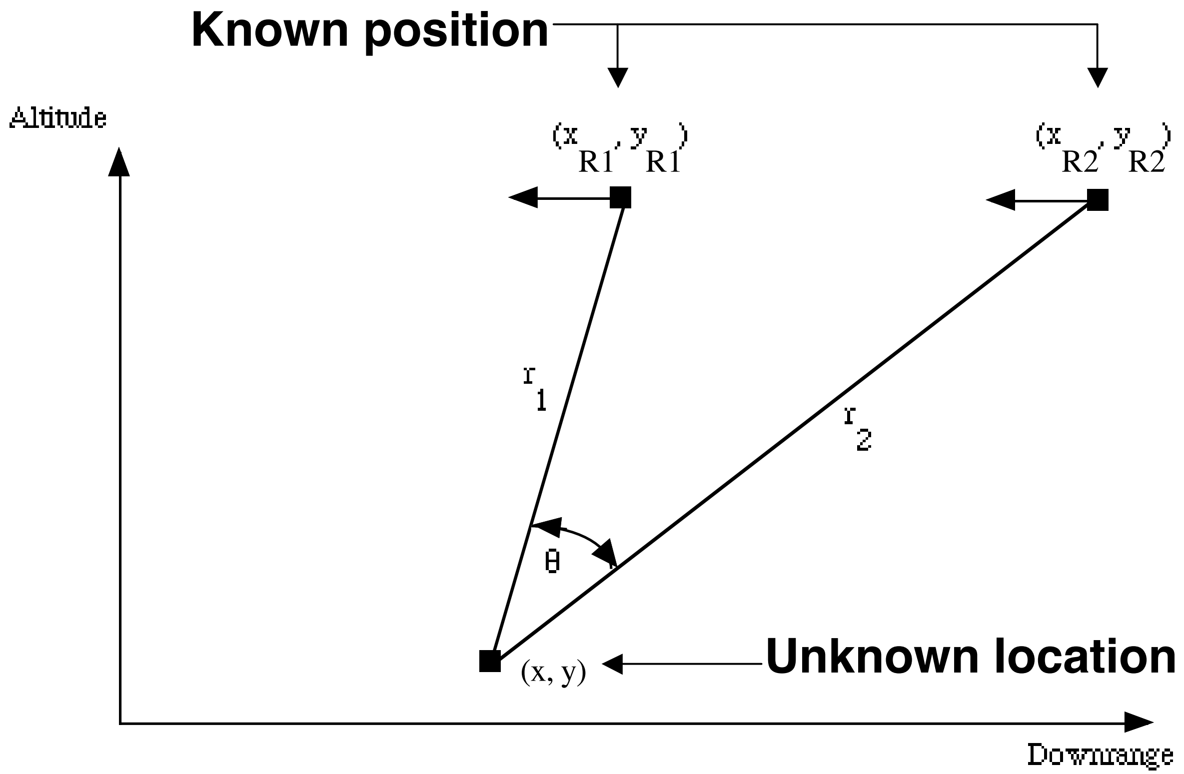
Satellite Navigation

Satellite Navigation
Overview

- Solving for receiver location based on perfect range measurements from two satellites
- Solving for receiver location based on noisy range measurements from two satellites (no filtering)
- Improvements with linear filtering of range
- Using extended Kalman filter
- Using extended Kalman filter with measurements from only one satellite
- Moving receiver
  - Constant velocity
  - Variable velocity

Solving for Receiver Location Based on Perfect Range Measurements From Two Satellites

Two Satellites Making Range Measurements to a Receiver



Satellite to Receiver Geometry-1

Range from each satellite to receiver

$$r_1 = \sqrt{(x_{R1} - x)^2 + (y_{R1} - y)^2}$$

$$r_2 = \sqrt{(x_{R2} - x)^2 + (y_{R2} - y)^2}$$

←

2 equations with 2 unknowns

Solve for receiver location

Squaring both sides of each equation

$$r_1^2 = x_{R1}^2 - 2x_{R1}x + x^2 + y_{R1}^2 - 2y_{R1}y + y^2$$

$$r_2^2 = x_{R2}^2 - 2x_{R2}x + x^2 + y_{R2}^2 - 2y_{R2}y + y^2$$

Subtracting second equation from first and combining terms

$$r_1^2 - r_2^2 = 2x(x_{R2} - x_{R1}) + 2y(y_{R2} - y_{R1}) + x_{R1}^2 + y_{R1}^2 - x_{R2}^2 - y_{R2}^2$$

Solving for x

$$x = -\frac{y(y_{R2} - y_{R1})}{(x_{R2} - x_{R1})} + \frac{r_1^2 - r_2^2 - x_{R1}^2 - y_{R1}^2 + x_{R2}^2 + y_{R2}^2}{2(x_{R2} - x_{R1})}$$

Satellite to Receiver Geometry-2

By defining

$$A = -\frac{(y_{R2} - y_{R1})}{(x_{R2} - x_{R1})}$$

$$B = \frac{r_1^2 - r_2^2 - x_{R1}^2 - y_{R1}^2 + x_{R2}^2 + y_{R2}^2}{2(x_{R2} - x_{R1})}$$

We get

$$x = Ay + B$$

Substituting into square of first range equation yields

$$r_1^2 = x_{R1}^2 - 2x_{R1}(Ay + B) + (Ay + B)^2 + y_{R1}^2 - 2y_{R1}y + y^2$$

Rewriting preceding equation as quadratic

$$0 = y^2(1 + A^2) + y(-2Ax_{R1} + 2AB - 2y_{R1}) + x_{R1}^2 - 2x_{R1}B + y_{R1}^2 - r_1^2$$

Simplify by defining

$$a = 1 + A^2 \qquad b = -2Ax_{R1} + 2AB - 2y_{R1} \qquad c = x_{R1}^2 - 2x_{R1}B + y_{R1}^2 - r_1^2$$

Satellite to Receiver Geometry-3

Quadratic equation becomes

$$0 = ay^2 + by + c$$

Solve and use common sense to throw away extraneous root

$$y = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Once we have x we can get y from

$$x = Ay + B$$

If we know satellite position at any time and also have perfect range measurements to a receiver whose location is unknown we can <u>derive</u> receiver location

Deriving Formula for Angle Between Two Range Measurements

Angle between two range measurements can be expressed as

$$\theta = \cos^{-1} \frac{\overline{\mathbf{r}}_1 . \ \overline{\mathbf{r}}_2}{|\overline{\mathbf{r}}_1| \ |\overline{\mathbf{r}}_2|}$$

Range measurements can also be expressed as vectors

$$\overline{\mathbf{r}}_1 = (x_{R1} - x)\ \overline{\mathbf{i}} + (y_{R1} - y)\ \overline{\mathbf{j}}$$

$$\overline{\mathbf{r}}_2 = (x_{R2} - x)\ \overline{\mathbf{i}} + (y_{R2} - y)\ \overline{\mathbf{j}}$$

