Miscellaneous Topics

Miscellaneous Topics Overview

- Sinusoidal Kalman filter and signal to noise ratio
- When only a few measurements are available
- Detecting filter divergence in the real world
- Observability example
- Aiding

Sinusoidal Kalman Filter and Signal to Noise Ratio

Review

Measurement

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

Fundamental matrix for sinusoidal Kalman filter

$$\mathbf{\Phi_k} = \begin{bmatrix} \cos\omega T_s & \frac{\sin\omega T_s}{\omega} \\ -\omega\sin\omega T_s & \cos\omega T_s \end{bmatrix}$$

Measurement matrix

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

Kalman filtering equations

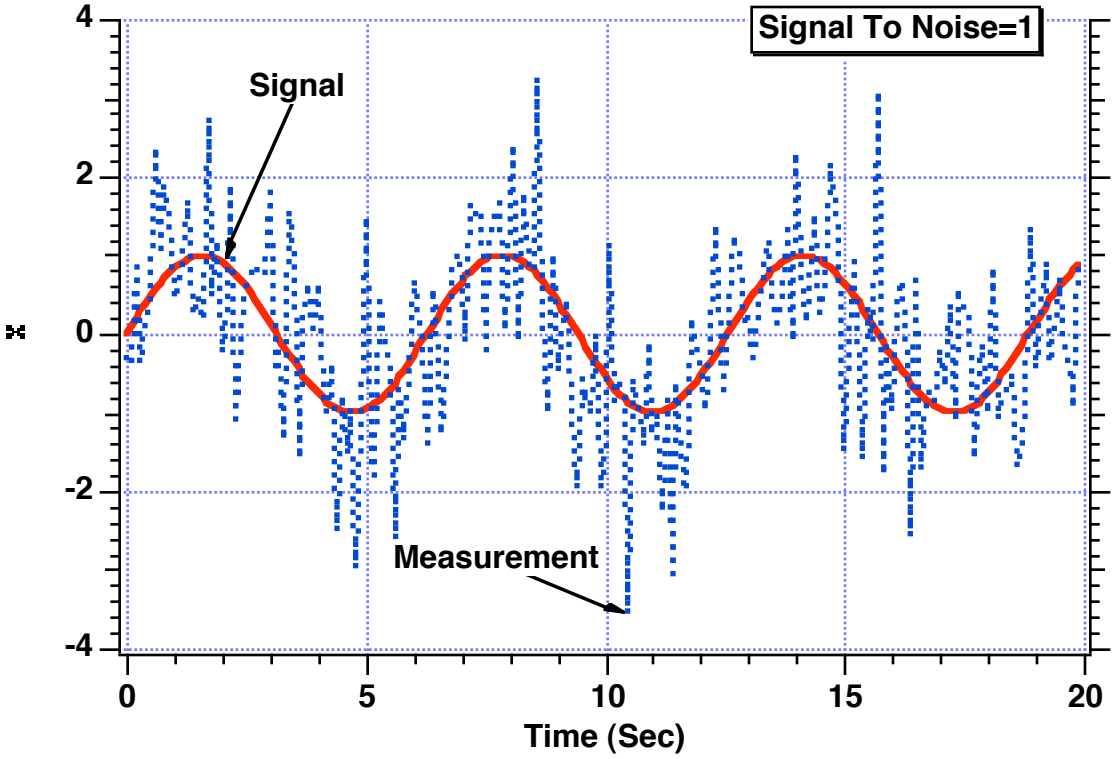
$$RES_k = x_k^* - \cos\omega T_s \hat{x}_{k-1} - \frac{\sin\omega T_s}{\omega} \hat{\dot{x}}_{k-1}$$

$$\hat{x}_k = \cos\omega T_s \hat{x}_{k-1} + \frac{\sin\omega T_s}{\omega} \hat{\dot{x}}_{k-1} + K_{1_k} RES_k$$

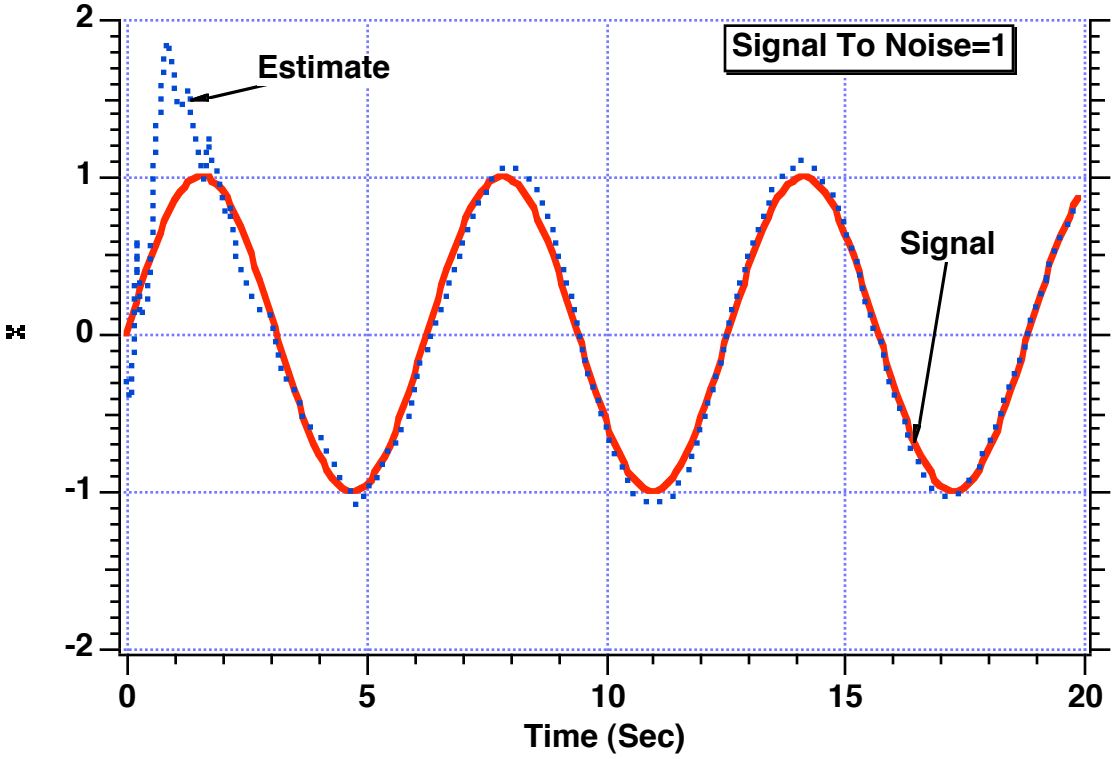
$$\hat{\dot{x}}_k = -\omega\sin\omega T_s \hat{x}_{k-1} + \cos\omega T_s \hat{\dot{x}}_{k-1} + K_{2_k} RES_k$$

Signal to noise ratio is amplitude of signal divided by standard deviation of the measurement noise

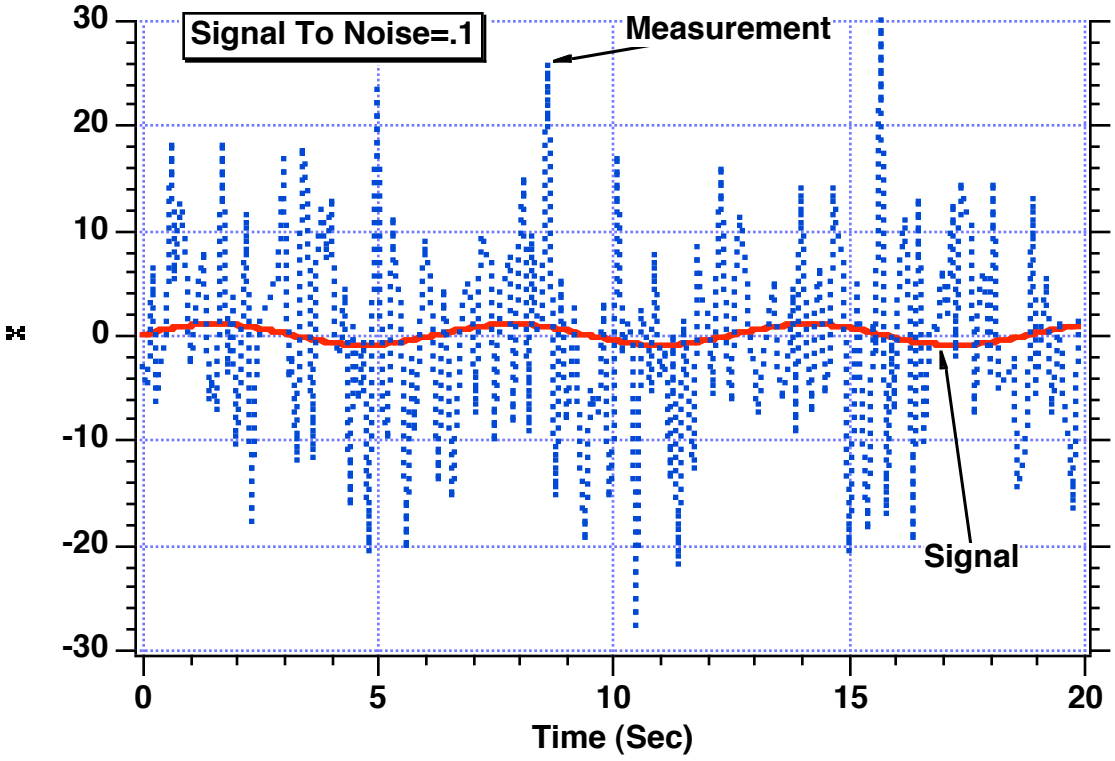
Measurement Appears to be a Noisy Sinusoid When the Signal to Noise Ratio is Unity



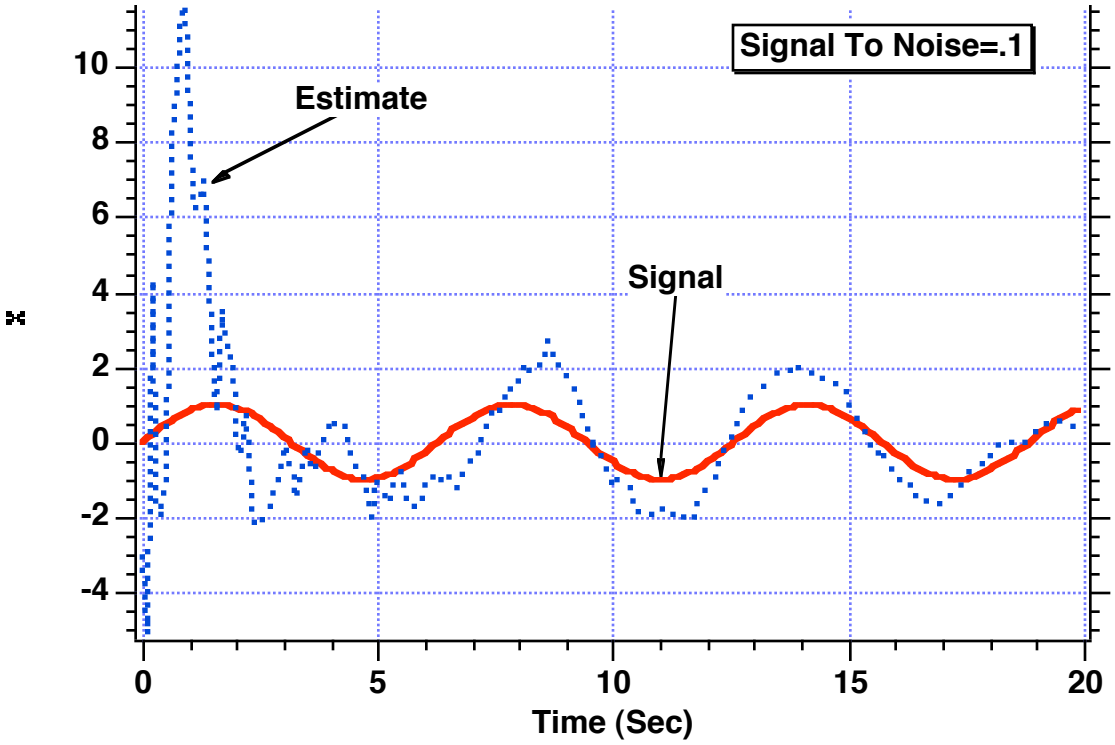
Sinusoidal Kalman Filter Yields Excellent Estimates of the Signal When the Signal to Noise Ratio is Unity



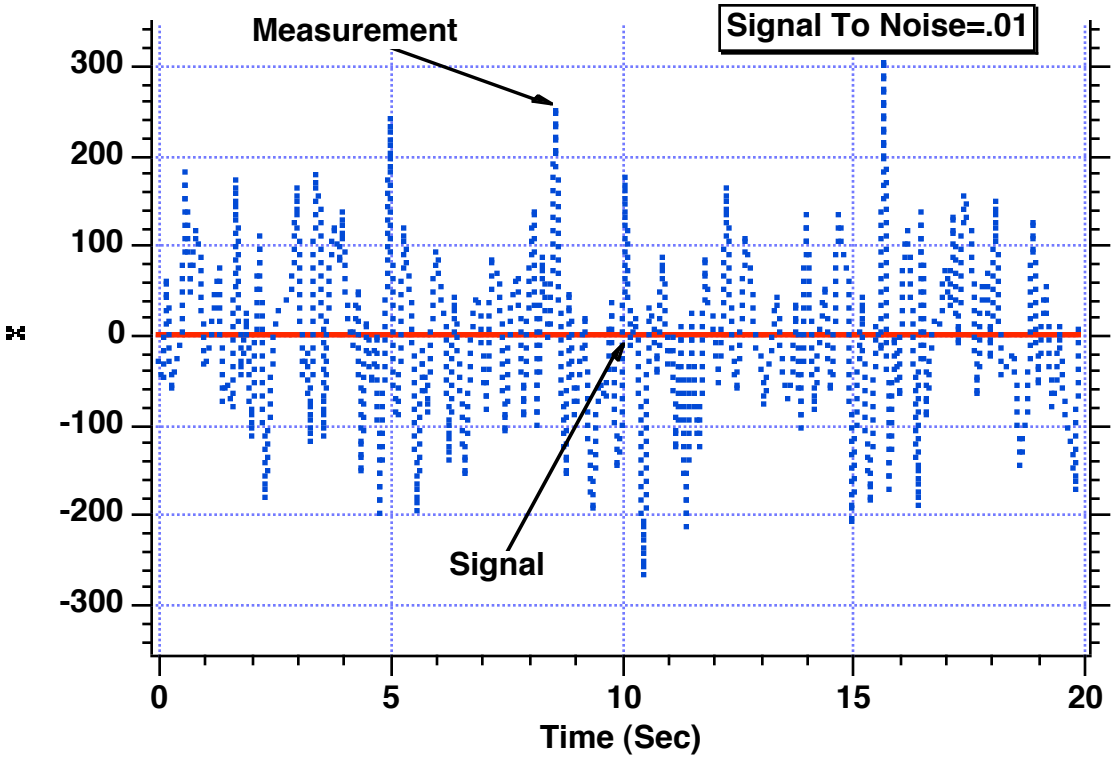
Measurement Appears to be Correlated Noise When Signal to Noise Ratio is .1



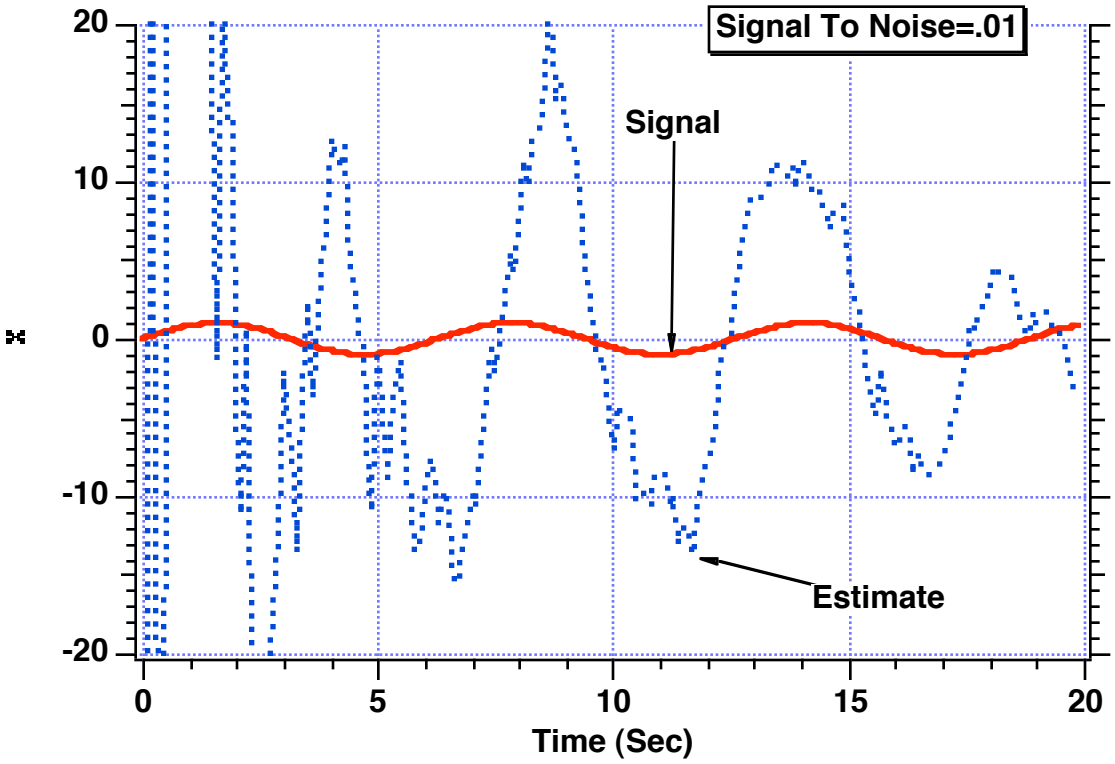
Filter Estimate Deteriorates and Lags Signal When Signal to Noise Ratio is .1



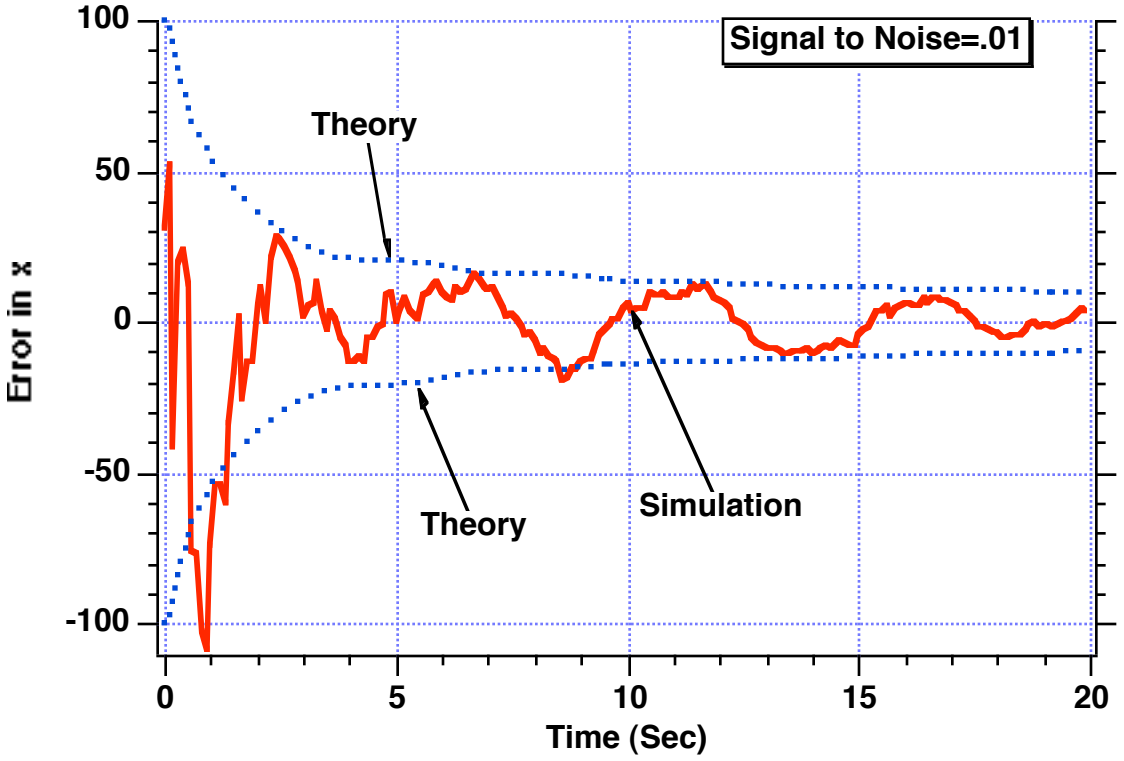
Measurement Appears to be Correlated Noise When Signal to Noise Ratio is .01



