Tracking a Sine Wave

Tracking a Sine Wave
Overview

- Initial formulation
  - Try and improve deficiencies by adding process noise or reducing measurement noise
- Academic experiment in which a priori information is assumed and filter state is eliminated
- Alternative formulation of extended Kalman filter
- Another formulation

Initial Formulation

Formulation of the Problem

We want to estimate amplitude and frequency of sine wave

$$x = A \sin \omega t$$

given that we have noisy measurements of x

Define a new variable

$$\phi = \omega t$$

If the frequency of the sinusoid is constant

$$\dot{\phi} = \omega$$

$$\dot{\omega} = 0$$

If the amplitude is also constant

$$\dot{A} = 0$$

Matrices for Extended Kalman Filter Formulation

Model of real world in state space form

$$\begin{bmatrix} \dot{\phi} \\ \dot{\omega} \\ \dot{A} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \omega \\ A \end{bmatrix} + \begin{bmatrix} 0 \\ u_{s1} \\ u_{s2} \end{bmatrix}$$

Continuous process noise matrix

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \Phi_{s1} & 0 \\ 0 & 0 & \Phi_{s2} \end{bmatrix}$$

Systems dynamics matrix

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

Finding Fundamental Matrix

Recall

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

Therefore

$$\mathbf{F}^2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Only two terms are needed to find fundamental matrix

$$\mathbf{\Phi} = \mathbf{I} + \mathbf{F}t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} t = \begin{bmatrix} 1 & t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Therefore discrete fundamental matrix given by

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

Finding Measurement Matrix

Since x is not a state we must linearize measurement equation

$$\Delta x^* = \begin{bmatrix} \frac{\partial x}{\partial \phi} & \frac{\partial x}{\partial \omega} & \frac{\partial x}{\partial A} \end{bmatrix} \begin{bmatrix} \Delta\phi \\ \Delta\omega \\ \Delta A \end{bmatrix} + v$$

Recall

$$x = A\sin\omega t = A \sin \phi$$

Therefore

$$\frac{\partial x}{\partial \phi} = A \cos \phi \qquad \frac{\partial x}{\partial \omega} = 0 \qquad \frac{\partial x}{\partial A} = \sin \phi$$

Measurement matrix

$$\mathbf{H} = \begin{bmatrix} A\cos\phi & 0 & \sin\phi \end{bmatrix}$$

Measurement noise matrix

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

Finding Discrete Process Noise Matrix

Recall continuous process noise matrix given by

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \Phi_{s1} & 0 \\ 0 & 0 & \Phi_{s2} \end{bmatrix}$$

Discrete process noise matrix found from

$$\mathbf{Q_k} = \int_0^{T_s} \mathbf{\Phi(\tau) Q \Phi^T(\tau) dt}$$

Substitution yields

$$\mathbf{Q_k} = \int_0^{T_s} \begin{bmatrix} 1 & \tau & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \Phi_{s1} & 0 \\ 0 & 0 & \Phi_{s2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ \tau & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} d\tau \longrightarrow \mathbf{Q_k} = \int_0^{T_s} \begin{bmatrix} \tau^2\Phi_{s1} & \tau\Phi_{s1} & 0 \\ \tau\Phi_{s1} & \Phi_{s1} & 0 \\ 0 & 0 & \Phi_{s2} \end{bmatrix} d\tau$$

Integration yields

$$\mathbf{Q_k} = \begin{bmatrix} \frac{\Phi_{s1} T_s^3}{3} & \frac{\Phi_{s1} T_s^2}{2} & 0 \\ \frac{\Phi_{s1} T_s^2}{2} & \Phi_{s1} T_s & 0 \\ 0 & 0 & \Phi_{s2} T_s \end{bmatrix}$$

Extended Kalman Filtering Equations

Since fundamental matrix is exact, state propagation is exact

$$\begin{bmatrix} \bar{\phi}_k \\ \bar{\omega}_k \\ \bar{A}_k \end{bmatrix} = \begin{bmatrix} 1 & T_s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\phi}_{k-1} \\ \hat{\omega}_{k-1} \\ \hat{A}_{k-1} \end{bmatrix}$$

Or in scalar form

$$\bar{\phi}_k = \hat{\phi}_{k-1} + \hat{\omega}_{k-1} T_s$$

$$\bar{\omega}_k = \hat{\omega}_{k-1}$$

$$\bar{A}_k = \hat{A}_{k-1}$$

We can use nonlinear measurement equation for residual

$$RES_k = x_k^* - \bar{A}_k \sin\bar{\phi}_k$$

So filtering equations become

$$\hat{\phi}_k = \bar{\phi}_k + K_{1_k} RES_k$$

$$\hat{\omega}_k = \bar{\omega}_k + K_{2_k} RES_k$$

$$\hat{A}_k = \bar{A}_k + K_{3_k} RES_k$$

MATLAB Version of Extended Kalman Filter for Sinusoidal Signal With Unknown Frequency-1

```
A=1.;
W=1.;
TS=.1;
ORDER=3;
PHIS1=0.;
PHIS2=0.;
SIGX=1.;
T=0.;
S=0.;
H=.001;
PHI=zeros(ORDER,ORDER);
P=zeros(ORDER,ORDER);
IDNP=eye(ORDER);
Q=zeros(ORDER,ORDER);
RMAT(1,1)=SIGX^2;
PHIH=0.;
WH=2.;
AH=3.;
P(1,1)=0.;
P(2,2)=(W-WH)^2;
P(3,3)=(A-AH)^2;
XT=0.;
XTD=A*W;
count=0;
while T<=20.
      XTOLD=XT;
      XTDOLD=XTD;
      XTDD=-W*W*XT;
      XT=XT+H*XTD;
      XTD=XTD+H*XTDD;
      T=T+H;
      XTDD=-W*W*XT;
      XT=.5*(XTOLD+XT+H*XTD);
      XTD=.5*(XTDOLD+XTD+H*XTDD);
      S=S+H;
```

Actual initial states (A, W)

Initial filter estimates (PHIH, WH, AH)

Initial covariance matrix (P(1,1), P(2,2), P(3,3))

Integrating second-order differential equation to get sine wave using second-order Runge-Kutta integration

MATLAB Version of Extended Kalman Filter for Sinusoidal Signal With Unknown Frequency-2

```
if S>=(TS-.00001)
        S=0.;
        PHI(1,1)=1.;
        PHI(1,2)=TS;
        PHI(2,2)=1.;
        PHI(3,3)=1.;
        Q(1,1)=TS*TS*TS*PHIS1/3.;
        Q(1,2)=.5*TS*TS*PHIS1;
        Q(2,1)=Q(1,2);
        Q(2,2)=PHIS1*TS;
        Q(3,3)=PHIS2*TS;
        PHIB=PHIH+WH*TS;
        HMAT(1,1)=AH*cos(PHIB);
        HMAT(1,2)=0.;
        HMAT(1,3)=sin(PHIB);
        PHIT=PHI';
        HT=HMAT';
        PHIP=PHI*P;
        PHIPPHIT=PHIP*PHIT;
        M=PHIPPHIT+Q;
        HM=HMAT*M;
        HMHT=HM*HT;
        HMHTR=HMHT+RMAT;
        HMHTRINV(1,1)=1./HMHTR(1,1);
        MHT=M*HT;
        K=MHT*HMHTRINV;
        KH=K*HMAT;
        IKH=IDNP-KH;
        P=IKH*M;
        XTNOISE=SIGX*randn;
        XTMEAS=XT+XTNOISE;
        RES=XTMEAS-AH*sin(PHIB);
        PHIH=PHIB+K(1,1)*RES;
        WH=WH+K(2,1)*RES;
        AH=AH+K(3,1)*RES;
```

Fundamental and process noise matrices (PHI and Q lines)

Linearized measurement matrix (PHIB and HMAT lines)

Riccati equations (PHIP through P=IKH*M)

Extended Kalman filter (XTMEAS through AH lines)

MATLAB Version of Extended Kalman Filter for Sinusoidal Signal With Unknown Frequency-3

```
ERRPHI=W*T-PHIH;
SP11=sqrt(P(1,1));
ERRW=W-WH;
SP22=sqrt(P(2,2));
ERRA=A-AH;
SP33=sqrt(P(3,3));
```

Compute actual and theoretical errors in the estimates

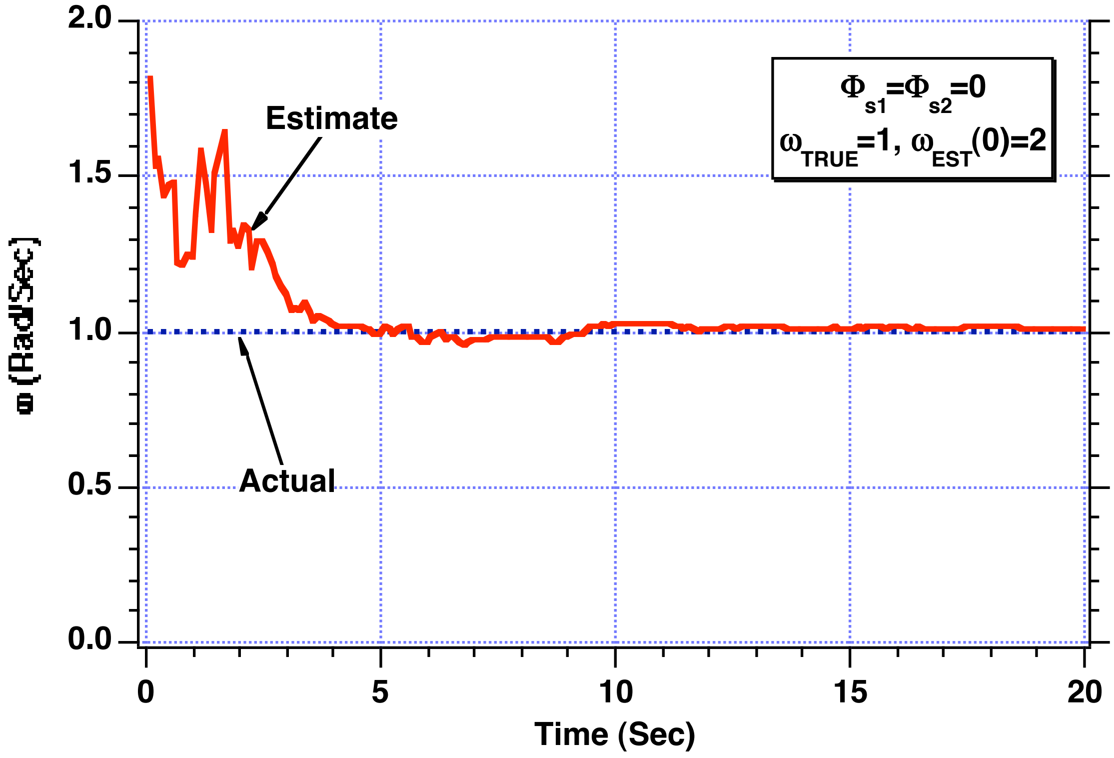
```
XTH=AH*sin(PHIH);
XTDH=AH*WH*cos(PHIH);
SP11P=-SP11;
SP22P=-SP22;
SP33P=-SP33;
count=count+1;
ArrayT(count)=T;
ArrayW(count)=W;
ArrayWH(count)=WH;
ArrayA(count)=A;
ArrayAH(count)=AH;
ArrayERRPHI(count)=ERRPHI;
ArraySP11(count)=SP11;
ArraySP11P(count)=SP11P;
ArrayERRW(count)=ERRW;
ArraySP22(count)=SP22;
ArraySP22P(count)=SP22P;
ArrayERRA(count)=ERRA;
ArraySP33(count)=SP33;
ArraySP33P(count)=SP33P;
end
```

Save some data as arrays for plotting and writing to files

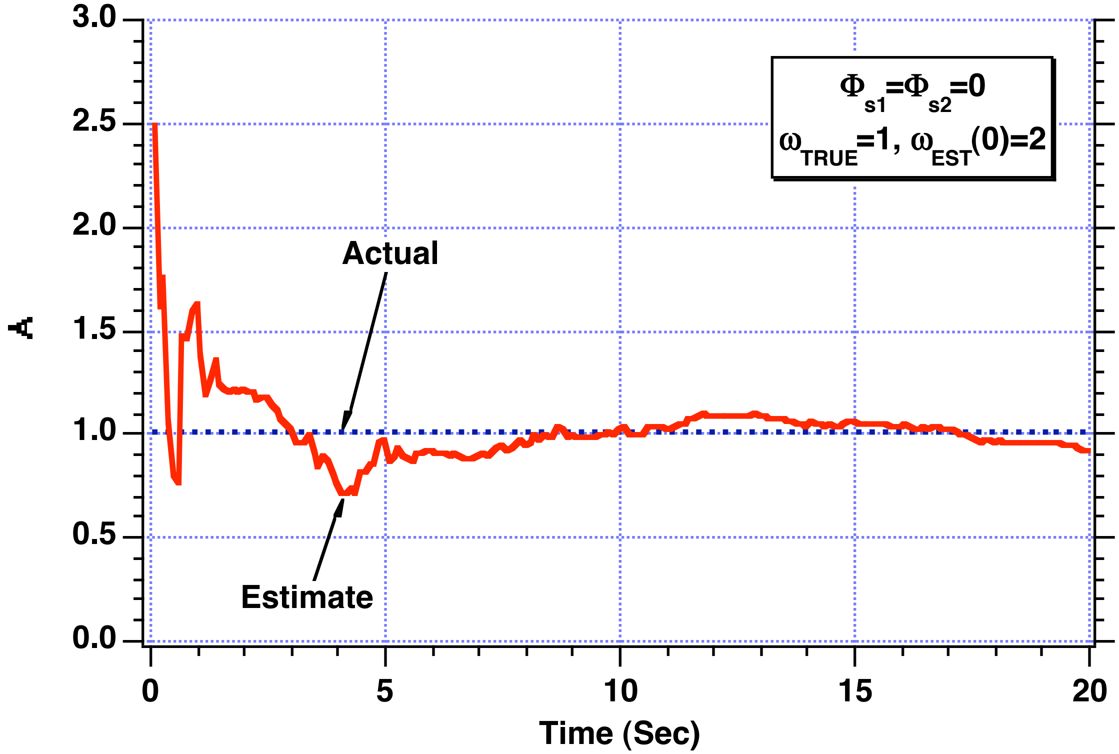
```
end
figure
plot(ArrayT,ArrayW,ArrayT,ArrayWH),grid
xlabel('Time (Sec)')
ylabel('Frequency (R/S)')
axis([0 20 0 2])
figure
plot(ArrayT,ArrayA,ArrayT,ArrayAH),grid
xlabel('Time (Sec)')
ylabel('Amplitude')
axis([0 20 0 3])
```

Plot some states and their estimates

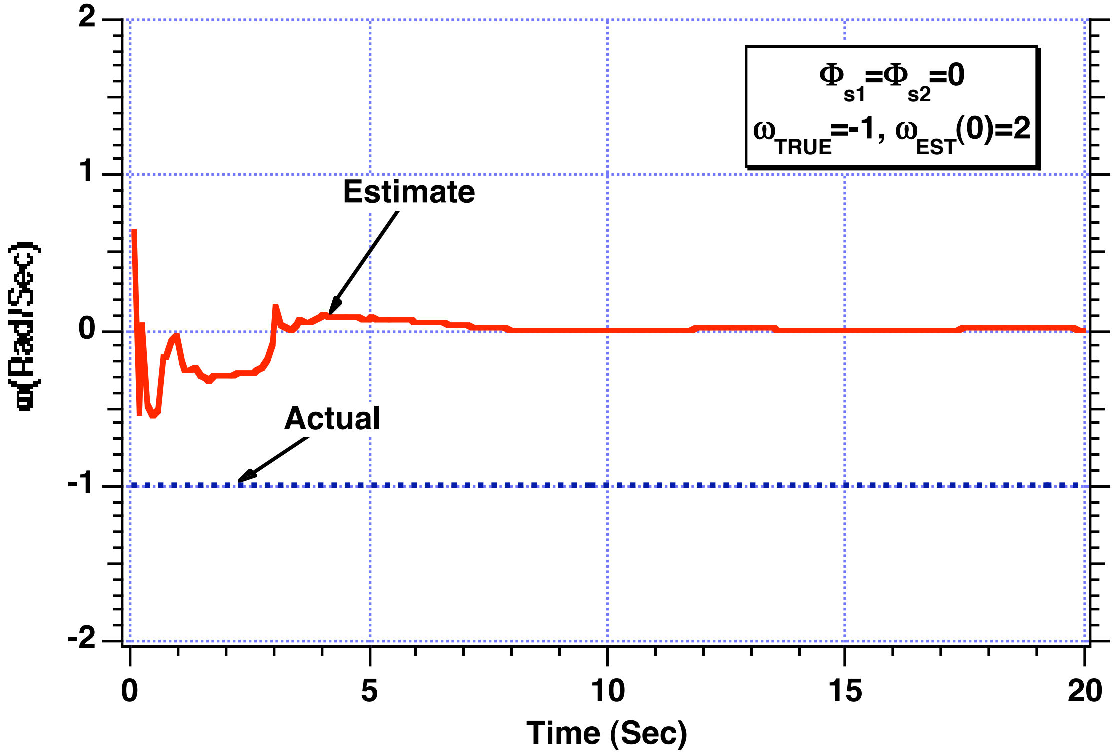
Extended Kalman Filter is Able to Estimate Positive Frequency When Initial Frequency Estimate is also Positive



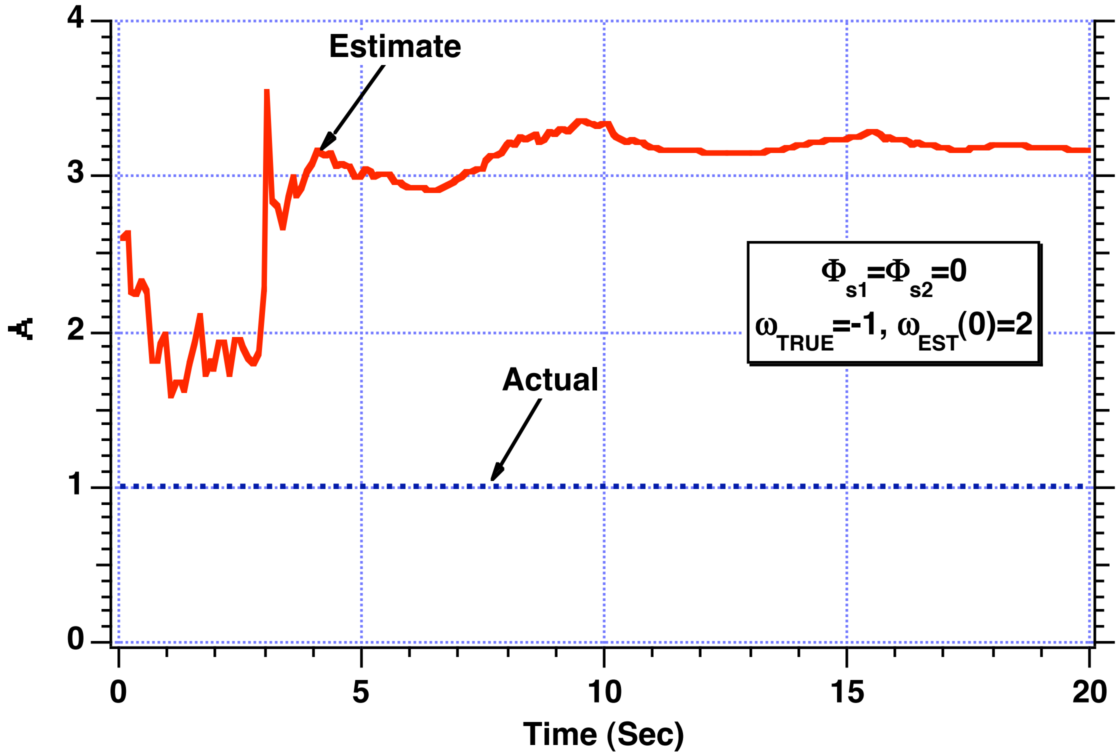
Extended Kalman Filter is Able to Estimate Amplitude When Actual Frequency is Positive and Initial Frequency Estimate is also Positive