Range magnitudes are simply

$$|\overline{\mathbf{r}}_1| = r_1$$

$$|\overline{\mathbf{r}}_2| = r_2$$

Substitution yields

$$\theta = \cos^{-1} \frac{(x_{R1} - x)(x_{R2} - x) + (y_{R1} - y)(y_{R2} - y)}{r_1 r_2}$$

Simplified Global Positioning System (GPS) Example

GPS satellites are at 20,000 km altitude and travel at 14,600 ft/sec

Satellite position can be derived from velocity

$$x_{R1} = \dot{x}_{R1}t + x_{R1}(0)$$

$$x_{R2} = \dot{x}_{R2}t + x_{R2}(0)$$

$$y_{R1} = y_{R1}(0)$$

$$y_{R2} = y_{R2}(0)$$

For this example

$x_{R1}(0) = 1{,}000{,}000$ ft

$x_{R2}(0) = 500{,}000$ ft

$y_{R1}(0) = 20{,}000{*}3280$ ft

$y_{R2}(0) = 20{,}000{*}3280$ ft

$\dot{x}_{R1} = -14{,}600$ ft/sec

$\dot{x}_{R2} = -14{,}600$ ft/sec

FORTRAN Code to See if Receiver Location can be Determined From Perfect Range Measurements-1

```
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
X=0.
Y=0.
XR1=1000000.
YR1=20000.*3280.
XR2=500000.
YR2=20000.*3280.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
TS=1.
TF=100.
T=0.
S=0.
H=.01
WHILE(T<=TF)
          XR1OLD=XR1
          XR2OLD=XR2
          XR1D=-14600.
          XR2D=-14600.
          XR1=XR1+H*XR1D
          XR2=XR2+H*XR2D
          T=T+H
          XR1D=-14600.
          XR2D=-14600.
          XR1=.5*(XR1OLD+XR1+H*XR1D)
          XR2=.5*(XR2OLD+XR2+H*XR2D)
          S=S+H
```

Receiver location (X=0., Y=0.)

Initial location of each satellite (XR1 through YR2)

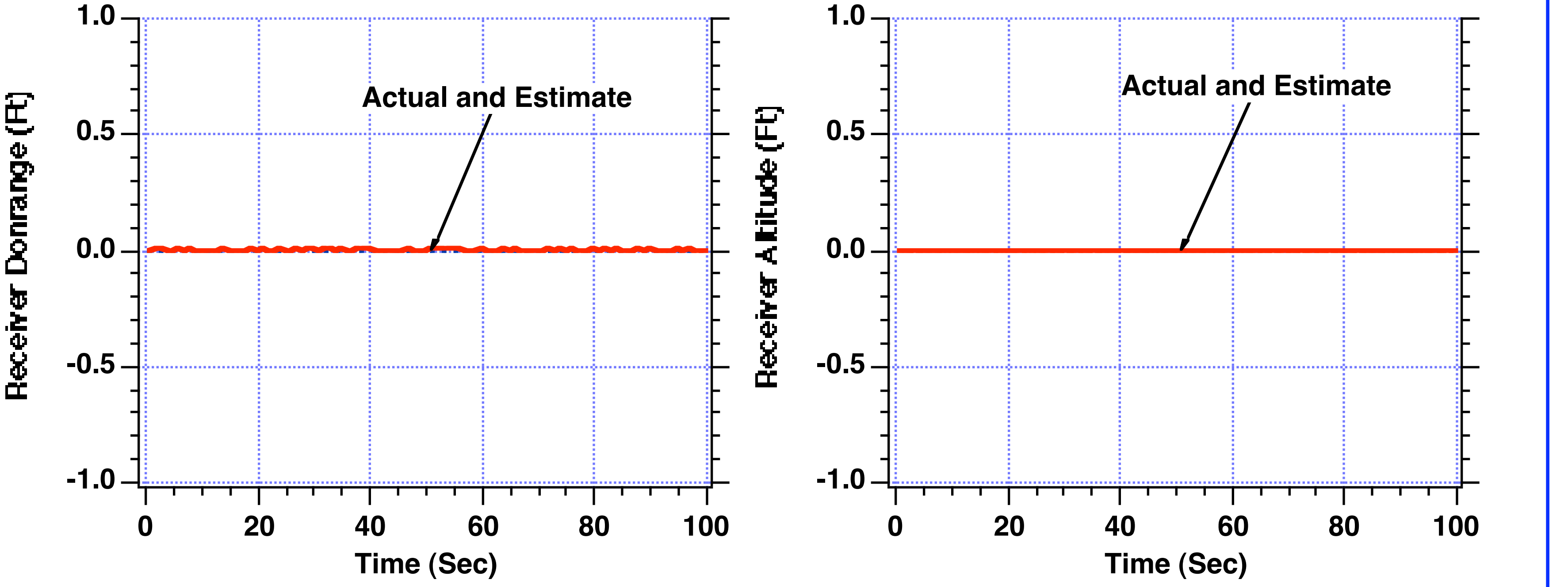
Numerical integration of satellite differential equations using second-order Runge-Kutta technique (XR1OLD=XR1 through XR2=.5*(XR2OLD+XR2+H*XR2D))

FORTRAN Code to See if Receiver Location can be Determined From Perfect Range Measurements-2