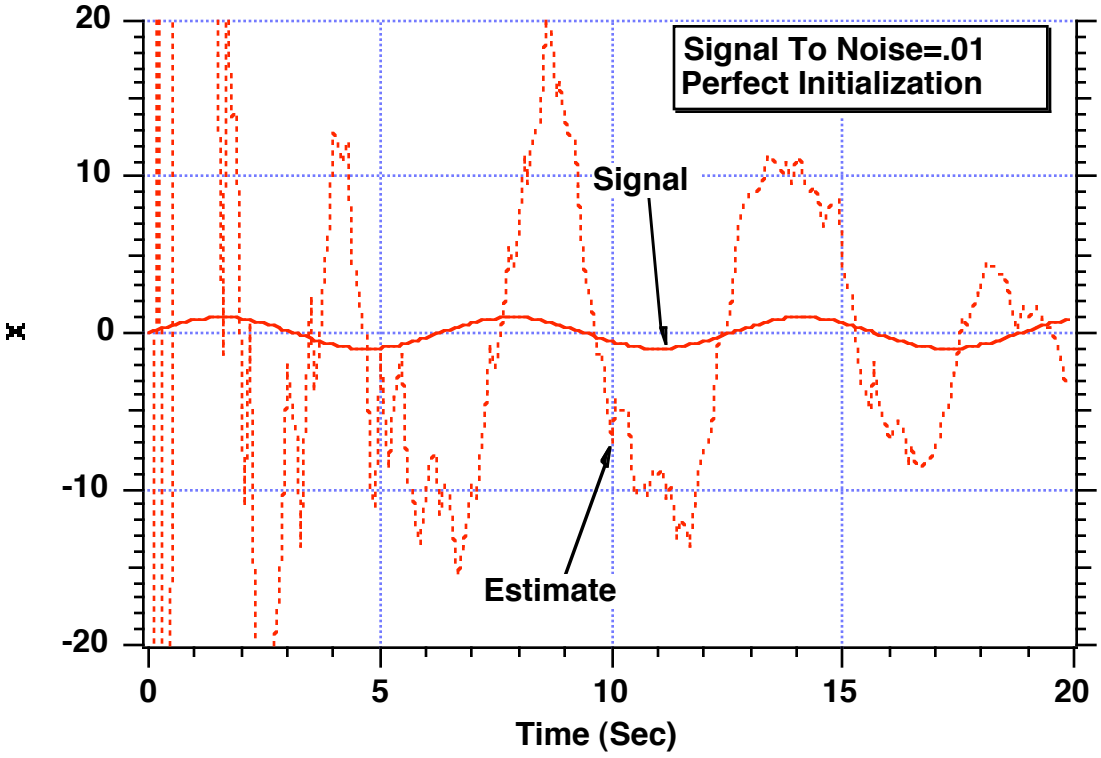
Sinusoidal Kalman Filter Estimate is Virtually Worthless When Signal to Noise Ratio is .01



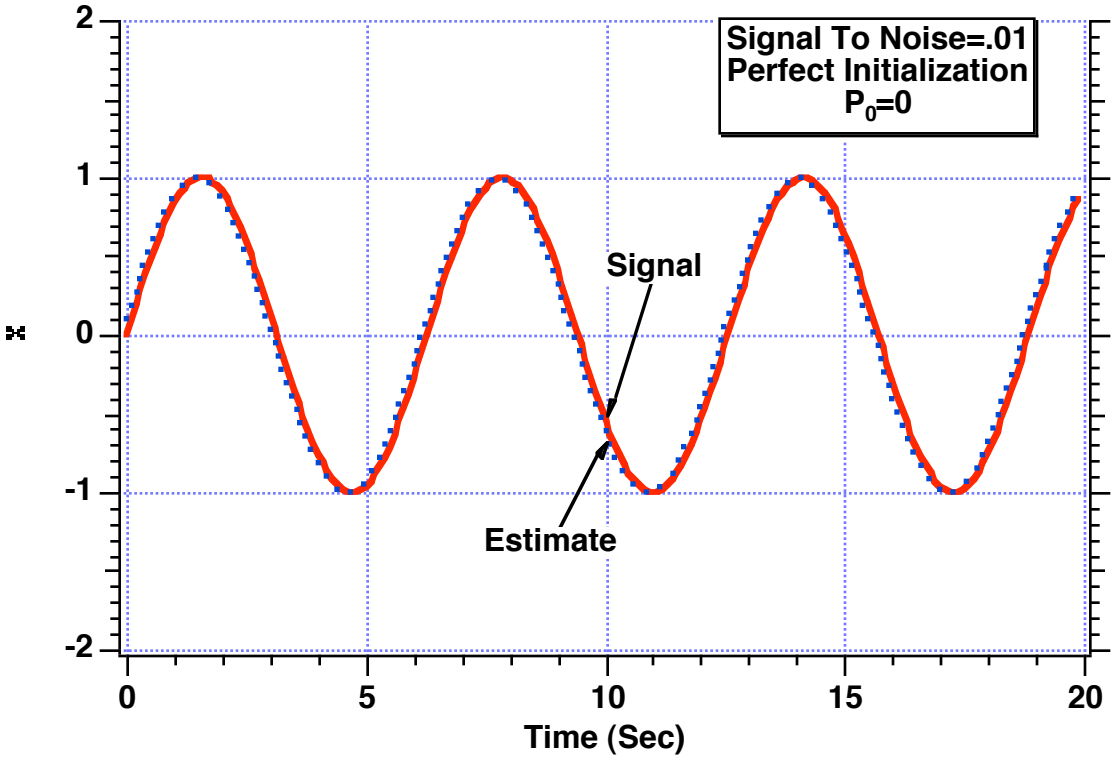
Error in the Estimate Results Indicate That Filter is Working Correctly When Signal to Noise Ratio is .01



Filter Estimate is Still Virtually Worthless if States are Perfectly Initialized When Signal to Noise Ratio is .01



Filter Estimate is Near Perfect if States are Perfectly Initialized and Initial Covariance Matrix is Zero



Summary

The initialization of the filter states and initial covariance matrix can be very important in reducing estimation errors and improving track accuracy

When Only a Few Measurements Are Available

Important Matrices for Different Order Linear Polynomial Kalman Filters

States	Fundamental Matrix	Measurement Matrix	Measurement Noise Matrix
3	$\mathbf{\Phi_k} = \begin{bmatrix} 1 & T_s & .5T_s^2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}$	$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$	$\mathbf{R_k} = \sigma_n^2$
4	$\mathbf{\Phi_k} = \begin{bmatrix} 1 & T_s & .5T_s^2 & \frac{T_s^3}{6} \\ 0 & 1 & T_s & .5T_s^2 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$	$\mathbf{R_k} = \sigma_n^2$
5	$\mathbf{\Phi_k} = \begin{bmatrix} 1 & T_s & .5T_s^2 & \frac{T_s^3}{6} & \frac{T_s^4}{24} \\ 0 & 1 & T_s & .5T_s^2 & \frac{T_s^3}{6} \\ 0 & 0 & 1 & T_s & .5T_s^2 \\ 0 & 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\mathbf{R_k} = \sigma_n^2$

Review

Polynomial Kalman filter

$$\hat{\mathbf{x}}_k = \mathbf{\Phi}_k \hat{\mathbf{x}}_{k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\mathbf{\Phi}_k \hat{\mathbf{x}}_{k-1})$$

Zero process noise

$$\mathbf{Q}_k = 0$$

Compute gains from Riccati equations

$$\mathbf{M}_k = \mathbf{\Phi}_k \mathbf{P}_{k-1} \mathbf{\Phi}_k^{\mathbf{T}} + \mathbf{Q}_k$$

$$\mathbf{K}_k = \mathbf{M}_k \mathbf{H}^{\mathbf{T}} (\mathbf{H}\mathbf{M}_k \mathbf{H}^{\mathbf{T}} + \mathbf{R}_k)^{-1}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H})\mathbf{M}_k$$

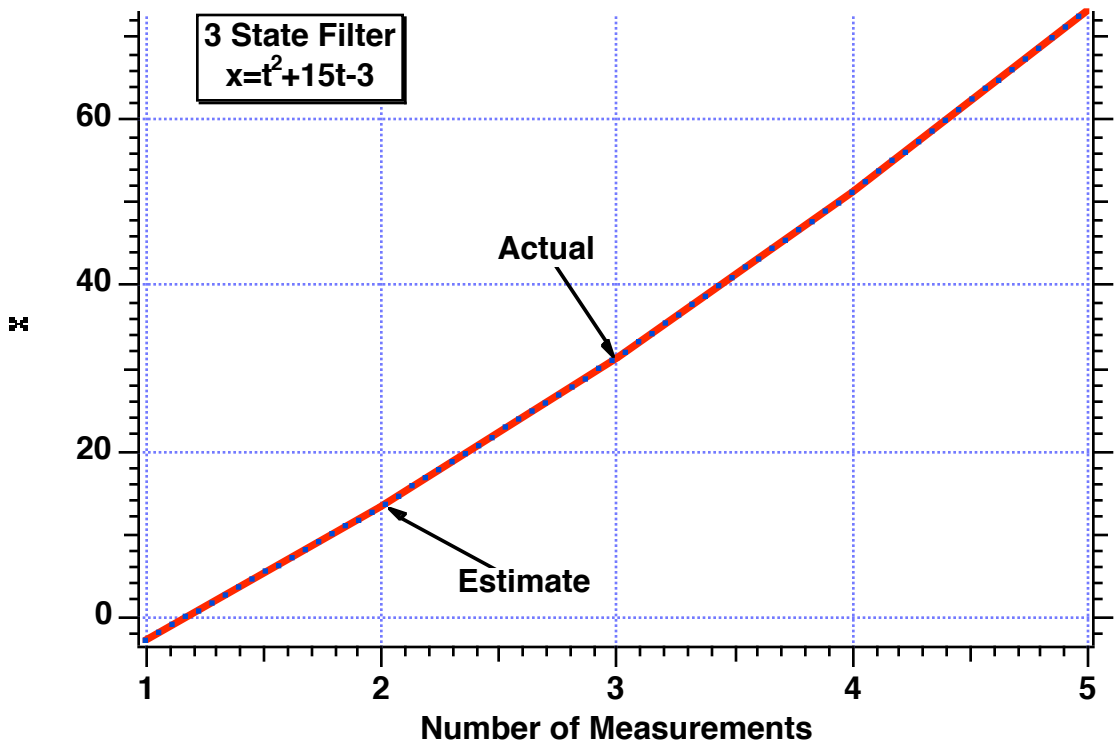
Problem

$$x = t^2 + 15t - 3 \longleftarrow$$ **Signal and measurement**

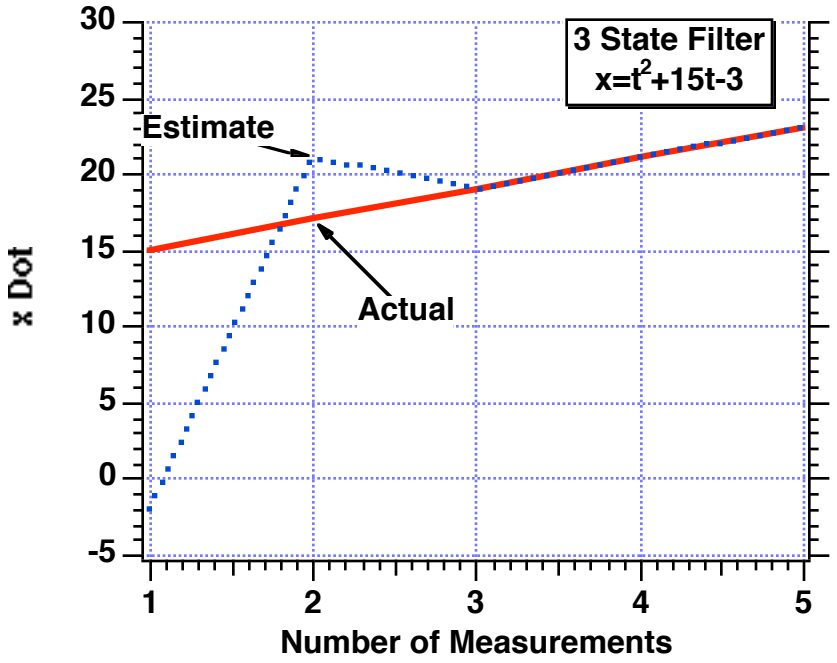
$$\dot{x} = 2t + 15$$

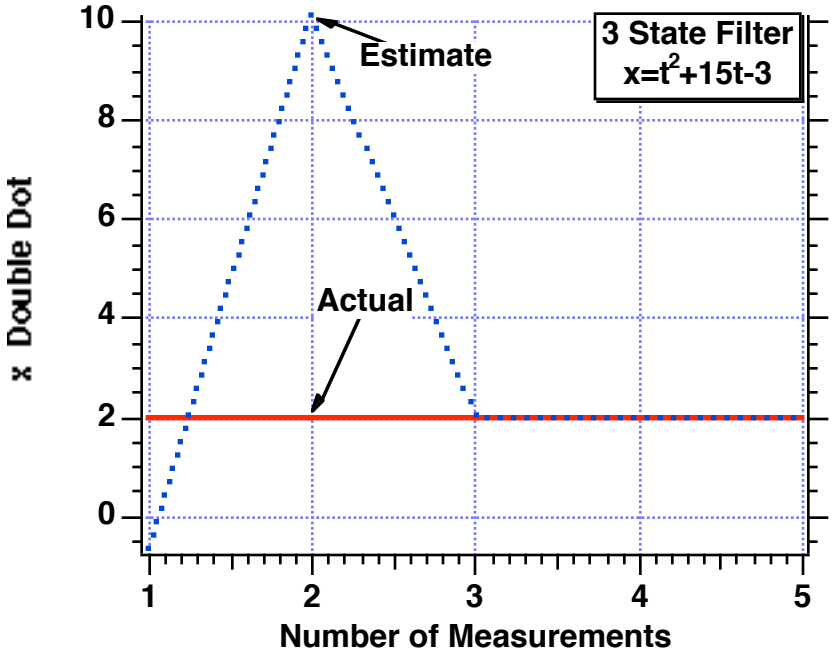
$$\ddot{x} = 2$$

Three-State Polynomial Kalman Filter Estimates Signal Perfectly After Only One Measurement

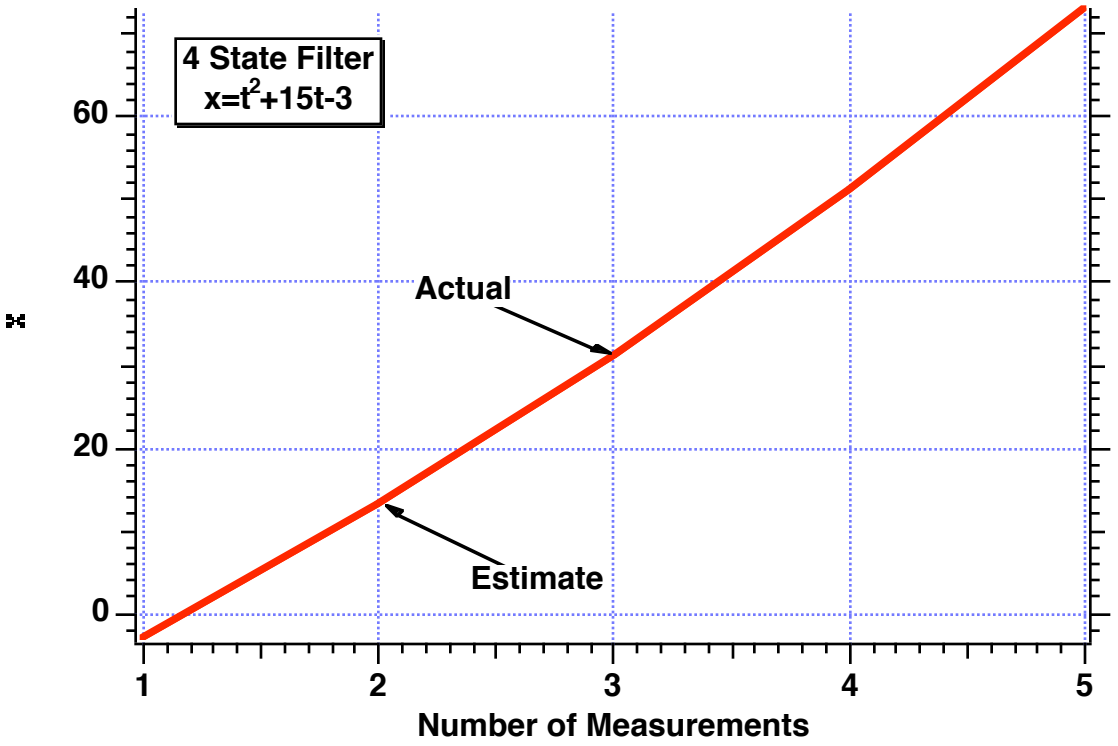


Three-State Filter Requires Three Measurements to Estimate First and Second Derivatives of the Signal

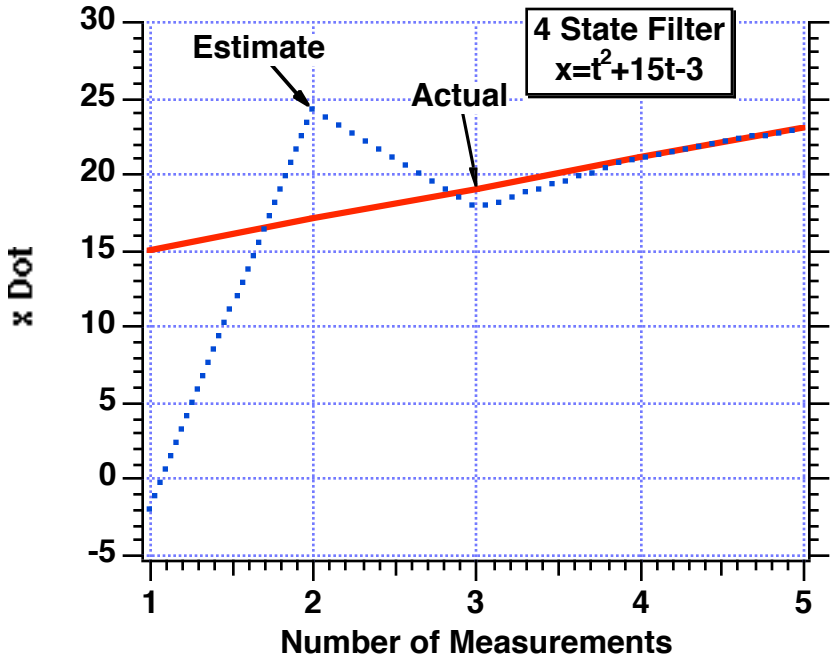


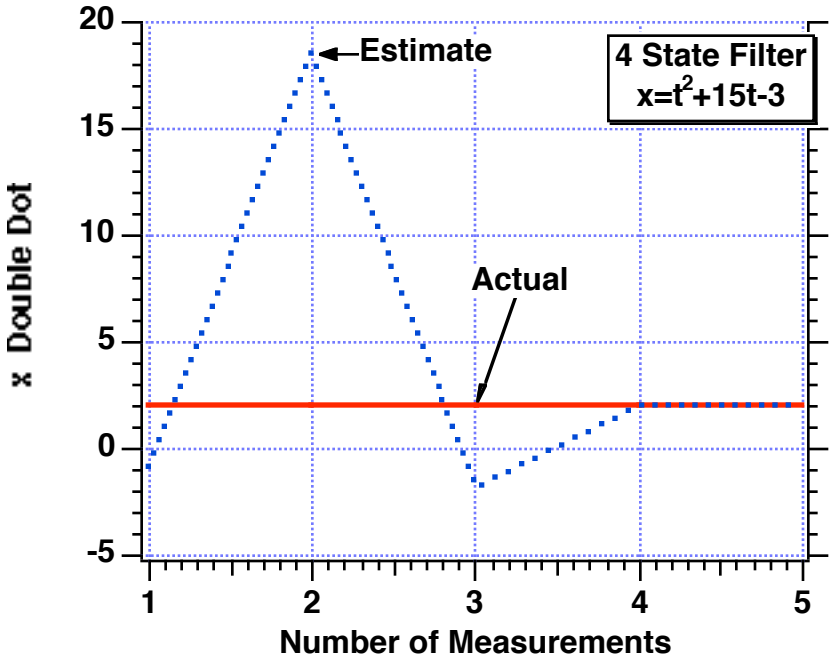


Four-State Polynomial Kalman Filter Estimates Signal Perfectly After Only One Measurement