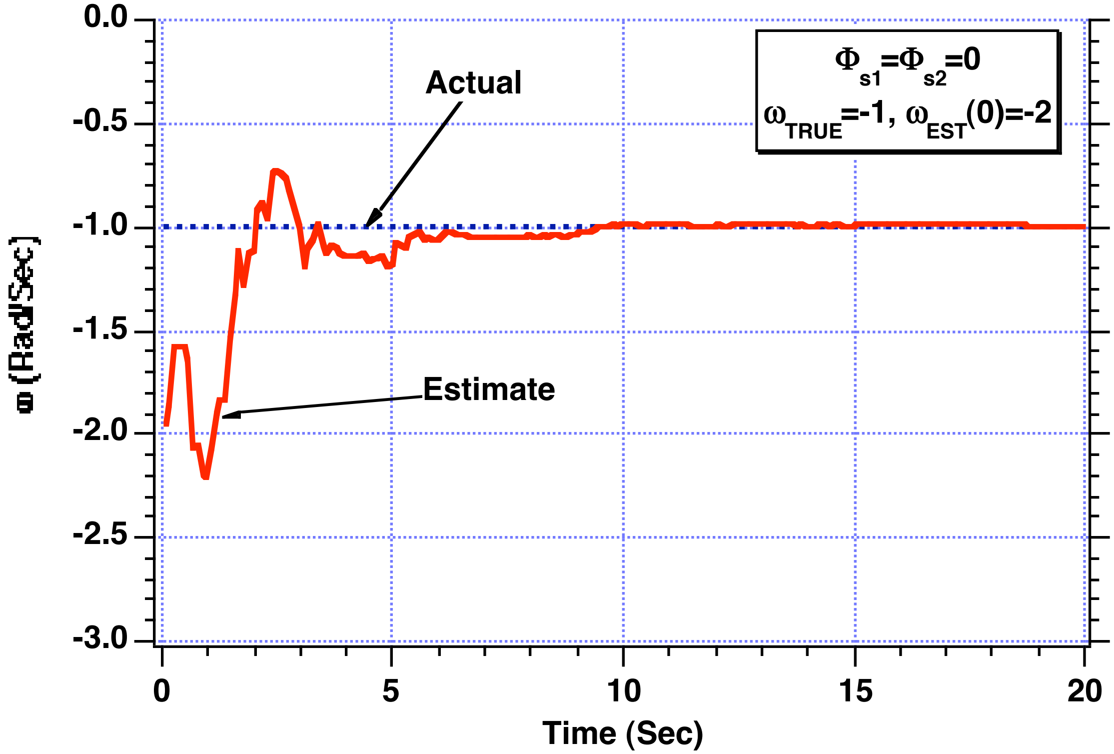
Extended Kalman Filter is Unable to Estimate Negative Frequency When Initial Frequency Estimate is Positive



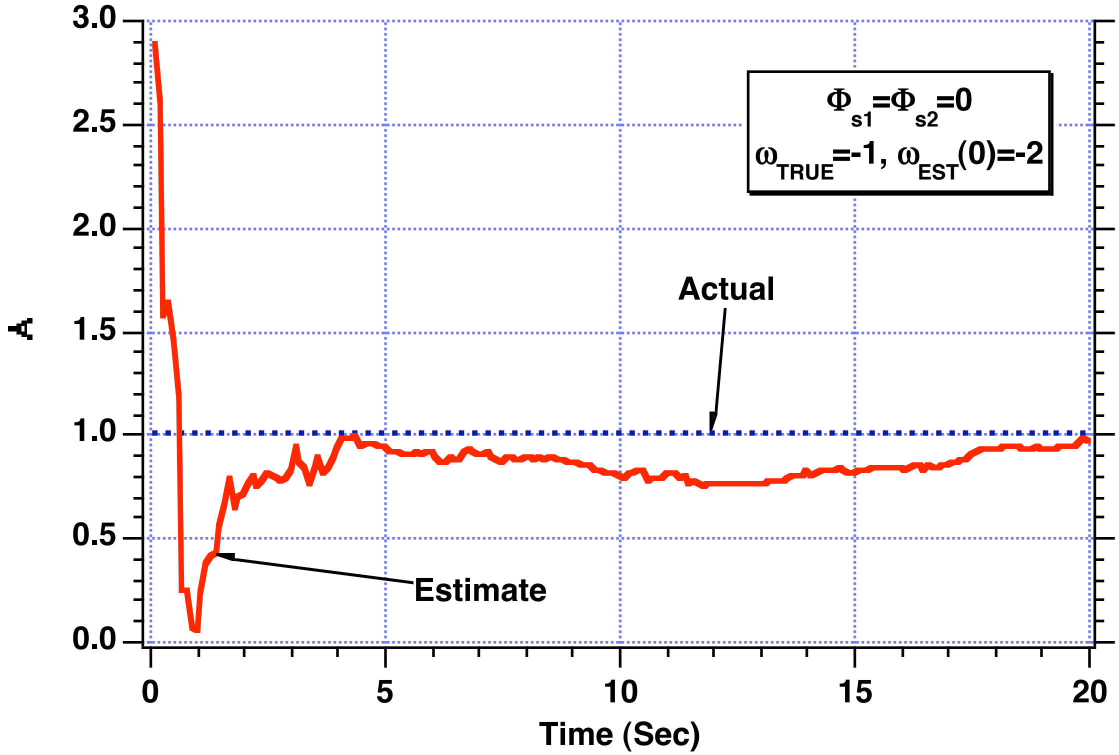
Extended Kalman Filter is Unable to Estimate Amplitude When Actual Frequency is Negative and Initial Frequency Estimate is Positive



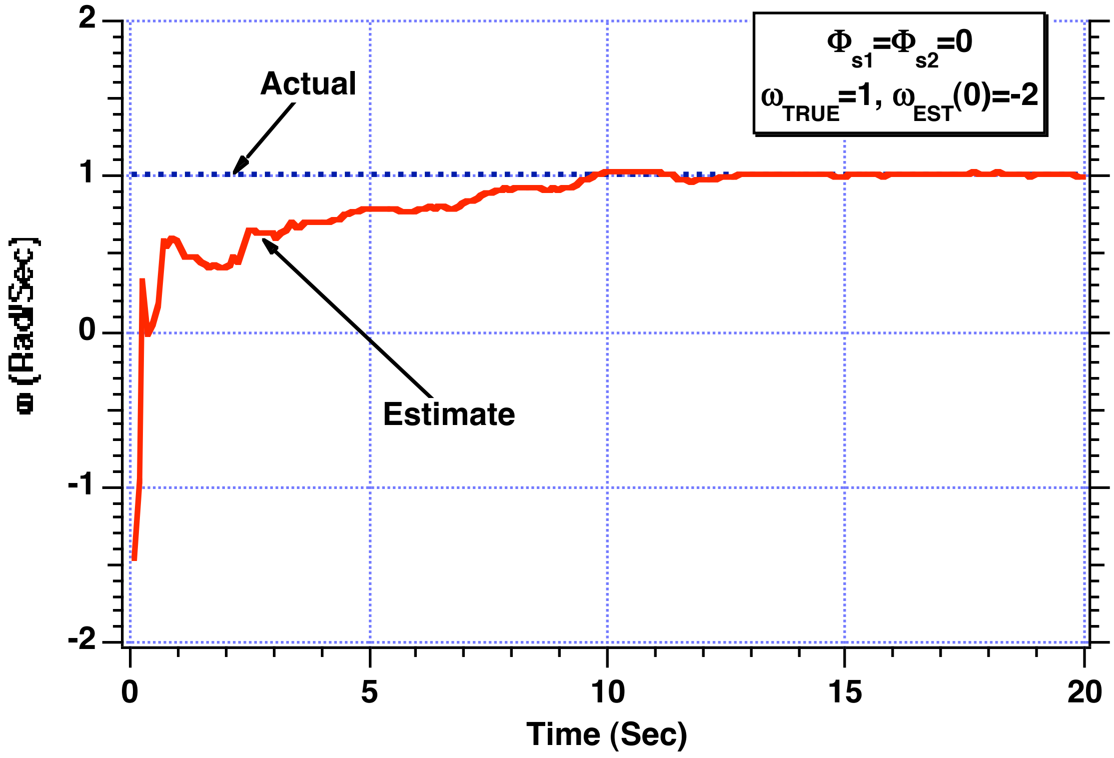
Extended Kalman Filter is Now Able to Estimate Negative Frequency When Initial Frequency Estimate is also Negative



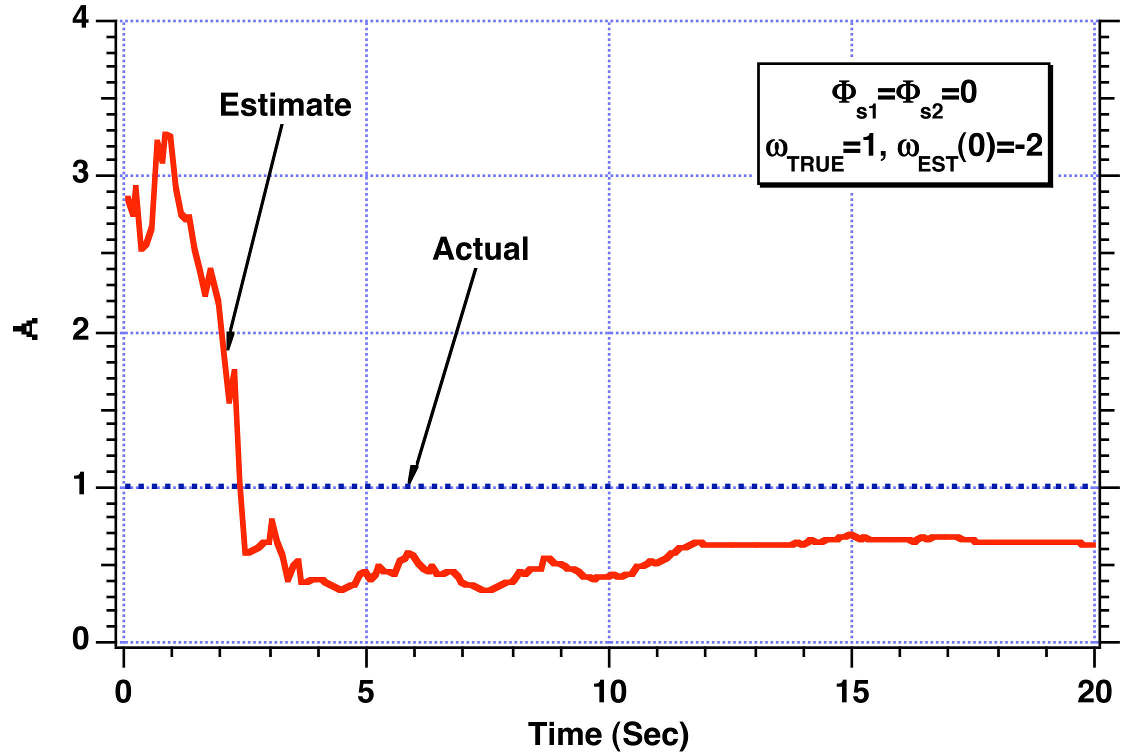
Extended Kalman Filter is Now Able to Estimate Amplitude When Actual Frequency is Negative and Initial Frequency Estimate is also Negative



Extended Kalman Filter is Able to Estimate Positive Frequency When Initial Frequency Estimate is Negative



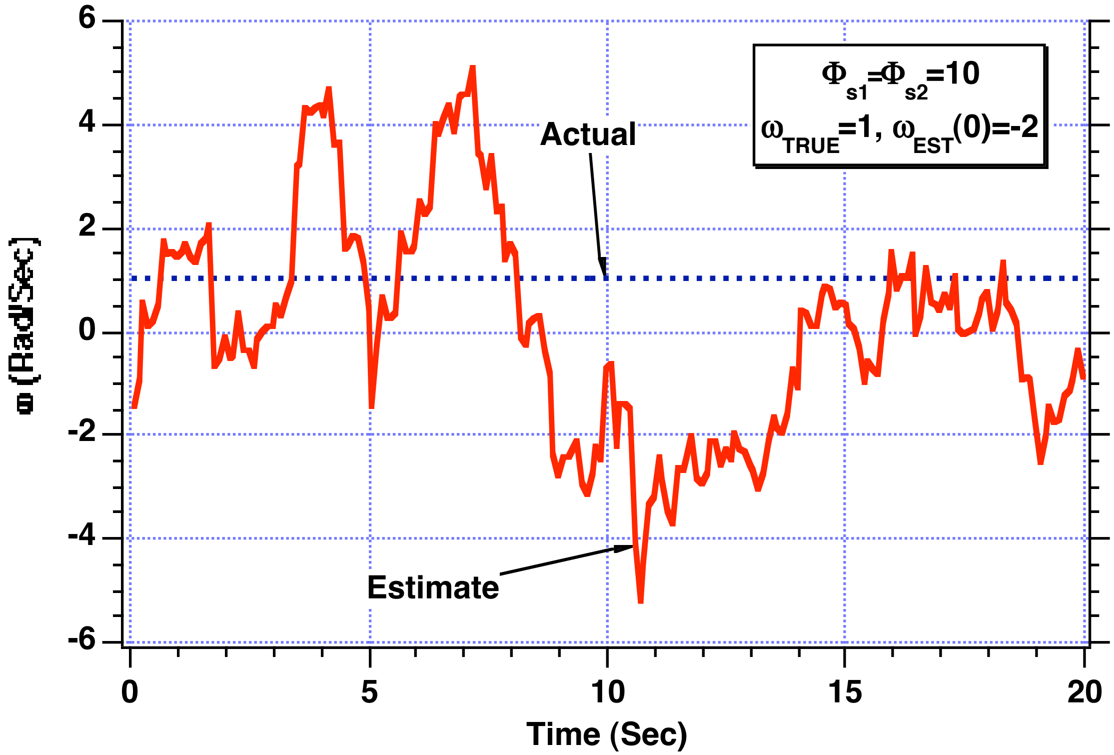
Extended Kalman Filter is Not Able to Estimate Amplitude When Actual Frequency is Positive and Initial Frequency Estimate is Negative



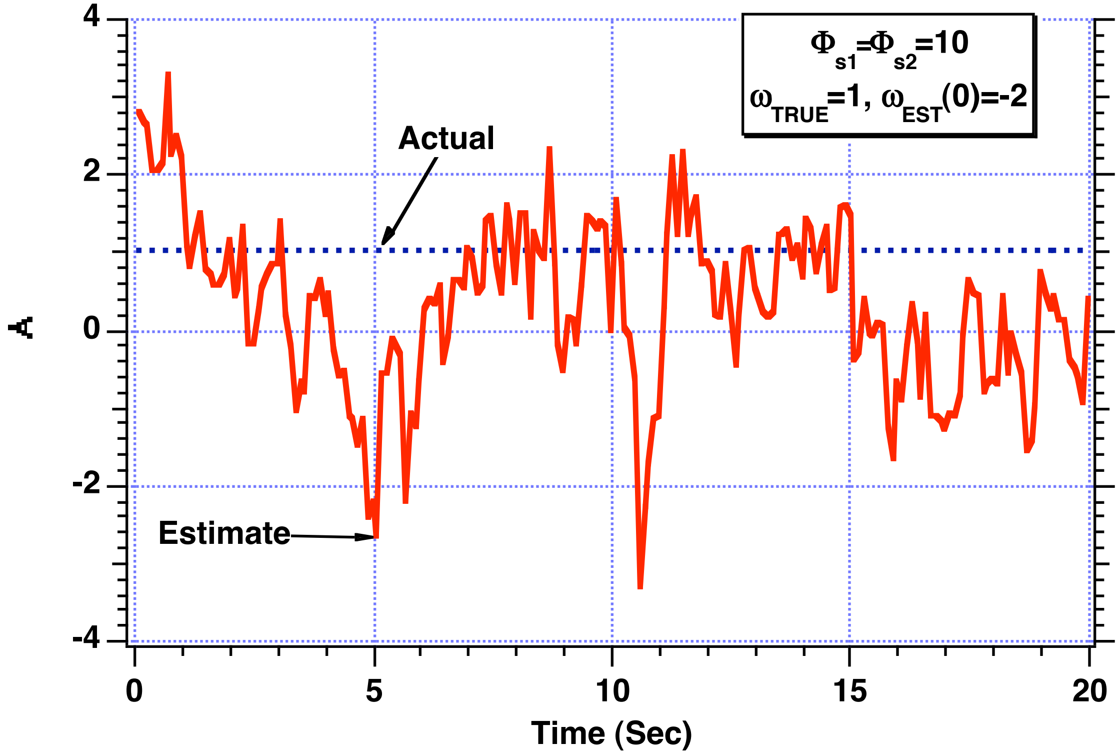
Thoughts

- It appears that extended Kalman filter only works when the sign of the initial frequency estimate matches the actual frequency
- Perhaps we should add process noise because that helped in the past
- Perhaps there is too much measurement noise

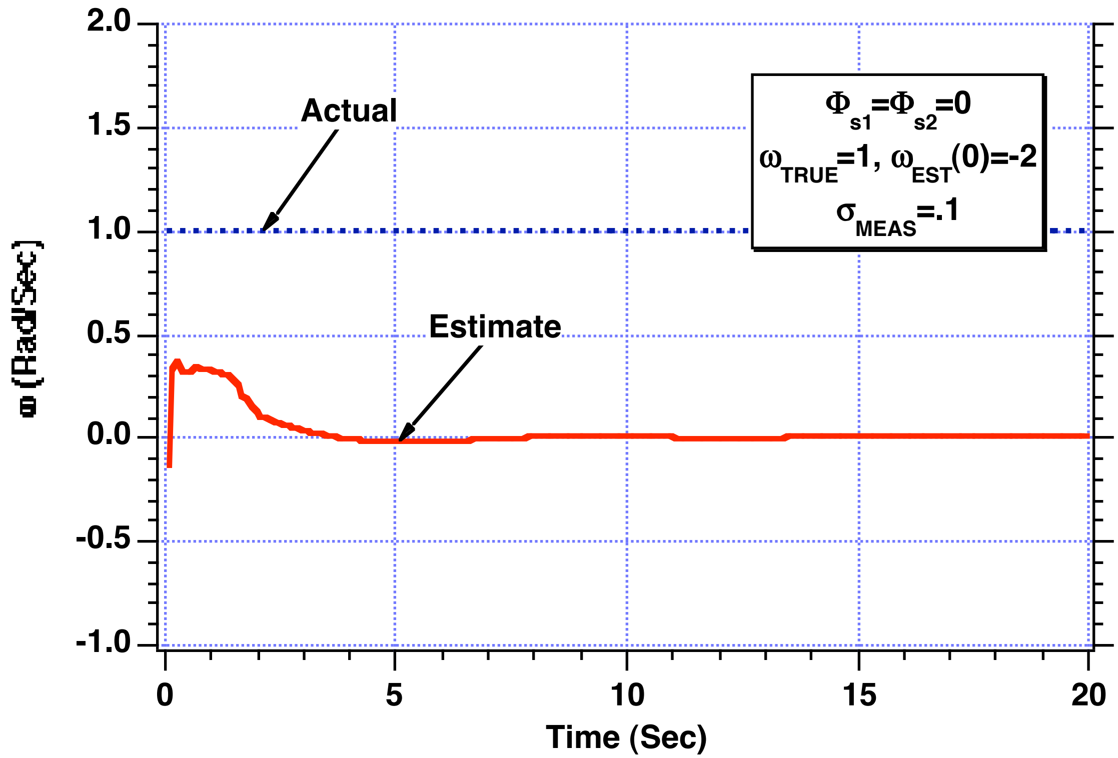
The Addition of Process Noise is Not the Engineering Fix to Enable Filter to Estimate Frequency