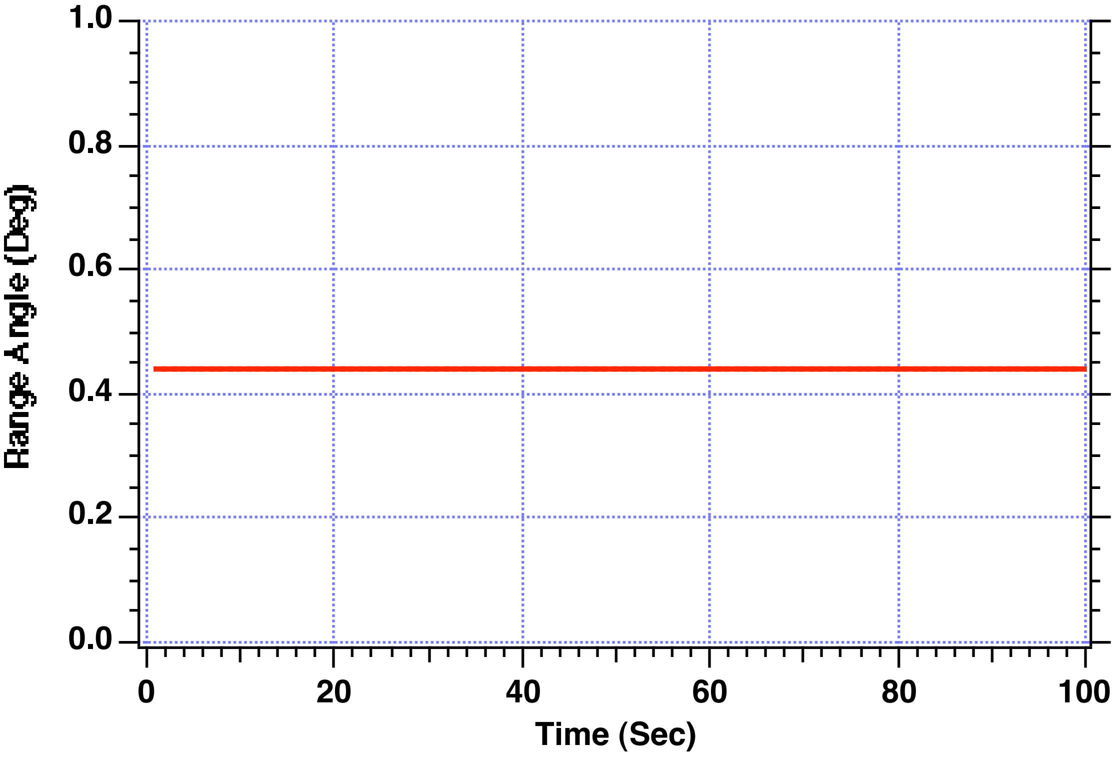
```
                IF(S>=(TS-.00001))THEN
                        S=0.
                        R1=SQRT((XR1-X)**2+(YR1-Y)**2)
                        R2=SQRT((XR2-X)**2+(YR2-Y)**2)
                        A1=(YR1-YR2)/(XR2-XR1)
                        B1=(R1**2-R2**2-XR1**2-YR1**2+XR2**2+YR2**2)/
     1                          (2.*(XR2-XR1))
                        A=1.+A1**2
                        B=2.*A1*B1-2.*A1*XR1-2.*YR1
                        C=XR1**2-2.*XR1*B1+YR1**2-R1**2
                        YH=(-B-SQRT(B**2-4.*A*C))/(2.*A)
                        XH=A1*YH+B1
                        THET=ACOS(((XR1-X)*(XR2-X)+(YR1-Y)*(YR2-Y))/
     1                          (R1*R2))
                        WRITE(9,*)T,X,XH,Y,YH,57.3*THET
                        WRITE(1,*)T,X,XH,Y,YH,57.3*THET
                ENDIF
        END DO
        PAUSE
        CLOSE(1)
        END
```

Finding receiver location from derived formulas

We Can Estimate Receiver Location Perfectly if Range Measurements are Perfect



Angle Between Range Vectors is Small and Approximately Constant For 100 Sec of Satellite Travel



Solving for Receiver Location Based on Noisy Range Measurements From Two Satellites (No Filtering)

FORTRAN Simulation to See if Receiver Location Can be Determined From Two Noisy Range Measurements-1

```
GLOBAL DEFINE
        INCLUDE 'quickdraw.inc'
END
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
SIGNOISE=300.
X=0.
Y=0.
XR1=1000000.
YR1=20000.*3280.
XR2=500000.
YR2=20000.*3280.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
TS=1.
TF=100.
T=0.
S=0.
H=.01
WHILE(T<=TF)
        XR1OLD=XR1
        XR2OLD=XR2
        XR1D=-14600.
        XR2D=-14600.
        XR1=XR1+H*XR1D
        XR2=XR2+H*XR2D
        T=T+H
        XR1D=-14600.
        XR2D=-14600.
        XR1=.5*(XR1OLD+XR1+H*XR1D)
        XR2=.5*(XR2OLD+XR2+H*XR2D)
        S=S+H
```

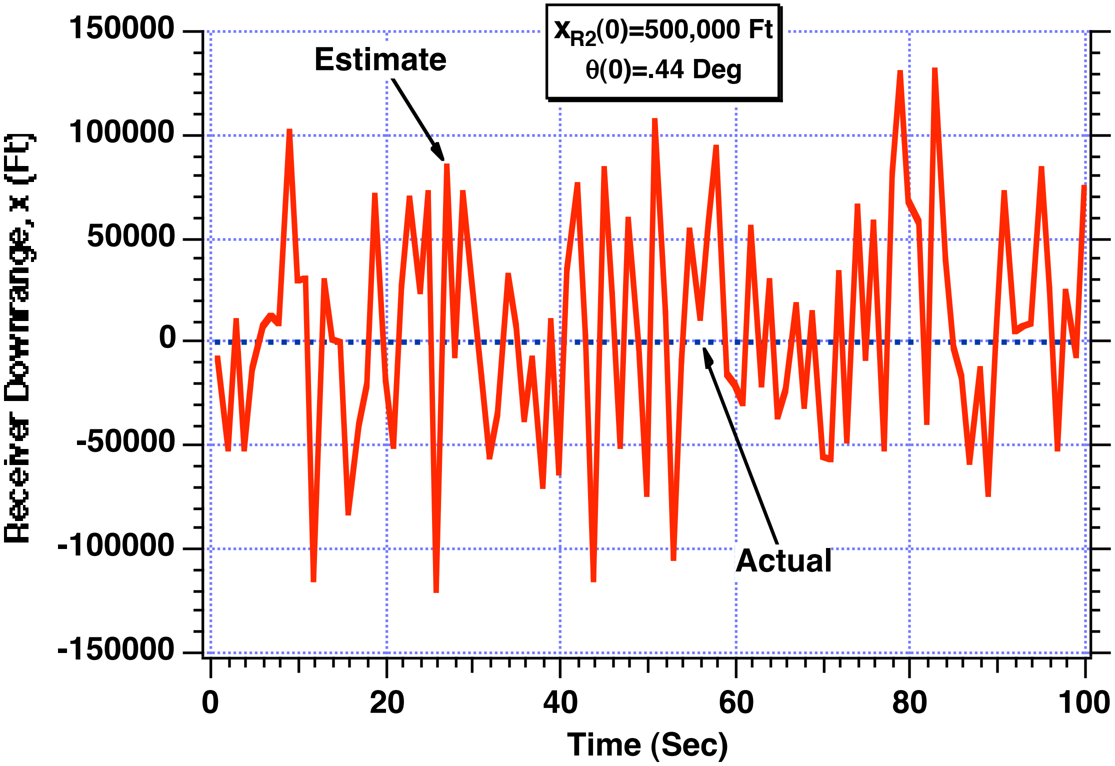
Standard deviation of noise on range

FORTRAN Simulation to See if Receiver Location Can be Determined From Two Noisy Range Measurements-2