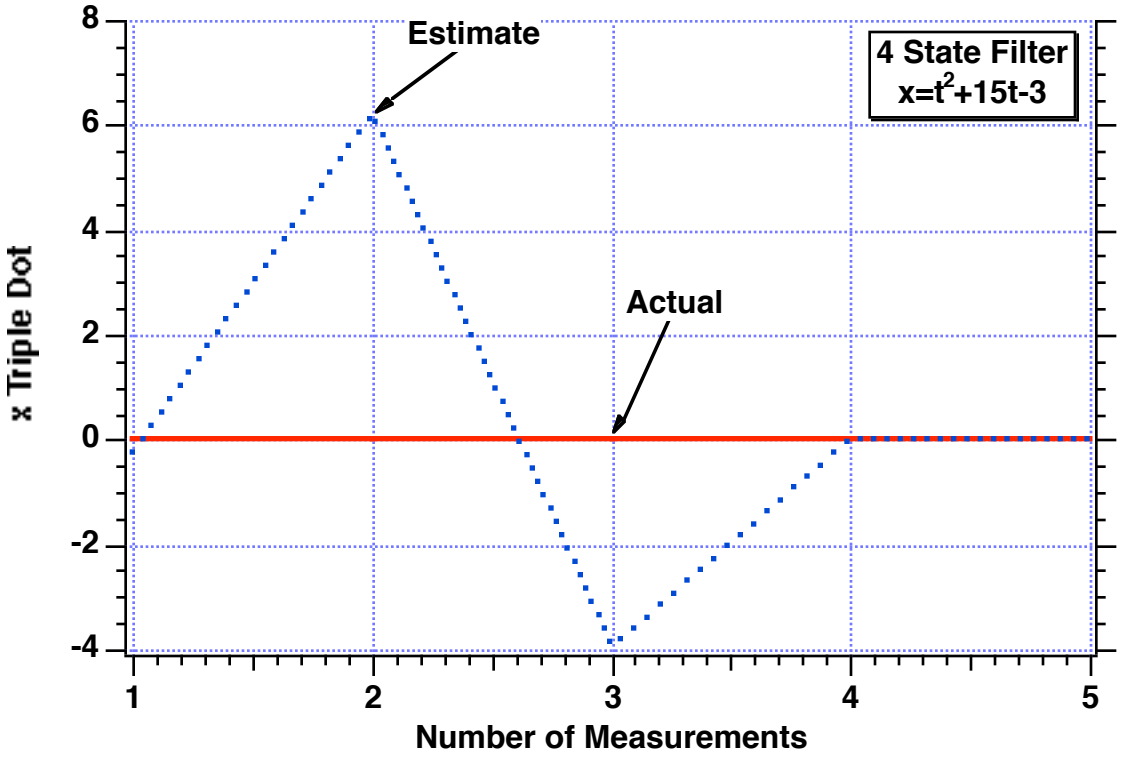


Four-State Filter Requires Four Measurements to Estimate First and Second Derivatives of the Signal

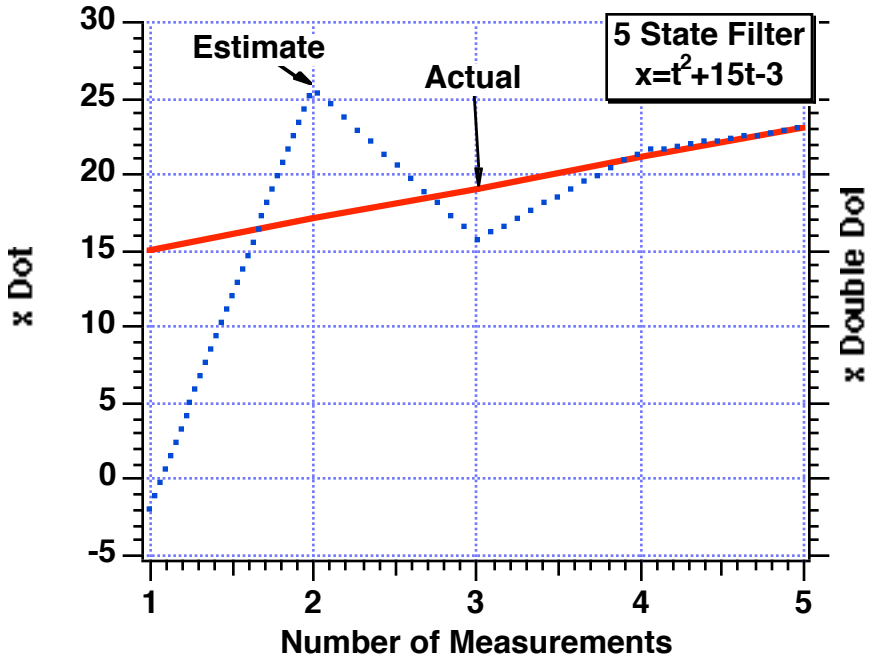


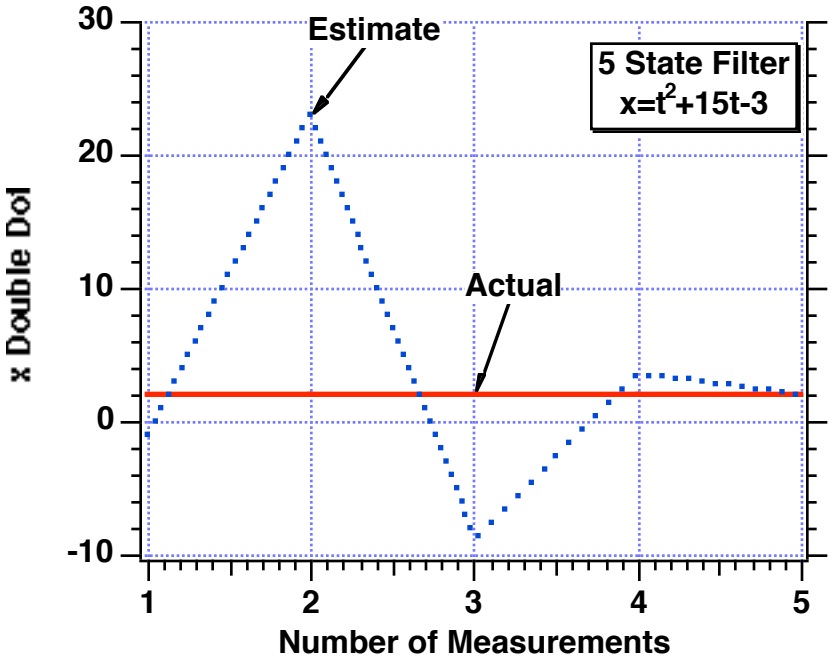


Unlike Lower Order Filter Four-State Filter Can Estimate Third Derivative of the Signal

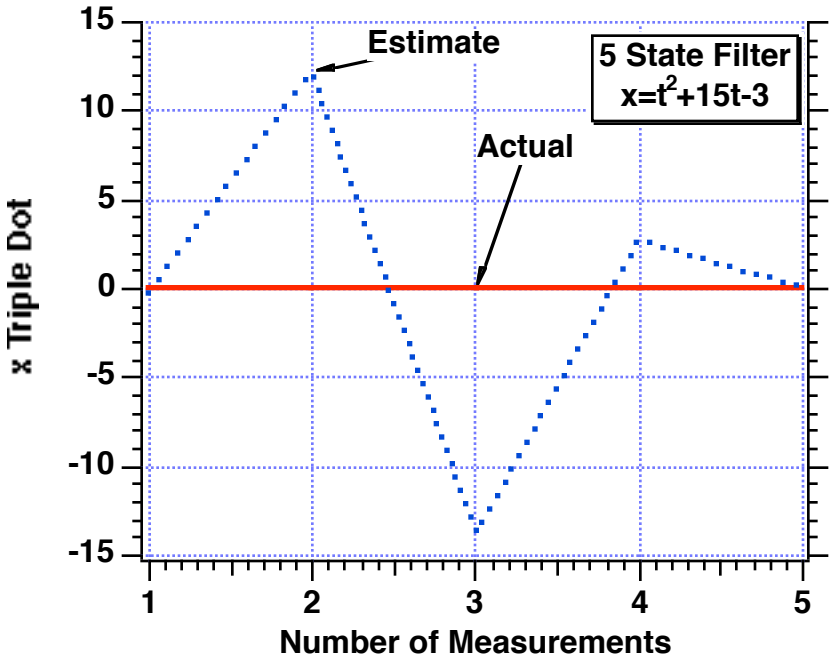


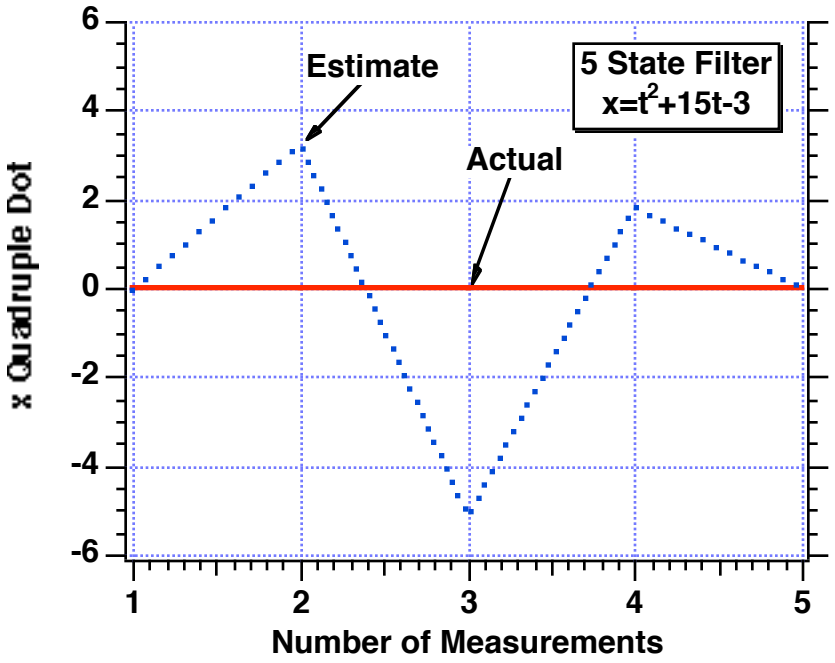
Five-State Filter Requires Five Measurements to Estimate First and Second Derivatives of the Signal





Unlike Lower Order Filter Five-State Filter Can Estimate Third and Fourth Derivatives of the Signal (With 5 Measurements)





Summary

- When only a few measurements are available the lower order polynomial Kalman filter is more accurate than the higher order filter in estimating the lower order derivatives of the measurement signal
- A three-state filter requires three measurements to get perfect estimates of the first and second derivatives of a signal
- A four-state filter requires four measurements to get perfect estimates of the first, second and third derivatives of the signal
- A five-state filter requires five measurement to get estimates of the first, second, third and fourth derivatives of the signal

Detecting Filter Divergence in the Real World

Deriving Formula For Residual

Kalman filter

$$\hat{\mathbf{x}}_k = \Phi_k \hat{\mathbf{x}}_{k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\Phi_k \hat{\mathbf{x}}_{k-1})$$

Definition of residual

$$\mathbf{RES} = \mathbf{z}_k - \mathbf{H}\Phi_k \hat{\mathbf{x}}_{k-1}$$

Measurement

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k$$

Projecting true states ahead

$$\mathbf{x}_k = \Phi_k \mathbf{x}_{k-1} + \mathbf{w}_k$$

Substitution yields

$$\mathbf{RES} = \mathbf{H}\Phi_k \mathbf{x}_{k-1} + \mathbf{H}\mathbf{w}_k + \mathbf{v}_k - \mathbf{H}\Phi_k \hat{\mathbf{x}}_{k-1}$$

Combining like terms

$$\mathbf{RES} = \mathbf{H}\Phi_k(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}) + \mathbf{H}\mathbf{w}_k + \mathbf{v}_k$$

Finding Covariance of Residual

Taking expectations

$$E(RES*RES^T) = H\Phi_k E[(x_{k-1} - \hat{x}_{k-1})(x_{k-1} - \hat{x}_{k-1})^T]\Phi_k^T H^T + HE(w_k w_k^T)H^T + E(v_k v_k^T)$$

Since

$$P_k = E[(x_{k-1} - \hat{x}_{k-1})(x_{k-1} - \hat{x}_{k-1})^T]$$

$$Q_k = E(w_k w_k^T)$$

$$R_k = E(v_k v_k^T)$$

Covariance of residual reduces to

$$E(RES*RES^T) = H\Phi_k P_k \Phi_k^T H^T + H Q_k H^T + R_k$$

Or more simply

$$E(RES*RES^T) = H(\Phi_k P_k \Phi_k^T + Q_k)H^T + R_k$$

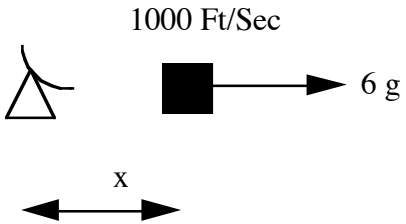
From first Riccati equation we know

$$M_k = \Phi_k P_k \Phi_k^T + Q_k$$

Therefore

$$E(RES*RES^T) = H_k M_k H^T + R_k$$ ← Important

Accelerating Object Traveling Away From Radar



From basic physics

$$x = 1000t + \frac{1}{2} 193.2t^2$$

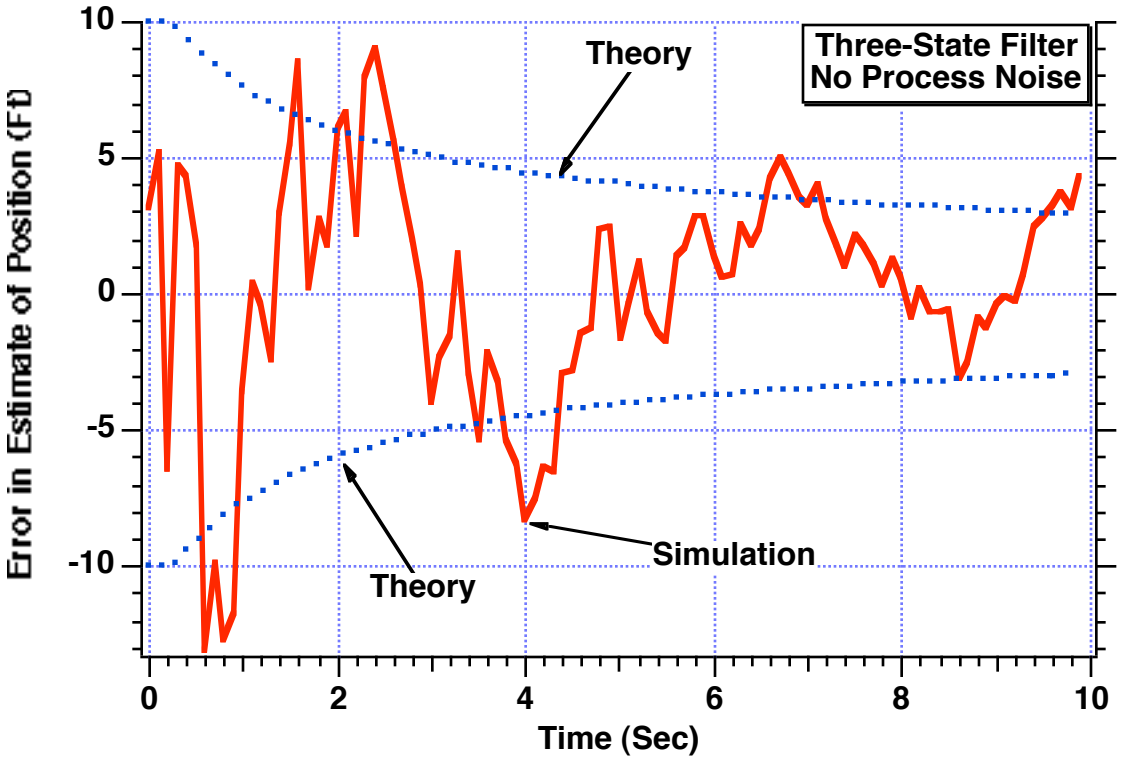
Taking derivatives

$$\dot{x} = 1000 + 193.2t$$

$$\ddot{x} = 193.2$$

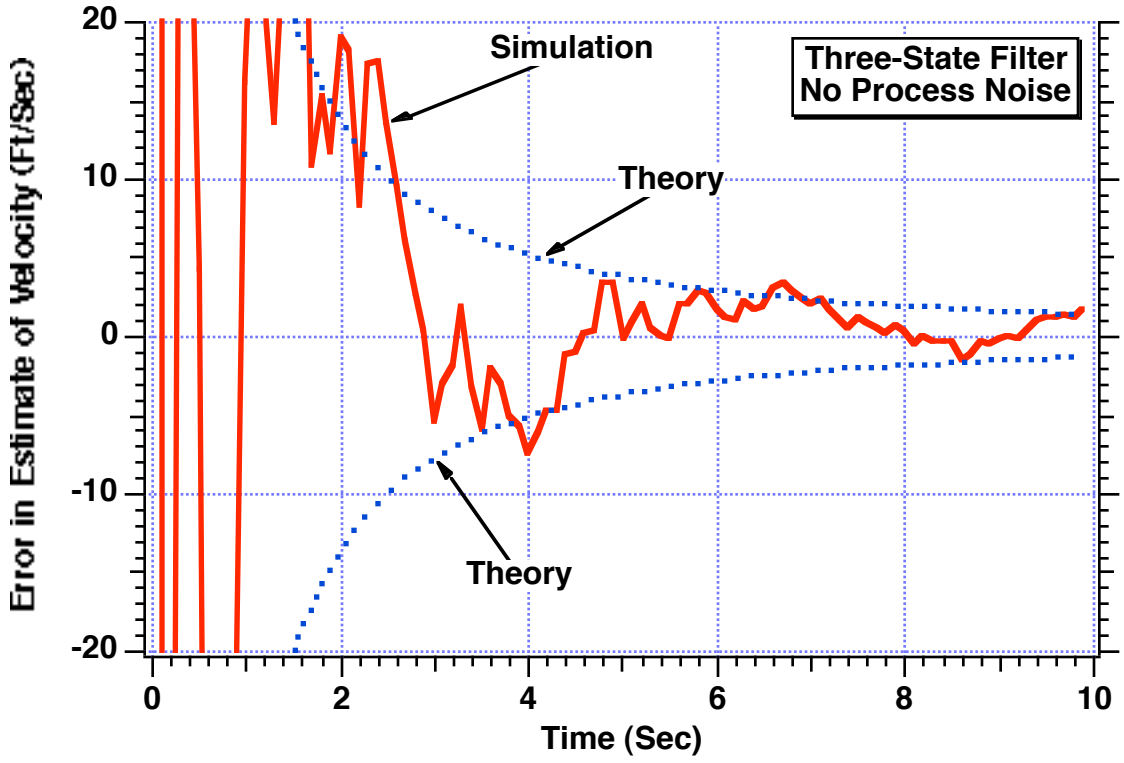
Three-State Kalman Filter Can be Used to Track Accelerating Object