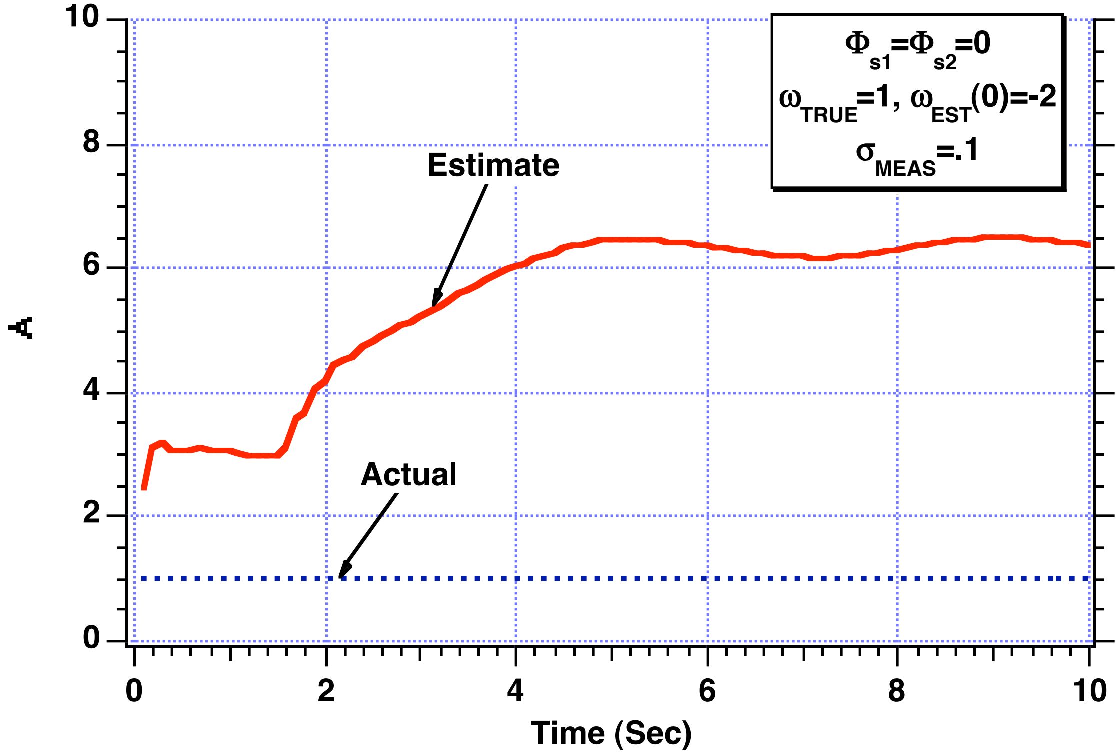
The Addition of Process Noise is Not the Engineering Fix to Enable Filter to Estimate Amplitude



Reducing Measurement Noise by an Order of Magnitude Does Not Yield Accurate Frequency Estimate



Reducing Measurement Noise by an Order of Magnitude Does Not Yield Accurate Amplitude Estimate



Two State Extended Kalman Filter With A Priori Information

New Problem Setup - Academic Experiment-1

It appears we can't estimate both frequency and amplitude

$$x = A\sin\omega t$$

If we know amplitude, model of real world becomes

$$\begin{bmatrix} \dot{\phi} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ u_s \end{bmatrix}$$

Continuous process noise matrix

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 \\ 0 & \Phi_s \end{bmatrix}$$

Systems dynamics matrix

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

Since F squared is zero

$$\boldsymbol{\Phi} = \mathbf{I} + \mathbf{F}t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} t = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}$$

New Problem Setup - Academic Experiment-2

Discrete fundamental matrix

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

Linearized measurement equation

$$\Delta x^* = \begin{bmatrix} \frac{\partial x}{\partial \phi} & \frac{\partial x}{\partial \omega} \end{bmatrix} \begin{bmatrix} \Delta\phi \\ \Delta\omega \end{bmatrix} + v$$

Partial derivatives can be evaluated as

$$x = A\sin\omega t = A \sin\phi \longrightarrow \quad \frac{\partial x}{\partial \phi} = A \cos\phi, \quad \frac{\partial x}{\partial \omega} = 0$$

Linearized measurement matrix

$$\mathbf{H} = \begin{bmatrix} A\cos\phi & 0 \end{bmatrix}$$

Measurement noise matrix is a scalar

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

Finding Discrete Process Noise Matrix

Recall

$$Q_k = \int_0^{T_s} \Phi(\tau) Q \Phi^T(\tau) dt \qquad \text{where} \qquad Q = \begin{bmatrix} 0 & 0 \\ 0 & \Phi_s \end{bmatrix}$$

Substitution yields

$$Q_k = \int_0^{T_s} \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \Phi_s \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \tau & 1 \end{bmatrix} d\tau$$

$$Q_k = \int_0^{T_s} \begin{bmatrix} \tau^2 \Phi_s & \tau \Phi_s \\ \tau \Phi_s & \Phi_s \end{bmatrix} d\tau$$

After integration we get

$$Q_k = \begin{bmatrix} \frac{\Phi_s T_s^3}{3} & \frac{\Phi_s T_s^2}{2} \\ \frac{\Phi_s T_s^2}{2} & \Phi_s T_s \end{bmatrix}$$

New Extended Kalman Filtering Equations

State propagation is exact

$$\begin{bmatrix} \bar{\phi}_k \\ \bar{\omega}_k \end{bmatrix} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\phi}_{k-1} \\ \hat{\omega}_{k-1} \end{bmatrix}$$

Multiplying out terms yields

$$\bar{\phi}_k = \hat{\phi}_{k-1} + \hat{\omega}_{k-1} T_s$$

$$\bar{\omega}_k = \hat{\omega}_{k-1}$$

Filtering equations

$$RES_k = x_k^* - A\sin\bar{\phi}_k$$

$$\hat{\phi}_k = \bar{\phi}_k + K_{1_k} RES_k$$

$$\hat{\omega}_k = \bar{\omega}_k + K_{2_k} RES_k$$

MATLAB Version of 2-State Extended Kalman Filter for Sinusoidal Signal With Unknown Frequency-1

```
TS=.1;
A=1.;
W=1.;
PHIS=0.;
SIGX=1.;
ORDER=2;
T=0.;
S=0.;
H=.001;
PHI=zeros(ORDER,ORDER);
P=zeros(ORDER,ORDER);
IDNP=eye(ORDER);
Q=zeros(ORDER,ORDER);
RMAT(1,1)=SIGX^2;
PHIH=0.;              ] Initial filter state estimates
WH=2.;
P(1,1)=0.^2;          ] Initial covariance matrix
P(2,2)=(W-WH)^2;
XT=0.;
XTD=A*W;
count=0;
while T<=20.
          XTOLD=XT;
          XTDOLD=XTD;
          XTDD=-W*W*XT;
          XT=XT+H*XTD;
          XTD=XTD+H*XTDD;
          T=T+H;
          XTDD=-W*W*XT;
          XT=.5*(XTOLD+XT+H*XTD);
          XTD=.5*(XTDOLD+XTD+H*XTDD);
          S=S+H;
```

Annotation on lines XTOLD=XT; through XTD=.5*(XTDOLD+XTD+H*XTDD); — Integrating second-order differential equation to get sinusoidal signal

MATLAB Version of 2-State Extended Kalman Filter for Sinusoidal Signal With Unknown Frequency-2

```
if S>=(TS-.00001)
        S=0.;
        PHI(1,1)=1.;
        PHI(1,2)=TS;
        PHI(2,2)=1.;
        Q(1,1)=TS*TS*TS*PHIS/3.;
        Q(1,2)=.5*TS*TS*PHIS;
        Q(2,1)=Q(1,2);
        Q(2,2)=PHIS*TS;
        PHIB=PHIH+WH*TS;
        HMAT(1,1)=cos(PHIB);
        HMAT(1,2)=0.;
        PHIT=PHI';
        HT=HMAT';
        PHIP=PHI*P;
        PHIPPHIT=PHIP*PHIT;
        M=PHIPPHIT+Q;
        HM=HMAT*M;
        HMHT=HM*HT;
        HMHTR=HMHT+RMAT;
        HMHTRINV(1,1)=1./HMHTR(1,1);
        MHT=M*HT;
        K=MHT*HMHTRINV;
        KH=K*HMAT;
        IKH=IDNP-KH;
        P=IKH*M;
        XTNOISE=SIGX*randn;
        XTMEAS=XT+XTNOISE;
        RES=XTMEAS-A*sin(PHIB);
        PHIH=PHIB+K(1,1)*RES;
        WH=WH+K(2,1)*RES;
```

Fundamental and process noise matrices (S=0. through Q(2,2)=PHIS*TS;)

Linearized measurement matrix (PHIB through HMAT(1,2))

Riccati equations (HT=HMAT' through P=IKH*M;)

Extended Kalman filter (XTMEAS through WH=WH+K(2,1)*RES;)

MATLAB Version of 2-State Extended Kalman Filter for Sinusoidal Signal With Unknown Frequency-3

```
        PHIREAL=W*T;
        ERRPHI=PHIREAL-PHIH;
        SP11=sqrt(P(1,1));
        ERRW=W-WH;
        SP22=sqrt(P(2,2));
        XTH=A*sin(PHIH);
        XTDH=A*WH*cos(PHIH);
        SP11P=-SP11;
        SP22P=-SP22;
        count=count+1;
        ArrayT(count)=T;
        ArrayW(count)=W;
        ArrayWH(count)=WH;
        ArrayERRPHI(count)=ERRPHI;
        ArraySP11(count)=SP11;
        ArraySP11P(count)=SP11P;
        ArrayERRW(count)=ERRW;
        ArraySP22(count)=SP22;
        ArraySP22P(count)=SP22P;
        end
end
figure
plot(ArrayT,ArrayW,ArrayT,ArrayWH),grid
xlabel('Time (Sec)')
ylabel('Frequency (R/S)')
axis([0 20 0 2])
clc
output=[ArrayT',ArrayW',ArrayWH'];
save datfil output  -ascii
output=[ArrayT',ArrayERRPHI',ArraySP11',ArraySP11P',ArrayERRW',ArraySP22',...
ArraySP22P'];
save covfil output  -ascii
disp 'simulation finished'
```

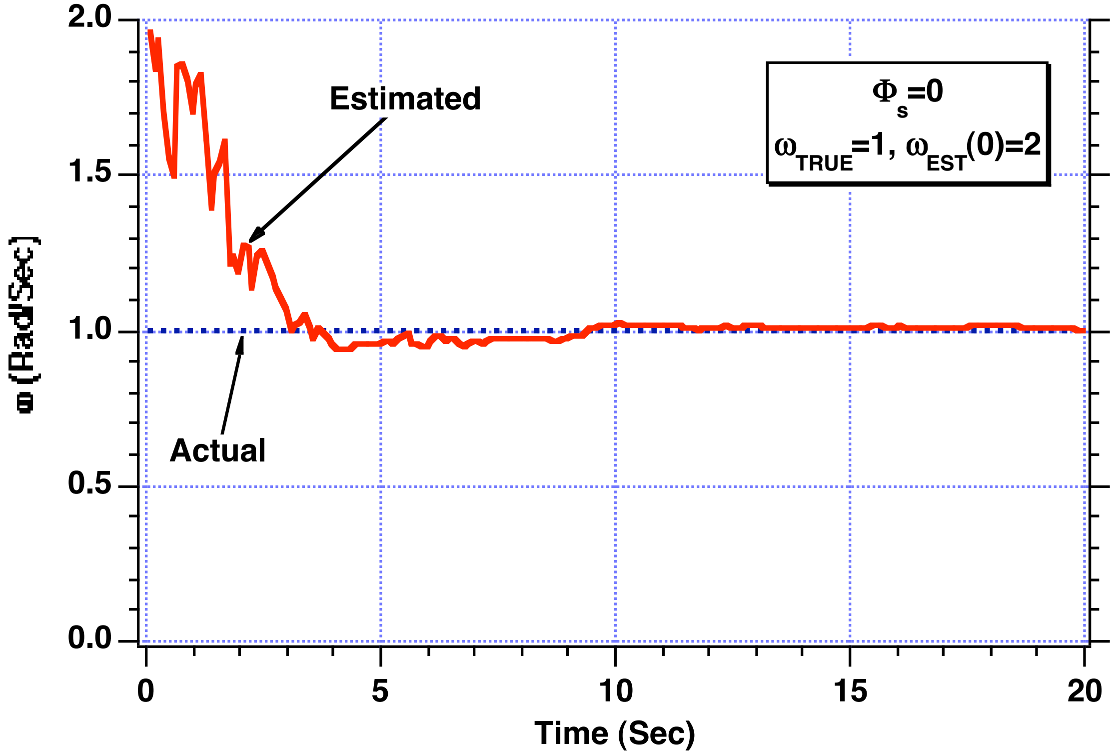
Actual and theoretical errors in estimates (PHIREAL through SP22)

Save data for plotting and writing to files (count through ArraySP22P)

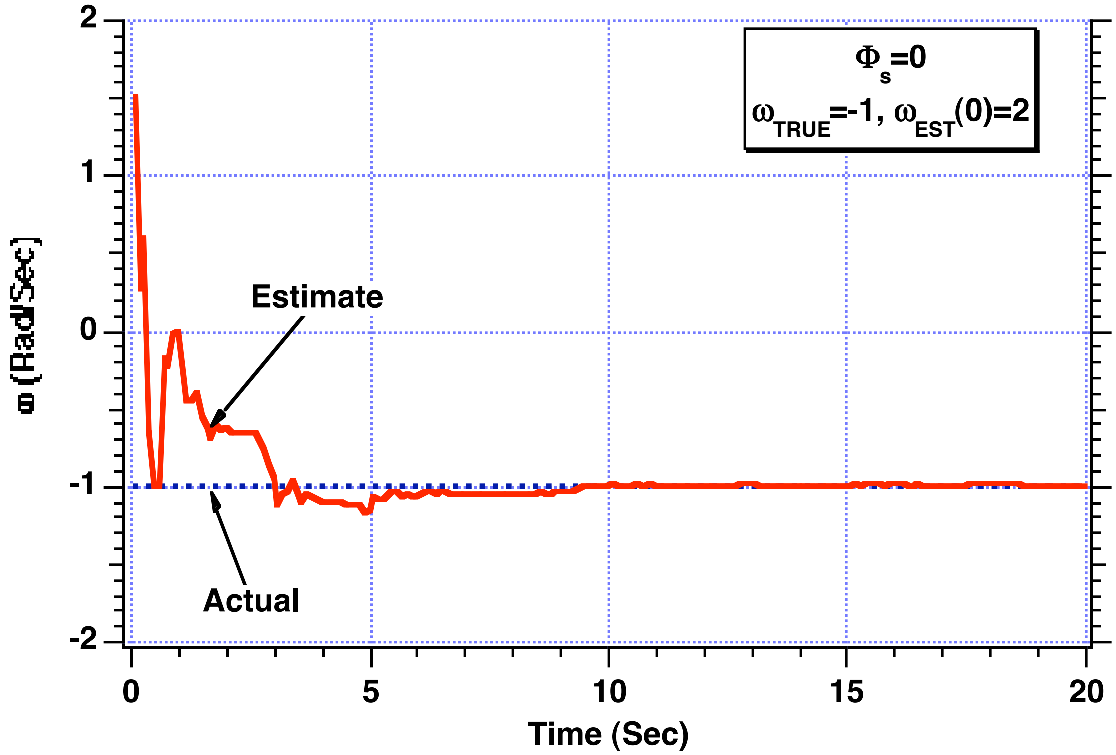
Plot some data (end through ylabel)

Write some data to files (axis through disp)

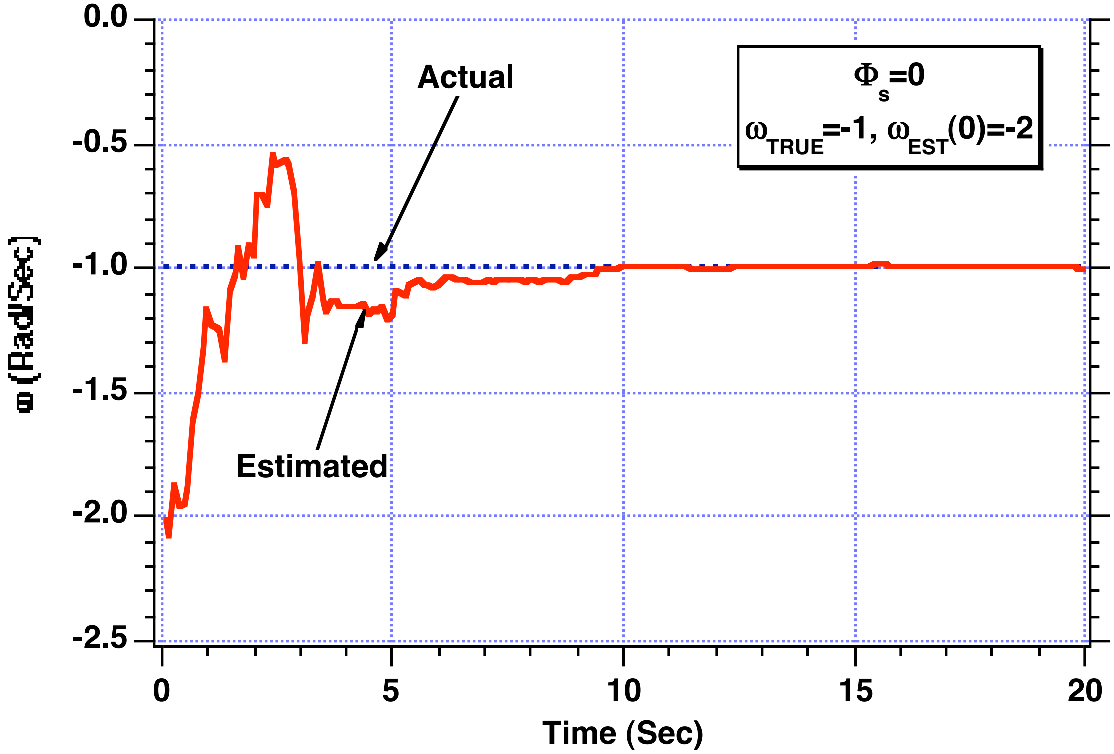
Two-State Extended Kalman Filter Estimates Positive Frequency When Initial Frequency Estimate is Also Positive