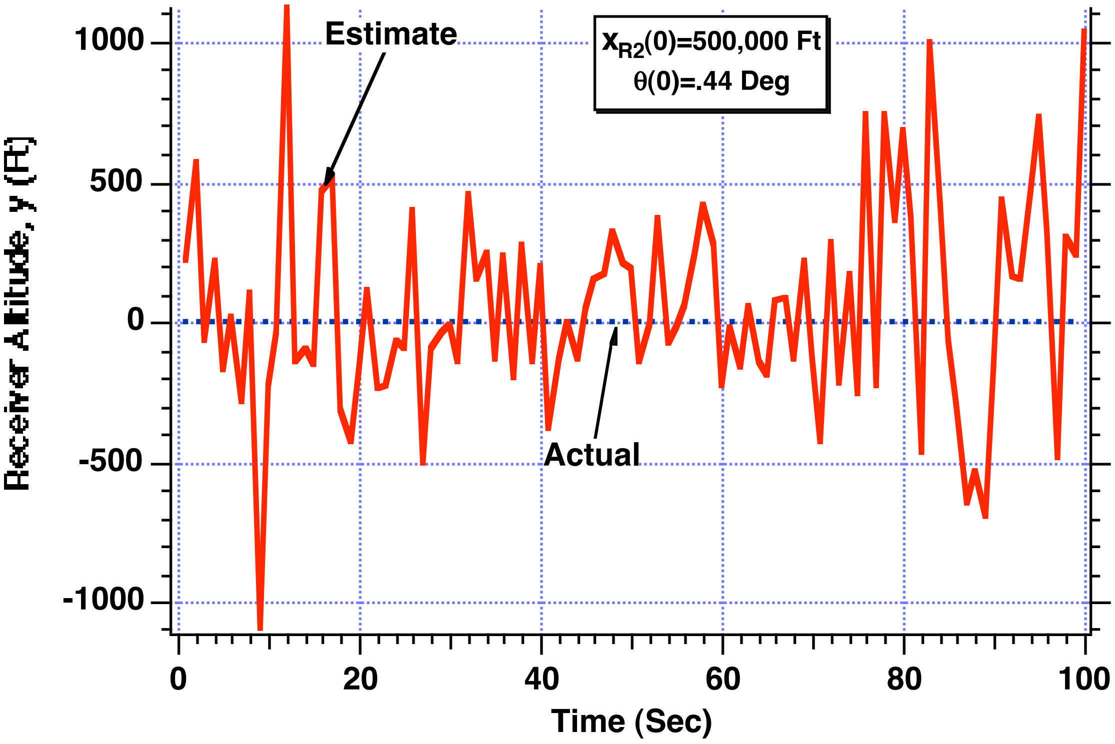
```
                    IF(S>=(TS-.00001))THEN
                              S=0.
                              CALL GAUSS(R1NOISE,SIGNOISE)
                              CALL GAUSS(R2NOISE,SIGNOISE)
                              R1=SQRT((XR1-X)**2+(YR1-Y)**2)
                              R2=SQRT((XR2-X)**2+(YR2-Y)**2)
                              R1S=R1+R1NOISE
                              R2S=R2+R2NOISE
                              A1=(YR1-YR2)/(XR2-XR1)
                              B1=(R1S**2-R2S**2-XR1**2-YR1**2+XR2**2+YR2**2)/
     1                                  (2.*(XR2-XR1))
                              A=1.+A1**2
                              B=2.*A1*B1-2.*A1*XR1-2.*YR1
                              C=XR1**2-2.*XR1*B1+YR1**2-R1S**2
                              YH=(-B-SQRT(B**2-4.*A*C))/(2.*A)
                              XH=A1*YH+B1
                              THET=ACOS(((XR1-X)*(XR2-X)+(YR1-Y)*(YR2-Y))/
     1                                  (R1*R2))
                              WRITE(9,*)T,X,XH,Y,YH,57.3*THET,R1,R2
                              WRITE(1,*)T,X,XH,Y,YH,57.3*THET,R1,R2
                    ENDIF
          END DO
          PAUSE
          CLOSE(1)
          END
```

Noisy range measurements

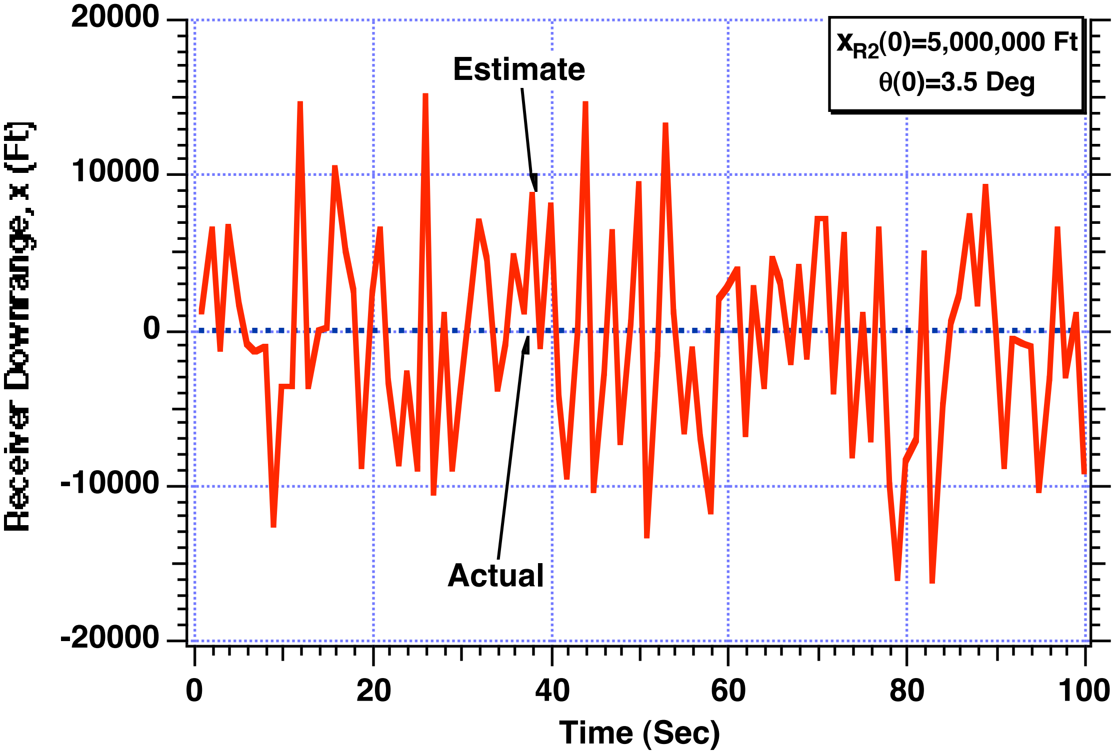
Using Raw Range Measurements Yields Large Downrange Receiver Location Errors