Error in the Estimate of Position is Within the Theoretical Bounds



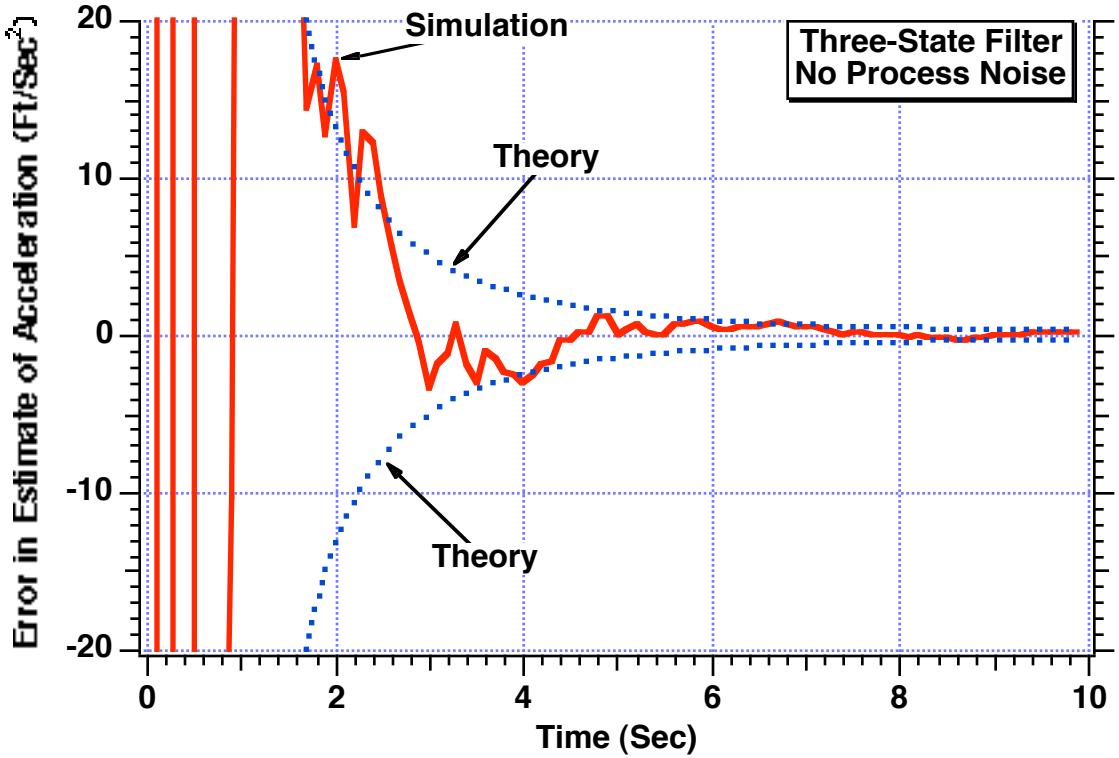
$$x = 1000t + \frac{1}{2}193.2t^2$$

Error in the Estimate of Velocity is Within the Theoretical Bounds



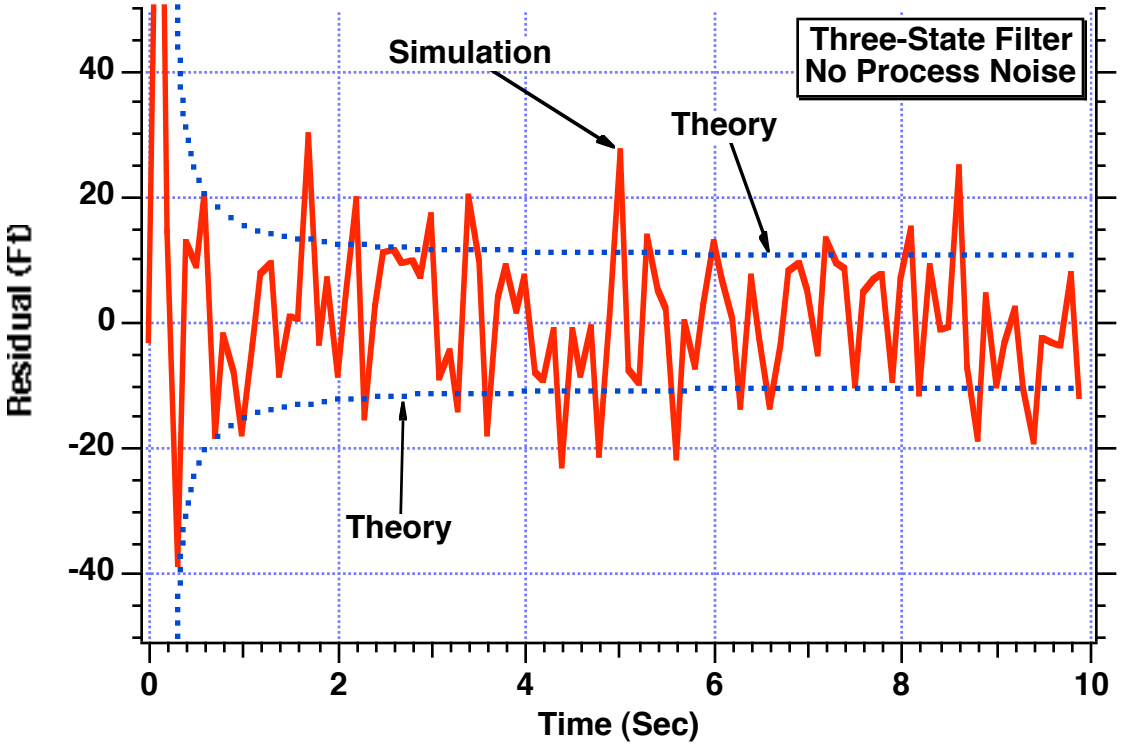
$$\dot{x} = 1000 + 193.2t$$

Error in the Estimate of Acceleration is Within the Theoretical Bounds



$\ddot{x} = 193.2$

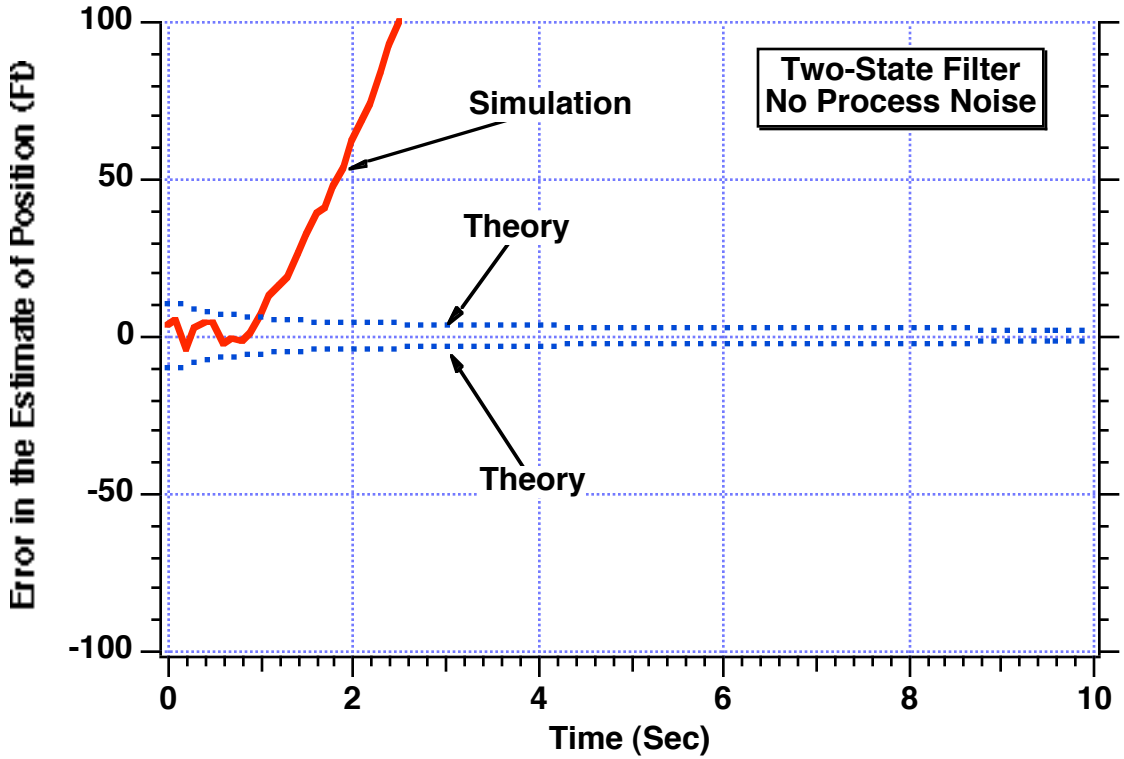
Residual is Within the Theoretical Bounds



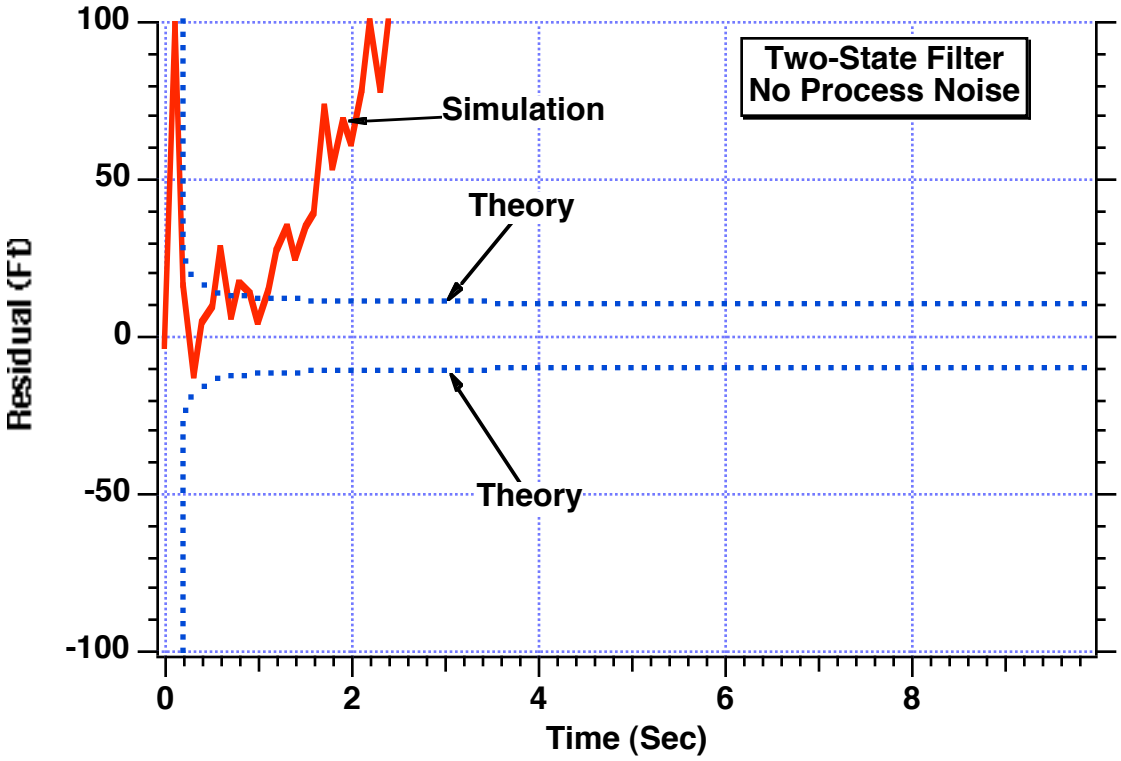
$$E(RES * RES^T) = H_k M_k H^T + R_k$$

Two-State Kalman Filter Without Process Noise Can Not be Used to Track Accelerating Object

Error in the Estimate of Position for Two-State Filter Without Process Noise is Diverging

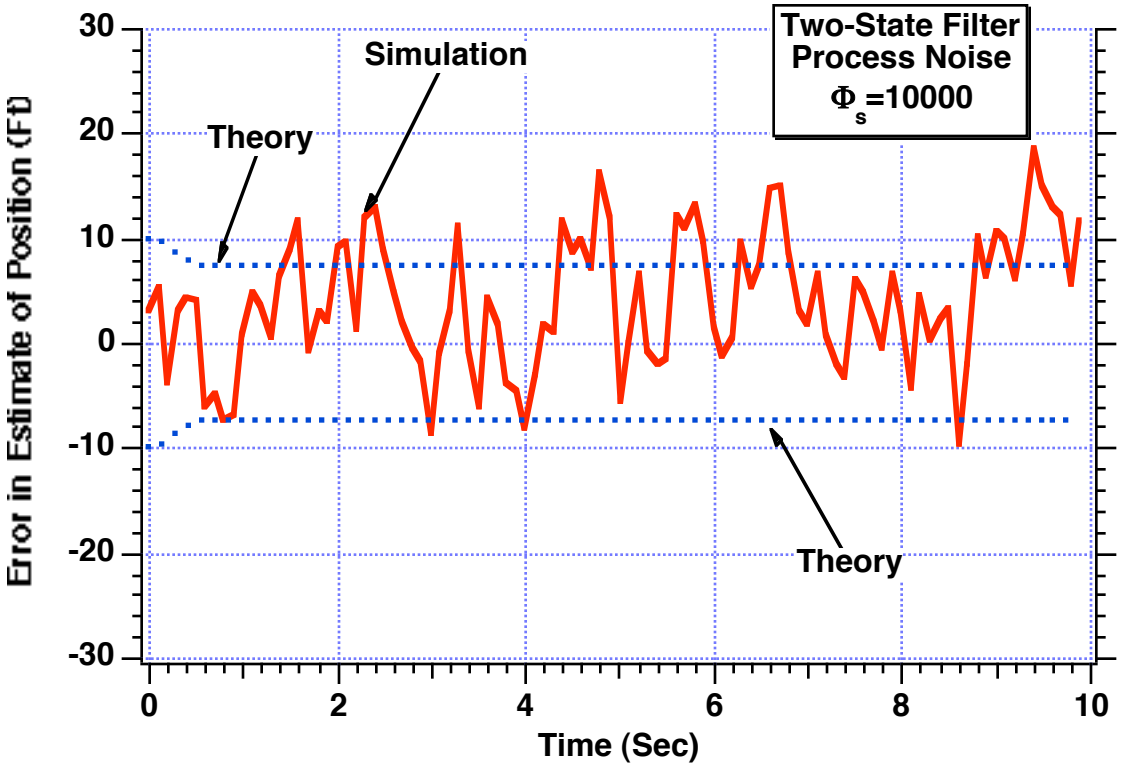


Residual Also Indicates That Two-State Filter Without Process Noise Can Not Track Accelerating Object

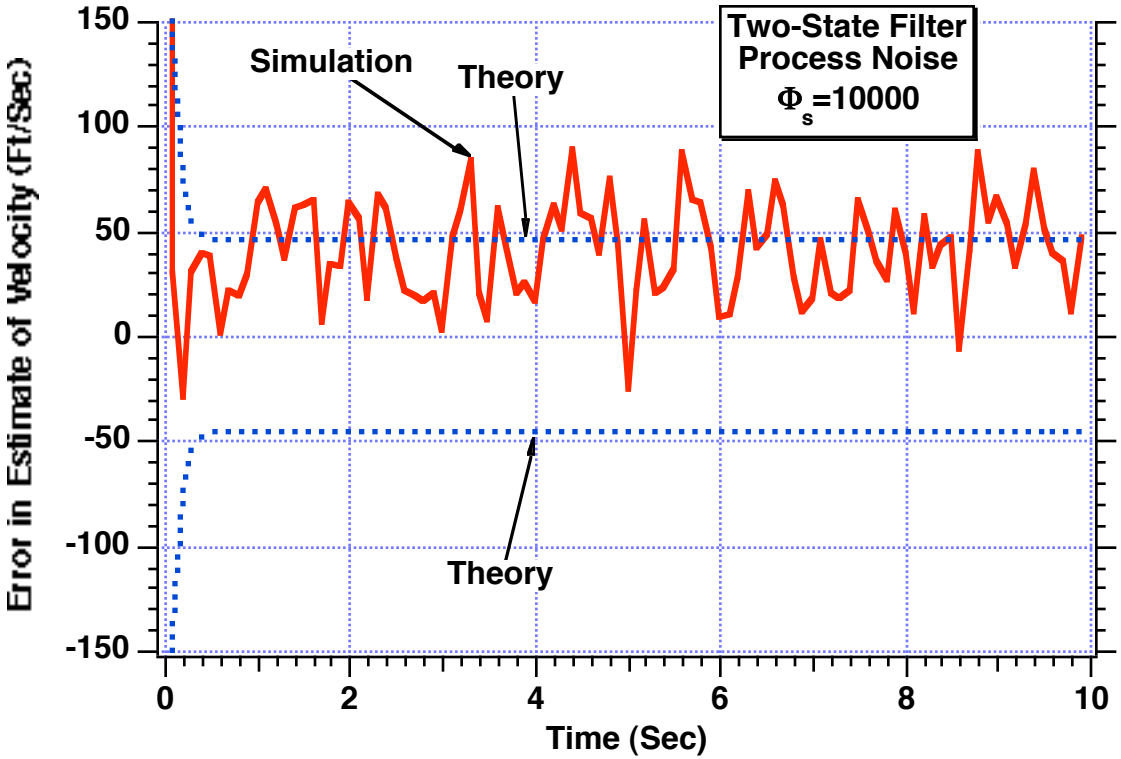


Two-State Kalman Filter With Process Noise Can be Used to Track Accelerating Object

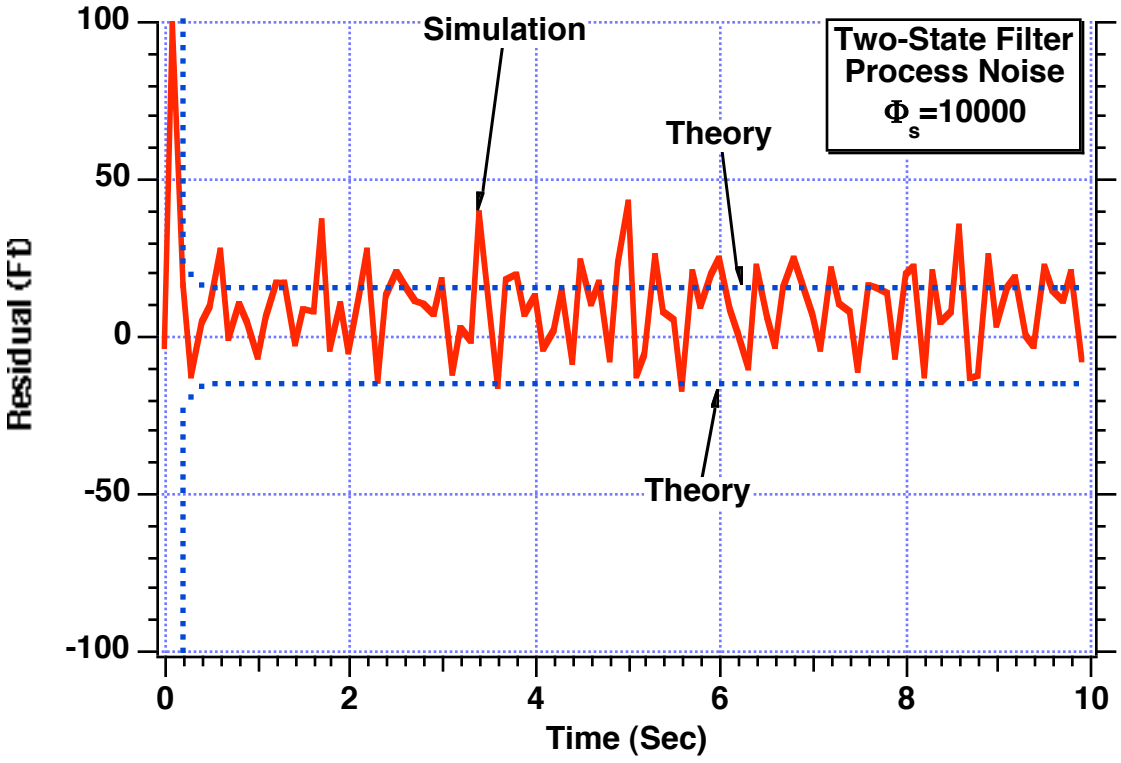
Addition of Process Noise Enables Errors in Estimate of Position to Lie Within Theoretical Bounds