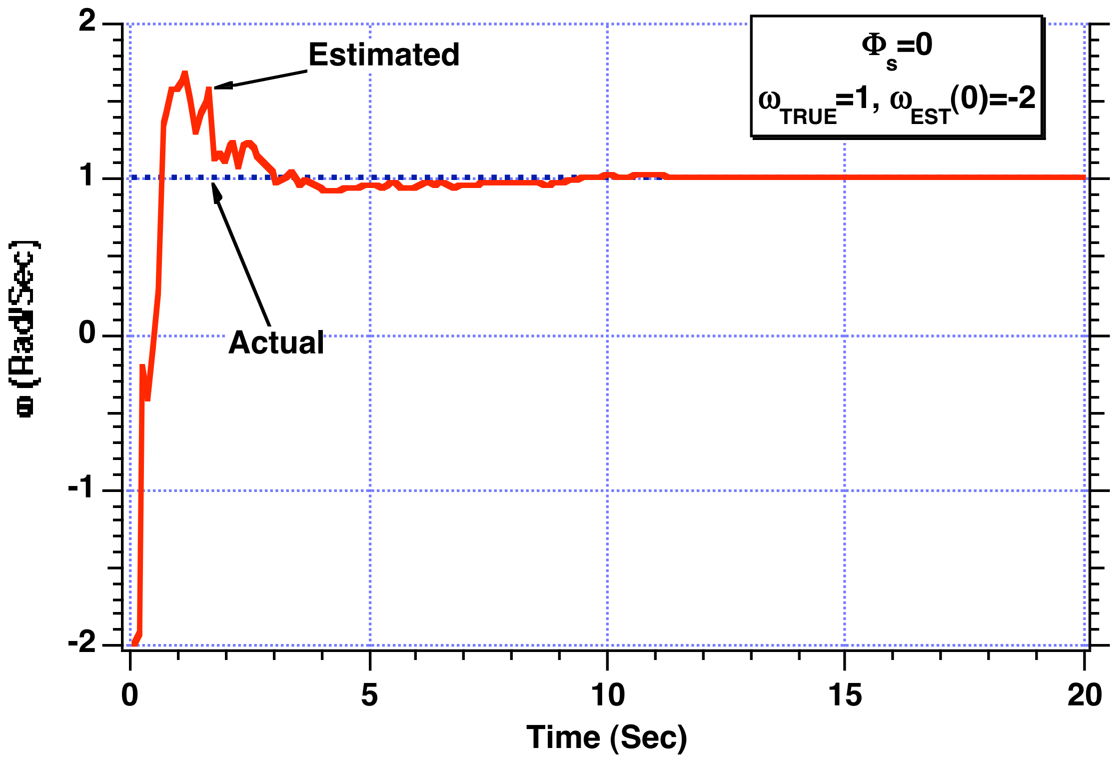
New Two-State Extended Kalman Filter Estimates Negative Frequency When Initialized Positive



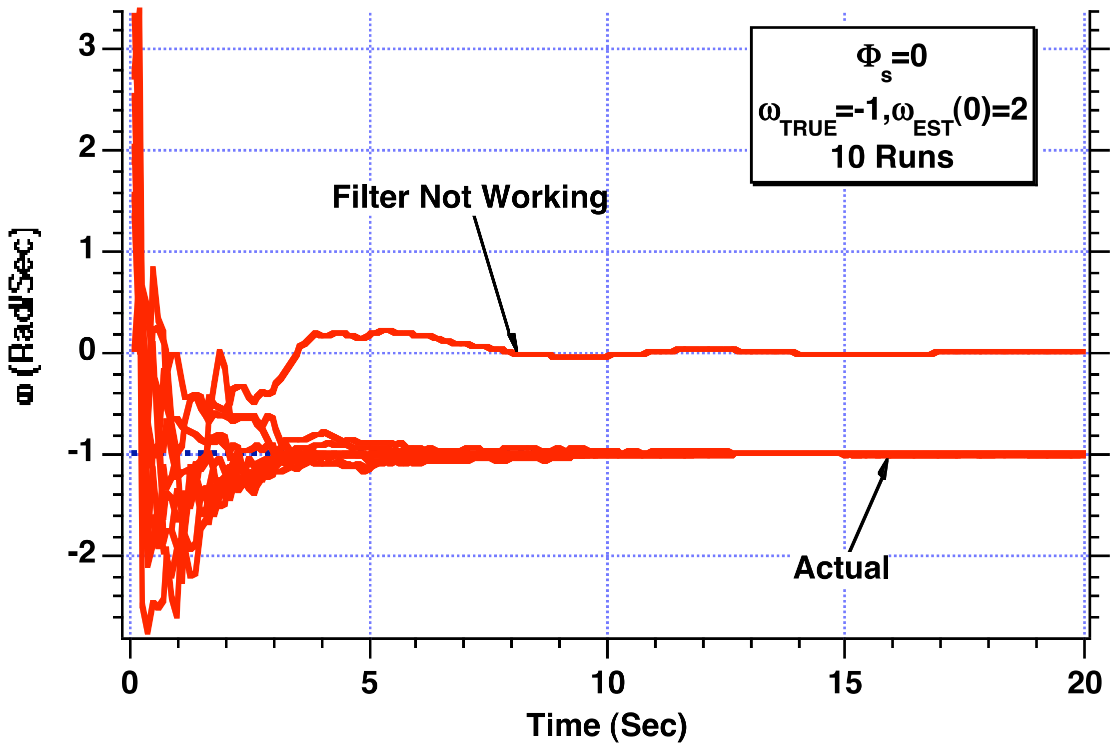
New Two-State Extended Kalman Filter Estimates Negative Frequency When Initialized Negative



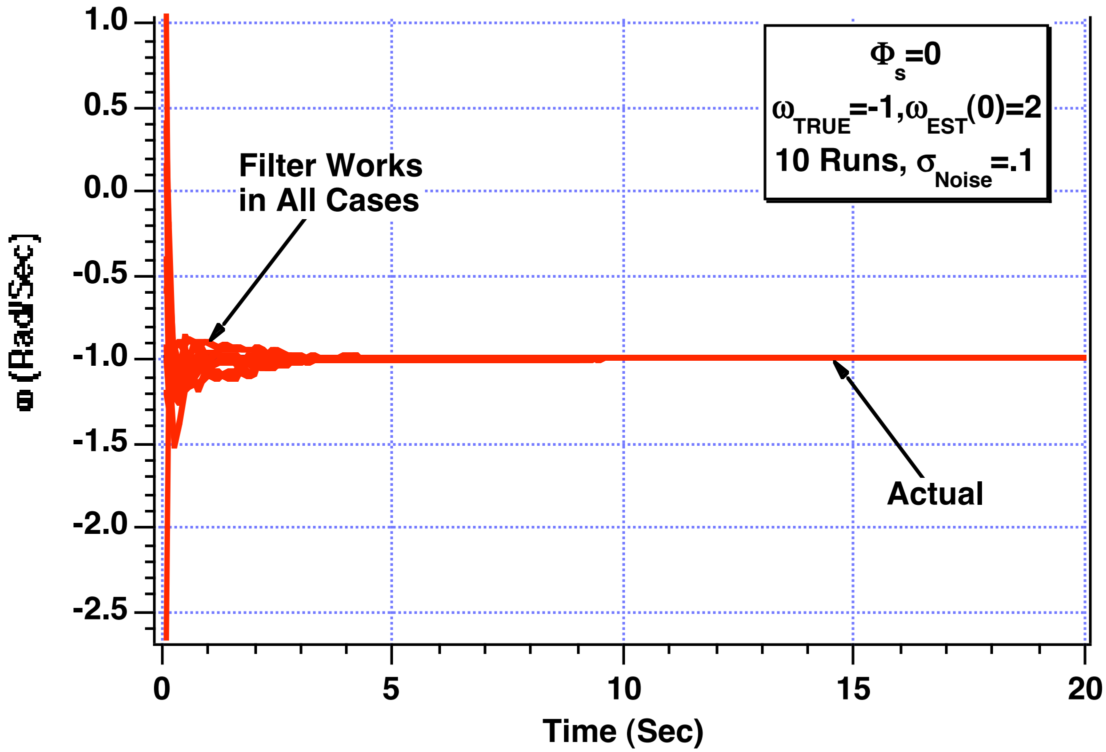
New Two-State Extended Kalman Filter Estimates Positive Frequency When Initialized Negative



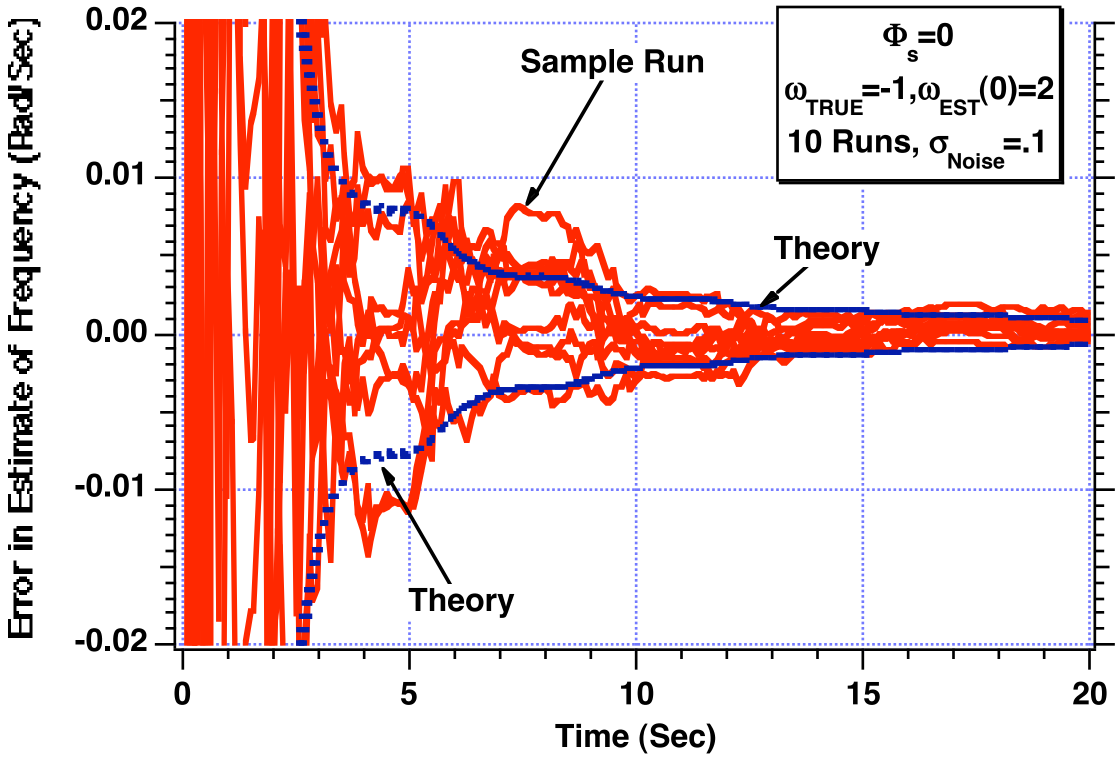
New Two-State Extended Kalman Filter Doesn't Estimate Negative Frequency all the Time When Filter is Initialized Positive



New Two-State Extended Kalman Filter Works <u>All</u> the Time When the Measurement Noise is Reduced by an Order of Magnitude



Error in the Estimate Results Indicate That New Two-State Extended Kalman Filter is Able to Estimate Frequency



Alternate Extended Kalman Filter For Sinusoidal Signal

Alternative Formulation

Recall

$$x = A\sin\omega t$$

Taking the derivative twice

$$\dot{x} = A\omega\cos\omega t$$

$$\ddot{x} = -A\omega^2\sin\omega t$$

Second derivative can be rewritten as

$$\ddot{x} = -\omega^2 x$$

Real world model

$$\ddot{x} = -\omega^2 x$$

$$\dot{\omega} = u_s$$

Or in state space form

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -\omega^2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ u_s \end{bmatrix}$$

Finding Important Matrices For Alternative Filter-1

Systems dynamics matrix

$$\mathbf{F} = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial \dot{x}} & \frac{\partial \dot{x}}{\partial \omega} \\ \frac{\partial \ddot{x}}{\partial x} & \frac{\partial \ddot{x}}{\partial \dot{x}} & \frac{\partial \ddot{x}}{\partial \omega} \\ \frac{\partial \dot{\omega}}{\partial x} & \frac{\partial \dot{\omega}}{\partial \dot{x}} & \frac{\partial \dot{\omega}}{\partial \omega} \end{bmatrix}$$

After taking partial derivatives we get

$$\mathbf{F} = \begin{bmatrix} 0 & 1 & 0 \\ -\hat{\omega}^2 & 0 & -2\hat{\omega}\hat{x} \\ 0 & 0 & 0 \end{bmatrix}$$

Approximating fundamental matrix with two terms

$$\mathbf{\Phi}(t) \approx I + \mathbf{F}t = \begin{bmatrix} 1 & t & 0 \\ -\hat{\omega}^2 t & 1 & -2\hat{\omega}\hat{x}t \\ 0 & 0 & 1 \end{bmatrix}$$

Finding Important Matrices For Alternative Filter-2

Discrete fundamental matrix

$$\Phi_k \approx \begin{bmatrix} 1 & T_s & 0 \\ -\hat{\omega}_{k-1}^2 T_s & 1 & -2\hat{\omega}_{k-1}\hat{x}_{k-1}T_s \\ 0 & 0 & 1 \end{bmatrix}$$

Fundamental matrix is not exact

Continuous process noise matrix

$$\mathbf{Q} = E(ww^T) = E\left[\begin{bmatrix} 0 \\ 0 \\ u_s \end{bmatrix}\begin{bmatrix} 0 & 0 & u_s \end{bmatrix}\right] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Phi_s \end{bmatrix}$$

Formula for discrete process noise matrix

$$\mathbf{Q_k} = \int_0^{T_s} \mathbf{\Phi(\tau)Q\Phi^T(\tau)dt}$$

Substitution yields

$$\mathbf{Q}_k = \int_0^{T_s} \begin{bmatrix} 1 & \tau & 0 \\ -\hat{\omega}^2\tau & 1 & -2\hat{\omega}\hat{x}\tau \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Phi_s \end{bmatrix}\begin{bmatrix} 1 & -\hat{\omega}^2\tau & 0 \\ \tau & 1 & 0 \\ 0 & -2\hat{\omega}\hat{x}\tau & 1 \end{bmatrix} d\tau$$

Finding Important Matrices For Alternative Filter-3

Multiplication yields

$$\mathbf{Q}_k = \int_0^{T_s} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4\hat{\omega}^2\hat{x}^2\tau^2\Phi_s & -2\hat{\omega}\hat{x}\tau\Phi_s \\ 0 & -2\hat{\omega}\hat{x}\tau\Phi_s & \Phi_s \end{bmatrix} d\tau$$

After integration we obtain

$$\mathbf{Q}_k = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.333\hat{\omega}^2\hat{x}^2T_s^3\Phi_s & -\hat{\omega}\hat{x}T_s^2\Phi_s \\ 0 & -\hat{\omega}\hat{x}T_s^2\Phi_s & T_s\Phi_s \end{bmatrix}$$

Measurement equation is now linear

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

Measurement and measurement noise matrices

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \qquad \mathbf{R}_k = E(\mathbf{v}_k\mathbf{v}_k^T) = \sigma_k^2$$

Filtering Equations For Alternative Filter

$$\hat{x}_k = \bar{x}_k + K_{1_k}(x_k^* - \bar{x}_k)$$

$$\hat{\dot{x}}_k = \bar{\dot{x}}_k + K_{2_k}(x_k^* - \bar{x}_k)$$

$$\hat{\omega}_k = \hat{\omega}_{k-1} + K_{3_k}(x_k^* - \bar{x}_k)$$

***Barred quantities are obtained by numerically integrating nonlinear differential equations. They are <u>not</u> obtained using fundamental matrix**

True BASIC Alternate Extended Kalman Filter-1

```
OPTION NOLET
REM UNSAVE "DATFIL"
REM UNSAVE "COVFIL"
OPEN #1:NAME "DATFIL",ACCESS OUTPUT,CREATE NEW, ORGANIZATION TEXT
OPEN #2:NAME "COVFIL",ACCESS OUTPUT,CREATE NEW, ORGANIZATION TEXT
SET #1: MARGIN 1000
SET #2: MARGIN 1000
DIM P(3,3),Q(3,3),M(3,3),PHI(3,3),HMAT(1,3),HT(3,1),PHIT(3,3)
DIM RMAT(1,1),IDNP(3,3),PHIP(3,3),PHIPPHIT(3,3),HM(1,3)
DIM HMHT(1,1),HMHTR(1,1),HMHTRINV(1,1),MHT(3,1),K(3,1),F(3,3)
DIM KH(3,3),IKH(3,3)
HP=.001
W=1.
A=1.
TS=.1
ORDER=3
PHIS=0.
SIGX=1.
T=0.
S=0.
H=.001
MAT F=ZER(ORDER,ORDER)
MAT PHI=ZER(ORDER,ORDER)
MAT P=ZER(ORDER,ORDER)
MAT IDNP=IDN(ORDER,ORDER)
MAT Q=ZER(ORDER,ORDER)
RMAT(1,1)=SIGX^2
P(1,1)=SIGX^2
P(2,2)=2.^2
P(3,3)=2.^2
XTH=0.
XTDH=0.
WH=2.
XT=0.
XTD=A*W
```

Initial covariance matrix (P(1,1), P(2,2), P(3,3))

Initial filter estimates (XTH, XTDH, WH)

True BASIC Alternate Extended Kalman Filter-2

```
DO WHILE T<=20.
        XTOLD=XT
        XTDOLD=XTD
        XTDD=-W*W*XT
        XT=XT+H*XTD
        XTD=XTD+H*XTDD
        T=T+H
        XTDD=-W*W*XT
        XT=.5*(XTOLD+XT+H*XTD)
        XTD=.5*(XTDOLD+XTD+H*XTDD)
        S=S+H
        IF S>=(TS-.00001) THEN
                S=0.
                F(1,2)=1.
                F(2,1)=-WH^2
                F(2,3)=-2.*WH*XTH
                PHI(1,1)=1.
                PHI(1,2)=TS
                PHI(2,1)=-WH*WH*TS
                PHI(2,2)=1.
                PHI(2,3)=-2.*WH*XTH*TS
                PHI(3,3)=1.
                Q(2,2)=4.*WH*WH*XTH*XTH*TS*TS*TS*PHIS/3.
                Q(2,3)=-2.*WH*XTH*TS*TS*PHIS/2.
                Q(3,2)=Q(2,3)
                Q(3,3)=PHIS*TS
                HMAT(1,1)=1.
                HMAT(1,2)=0.
                HMAT(1,3)=0.
                MAT PHIT=TRN(PHI)
                MAT HT=TRN(HMAT)
                MAT PHIP=PHI*P
                MAT PHIPPHIT=PHIP*PHIT
                MAT M=PHIPPHIT+Q
                MAT HM=HMAT*M
                MAT HMHT=HM*HT
                MAT HMHTR=HMHT+RMAT
                HMHTRINV(1,1)=1./HMHTR(1,1)
                MAT MHT=M*HT
                MAT K=MHT*HMHTRINV
                MAT KH=K*HMAT
                MAT IKH=IDNP-KH
                MAT P=IKH*M
```

Integrate second-order equation representing the real world (XTOLD=XT through XTD=.5*(XTDOLD+XTD+H*XTDD))

Systems dynamics matrix (S=0. through F(2,3)=-2.*WH*XTH)

Fundamental matrix (PHI(1,1)=1. through PHI(3,3)=1.)