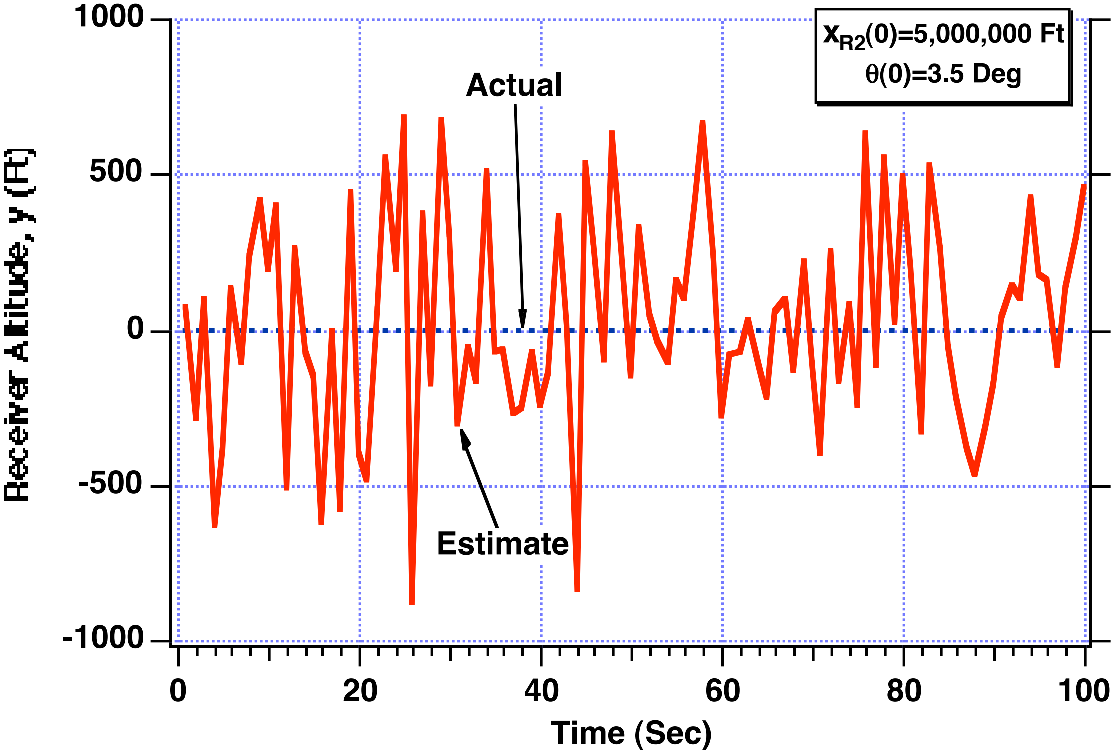
Using Raw Range Measurements Yields Large Altitude Receiver Location Errors



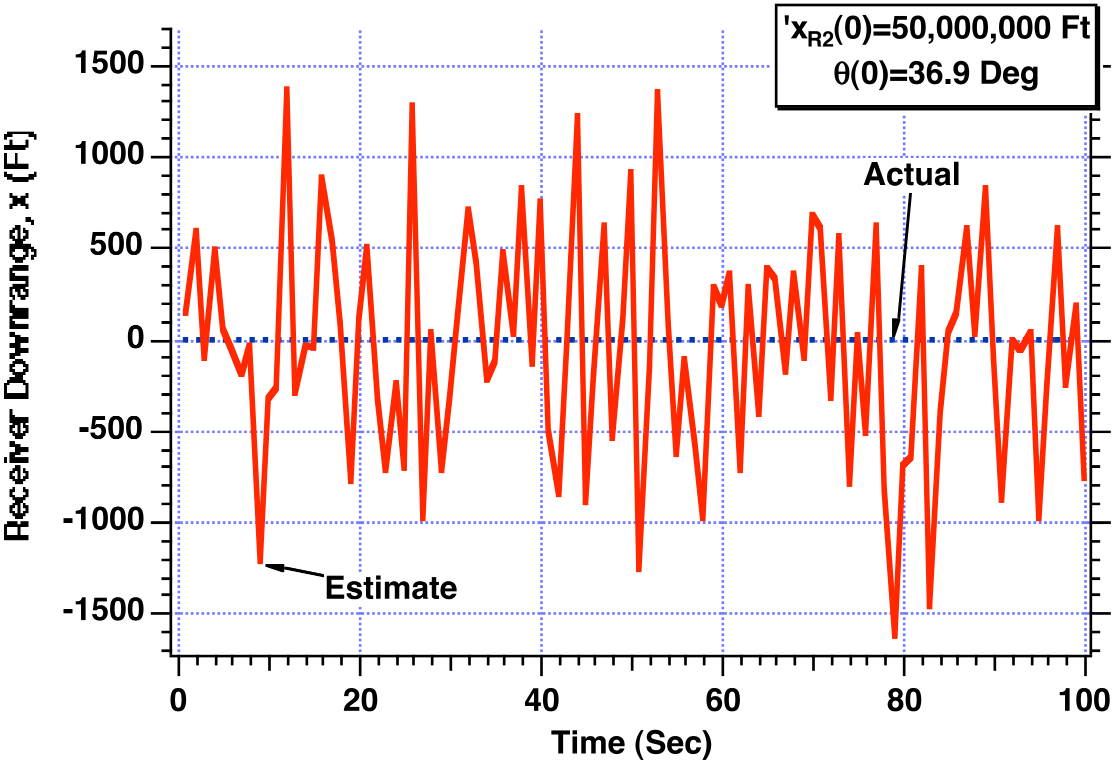
Receiver Downrange Errors Decrease by an Order of Magnitude When Geometrical Angle Between Satellites is Increased



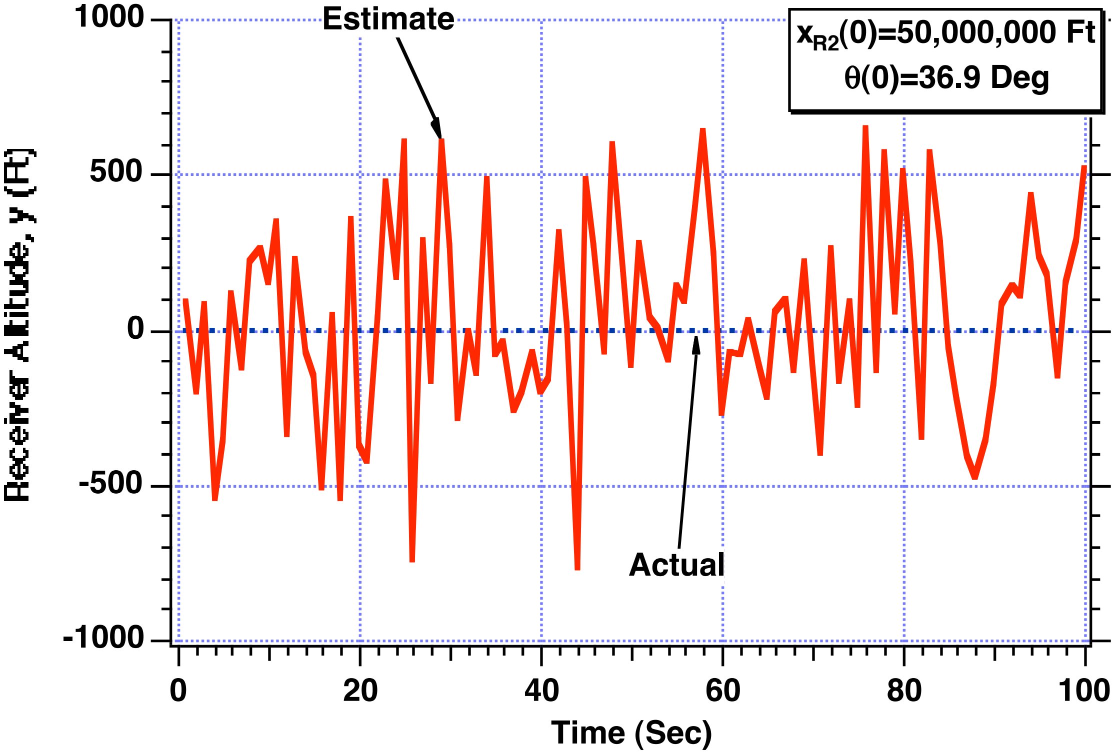
Receiver Altitude Errors Decrease Slightly When Geometrical Angle Between Satellites is Increased