Addition of Process Noise Enables Errors in Estimate of Velocity to Lie Within Theoretical Bounds



Addition of Process Noise Enables Residual to Lie Within Theoretical Bounds



Summary

- We have demonstrated that the residual and its theoretical bounds can also be used as an indication of whether or not the filter is working properly
- The residual test not only appears to be a useful test that can be used in the actual operation of the filter to see if it is performing according to theory but can also be used to determine if corrective actions to the filter are required

Observability Example

Traditional Two-State Polynomial Kalman Filter

Measurement is polynomial plus noise

$$x^* = a_0 + a_1 t + v = x + v$$

Model of real world in state space form

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

Measurement equation

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

Fundamental and measurement matrices

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

Kalman filtering equation

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

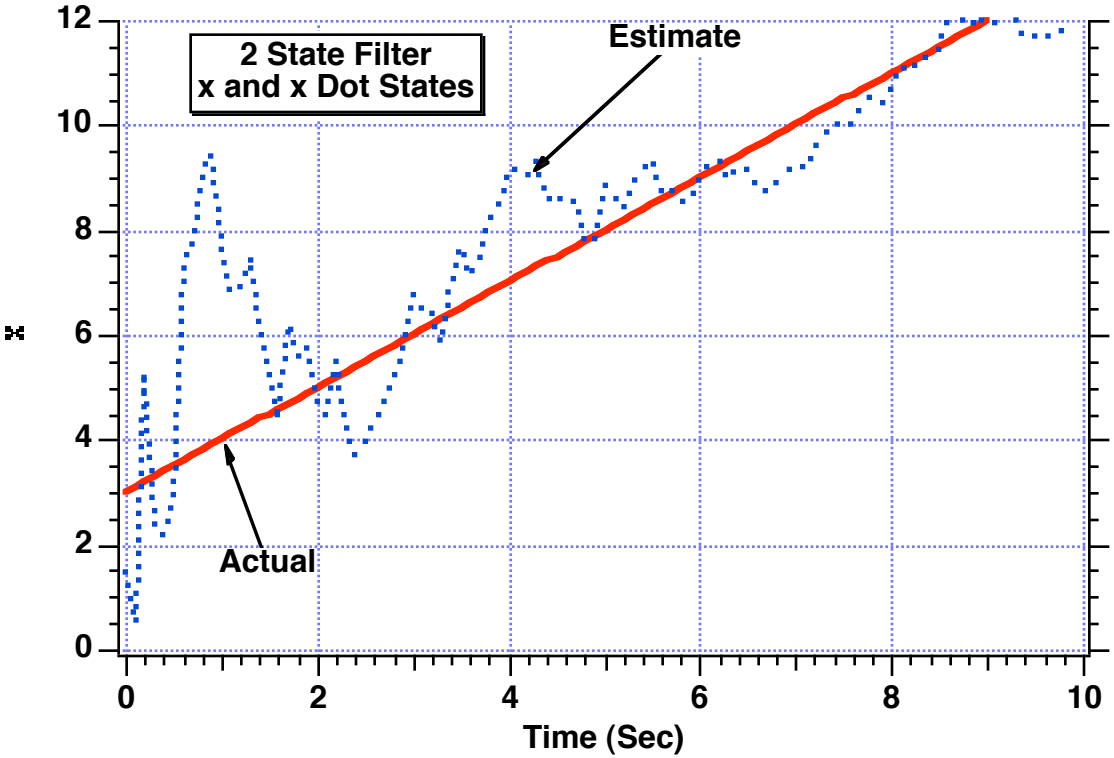
$$\begin{bmatrix} \hat{x}_k \\ \hat{\dot{x}}_k \end{bmatrix} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_{k-1} \\ \hat{\dot{x}}_{k-1} \end{bmatrix} + \begin{bmatrix} K_{1_k} \\ K_{2_k} \end{bmatrix} \left(x_k^* - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_{k-1} \\ \hat{\dot{x}}_{k-1} \end{bmatrix} \right)$$

Or in scalar form

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

$$\hat{\dot{x}}_k = \hat{\dot{x}}_{k-1} + K_{2_k}(x_k^* - \hat{x}_{k-1} - \hat{\dot{x}}_{k-1} T_s)$$

After a Brief Transient Period the Two-State Filter is Able to Estimate Signal

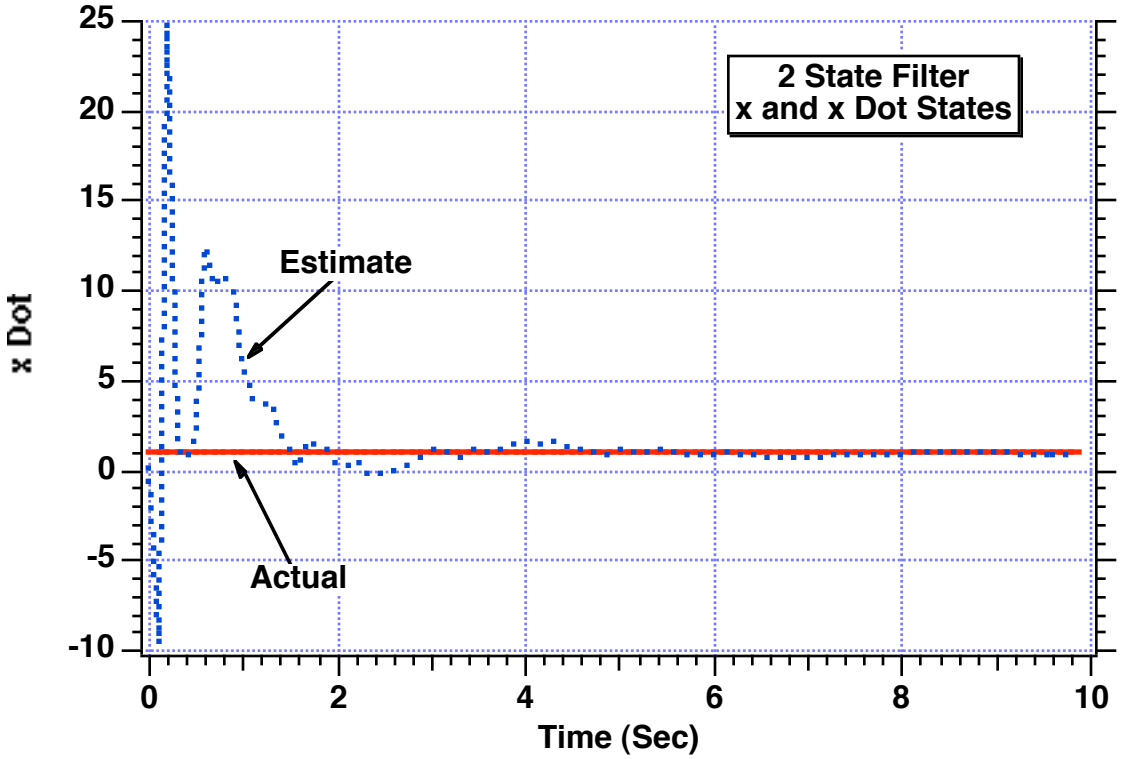


$x^* = 3 + t + noise$

$\sigma_{noise} = 5$

$x = 3 + t$

After a Brief Transient Period the Two-State Filter is Able to Estimate the Derivative of the Signal



$x^* = 3 + t + noise$

$\sigma_{noise} = 5$

$\dot{x} = 1$

Alternative Two-State Kalman Filter

Measurement is polynomial plus noise

$$x^* = a_0 + a_1 t + v$$

Model of real world in state space form

$$\begin{bmatrix} \dot{a}_0 \\ \dot{a}_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}$$

Measurement equation

$$x^* = \begin{bmatrix} 1 & t \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} + v$$

Fundamental and measurement matrices

$$\mathbf{\Phi}_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \mathbf{H} = \begin{bmatrix} 1 & t \end{bmatrix}$$

Kalman filtering equation

$$\hat{\mathbf{x}}_\mathbf{k} = \mathbf{\Phi}_\mathbf{k} \hat{\mathbf{x}}_{\mathbf{k}-1} + \mathbf{K}_\mathbf{k} (\mathbf{z}_\mathbf{k} - \mathbf{H} \mathbf{\Phi}_\mathbf{k} \hat{\mathbf{x}}_{\mathbf{k}-1})$$

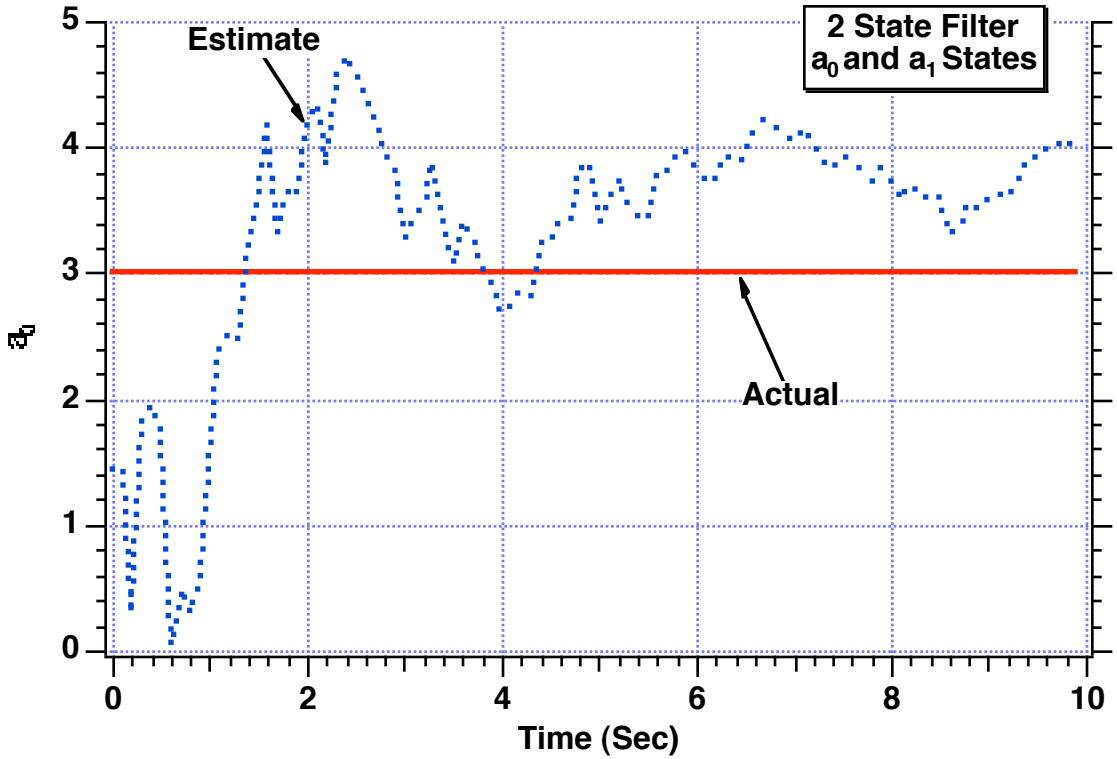
$$\begin{bmatrix} \hat{a}_{0_k} \\ \hat{a}_{1_k} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{a}_{0_{k-1}} \\ \hat{a}_{1_{k-1}} \end{bmatrix} + \begin{bmatrix} K_{1_k} \\ K_{2_k} \end{bmatrix} \left(x^* - \begin{bmatrix} 1 & t \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{a}_{0_{k-1}} \\ \hat{a}_{1_{k-1}} \end{bmatrix} \right)$$

Or in scalar form

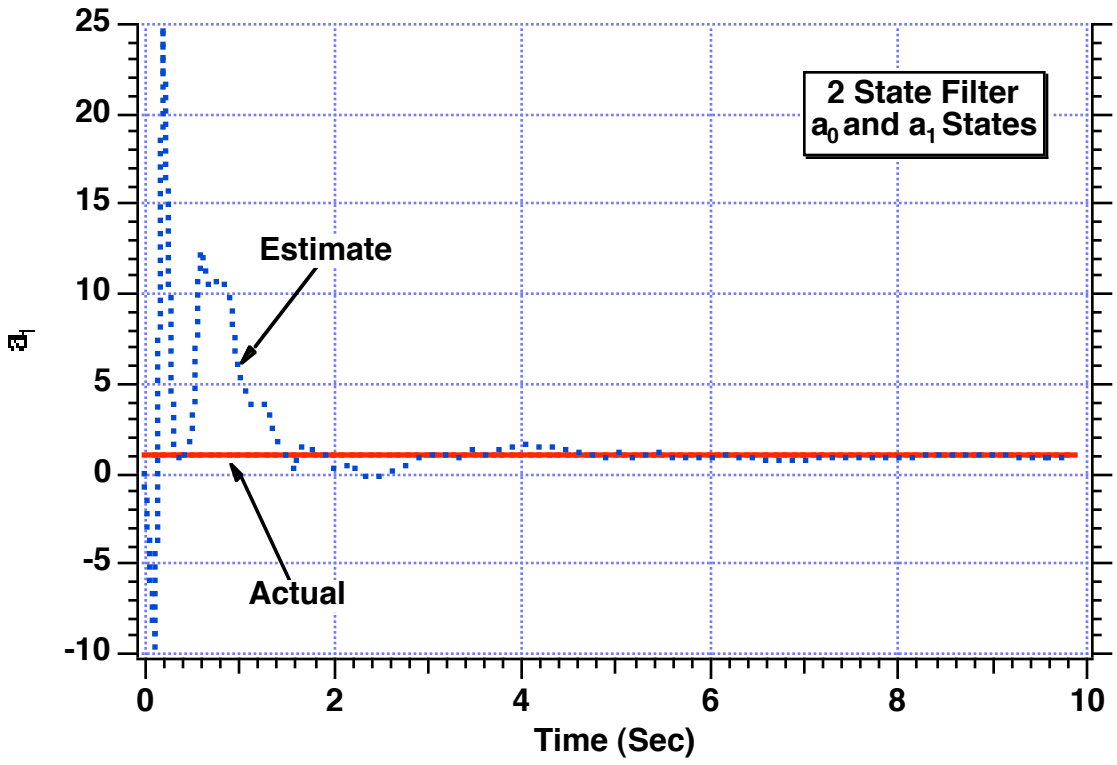
$$\hat{a}_{0_k} = \hat{a}_{0_{k-1}} + K_{1_k} (x_k^* - \hat{a}_{0_{k-1}} - \hat{a}_{1_{k-1}} t)$$

$$\hat{a}_{1_k} = \hat{a}_{1_{k-1}} + K_{2_k} (x_k^* - \hat{a}_{0_{k-1}} - \hat{a}_{1_{k-1}} t)$$

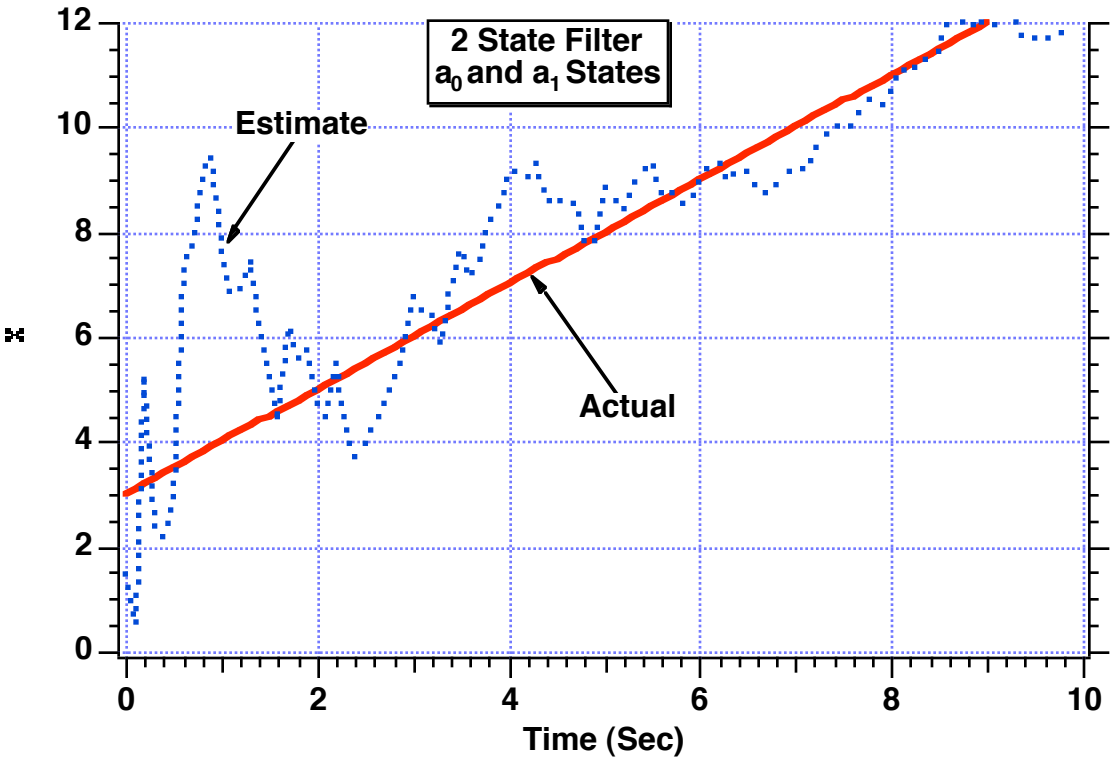
Alternative Kalman Filter Estimates First Polynomial Coefficient to Within 33%



Alternative Kalman Filter Does Very Well in Estimating Second Polynomial Coefficient