Discrete process noise matrix (Q(2,2) through Q(3,3))

Measurement matrix (HMAT(1,1) through HMAT(1,3))

Riccati equations (MAT PHIP=PHI*P through MAT P=IKH*M)

True BASIC Alternate Extended Kalman Filter-3

```
                    CALL GAUSS(XTNOISE,SIGX)
                    XTMEAS=XT+XTNOISE
                    CALL PROJECT(T,TS,XTH,XTDH,XTB,XTDB,HP,WH)
                    RES=XTMEAS-XTB
                    XTH=XTB+K(1,1)*RES
                    XTDH=XTDB+K(2,1)*RES
                    WH=WH+K(3,1)*RES
                    ERRX=XT-XTH
                    SP11=SQR(P(1,1))
                    ERRXD=XTD-XTDH
                    SP22=SQR(P(2,2))
                    ERRW=W-WH
                    SP33=SQR(P(3,3))
                    PRINT T,XT,XTH,XTD,XTDH,W,WH
                    PRINT #1:T,XT,XTH,XTD,XTDH,W,WH
                    PRINT #2:T,ERRX,SP11,-SP11,ERRXD,SP22,-SP22,ERRW,SP33,-SP33
          END IF
LOOP
CLOSE #1
CLOSE #2
END

SUB PROJECT(TP,TS,XTP,XTDP,XTH,XTDH,HP,W)
T=0.
XT=XTP
XTD=XTDP
H=HP
DO WHILE T<=(TS-.0001)
          XTDD=-W*W*XT
          XTD=XTD+H*XTDD
          XT=XT+H*XTD
          T=T+H
LOOP
XTH=XT
XTDH=XTD
END SUB
```

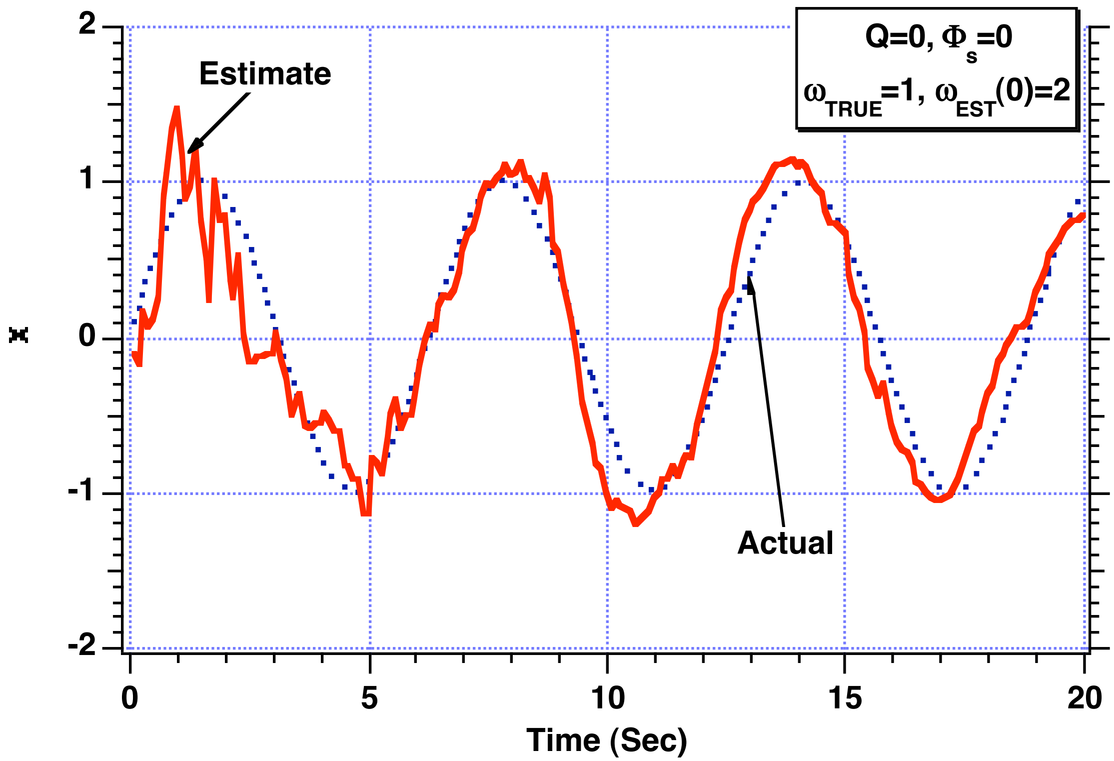
← Project states ahead (CALL PROJECT line)

Filter (RES through WH lines)

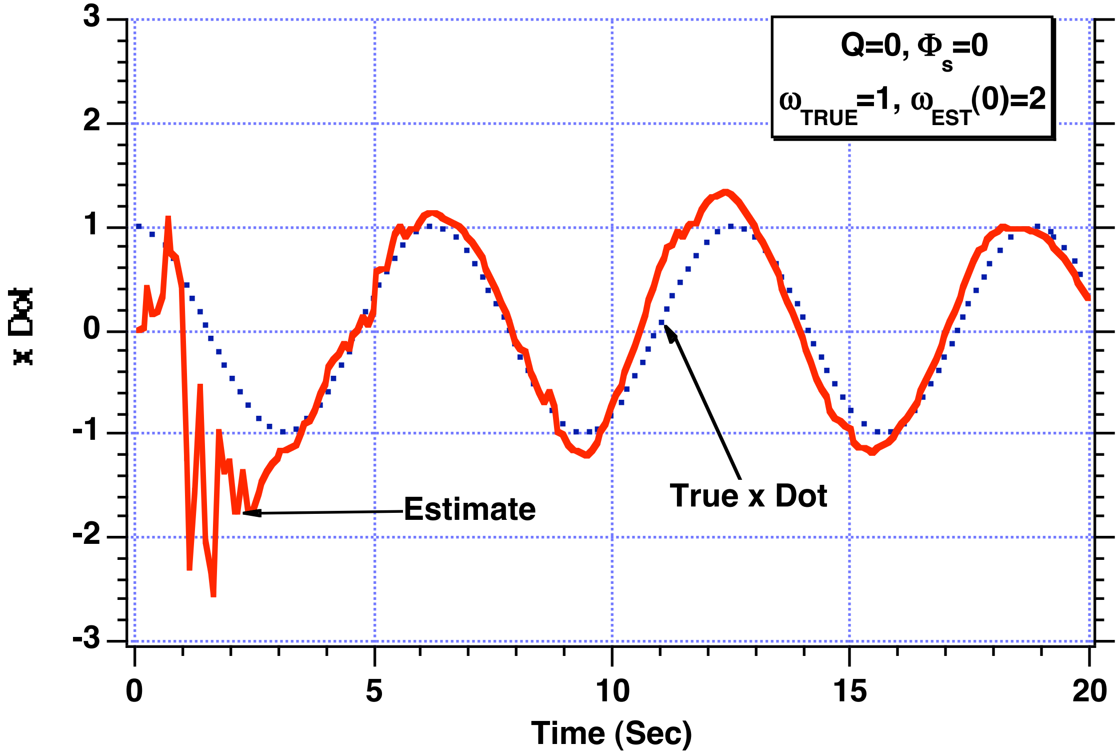
Actual and theoretical errors in estimates (ERRX through SP33 lines)

Subroutine to propagate states ahead one sampling interval using Euler integration (SUB PROJECT through END SUB)

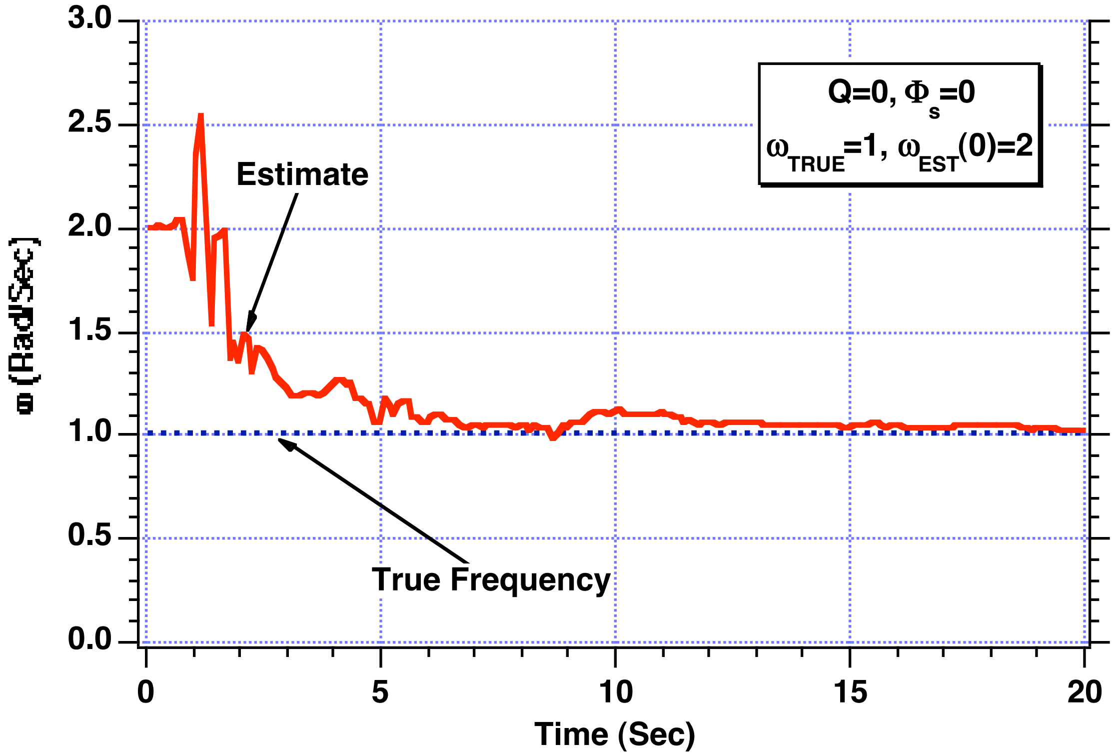
Alternate Three-State Extended Kalman Filter Estimates First State Well



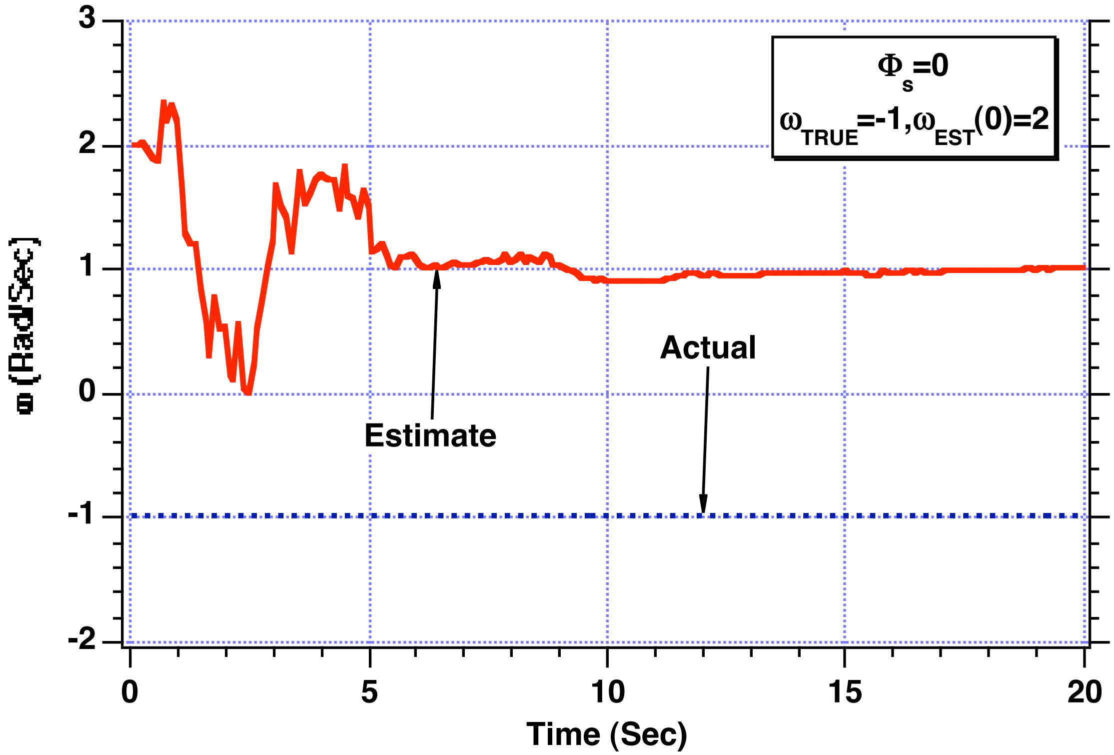
Alternate Three-State Extended Kalman Filter Estimates Second State Well Even Though Filter is Not Initialized Correctly



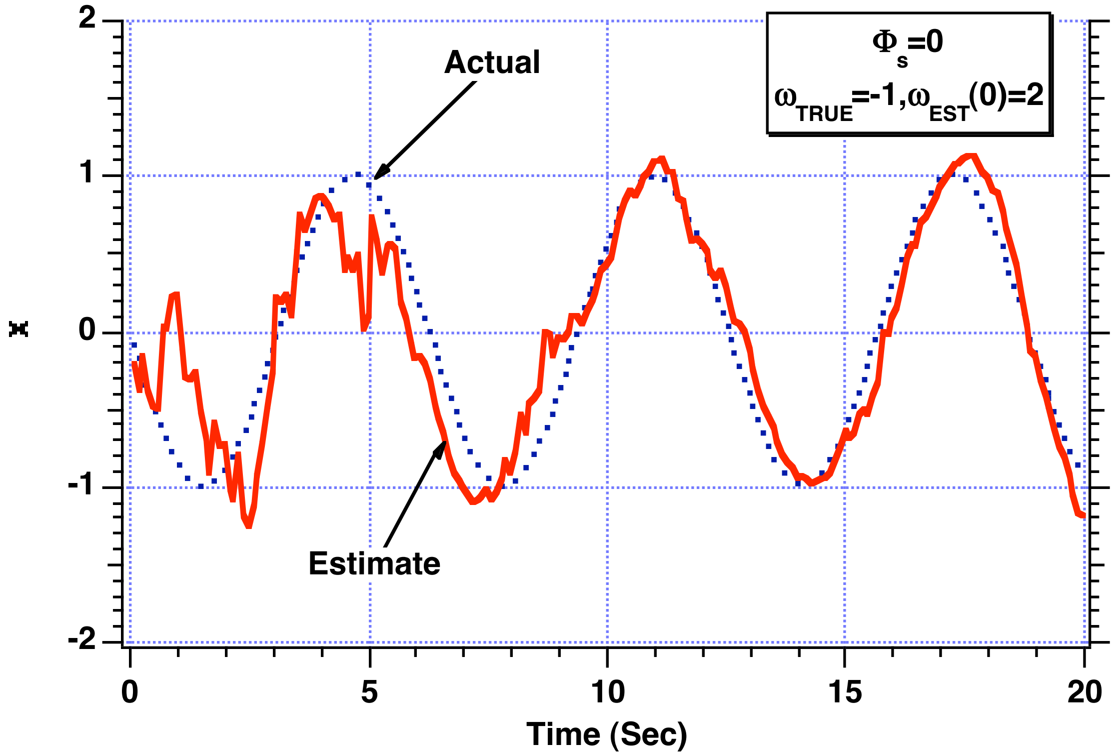
Alternate Extended Kalman Filter Appears Able to Estimate the Frequency of the Sinusoid Even Though Filter is Not Initialized Correctly



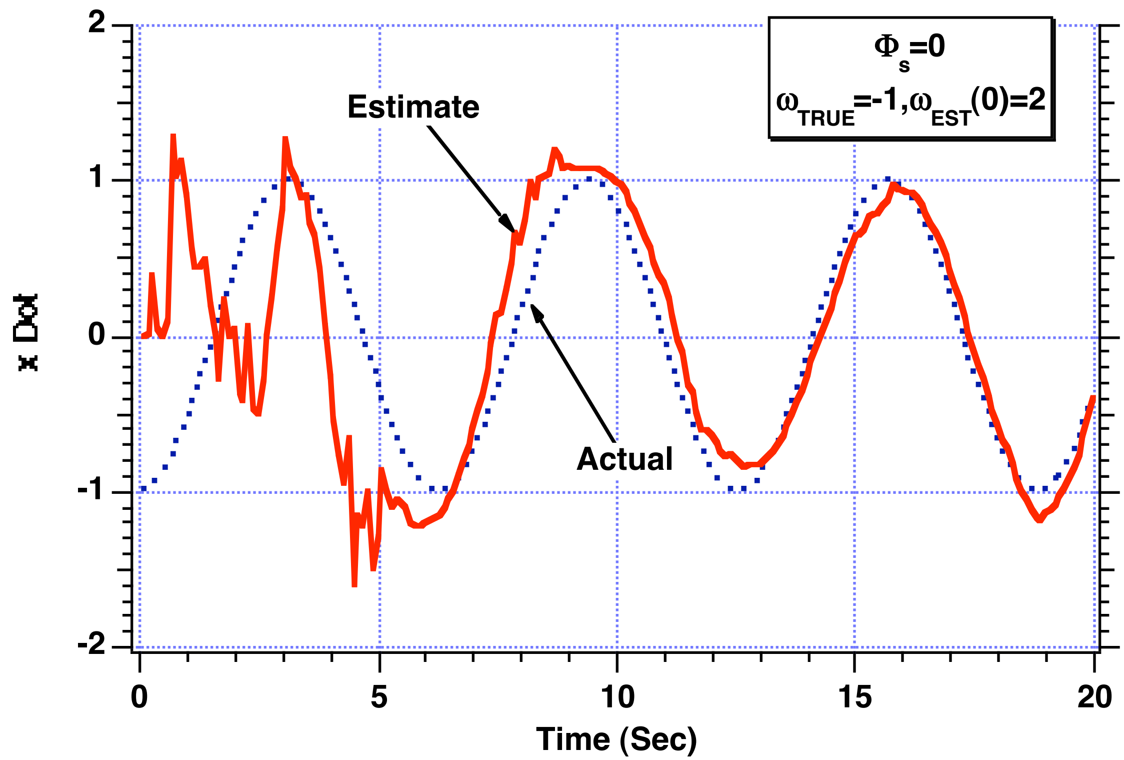
Alternate Extended Filter Appears Able to Estimate the Magnitude of the Frequency But Not it's Sign When the Filter is Not Initialized Correctly