Downrange Errors Decrease by Two Orders of Magnitude When Geometrical Angle is Increased by Two Orders of Magnitude



Altitude Errors Decrease Slightly When Geometrical Angle is Increased by Two Orders of Magnitude

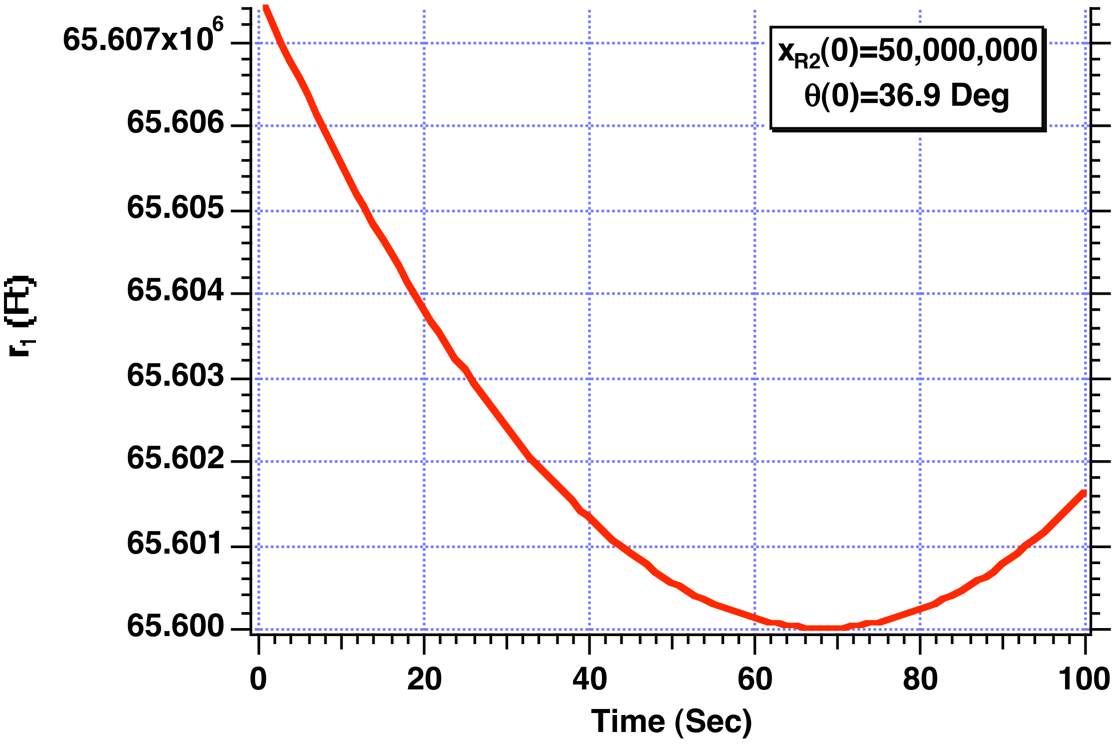


Summary When Filtering is Not Used

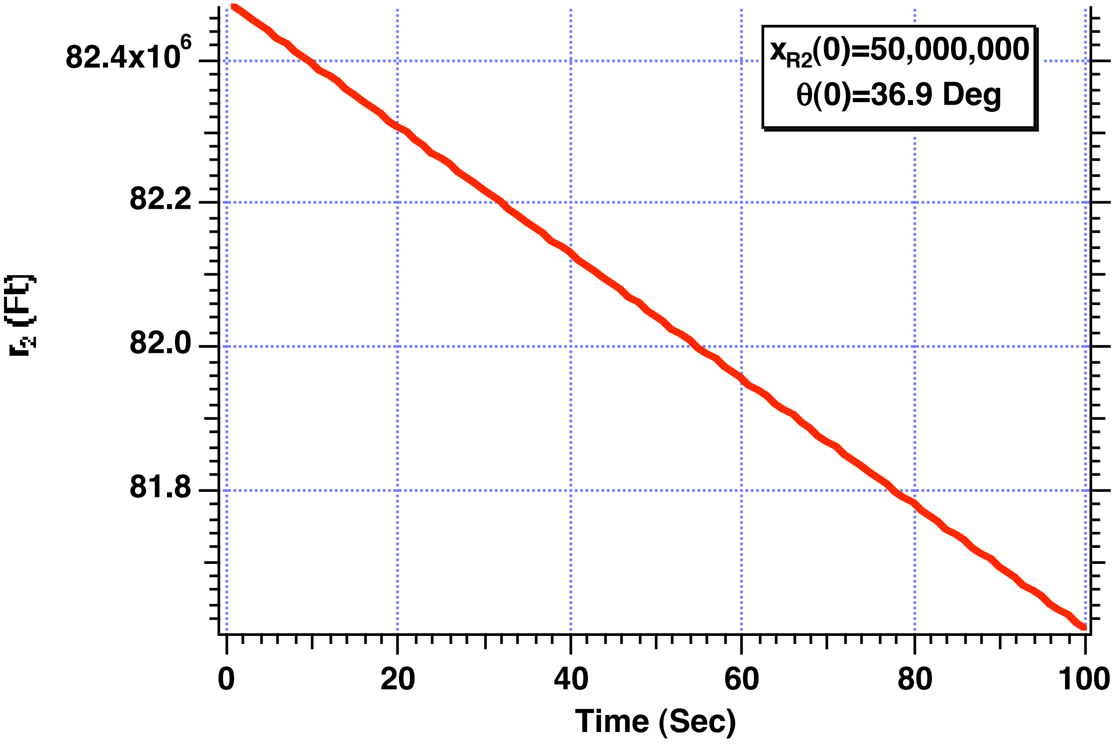
- Downrange errors decrease when range angle increases
- Altitude errors have weak dependence on range angle
- In best geometry range angle approaches 90 deg