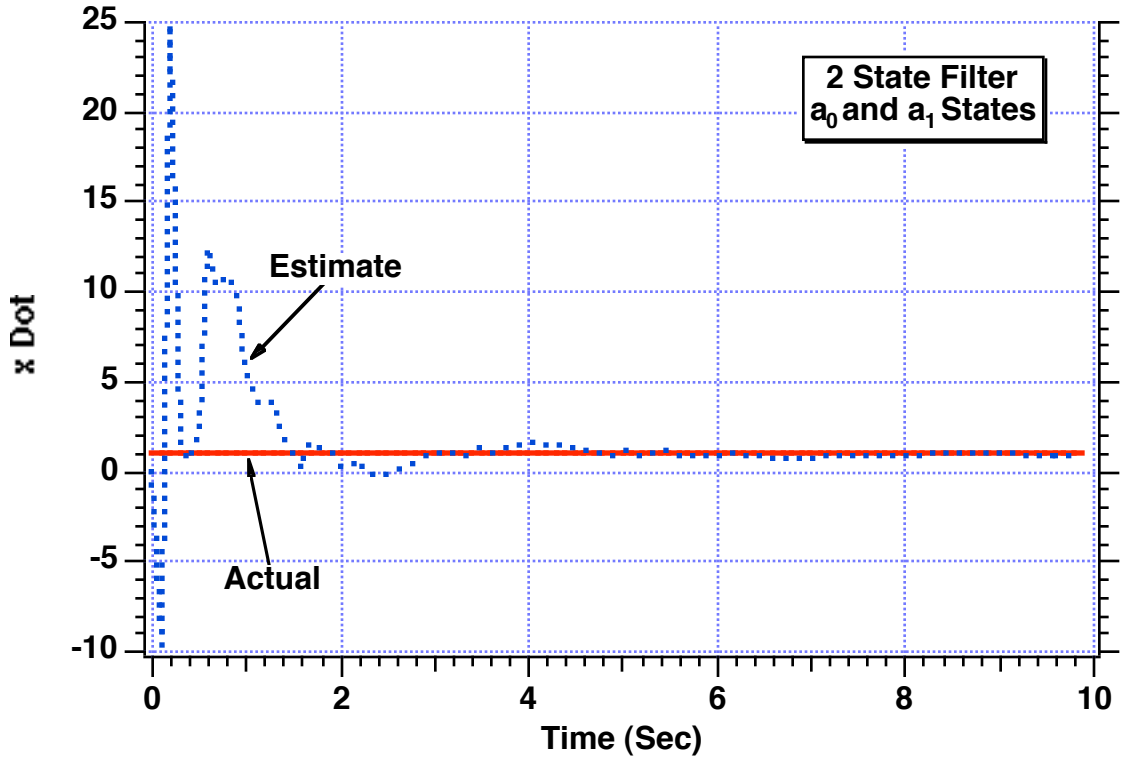


Using Alternative Kalman Filter Estimates Yields Good Estimates of Signal



$$\hat{x}_k = \hat{a}_{0_k} + \hat{a}_{1_k} t$$

Using Alternative Kalman Filter Estimates Yields Good Estimates of Derivative of the Signal



$$\hat{\dot{x}}_k = \hat{a}_{1_k}$$

Three-State Kalman Filter

Measurement is polynomial plus noise

$$x^* = a_0 + a_1 t + a_2 t + v$$ ← Less efficient form

Model of real world in state space form

$$\begin{bmatrix} \dot{a}_0 \\ \dot{a}_1 \\ \dot{a}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

Measurement equation

$$x^* = \begin{bmatrix} 1 & t & t \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} + v$$

Fundamental and measurement matrices

$$\mathbf{\Phi_k} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \mathbf{H} = \begin{bmatrix} 1 & t & t \end{bmatrix}$$

Kalman filtering equation

$$\begin{bmatrix} \hat{a}_{0_k} \\ \hat{a}_{1_k} \\ \hat{a}_{2_k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{a}_{0_{k-1}} \\ \hat{a}_{1_{k-1}} \\ \hat{a}_{2_{k-1}} \end{bmatrix} + \begin{bmatrix} K_{1_k} \\ K_{2_k} \\ K_{3_k} \end{bmatrix} \left(x^* - \begin{bmatrix} 1 & t & t \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{a}_{0_{k-1}} \\ \hat{a}_{1_{k-1}} \\ \hat{a}_{2_{k-1}} \end{bmatrix} \right)$$

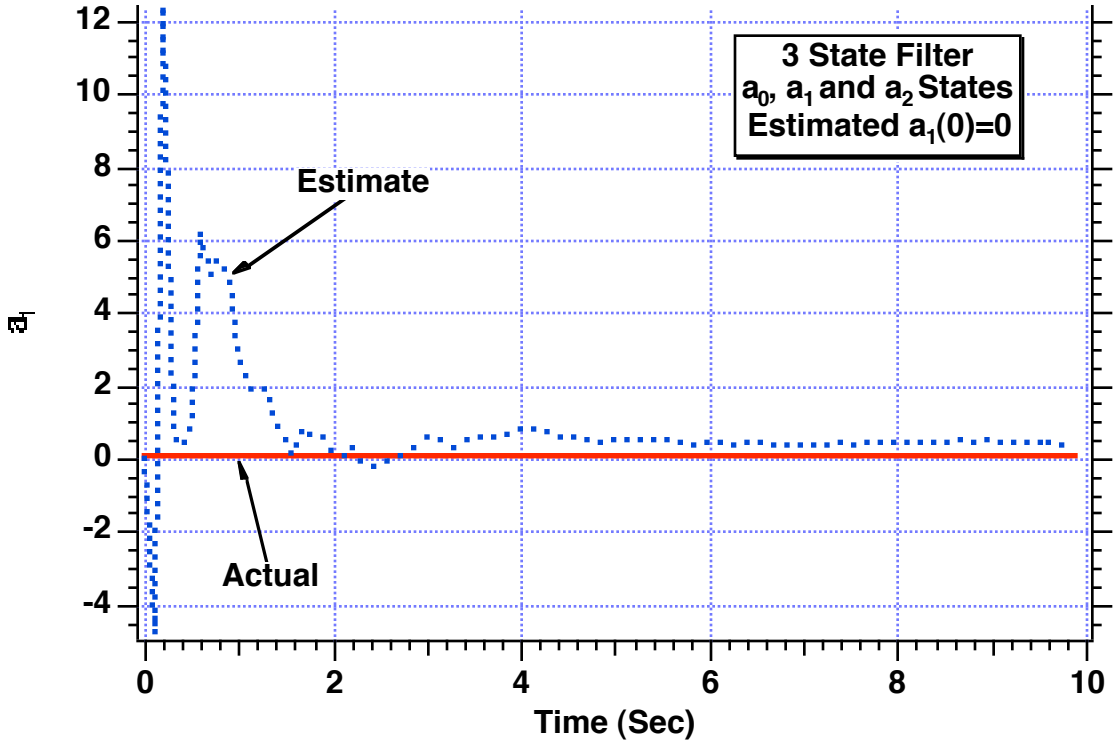
Or in scalar form

$$\hat{a}_{0_k} = \hat{a}_{0_{k-1}} + K_{1_k}(x_k^* - \hat{a}_{0_{k-1}} - \hat{a}_{1_{k-1}} t - \hat{a}_{2_{k-1}} t)$$

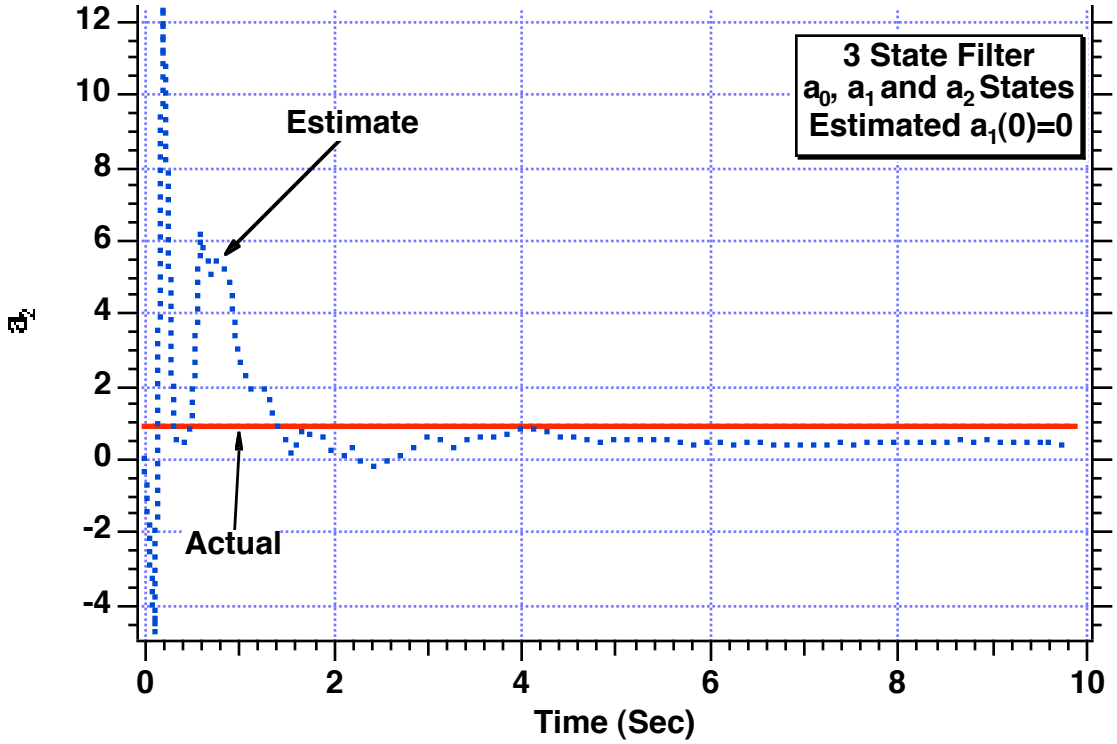
$$\hat{a}_{1_k} = \hat{a}_{1_{k-1}} + K_{2_k}(x_k^* - \hat{a}_{0_{k-1}} - \hat{a}_{1_{k-1}} t - \hat{a}_{2_{k-1}} t)$$

$$\hat{a}_{2_k} = \hat{a}_{2_{k-1}} + K_3(x_k^* - \hat{a}_{0_{k-1}} - \hat{a}_{1_{k-1}} t - \hat{a}_{2_{k-1}} t)$$

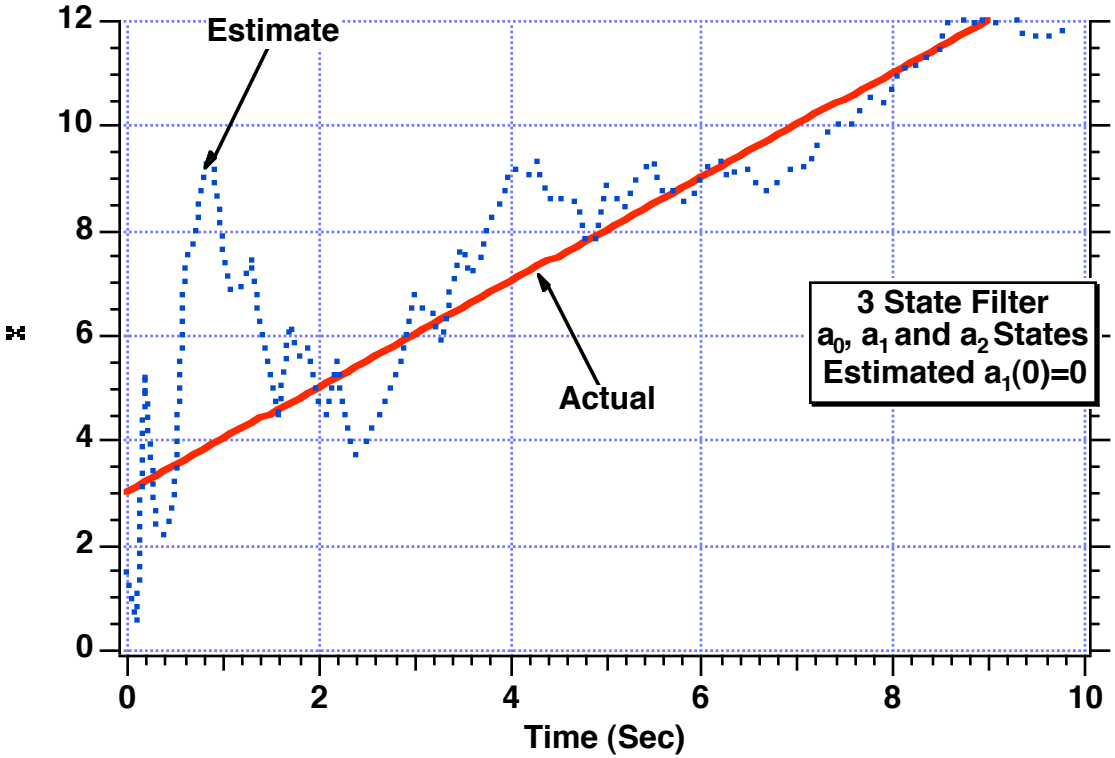
Three-State Filter Estimate of a_1 Asymptotically Approaches a Constant That is on the High Side



Three-State Filter Estimate of a_2 Asymptotically Approaches a Constant That is on the Low Side

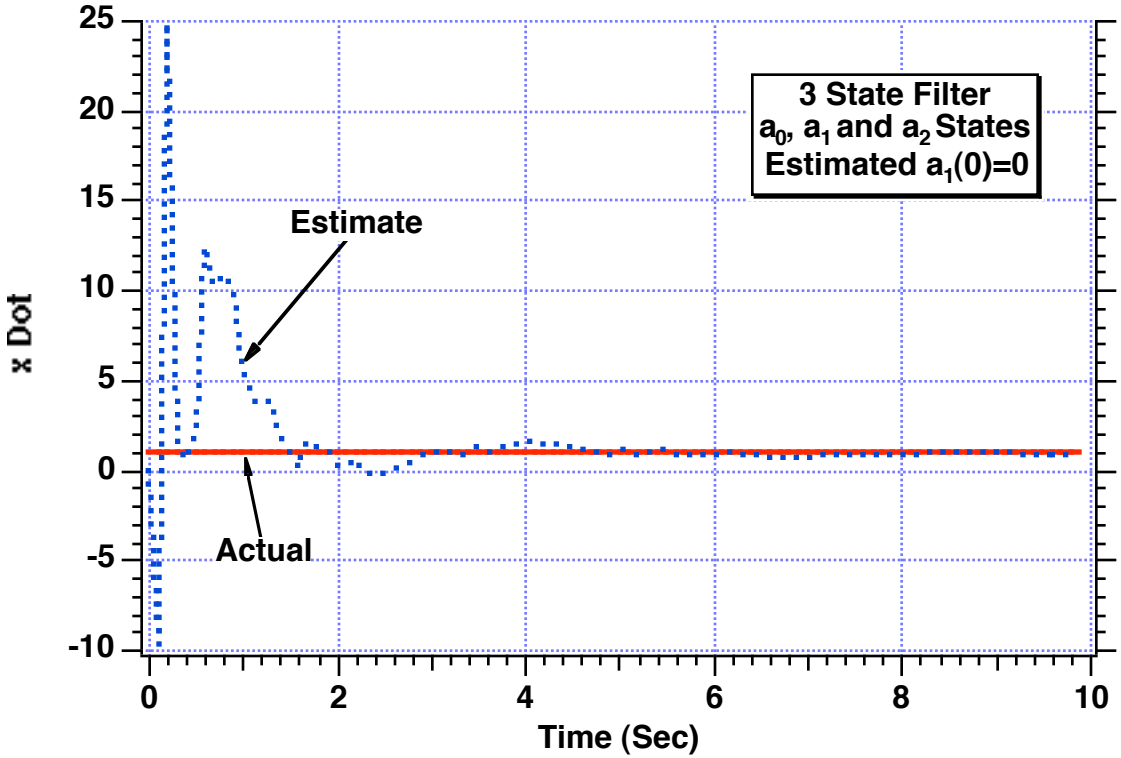


Using Three-State Kalman Filter Yields Good Estimates of Signal



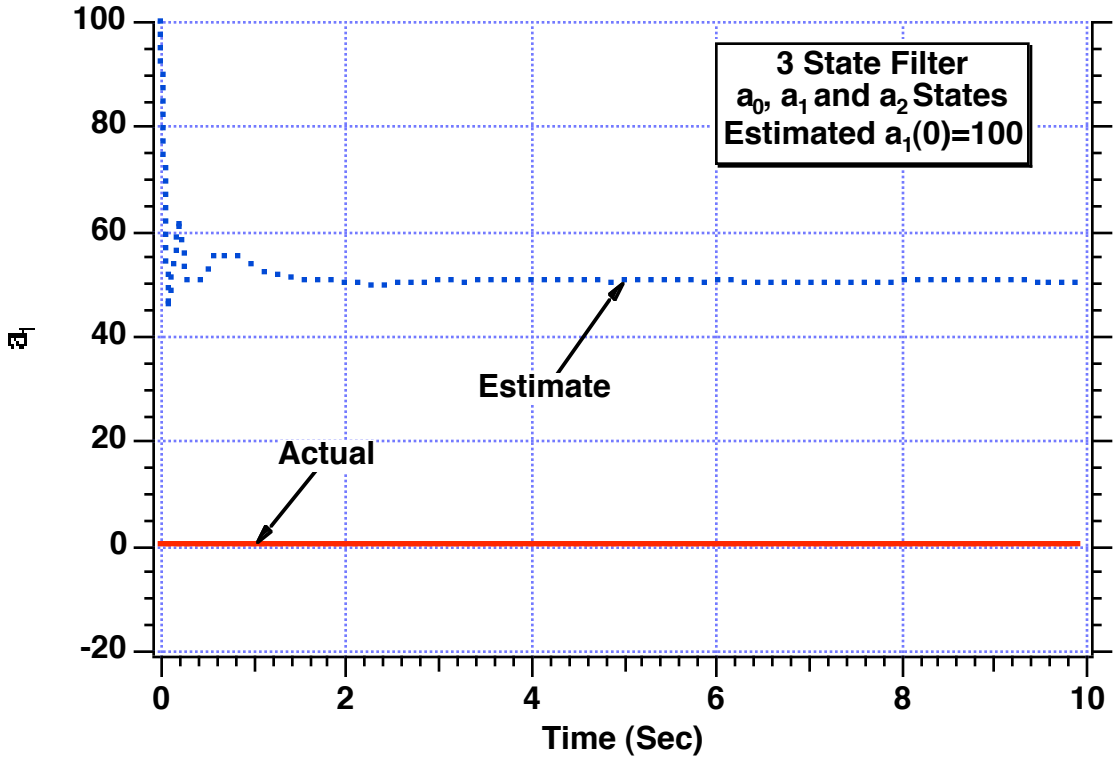
$$\hat{x}_k = \hat{a}_{0_k} + \hat{a}_{1_k} t + \hat{a}_{2_k} t$$