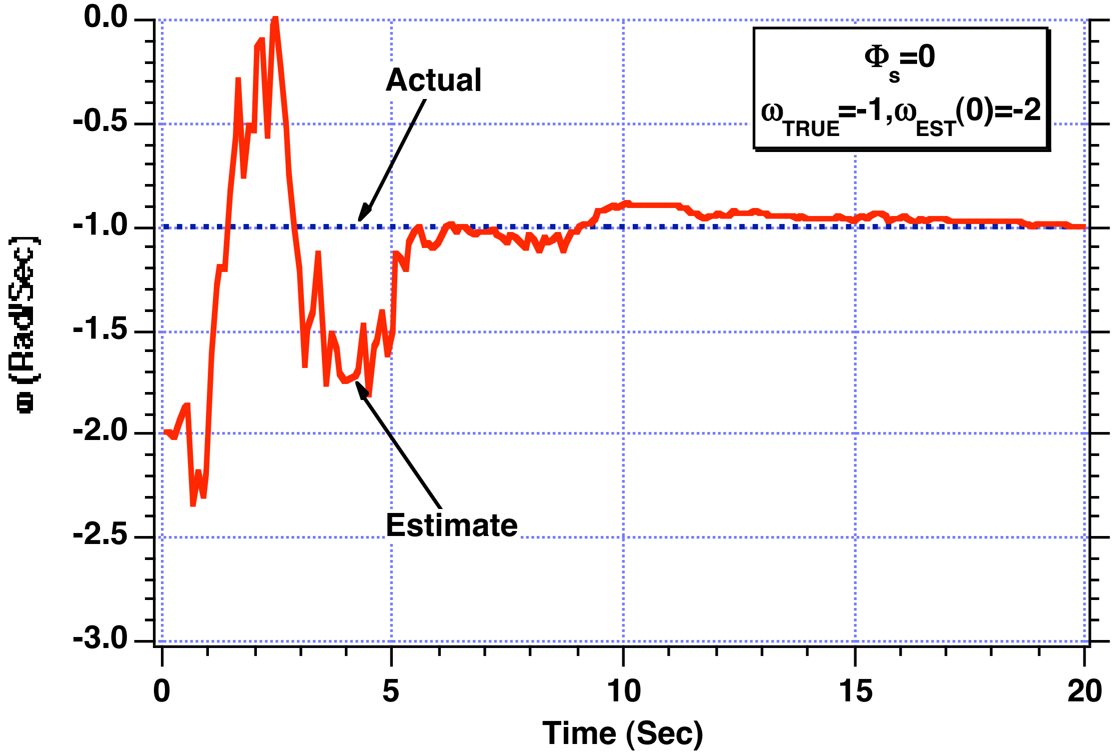
<u>However</u> Alternate Three-State Extended Kalman Filter Appears Able to Estimate the Signal When the Filter is Not Initialized Correctly



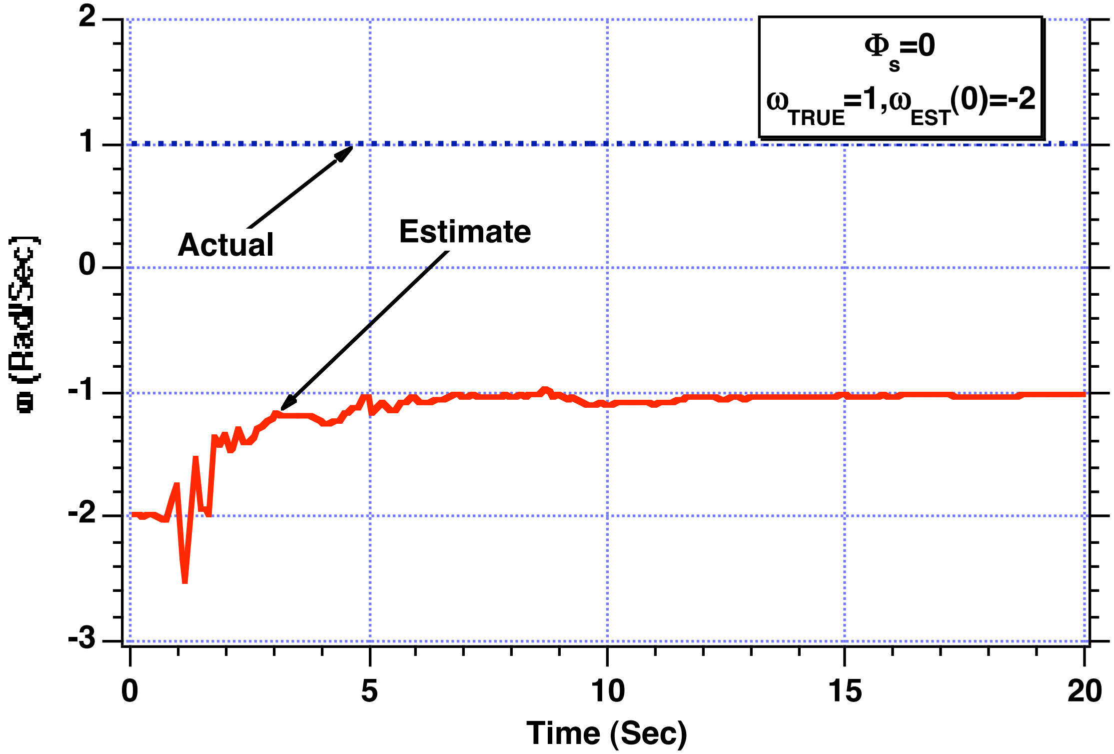
Alternate Three-State Extended Kalman Filter Appears Able to Estimate the Derivative of the Signal When the Filter is Not Initialized Correctly



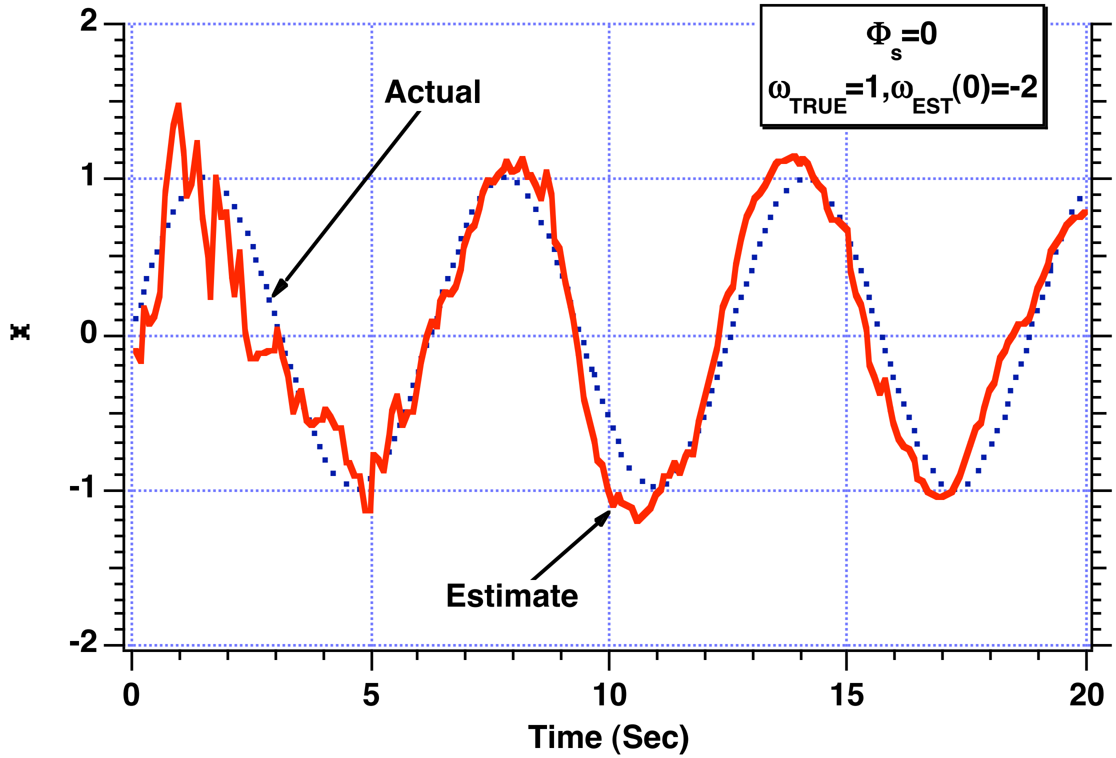
Alternate Extended Kalman Filter Appears Able to Estimate the Frequency Correctly When Frequency and Initial Estimate are Both Negative



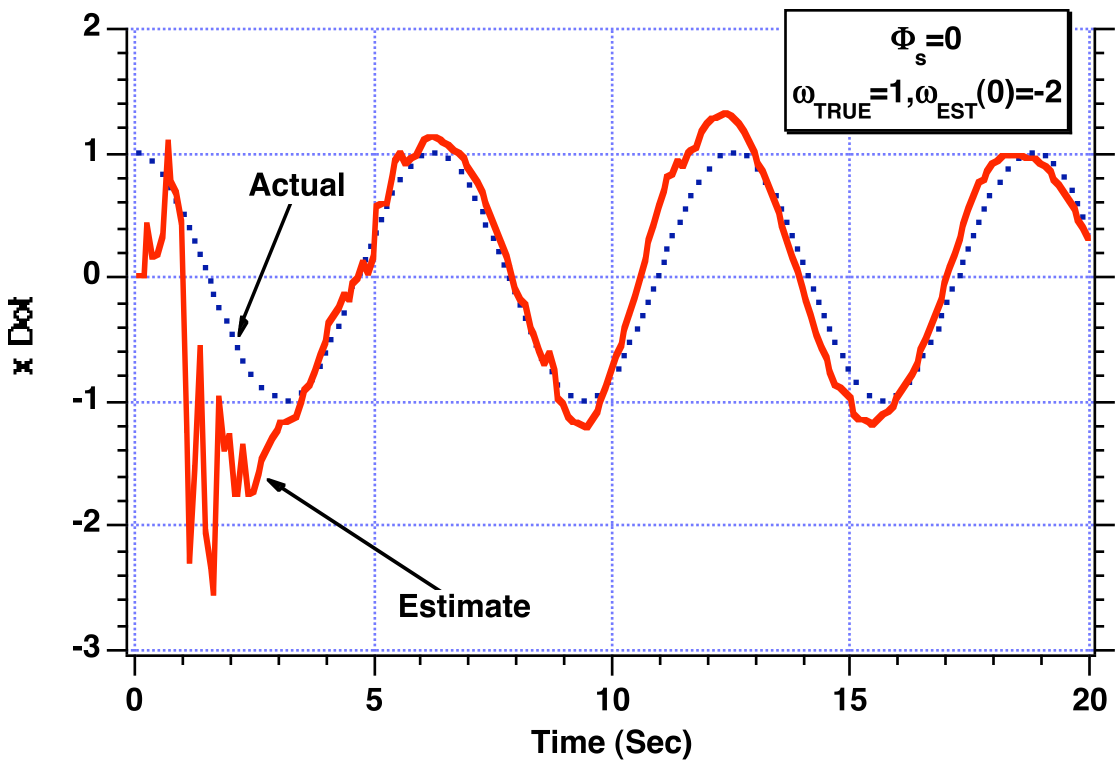
Alternate Extended Kalman Filter Appears Able to Estimate the Magnitude of the Frequency But Not it's Sign When the Filter is Not Initialized Correctly



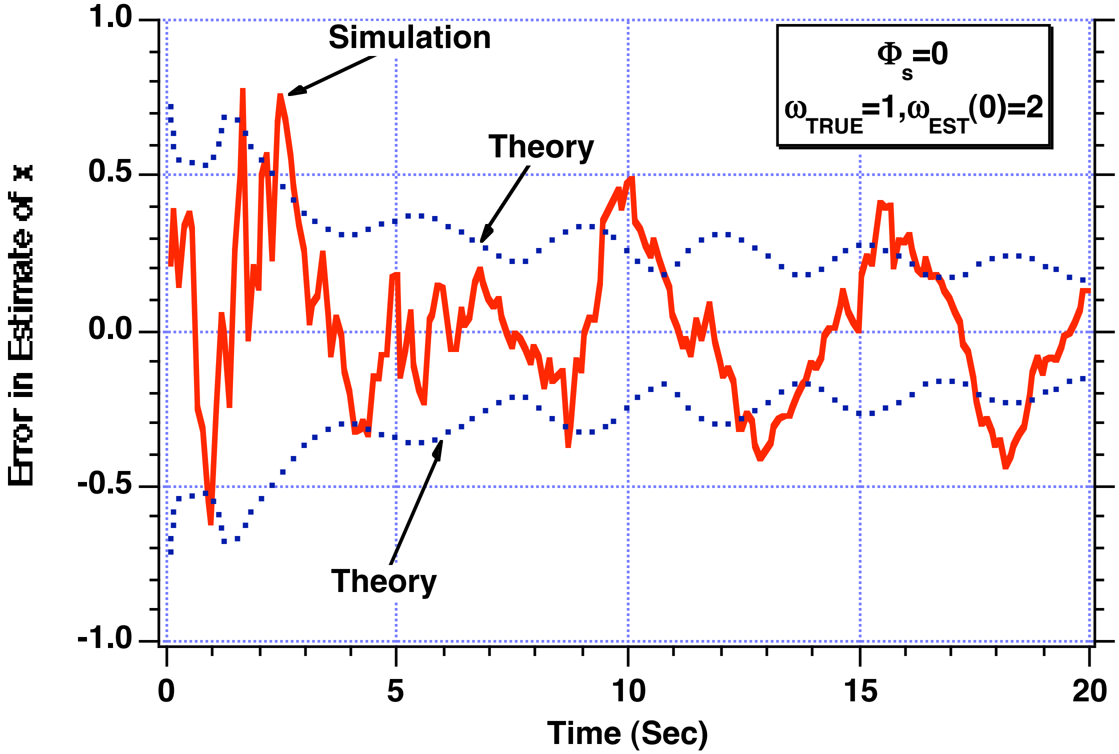
<u>However</u> Alternate Three-State Extended Kalman Filter Appears Able to Estimate the Signal When the Filter is Not Initialized Correctly



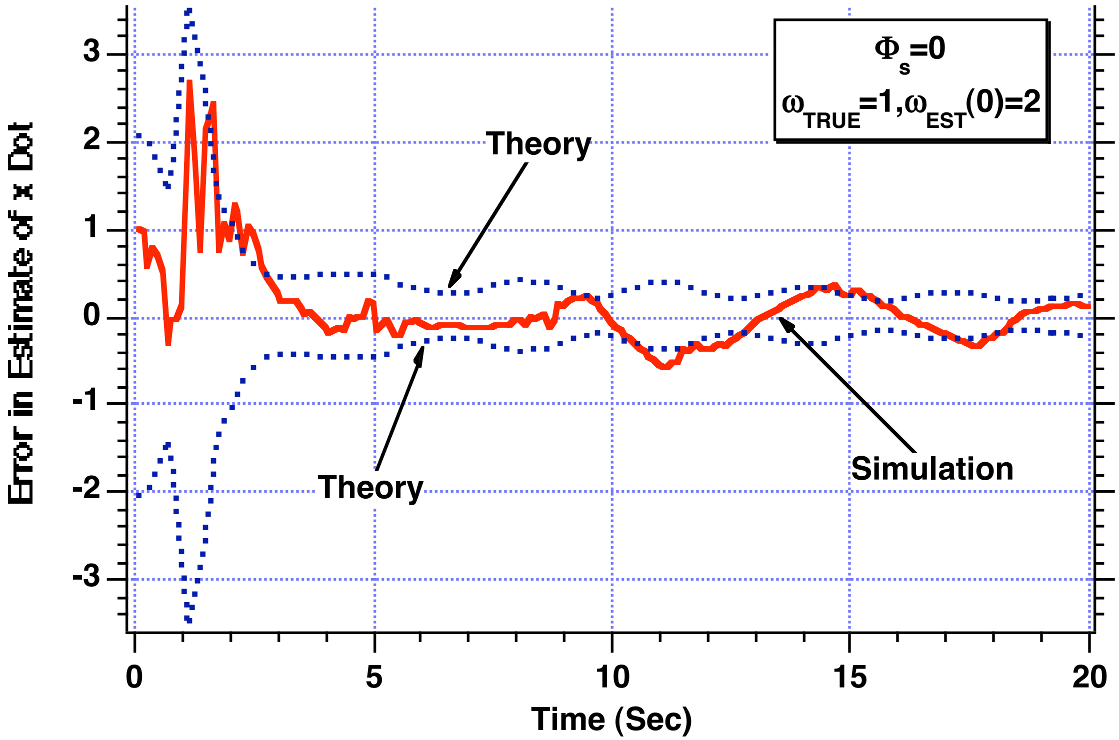
Alternate Three-State Extended Kalman Filter Appears Able to Estimate the Derivative of the Signal When the Filter is Not Initialized Correctly



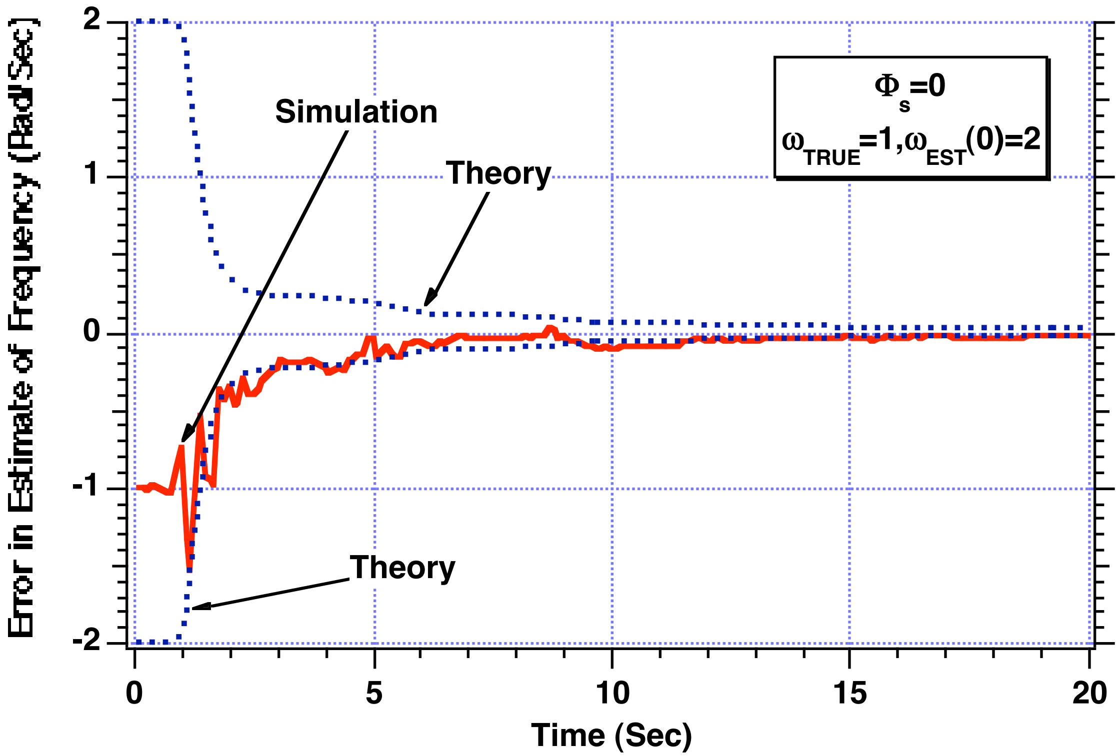
Error in the Estimate of First State Agrees With Theory



