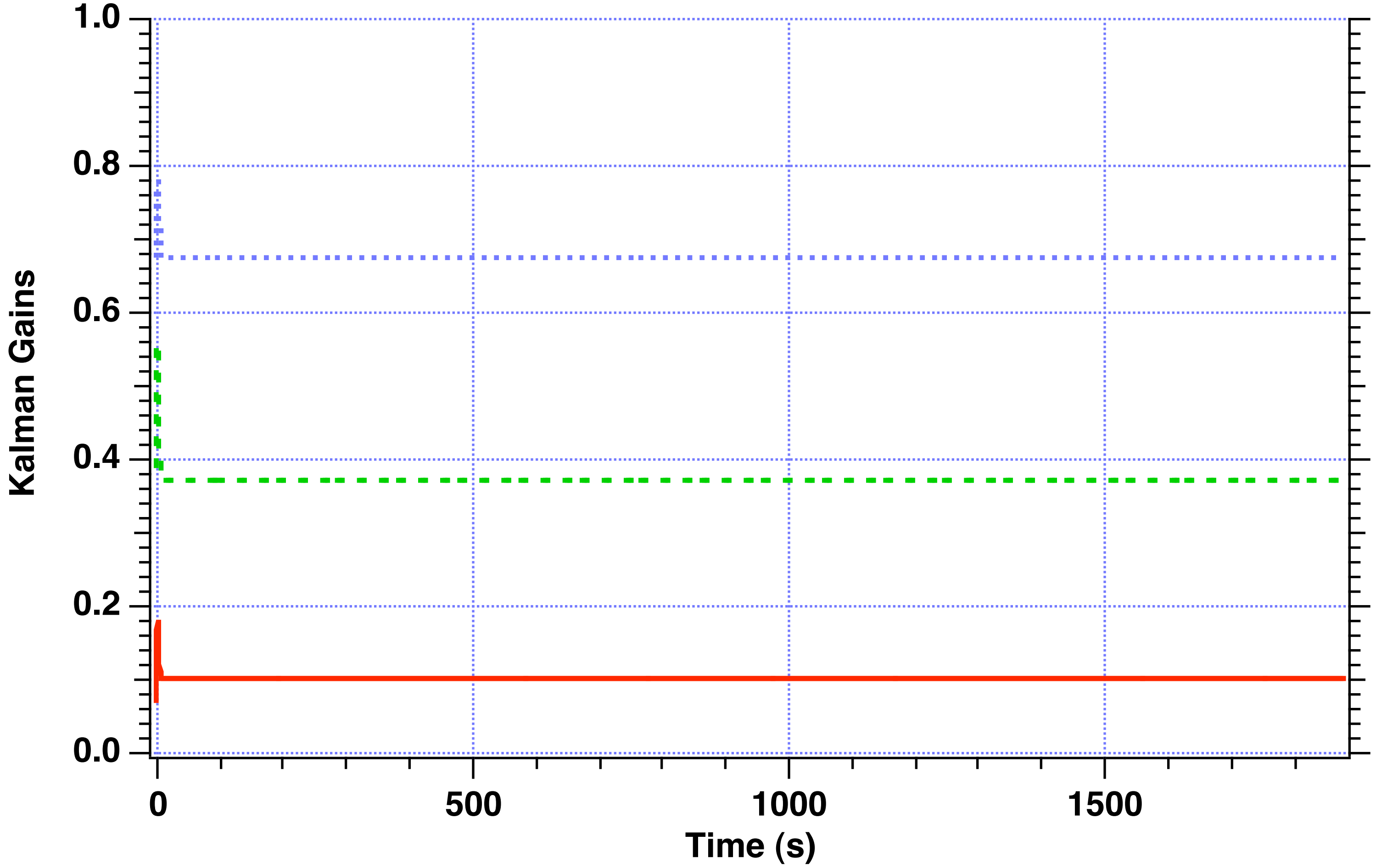
Three-State Filter With Process Noise is Able To Track Target After Half Hour Launch Delay


While Booster on Launch Pad For Half an Hour

From 100 s Before Launch Until a Little After Launch

*100 ft measurement noise and 1 s sampling time

Kalman Gains Do Not Go To Zero For Three-State Filter With Process Noise



When Kalman gains do not get too small filter always pays attention to measurements

A Tracking Disaster
Summary

Making process noise too small or zero be very dangerous, even when computer results are very encouraging