Improvements With Linear Filtering of Range

Range from Receiver to First Satellite is Parabolic



Range From Receiver to Second Satellite is a Straight Line



We will play it safe and use second-order recursive least squares filter on each set of range measurements

Recursive Second-Order Least Squares Filter Review

Gains

$$K_{1_k} = \frac{3(3k^2-3k+2)}{k(k+1)(k+2)} \quad k=1,2,...,n$$

$$K_{2_k} = \frac{18(2k-1)}{k(k+1)(k+2)T_s}$$

$$K_{3_k} = \frac{60}{k(k+1)(k+2)T_s^2}$$

Filter

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

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

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

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

FORTRAN Simulation With Filtering on Noisy Range Measurements-1

```
GLOBAL DEFINE
          INCLUDE 'quickdraw.inc'
END
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
REAL*8 K1,K2,K3
SIGNOISE=300.
X=0.
Y=0.
XR1=1000000.
YR1=20000.*3280.
XR2=50000000.
YR2=20000.*3280.
R1H=0.
R1DH=0.
R1DDH=0.
R2H=0.
R2DH=0.
R2DDH=0.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
TS=1.
TF=100.
T=0.
S=0.
H=.01
XN=0.
WHILE(T<=TF)
          XR1OLD=XR1
          XR2OLD=XR2
          XR1D=-14600.
          XR2D=-14600.
          XR1=XR1+H*XR1D
          XR2=XR2+H*XR2D
          T=T+H
          XR1D=-14600.
          XR2D=-14600.
          XR1=.5*(XR1OLD+XR1+H*XR1D)
          XR2=.5*(XR2OLD+XR2+H*XR2D)
          S=S+H
```

Initial state estimates of both range filters

FORTRAN Simulation With Filtering on Noisy Range Measurements-2

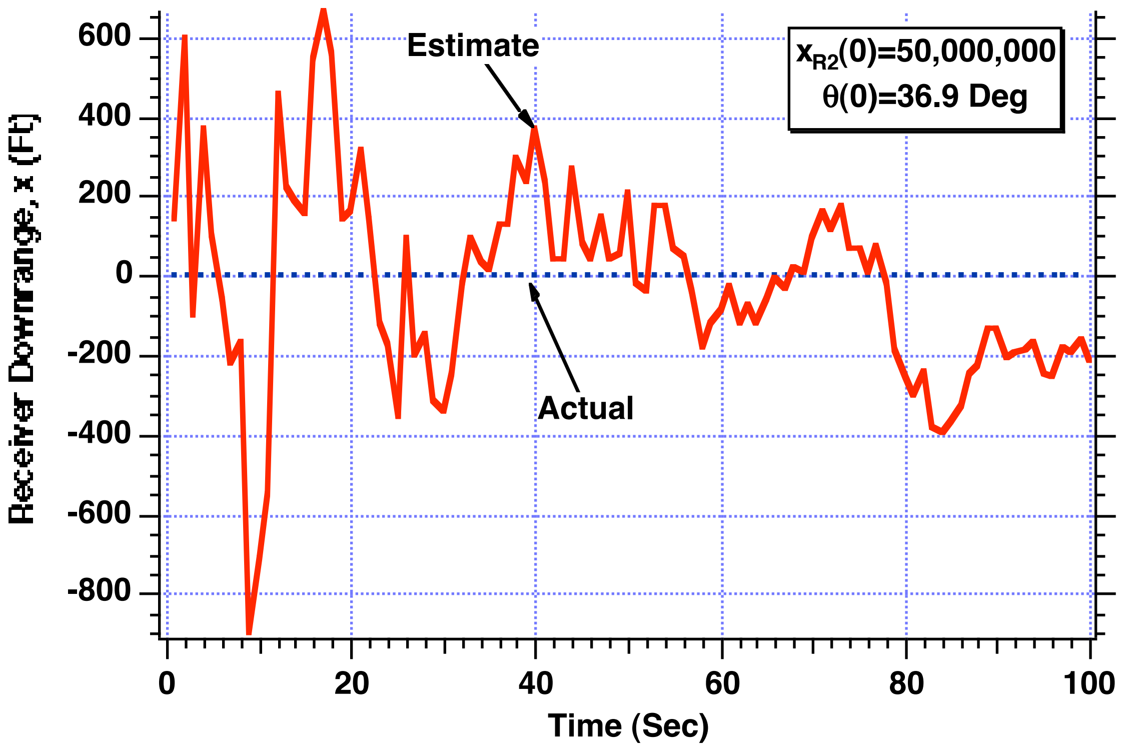
```
                        IF(S>=(TS-.00001))THEN
                                S=0.
                                XN=XN+1.
                                K1=3*(3*XN*XN-3*XN+2)/(XN*(XN+1)*(XN+2))
                                K2=18*(2*XN-1)/(XN*(XN+1)*(XN+2)*TS)
                                K3=60/(XN*(XN+1)*(XN+2)*TS*TS)
                                CALL GAUSS(R1NOISE,SIGNOISE)
                                CALL GAUSS(R2NOISE,SIGNOISE)
                                R1=SQRT((XR1-X)**2+(YR1-Y)**2)
                                R2=SQRT((XR2-X)**2+(YR2-Y)**2)
                                R1S=R1+R1NOISE
                                R2S=R2+R2NOISE
                                RES1=R1S-R1H-TS*R1DH-.5*TS*TS*R1DDH
                                R1H=R1H+R1DH*TS+.5*R1DDH*TS*TS+K1*RES1
                                R1DH=R1DH+R1DDH*TS+K2*RES1
                                R1DDH=R1DDH+K3*RES1
                                RES2=R2S-R2H-TS*R2DH-.5*TS*TS*R2DDH
                                R2H=R2H+R2DH*TS+.5*R2DDH*TS*TS+K1*RES2
                                R2DH=R2DH+R2DDH*TS+K2*RES2
                                R2DDH=R2DDH+K3*RES2
                                A1=(YR1-YR2)/(XR2-XR1)
                                B1=(R1H**2-R2H**2-XR1**2-YR1**2+XR2**2+YR2**2)/
1                                       (2.*(XR2-XR1))
                                A=1.+A1**2
                                B=2.*A1*B1-2.*A1*XR1-2.*YR1
                                C=XR1**2-2.*XR1*B1+YR1**2-R1H**2
                                YH=(-B-SQRT(B**2-4.*A*C))/(2.*A)
                                XH=A1*YH+B1
                                THET=ACOS(((XR1-X)*(XR2-X)+(YR1-Y)*(YR2-Y))/
1                                       (R1*R2))
                                WRITE(9,*)T,X,XH,Y,YH,57.3*THET,R1,R2
                                WRITE(1,*)T,X,XH,Y,YH,57.3*THET,R1,R2
                        ENDIF
                END DO
                PAUSE
                CLOSE(1)
                END
```