Using Three-State Kalman Filter Yields Good Estimate of Derivative of Signal

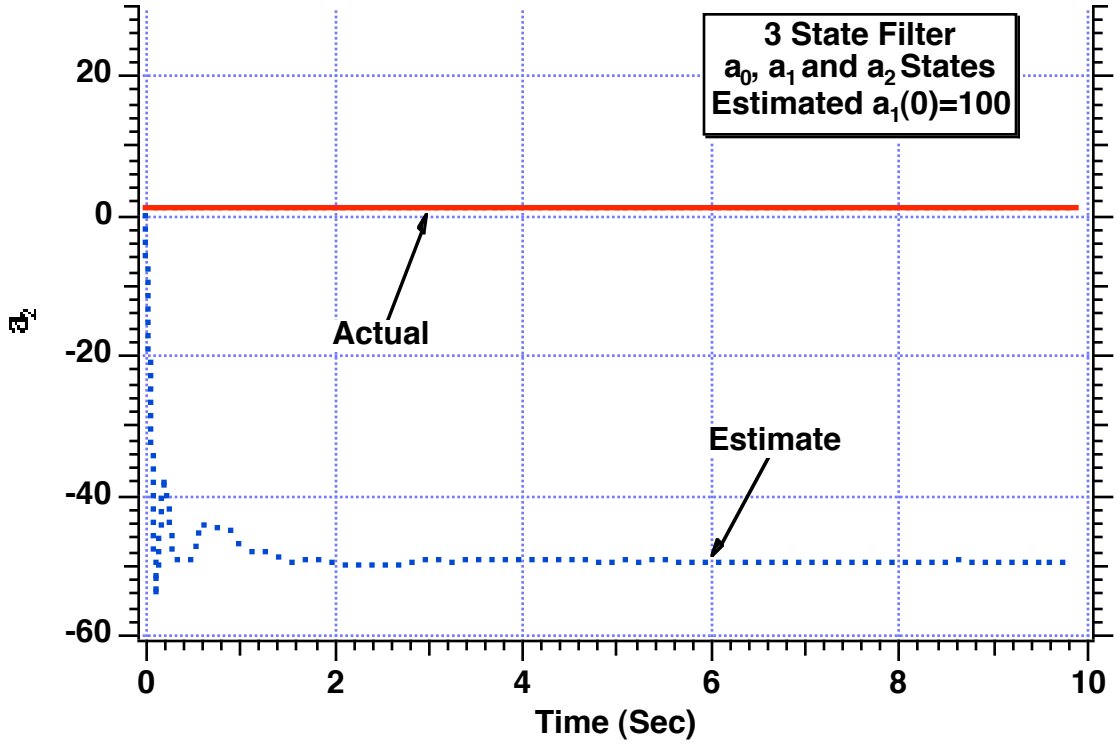


$$\hat{\dot{x}}_k = \hat{a}_{1_k} + \hat{a}_{2_k}$$

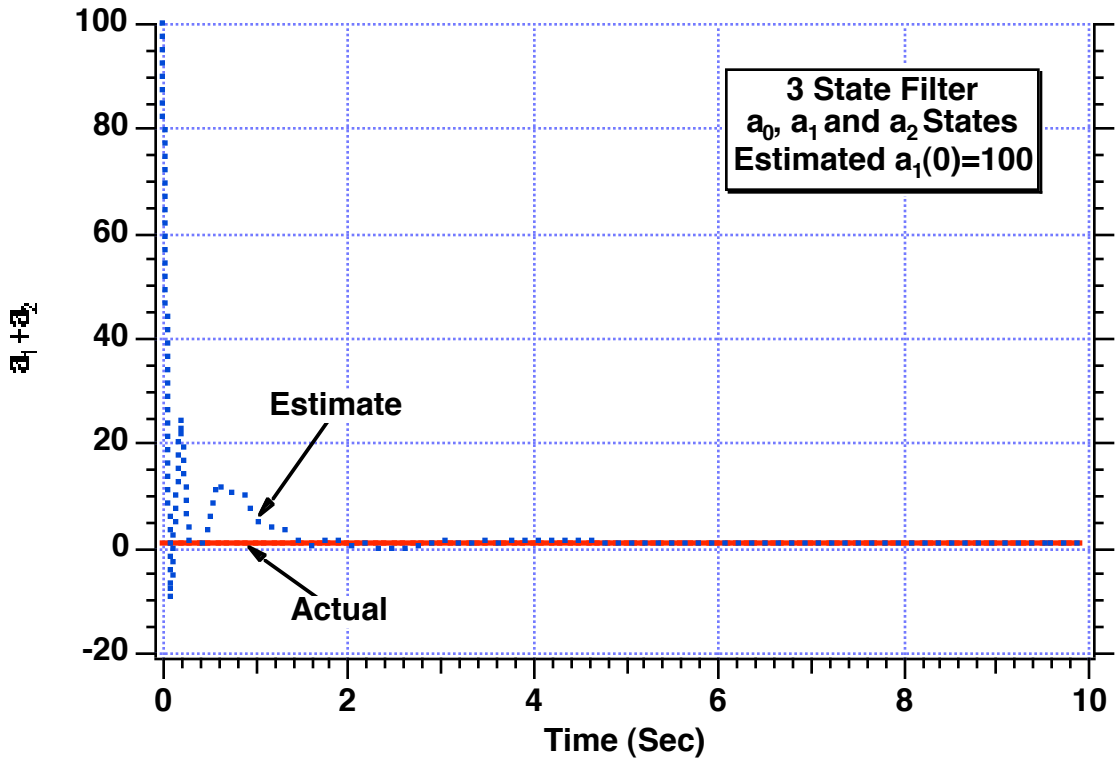
With Bad Initial Conditions the Three-State Filter is Unable to Estimate a_1



With Bad Initial Conditions the Three-State Filter is Unable to Estimate a_2



Sum of a_1 and a_2 Can be Estimated With Three-State Filter When the Initialization is Poor



Summary

- Even if states are not observable, a Kalman filter can be designed and built which might appear to work if the designer is not careful
- We showed that when filter states were not observable, the filter state estimates were a function of the initial conditions and the state estimates did not improve as more measurements were taken.
  - If the initial conditions were accurate, the state estimates of the unobservable states would be accurate
  - If the initial conditions were inaccurate, the state estimates of the unobservable states would also be inaccurate

Aiding

Model and Desires

Vehicle travels down a road with an erratic velocity

$$\dot{x} = 100 + 20\sin\omega t$$

If vehicle initially at origin, integration yields position

$$x = 100t - \frac{20\cos\omega t}{\omega} + \frac{20}{\omega}$$

Differentiation of velocity yields vehicles acceleration

$$\ddot{x} = 20\omega\cos\omega t$$

We want to build a Kalman filter to track the object given GPS position measurements every second with one sigma accuracy of 50 f/s

Since motion is unknown to us we will build three-state polynomial Kalman filter to be safe

MATLAB Simulation of Three-State Polynomial Kalman Filter Using GPS Information-1

```
ORDER =3;
XLOSE=9999999999.;
W=.1;
PHIS=1.;
TS=1.;
XH=0;
XDH=0;
XDDH=0;
SIGNOISE=50.;
PHI=zeros(ORDER,ORDER);
P=zeros(ORDER,ORDER);
IDNP=eye(ORDER);
Q=zeros(ORDER,ORDER);
P(1,1)=99999999999999.;
P(2,2)=99999999999999.;
P(3,3)=99999999999999.;
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;
H=zeros(1,ORDER);
H(1,1)=1;
HT=H';
R(1,1)=SIGNOISE^2;
PHIT=PHI';
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;
count=0;
```

Parameter for losing data

Process noise not zero

Initial filter state estimates

Initial covariance matrix

Fundamental matrix

Process noise matrix

MATLAB Simulation of Three-State Polynomial Kalman Filter Using GPS Information-2

```
for T=0:TS:200
        if T>XLOSE
                R(1,1)=99999999999999.;
        end
        PHIP=PHI*P;
        PHIPPHIT=PHIP*PHIT;
        M=PHIPPHIT+Q;
        MHT=M*HT;
        HMHT=H*MHT;
        HMHTR(1,1)=HMHT(1,1)+R(1,1);
        HMHTRINV(1,1)=1./HMHTR(1,1);
        K=MHT*HMHTRINV;
        KH=K*H;
        IKH=IDNP-KH;
        P=IKH*M;
        XNOISE=SIGNOISE*randn;
        X=100.*T-20.*cos(W*T)/W+20./W;
        XD=100.+20.*sin(W*T);
        XDD=20.*W*cos(W*T);
        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;
        SP11=sqrt(P(1,1));
        SP22=sqrt(P(2,2));
        SP33=sqrt(P(3,3));
        SP11P=-SP11;
        SP22P=-SP22;
        SP33P=-SP33;
        XHERR=X-XH;
        XDHERR=XD-XDH;
        XDDHERR=XDD-XDDH;
```

Simulating losing data

Riccati equations

True states of vehicle

Filter

Actual and theoretical errors in estimates

MATLAB Simulation of Three-State Polynomial Kalman Filter Using GPS Information-3

```
count=count+1;
ArrayT(count)=T;
ArrayX(count)=X;
ArrayXH(count)=XH;
ArrayXD(count)=XD;
ArrayXDH(count)=XDH;
ArrayXDD(count)=XDD;
ArrayXDDH(count)=XDDH;
ArrayXHERR(count)=XHERR;
ArraySP11(count)=SP11;
ArraySP11P(count)=SP11P;
ArrayXDHERR(count)=XDHERR;
ArraySP22(count)=SP22;
ArraySP22P(count)=SP22P;
ArrayXDDHERR(count)=XDDHERR;
ArraySP33(count)=SP33;
ArraySP33P(count)=SP33P;
end
```

Save data as arrays for plotting and writing data to files

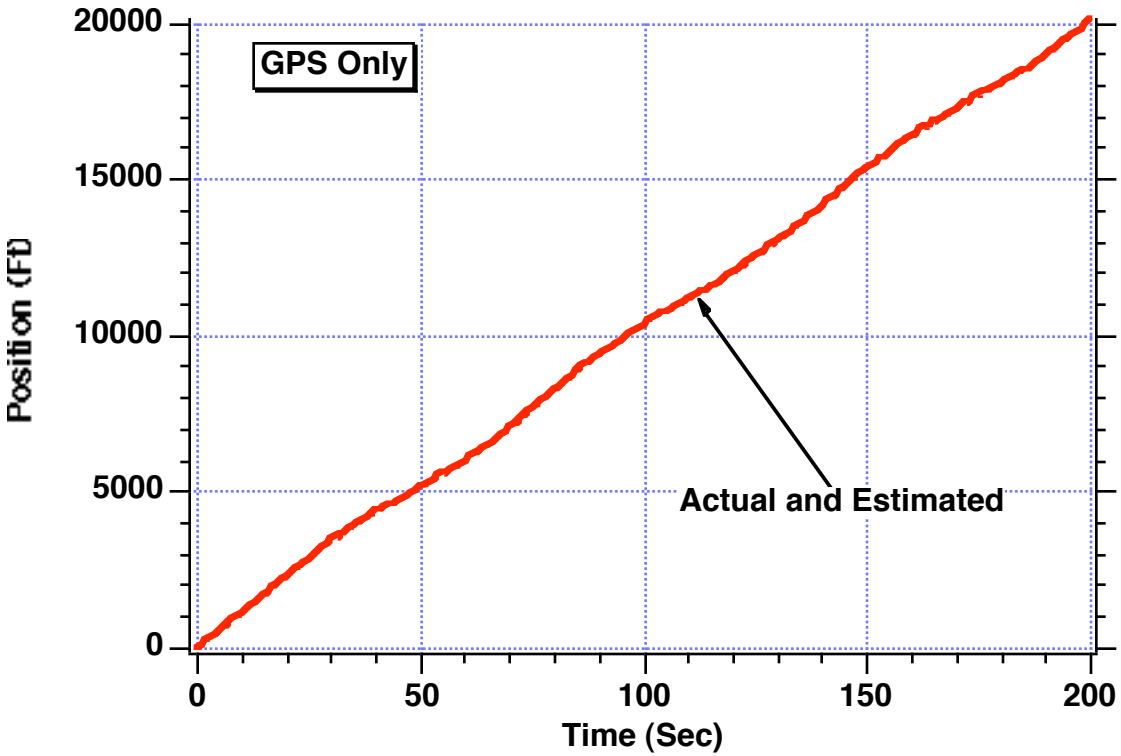
```
figure
plot(ArrayT,ArrayX,ArrayT,ArrayXH),grid
xlabel('Time (Sec)')
ylabel('Position (Ft)')
axis([0 200 0 20000])
figure
plot(ArrayT,ArrayXD,ArrayT,ArrayXDH),grid
xlabel('Time (Sec)')
ylabel('Velocity (Ft/Sec)')
axis([0 200 0 150])
figure
plot(ArrayT,ArrayXHERR,ArrayT,ArraySP11,ArrayT,ArraySP11P),grid
xlabel('Time (Sec)')
ylabel('Error in Estimate of Position (Ft)')
axis([0 200 -80 80])
```

Plot some results

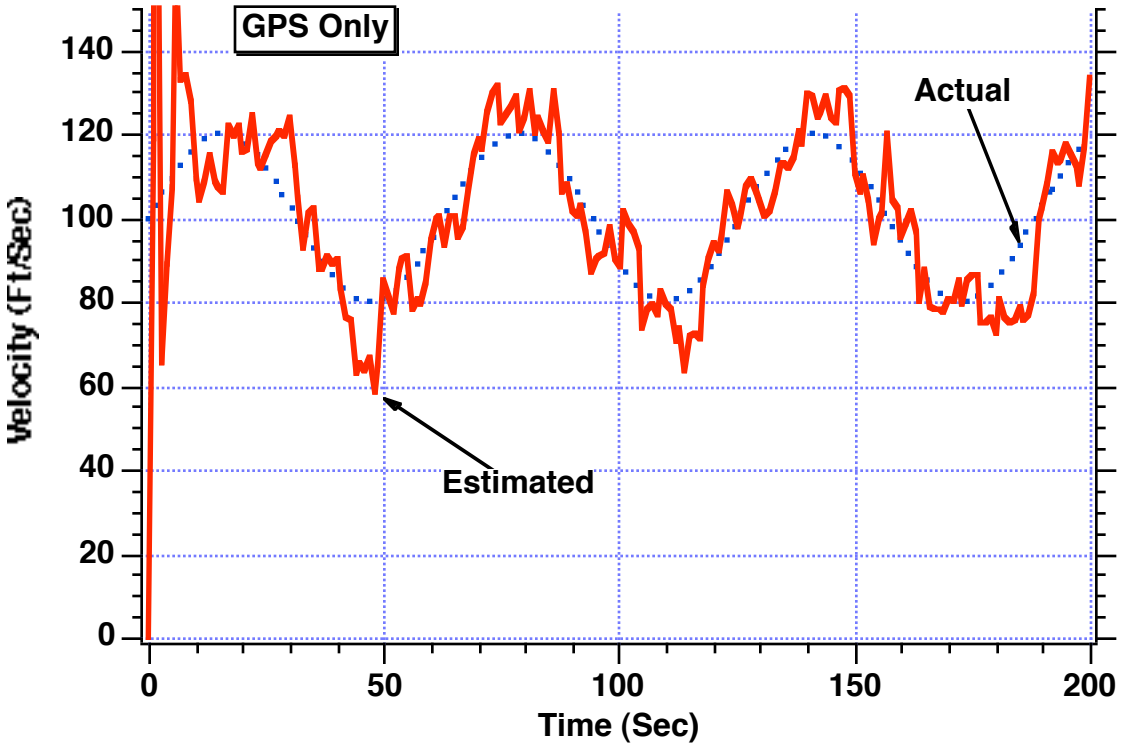
```
clc
output=[ArrayT',ArrayX',ArrayXH',ArrayXD',ArrayXDH',ArrayXDD',ArrayXDDH'];
save datfil output  -ascii
output=[ArrayT',ArrayXHERR',ArraySP11',ArraySP11P',ArrayXDHERR',ArraySP22',...
ArraySP22P',ArrayXDDHERR',ArraySP33',ArraySP33P'];
save covfil output  -ascii
disp 'simulation finished'
```

Write data to files

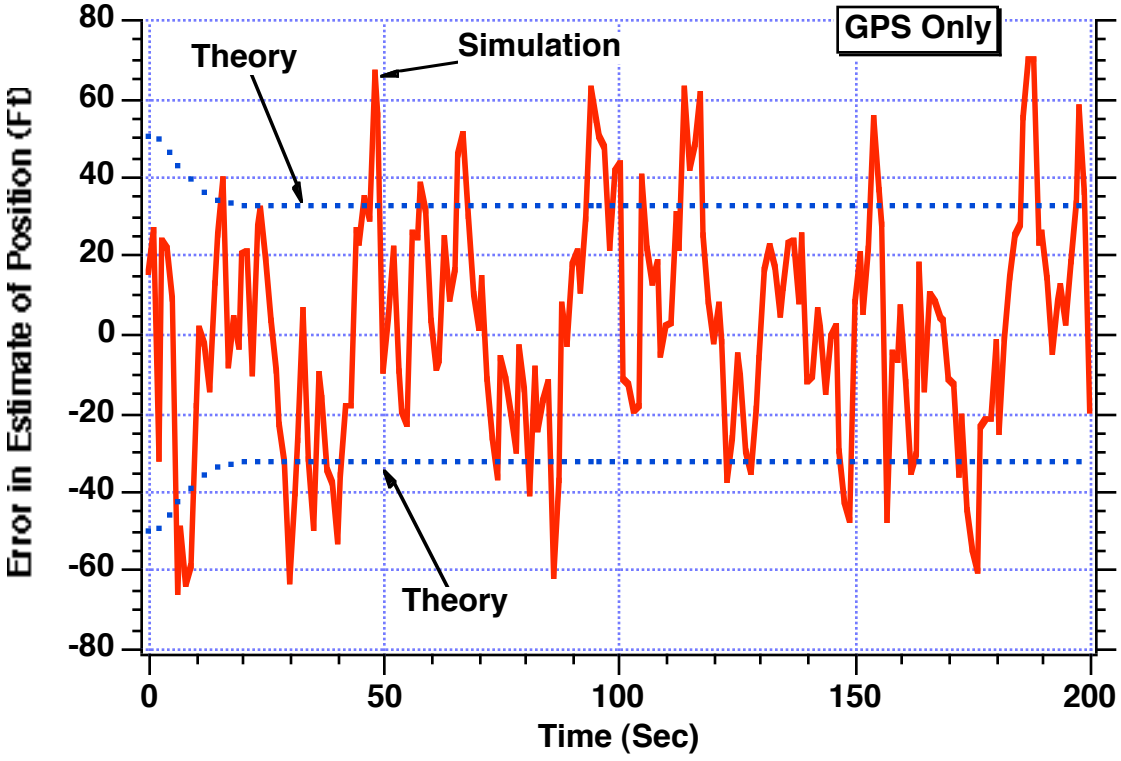
With All the GPS Data We are Able to Track Vehicles' Position Extremely Well



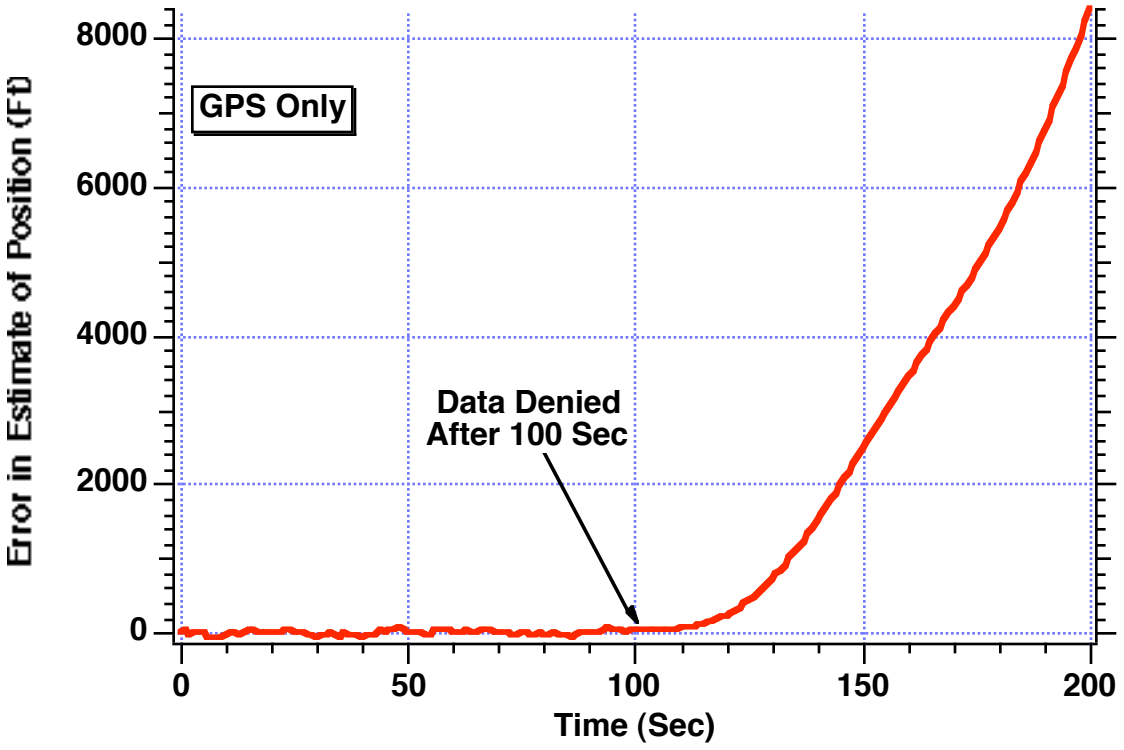
With All the GPS Data We are Able to Follow the Vehicles' Sinusoidal Motion



We are Able to Locate Vehicle to Within 30 Ft When We Continually Get GPS Measurements



Filter is Unable to Track Vehicle After GPS Data is Lost (XLOSE=100 in Simulation)