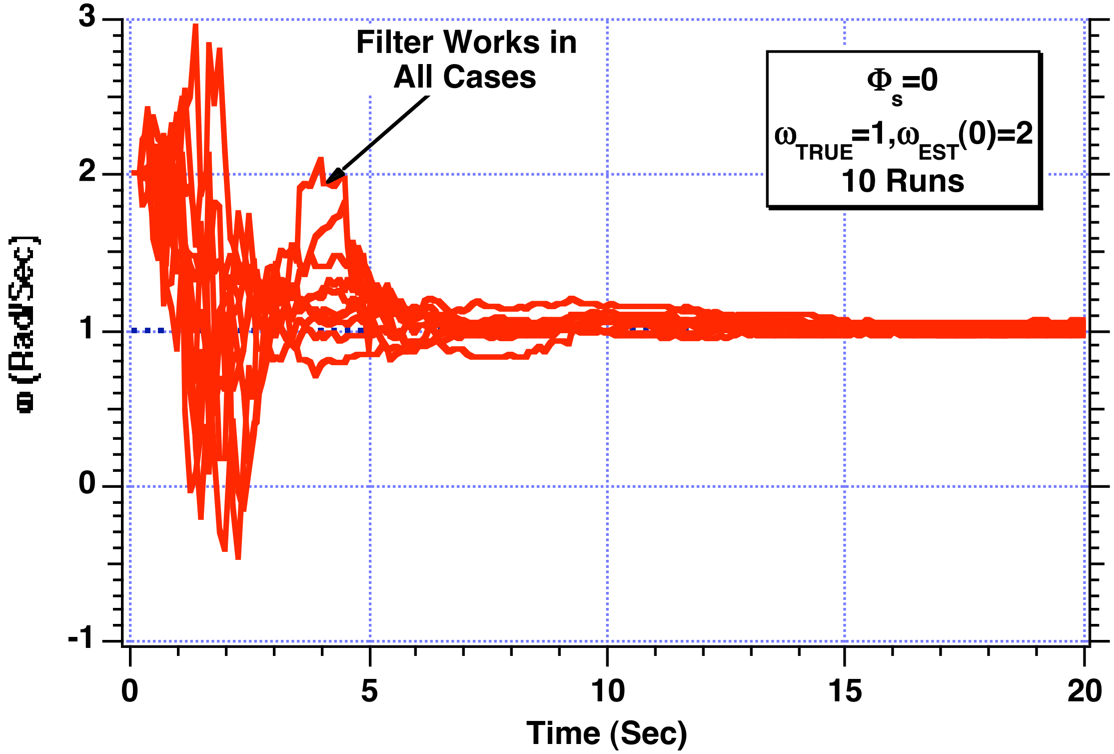
Error in the Estimate of Second State Agrees With Theory



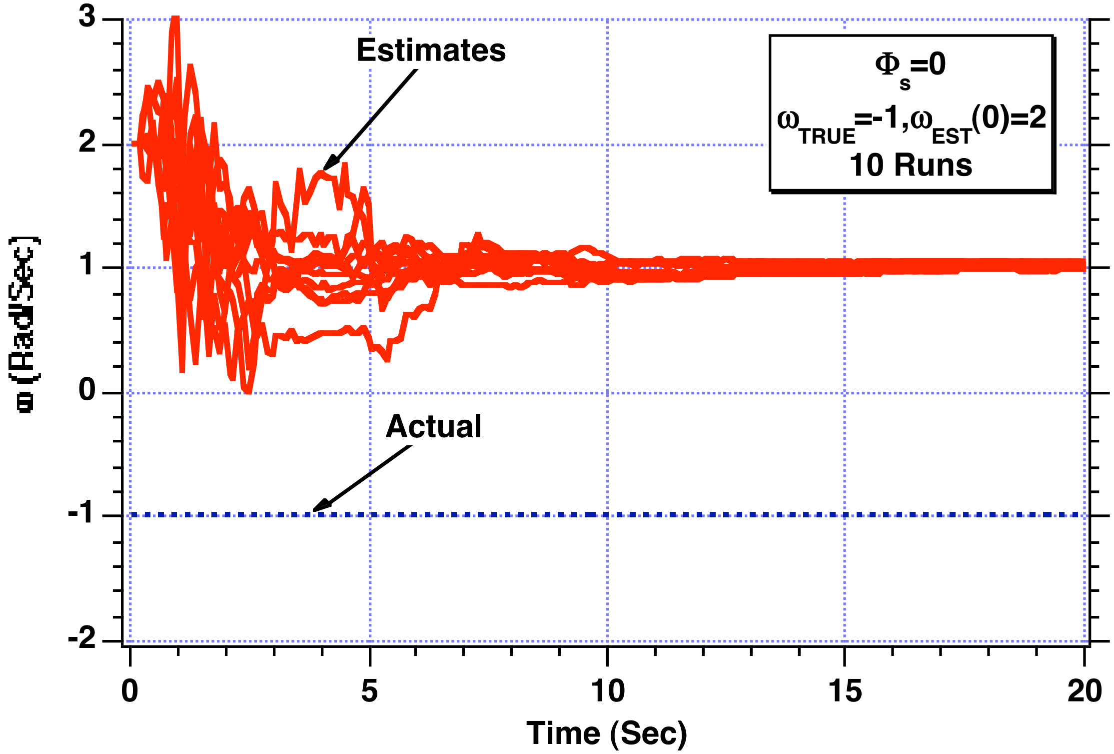
Error in the Estimate of Third State Agrees With Theory



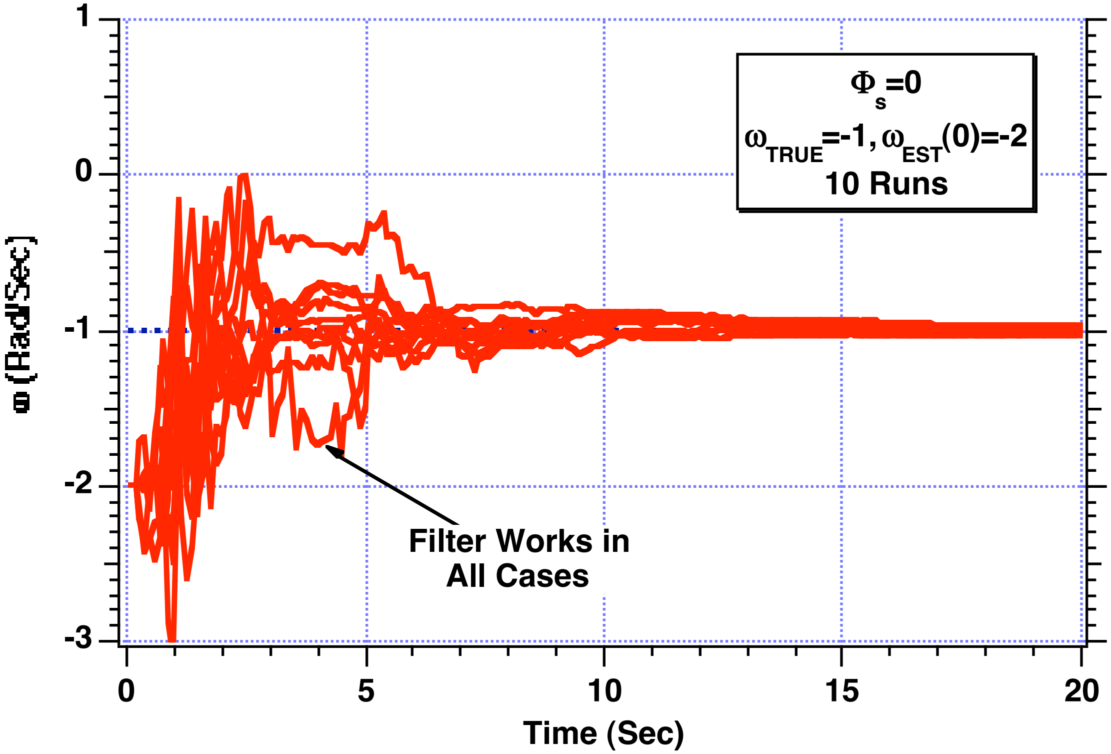
Alternate Three-State Extended Kalman Filter Estimates Correct Frequency When Initial Frequency Estimate is of Correct Sign



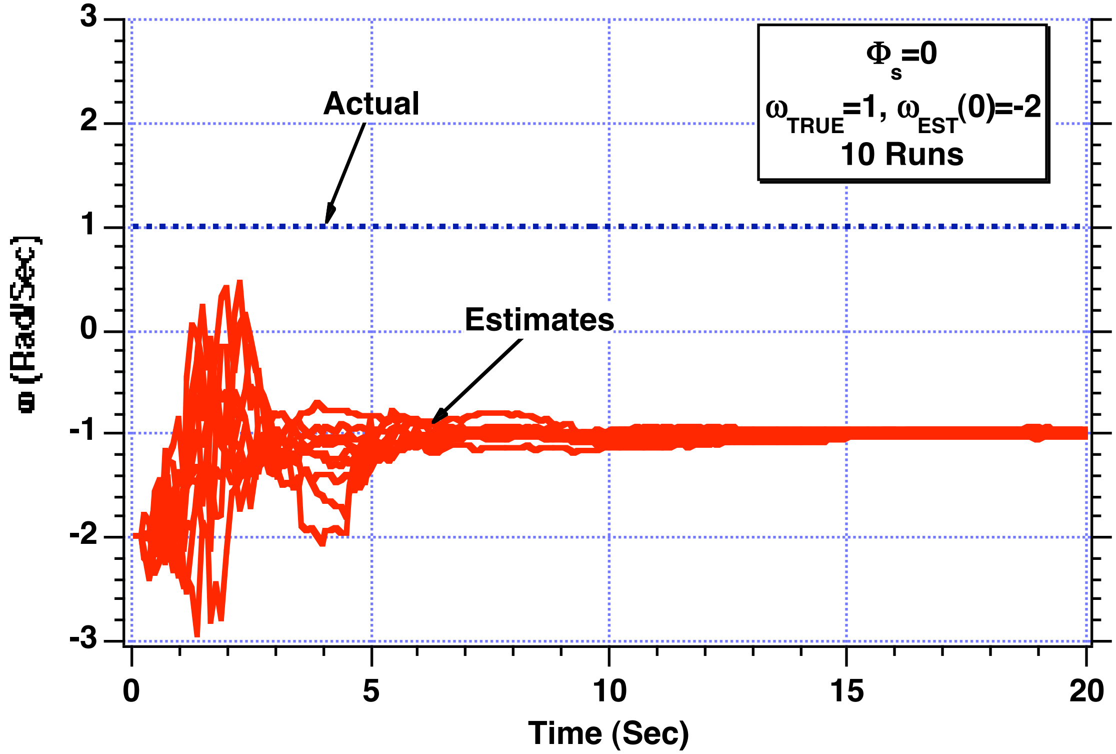
Alternate Three-State Extended Kalman Filter Estimates Correct Frequency <u>Magnitude</u> When Initial Frequency Estimate is of the Wrong Sign



Alternate Three-State Extended Kalman Filter Estimates Correct Frequency When Initial Frequency Estimate is of Correct Sign



Alternate Three-State Extended Kalman Filter Estimates Correct Frequency Magnitude When Initial Frequency Estimate is of the Wrong Sign



Summary For Alternate Filter

- Estimate frequency when initial frequency estimate is of same sign as actual frequency
- If initial frequency estimate is of different sign than actual frequency we are able to estimate magnitude but not sign of frequency
    - However we are able to estimate x and x dot in all cases
- Possible we can not distinguish between positive and negative frequencies because $\ddot{x} = -\omega^2 x$ only frequency squared term shows up in our model of the real world

Another Extended Kalman Filter For Sinusoidal Model

Another Extended Kalman Filter-1

Since

$$\ddot{x} = -\omega^2 x$$

We can define

$$z = \omega^2$$

And get new model of real world

$$\ddot{x} = -zx$$

$$\dot{z} = u_s$$

State space model of real world

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

Systems dynamics matrix

$$\mathbf{F} = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial \dot{x}} & \frac{\partial \dot{x}}{\partial z} \\ \frac{\partial \ddot{x}}{\partial x} & \frac{\partial \ddot{x}}{\partial \dot{x}} & \frac{\partial \ddot{x}}{\partial z} \\ \frac{\partial \dot{z}}{\partial x} & \frac{\partial \dot{z}}{\partial \dot{x}} & \frac{\partial \dot{z}}{\partial z} \end{bmatrix}$$

Another Extended Kalman Filter-2

After evaluating partial derivatives

$$\mathbf{F} = \begin{bmatrix} 0 & 1 & 0 \\ -\hat{z} & 0 & -\hat{x} \\ 0 & 0 & 0 \end{bmatrix}$$

Two term approximation for fundamental matrix

$$\mathbf{\Phi}(t) \approx I + \mathbf{F}t = \begin{bmatrix} 1 & t & 0 \\ -\hat{z}t & 1 & -\hat{x}t \\ 0 & 0 & 1 \end{bmatrix}$$

Discrete fundamental matrix

$$\mathbf{\Phi}_k \approx \begin{bmatrix} 1 & T_s & 0 \\ -\hat{z}_{k-1}T_s & 1 & -\hat{x}_{k-1}T_s \\ 0 & 0 & 1 \end{bmatrix}$$

Continuous process noise matrix

$$\mathbf{Q} = E(ww^T) = E\left[\begin{bmatrix} 0 \\ 0 \\ u_s \end{bmatrix} \begin{bmatrix} 0 & 0 & u_s \end{bmatrix}\right] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Phi_s \end{bmatrix}$$

Another Extended Kalman Filter-3

Recall

$$Q_k = \int_0^{T_s} \Phi(\tau) Q \Phi^T(\tau) dt$$

Substitution yields

$$Q_k = \int_0^{T_s} \begin{bmatrix} 1 & \tau & 0 \\ -\hat{z}\tau & 1 & -\hat{x}\tau \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Phi_s \end{bmatrix} \begin{bmatrix} 1 & -\hat{z}\tau & 0 \\ \tau & 1 & 0 \\ 0 & -\hat{x}\tau & 1 \end{bmatrix} d\tau$$

Multiply out the matrices

$$Q_k = \int_0^{T_s} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \hat{x}^2\tau^2\Phi_s & -\hat{x}\tau\Phi_s \\ 0 & -\hat{x}\tau\Phi_s & \Phi_s \end{bmatrix} d\tau$$

And integrate to get discrete process noise matrix

$$Q_k = \begin{bmatrix} 0 & 0 & 0 \\ 0 & .333\hat{x}^2 T_s^3 \Phi_s & -.5\hat{x} T_s^2 \Phi_s \\ 0 & -.5\hat{x} T_s^2 \Phi_s & T_s \Phi_s \end{bmatrix}$$

Another Extended Kalman Filter-4

Measurement equation is linear function of states

$$x_k^* = x_k + v_k$$

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

Measurement matrix

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

Measurement noise matrix is scalar

$$\mathbf{R}_k = E(\mathbf{v}_k \mathbf{v}_k^T) = \sigma_k^2$$

Filtering equations

$$\hat{x}_k = \bar{x}_k + K_{1_k}(x_k^* - \bar{x}_k)$$

$$\hat{\dot{x}}_k = \bar{\dot{x}}_k + K_{2_k}(x_k^* - \bar{x}_k)$$

$$\hat{z}_k = \hat{z}_{k-1} + K_{3_k}(x_k^* - \bar{x}_k)$$

***Barred quantities are obtained by numerically integrating nonlinear differential equations. They are <u>not</u> obtained using fundamental matrix**

Another True BASIC Extended Kalman Filter-1

```
OPTION NOLET
REM UNSAVE "DATFIL"
REM UNSAVE "COVFIL"
OPEN #1:NAME "DATFIL",ACCESS OUTPUT,CREATE NEW, ORGANIZATION TEXT
OPEN #2:NAME "COVFIL",ACCESS OUTPUT,CREATE NEW, ORGANIZATION TEXT
SET #1: MARGIN 1000
SET #2: MARGIN 1000
DIM P(3,3),Q(3,3),M(3,3),PHI(3,3),HMAT(1,3),HT(3,1),PHIT(3,3)
DIM RMAT(1,1),IDNP(3,3),PHIP(3,3),PHIPPHIT(3,3),HM(1,3)
DIM HMHT(1,1),HMHTR(1,1),HMHTRINV(1,1),MHT(3,1),K(3,1),F(3,3)
DIM KH(3,3),IKH(3,3)
HP=.001
W=1.
WH=2.
A=1.
TS=.1
ORDER=3
PHIS=0.
SIGX=1.
T=0.
S=0.
H=.001
MAT F=ZER(ORDER,ORDER)
MAT PHI=ZER(ORDER,ORDER)
MAT P=ZER(ORDER,ORDER)
MAT IDNP=IDN(ORDER,ORDER)
MAT Q=ZER(ORDER,ORDER)
RMAT(1,1)=SIGX^2
P(1,1)=SIGX^2      ┐
P(2,2)=2.^2        │ Initial covariance matrix
P(3,3)=4.^2        ┘
XTH=0.      ┐
XTDH=0.     │ Initial state estimates
ZH=WH^2     ┘
XT=0.
XTD=A*W
```

Another True BASIC Extended Kalman Filter-2

```
DO WHILE T<=20.
        XTOLD=XT
        XTDOLD=XTD
        XTDD=-W*W*XT
        XT=XT+H*XTD
        XTD=XTD+H*XTDD
        T=T+H
        XTDD=-W*W*XT
        XT=.5*(XTOLD+XT+H*XTD)
        XTD=.5*(XTDOLD+XTD+H*XTDD)
        S=S+H
        IF S>=(TS-.00001) THEN
                S=0.
                PHI(1,1)=1.
                PHI(1,2)=TS
                PHI(2,1)=-ZH*TS
                PHI(2,2)=1.
                PHI(2,3)=-XTH*TS
                PHI(3,3)=1.
                Q(2,2)=XTH*XTH*TS*TS*TS*PHIS/3.
                Q(2,3)=-XTH*TS*TS*PHIS/2.
                Q(3,2)=Q(2,3)
                Q(3,3)=PHIS*TS
                HMAT(1,1)=1.
                HMAT(1,2)=0.
                HMAT(1,3)=0.
                MAT PHIT=TRN(PHI)
                MAT HT=TRN(HMAT)
                MAT PHIP=PHI*P
                MAT PHIPPHIT=PHIP*PHIT
                MAT M=PHIPPHIT+Q
                MAT HM=HMAT*M
                MAT HMHT=HM*HT
                MAT HMHTR=HMHT+RMAT
                HMHTRINV(1,1)=1./HMHTR(1,1)
                MAT MHT=M*HT
                MAT K=MHT*HMHTRINV
                MAT KH=K*HMAT
                MAT IKH=IDNP-KH
                MAT P=IKH*M
```

Integrate second-order differential equation with second-order Runge-Kutta technique (XTOLD=XT through S=S+H)

Fundamental matrix (PHI(1,1)=1. through PHI(3,3)=1.)

Process noise matrix (Q(2,2) through Q(3,3))

Measurement matrix (HMAT(1,1) through HMAT(1,3))

Riccati equations (MAT PHIP=PHI*P through MAT P=IKH*M)

Another True BASIC Extended Kalman Filter-3

```
                    CALL GAUSS(XTNOISE,SIGX)
                    XTMEAS=XT+XTNOISE
                    CALL PROJECT(T,TS,XTH,XTDH,XTB,XTDB,HP,ZH)
                    RES=XTMEAS-XTB
                    XTH=XTB+K(1,1)*RES
                    XTDH=XTDB+K(2,1)*RES
                    ZH=ZH+K(3,1)*RES
                    ERRX=XT-XTH
                    SP11=SQR(P(1,1))
                    ERRXD=XTD-XTDH
                    SP22=SQR(P(2,2))
                    Z=W*W
                    ERRZ=Z-ZH
                    SP33=SQR(P(3,3))
                    PRINT T,XT,XTH,XTD,XTDH,Z,ZH
                    PRINT #1:T,XT,XTH,XTD,XTDH,Z,ZH
                    PRINT #2:T,ERRX,SP11,-SP11,ERRXD,SP22,-SP22,ERRZ,SP33,-SP33
          END IF
LOOP
CLOSE #1
CLOSE #2
END

SUB PROJECT(TP,TS,XTP,XTDP,XTH,XTDH,HP,Z)
T=0.
XT=XTP
XTD=XTDP
H=HP
DO WHILE T<=(TS-.0001)
          XTDD=-Z*XT
          XTD=XTD+H*XTDD
          XT=XT+H*XTD
          T=T+H
LOOP
XTH=XT
XTDH=XTD
END SUB
```

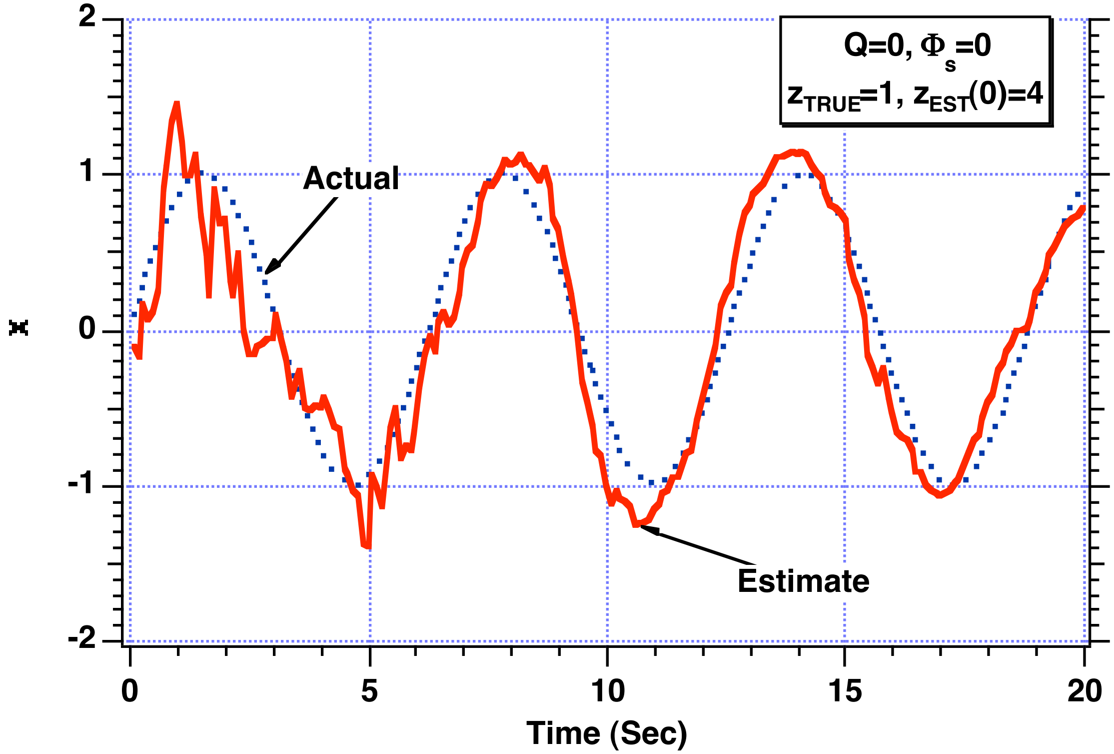
Project states ahead (CALL PROJECT line)

Filtering equations (RES through XTDH)

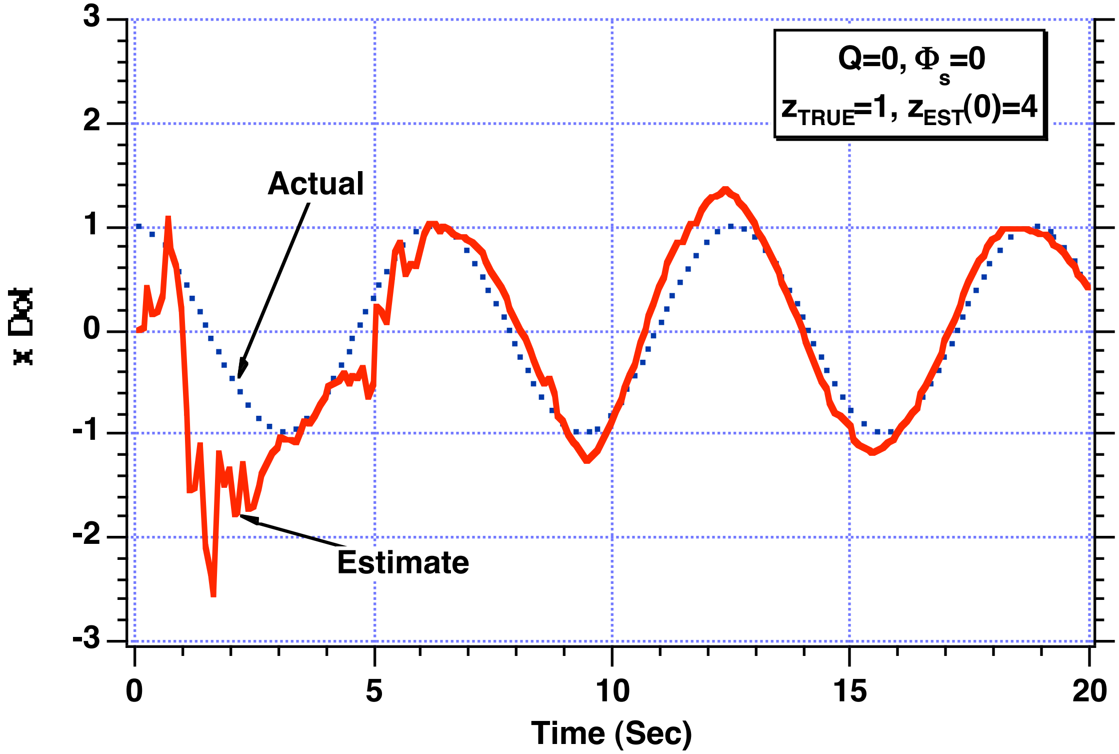
Actual and theoretical errors in estimates (ERRX through SP33)

Project states ahead one sampling interval by integrating second-order differential equation with Euler technique (SUB PROJECT)

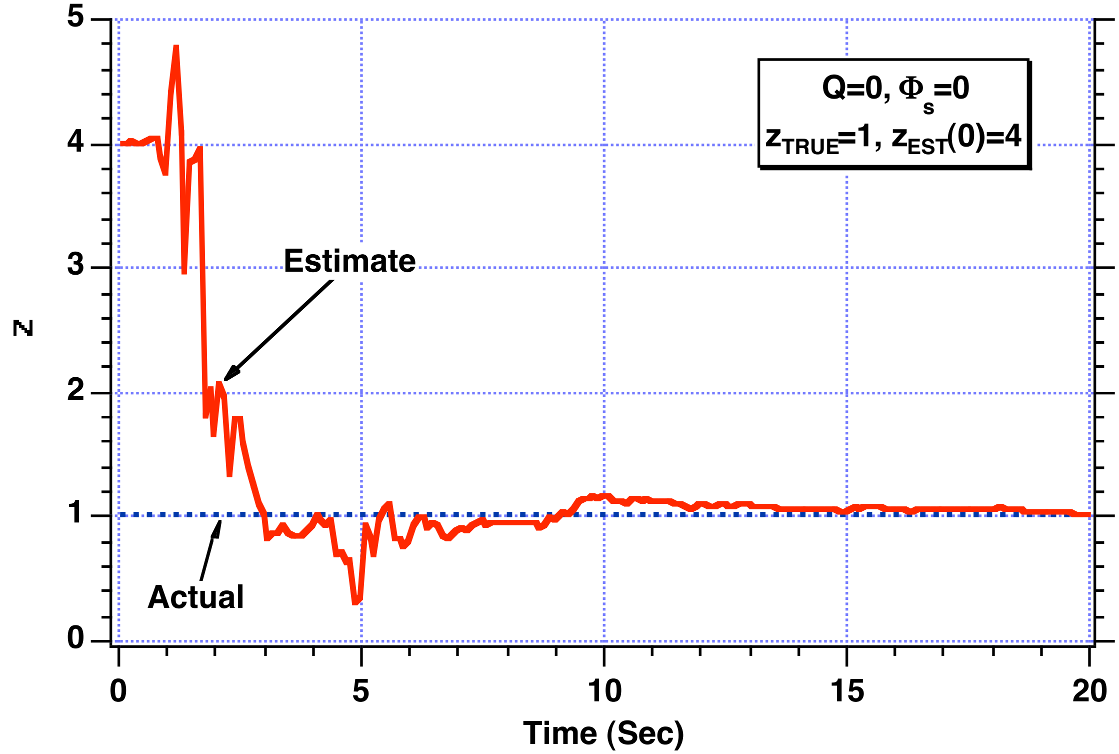
New Extended Kalman Filter Estimates First State Quite Well



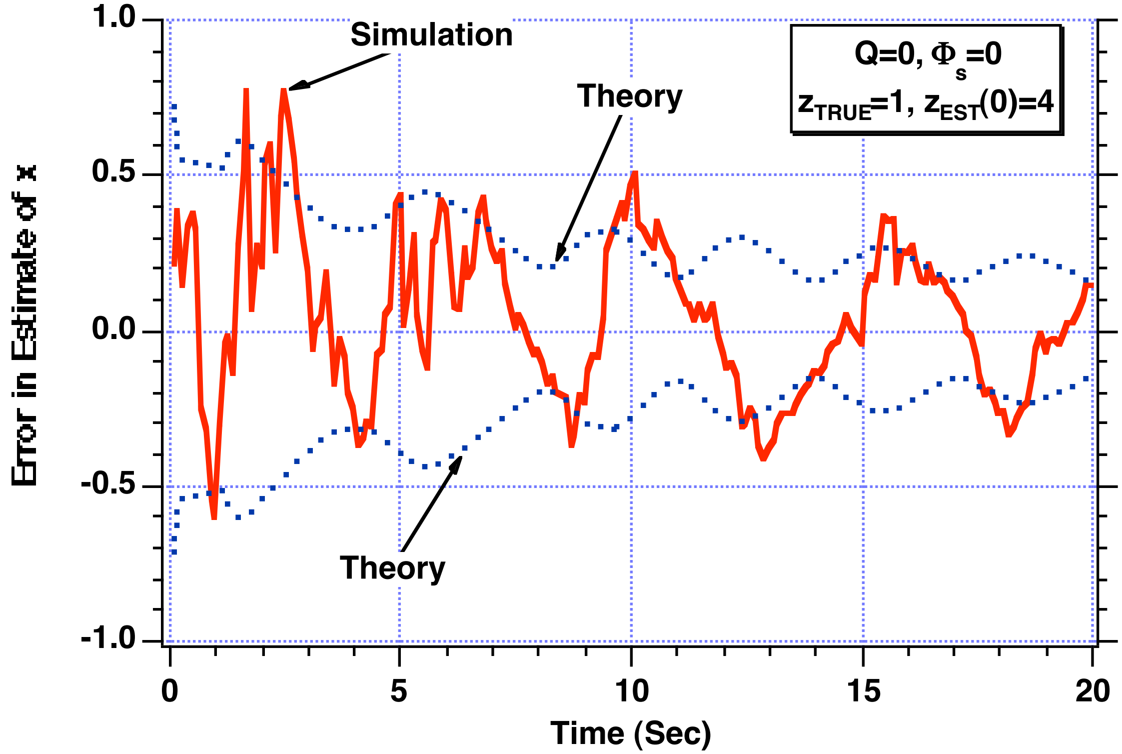
New Extended Kalman Filter Estimates Second State Well Even Though That State is Not Correctly Initialized



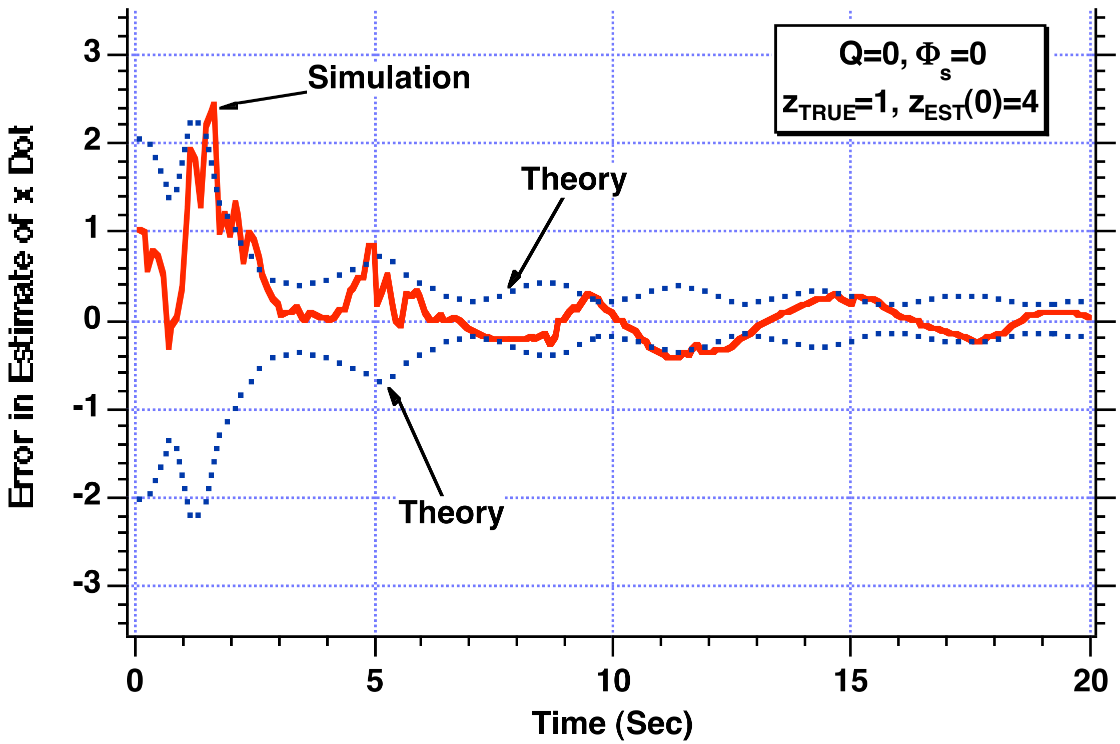
After 5 Sec New Extended Kalman Filter is Able to Estimate the Square of Frequency



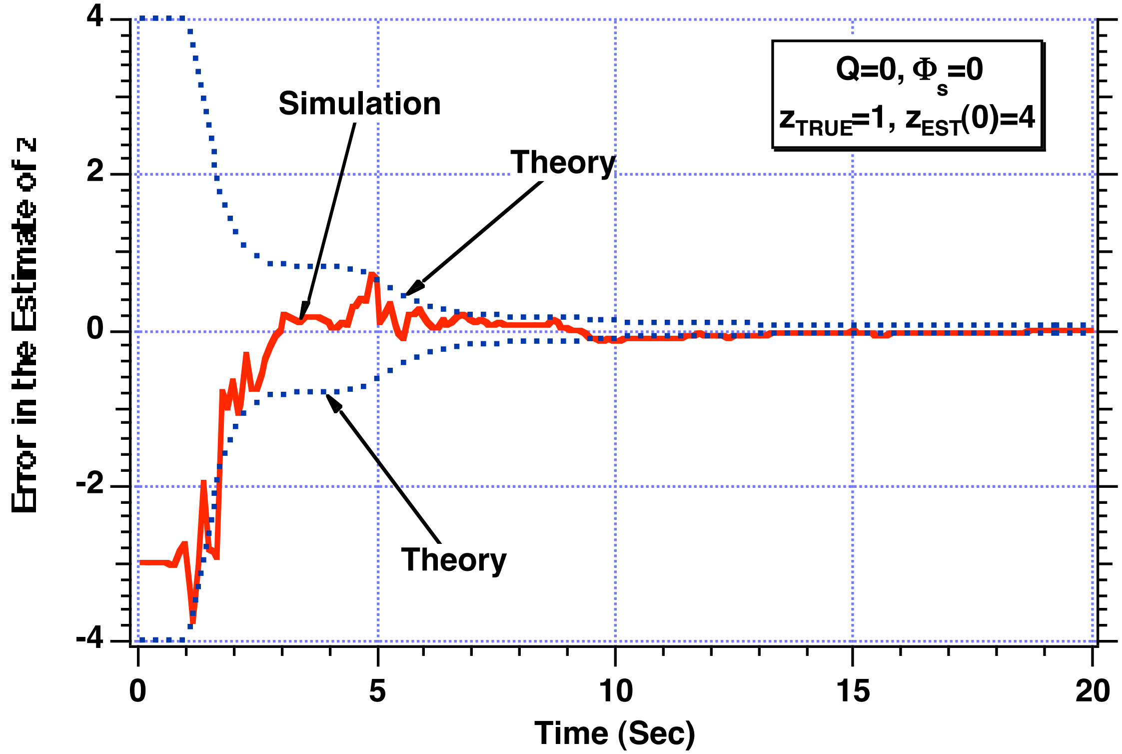
Error in the Estimate of First State Agrees With Theory



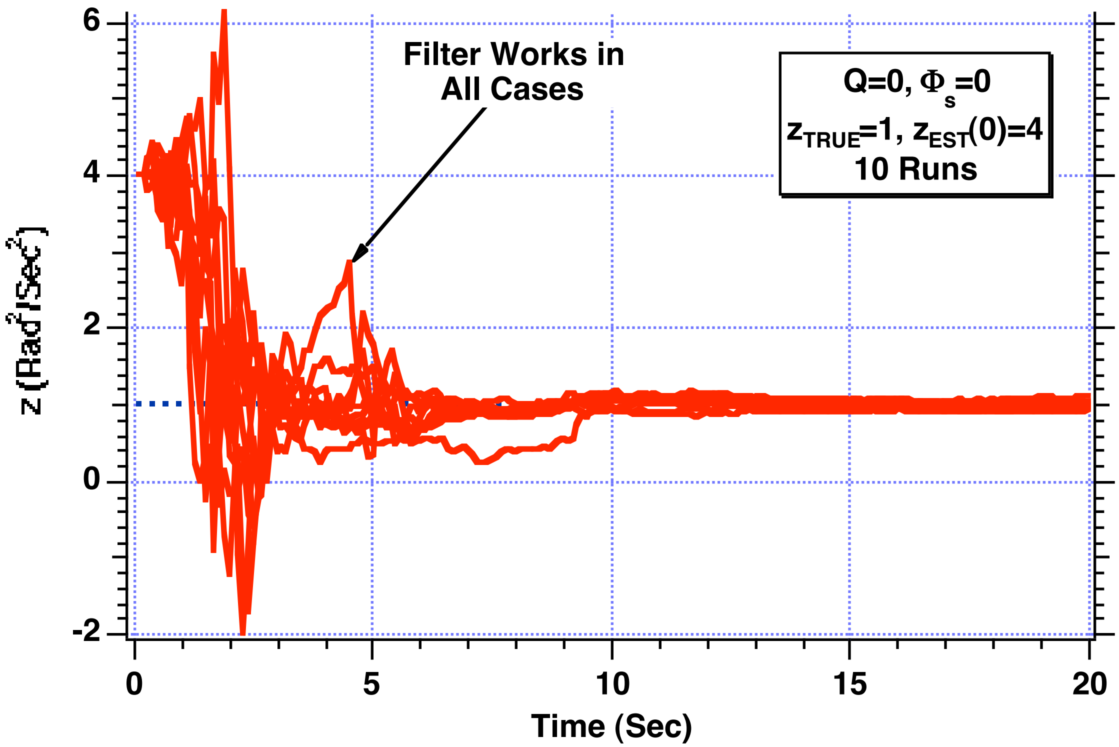
Error in the Estimate of Second State Agrees With Theory



Error in the Estimate of Third State Agrees With Theory



New Three-State Extended Kalman Filter is also Effective in Estimating the Square of the Frequency of a Sinusoid



Tracking a Sine Wave
Summary

- Arbitrarily choosing states for filter does not guarantee that it will work if programmed correctly
- Various extended Kalman filters designed to highlight issues
  - Initialization experiments were used to illustrate robustness of various filter designs