Combine GPS and INS Data

GPS measurement is corrupted by noise

$$z_{GPS} = x + v_{GPS}$$

INS measurement is corrupted by bias

$$z_{INS} = x + .5bt^2$$

Form a difference measurement

$$\Delta z = z_{INS} - z_{GPS} = x + .5bt^2 - x - v_{GPS} = .5bt^2 - v_{GPS}$$

Build one-state filter to estimate bias

$\dot{b} = u_s$ ← **State equation**

$\mathbf{\Phi_k} = 1 \rightarrow \mathbf{Q_k} = \Phi_s T_s$

Measurement matrix is scalar

$\Delta z = .5bt^2 - v_{GPS} \rightarrow \mathbf{H} = .5t^2 \rightarrow \mathbf{R_k} = \sigma^2_{GPS}$

Linear Kalman filter

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

$$\hat{b}_k = \hat{b}_{k-1} + K_k(\Delta z_k - .5\hat{b}_{k-1}t^2)$$

MATLAB Kalman Filter for Estimating INS Bias by Using Both GPS and INS Measurement Data-1

```
ORDER =1;
XLOSE=99999999.;
W=.1;
PHIS=10.;
TS=1.;
BIASH=0.;
BIAS=1.;
SIGNOISE=50.;
PHI=zeros(ORDER,ORDER);
P=zeros(ORDER,ORDER);
IDNP=eye(ORDER);
Q=zeros(ORDER,ORDER);
P(1,1)=99999999999999.;
PHI(1,1)=1;
H=zeros(1,ORDER);
R(1,1)=SIGNOISE^2;
PHIT=PHI';
Q(1,1)=PHIS*TS;
count=0;
for T=0:TS:200
        if T>XLOSE
                R(1,1)=99999999999999.;
        end
        H(1,1)=.5*T*T;
        HT=H';
        PHIP=PHI*P;
        PHIPPHIT=PHIP*PHIT;
        M=PHIPPHIT+Q;
        MHT=M*HT;
        HMHT=H*MHT;
        HMHTR(1,1)=HMHT(1,1)+R(1,1);
        HMHTRINV(1,1)=1./HMHTR(1,1);
        K=MHT*HMHTRINV;
        KH=K*H;
        IKH=IDNP-KH;
        P=IKH*M;
```

Actual and initial state estimate

Riccati equations

MATLAB Kalman Filter for Estimating INS Bias by Using Both GPS and INS Measurement Data-2

```
XNOISE=SIGNOISE*randn;
X=100.*T-20.*cos(W*T)/W+20./W;
XD=100.+20.*sin(W*T);
XDD=20.*W*cos(W*T);
XGPSS=X+XNOISE;
XINSS=X+.5*BIAS*T*T;
XS=XINSS-XGPSS;
RES=XS-.5*BIASH*T*T;
BIASH=BIASH+K(1,1)*RES;
SP11=sqrt(P(1,1));
BIASERR=BIAS-BIASH;
XH=XINSS-.5*BIASH*T*T;
SP11P=-SP11;
count=count+1;
ArrayT(count)=T;
ArrayBIAS(count)=BIAS;
ArrayBIASH(count)=BIASH;
ArrayBIASERR(count)=BIASERR;
ArraySP11(count)=SP11;
ArraySP11P(count)=SP11P;
end
figure
plot(ArrayT,ArrayBIAS,ArrayT,ArrayBIASH),grid
xlabel('Time (Sec)')
ylabel('Bias (Ft/Sec^2)')
axis([0 200 0 2])
figure
plot(ArrayT,ArrayBIASERR,ArrayT,ArraySP11,ArrayT,ArraySP11P),grid
xlabel('Time (Sec)')
ylabel('Error in Estimate of Bias (Ft/Sec^2)')
axis([0 200 -.1 .1])
clc
output=[ArrayT',ArrayBIAS',ArrayBIASH',ArrayBIASERR',ArraySP11',ArraySP11P'];
save datfil output  -ascii
disp 'simulation finished'
```

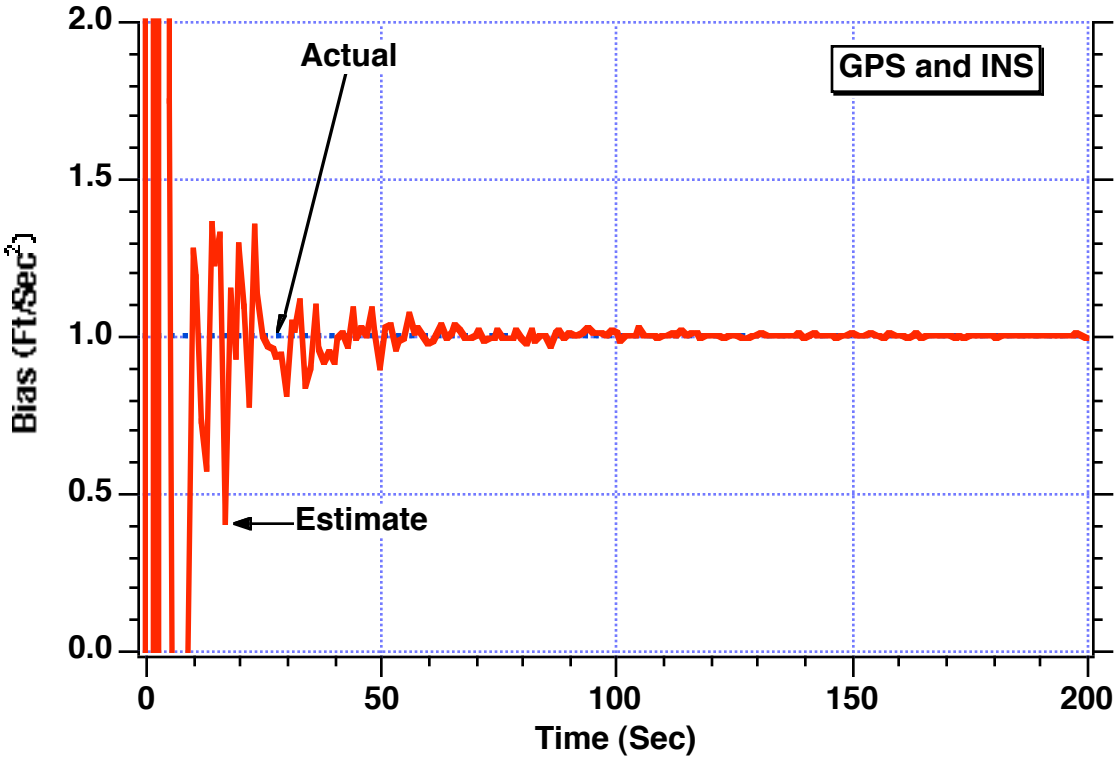
Vehicle position, velocity and acceleration

Actual measurements

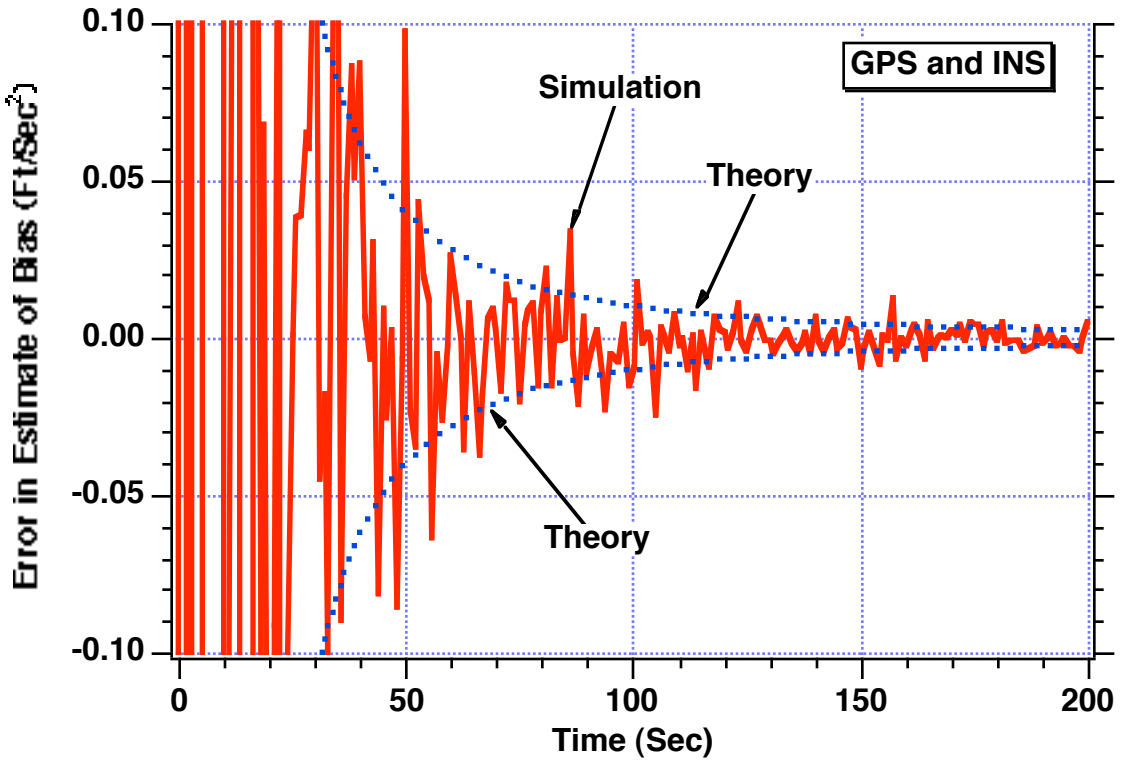
Filter to estimate bias

Estimate vehicle position

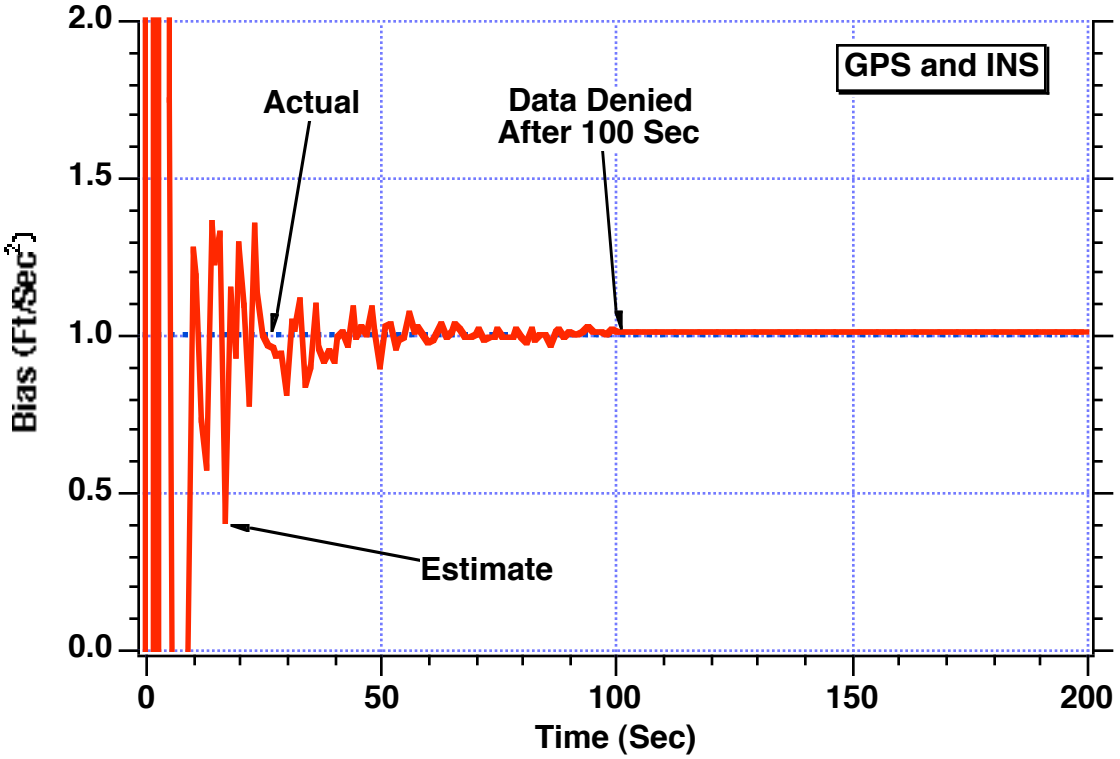
One-State Kalman Filter is Able to Estimate Bias in Less Than 50 Sec



One-State Kalman Filter Results Match Theory



New Kalman Filter is Able to Estimate Bias After GPS Data is Lost



When data denied coast using $\hat{x}_k = x_{k_{INS}} - .5\hat{b}_k t^2$

MATLAB GPS Kalman Filter and New Kalman Filter Running in Parallel-1

```
ORDER =3;
ORDERPZ=1;
XLOSE=100.;
W=.1;
PHIS=1.;
PHISPZ=10.;
TS=1.;
XH=0;
XDH=0;
XDDH=0;
BIASH=0.;
BIAS=1.;
SIGNOISE=50.;
PHI=zeros(ORDER,ORDER);
P=zeros(ORDER,ORDER);
IDNP=eye(ORDER);
Q=zeros(ORDER,ORDER);
IDNPZ(1,1)=1.;
P(1,1)=99999999999999.;
P(2,2)=99999999999999.;
P(3,3)=99999999999999.;
PPZ(1,1)=99999999999999.;
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;
PHIPZ(1,1)=1;
H=zeros(1,ORDER);
H(1,1)=1;
HT=H';
R(1,1)=SIGNOISE^2;
PHIT=PHI';
PHITPZ=PHIPZ';
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);
```

State estimates of both filters

Initial covariance matrices of both filters

Fundamental matrices of both filters

Measurement noise matrix for both filters

MATLAB GPS Kalman Filter and New Kalman Filter Running in Parallel-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;
QPZ(1,1)=PHISPZ*TS;
count=0;
for T=0:TS:200
        if T>XLOSE
                R(1,1)=99999999999999.;
        end
        PHIP=PHI*P;
        PHIPPHIT=PHIP*PHIT;
        M=PHIPPHIT+Q;
        MHT=M*HT;
        HMHT=H*MHT;
        HMHTR(1,1)=HMHT(1,1)+R(1,1);
        HMHTRINV(1,1)=1./HMHTR(1,1);
        K=MHT*HMHTRINV;
        KH=K*H;
        IKH=IDNP-KH;
        P=IKH*M;
        HPZ(1,1)=.5*T*T;
        HTPZ=HPZ';
        PHIPPZ=PHIPZ*PPZ;
        PHIPPHITPZ=PHIPPZ*PHITPZ;
        MPZ=PHIPPHITPZ+QPZ;
        MHTPZ=MPZ*HTPZ;
        HMHTPZ=HPZ*MHTPZ;
        HMHTRPZ(1,1)=HMHTPZ(1,1)+R(1,1);
        HMHTRINVPZ(1,1)=1./HMHTRPZ(1,1);
        KPZ=MHTPZ*HMHTRINVPZ;
        KHPZ=KPZ*HPZ;
        IKHPZ=IDNPZ-KHPZ;
        PPZ=IKHPZ*MPZ;
        XNOISE=SIGNOISE*randn;
        X=100.*T-20.*cos(W*T)/W+20./W;
        XD=100.+20.*sin(W*T);
        XDD=20.*W*cos(W*T);
```

Process noise matrices of both filters

Riccati equations for GPS Kalman filter

Riccati equations for new Kalman filter

Position, velocity and acceleration of vehicle

MATLAB GPS Kalman Filter and New Kalman Filter Running in Parallel-3

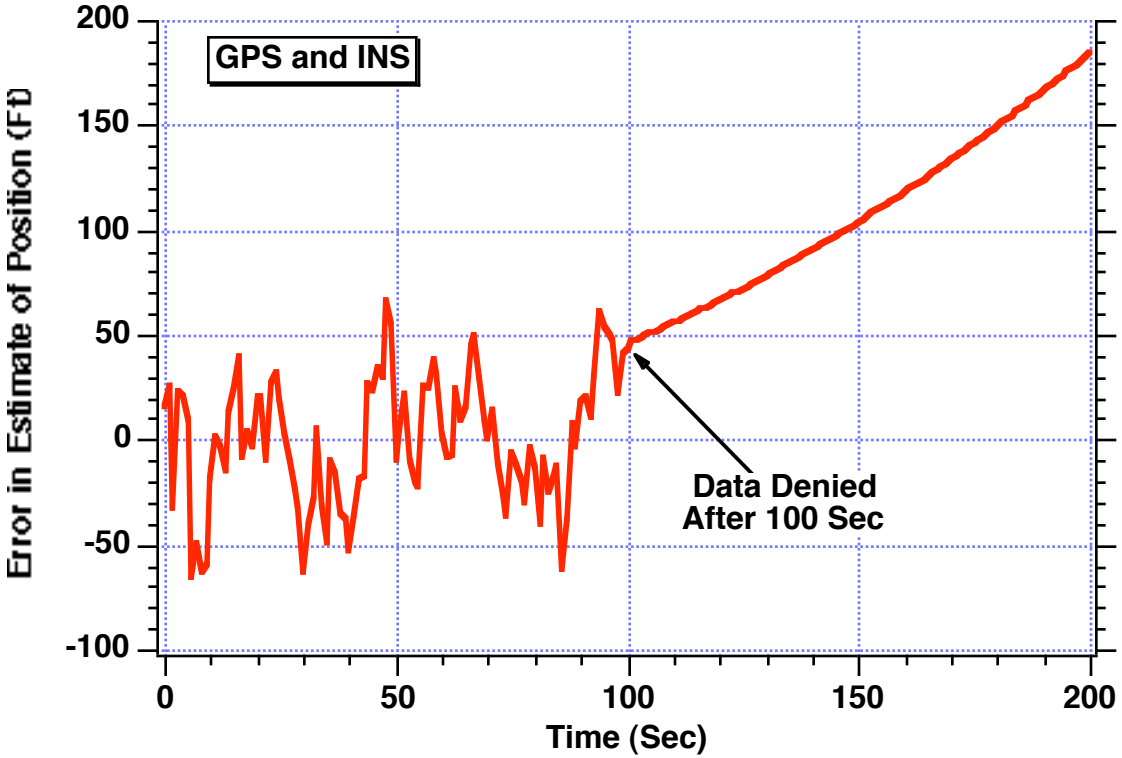
```
XS=X+XNOISE;
RES=XS-XH-TS*XDH-.5*TS*TS*XDDH;
XH=XH+XDH*TS+.5*TS*TS*XDDH+K(1,1)*RES;
XDH=XDH+XDDH*TS+K(2,1)*RES;
XDDH=XDDH+K(3,1)*RES;
XGPSS=XS;
XINSS=X+.5*BIAS*T*T;
XSPZ=XINSS-XGPSS;
RESPZ=XSPZ-.5*BIASH*T*T;
BIASH=BIASH+KPZ(1,1)*RESPZ;
if T>XLOSE
                XH=XINSS-.5*BIASH*T*T;
end
SP11=sqrt(P(1,1));
SP22=sqrt(P(2,2));
SP33=sqrt(P(3,3));
SP11P=-SP11;
SP22P=-SP22;
SP33P=-SP33;
XHERR=X-XH;
XDHERR=XD-XDH;
XDDHERR=XDD-XDDH;
SP11PZ=sqrt(PPZ(1,1));
SP11PZP=-SP11PZ;
BIASERR=BIAS-BIASH;
count=count+1;
ArrayT(count)=T;
ArrayX(count)=X;
ArrayXH(count)=XH;
ArrayXD(count)=XD;
ArrayXDH(count)=XDH;
ArrayXDD(count)=XDD;
ArrayXDDH(count)=XDDH;
end
```

GPS Kalman filter

New Kalman filter

Coasting and combining data when data is lost

Uncertainty in Vehicle Location has Been Dramatically Reduced With Aiding



Miscellaneous Topics
Summary

- Filter performance degrades as signal to noise ratio degrades
  - Filter initialization can be very important
- Lower order filters provide better estimates more rapidly when only a few measurements are available
- Residual and theoretical bound can be used in real world to determine if filter is working properly
- Practical tests were illustrated to see if states were observable
- Two sensors, both of which do not work satisfactorily by themselves, can be combined to work well under adverse circumstances (Aiding)