Filter gains

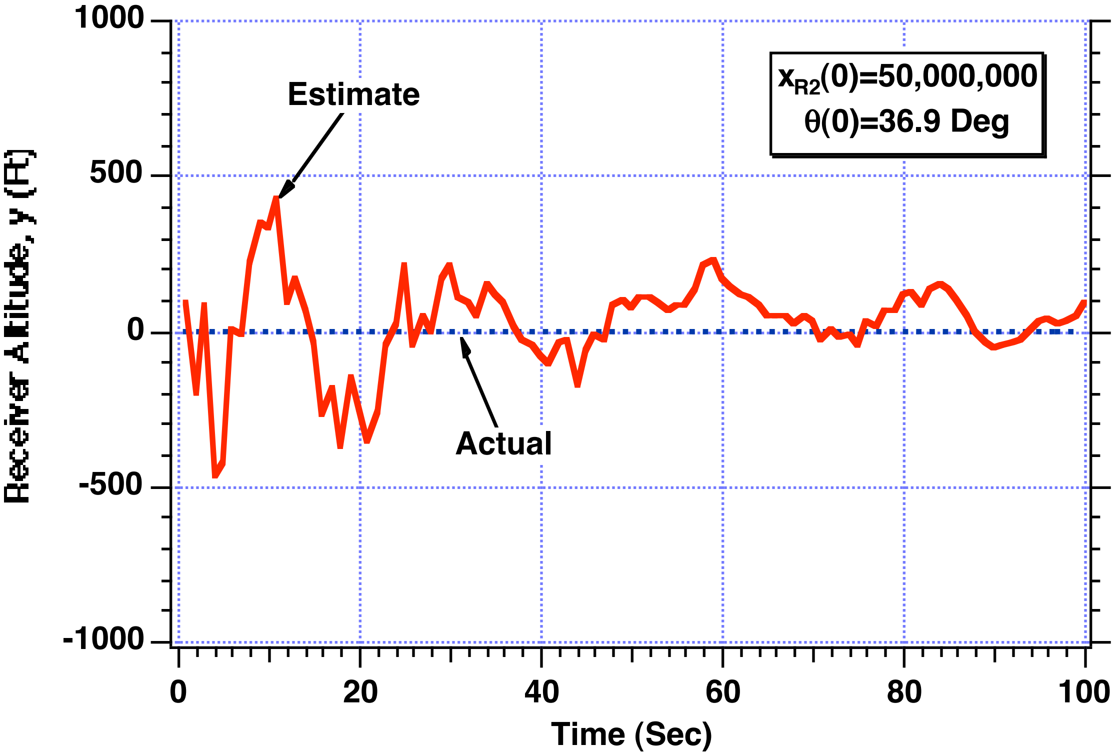
Filter equations

Deriving receiver location based on range estimates

Filtering Range Reduces Receiver Downrange Location Errors



Filtering Range Reduces Receiver Altitude Location Errors



Using Extended Kalman Filtering

Setting Up Problem-1

Receiver is stationary

$$\dot{x} = 0$$

$$\dot{y} = 0$$

Or in state space form without process noise

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Therefore systems dynamics matrix is zero

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

Fundamental matrix is identity matrix

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

Ranges from each satellite to receiver

$$r_1 = \sqrt{(x_{R1} - x)^2 + (y_{R1} - y)^2}$$

$$r_2 = \sqrt{(x_{R2} - x)^2 + (y_{R2} - y)^2}$$

Setting Up Problem-2

Linearized measurement equation

$$\begin{bmatrix} \Delta r_1^* \\ \Delta r_2^* \end{bmatrix} = \begin{bmatrix} \frac{\partial r_1}{\partial x} & \frac{\partial r_1}{\partial y} \\ \frac{\partial r_2}{\partial x} & \frac{\partial r_2}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} + \begin{bmatrix} v_{r1} \\ v_{r2} \end{bmatrix}$$

Measurement noise matrix

$$\mathbf{R_k} = \begin{bmatrix} \sigma_{r1}^2 & 0 \\ 0 & \sigma_{r2}^2 \end{bmatrix}$$

Linearized measurement matrix

$$\mathbf{H_k} = \begin{bmatrix} \frac{\partial r_1}{\partial x} & \frac{\partial r_1}{\partial y} \\ \frac{\partial r_2}{\partial x} & \frac{\partial r_2}{\partial y} \end{bmatrix}$$

Evaluate partial derivatives

$$r_1 = \sqrt{(x_{R1} - x)^2 + (y_{R1} - y)^2} \longrightarrow$$

$$\frac{\partial r_1}{\partial x} = .5\left[(x_{R1}-x)^2+(y_{R1}-y)^2\right]^{-.5} 2(x_{R1}-x)(-1) = \frac{-(x_{R1}-x)}{r_1}$$

$$\frac{\partial r_1}{\partial y} = .5\left[(x_{R1}-x)^2+(y_{R1}-y)^2\right]^{-.5} 2(y_{R1}-y)(-1) = \frac{-(y_{R1}-y)}{r_1}$$

$$r_2 = \sqrt{(x_{R2} - x)^2 + (y_{R2} - y)^2} \longrightarrow$$

$$\frac{\partial r_2}{\partial x} = .5\left[(x_{R1}-x)^2+(y_{R1}-y)^2\right]^{-.5} 2(x_{R2}-x)(-1) = \frac{-(x_{R2}-x)}{r_2}$$

$$\frac{\partial r_2}{\partial y} = .5\left[(x_{R1}-x)^2+(y_{R1}-y)^2\right]^{-.5} 2(y_{R2}-y)(-1) = \frac{-(y_{R2}-y)}{r_2}$$

Setting Up Problem-3

Partial derivatives are evaluated at projected state estimates which are also current state estimates in this example

$$\mathbf{H_k} = \begin{bmatrix} \frac{-(x_{R1}-x)}{r_1} & \frac{-(y_{R1}-y)}{r_1} \\ \frac{-(x_{R2}-x)}{r_2} & \frac{-(y_{R2}-y)}{r_2} \end{bmatrix}$$

Projected state estimates

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

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

Since fundamental matrix is identity matrix

Projected ranges from each satellite to receiver

$$\bar{r}_{1_k} = \sqrt{(x_{R1_k} - \bar{x}_k)^2 + (y_{R1_k} - \bar{y}_k)^2}$$

$$\bar{r}_{2_k} = \sqrt{(x_{R2_k} - \bar{x}_k)^2 + (y_{R2_k} - \bar{y}_k)^2}$$

Residual is calculated from nonlinear equation

$$RES_{1_k} = r^*_{1_k} - \bar{r}_{1_k}$$

$$RES_{2_k} = r^*_{2_k} - \bar{r}_{2_k}$$

Filtering equations

$$\hat{x}_k = \bar{x}_k + K_{11_k} RES_{1_k} + K_{12_k} RES_{2_k}$$

$$\hat{y}_k = \bar{y}_k + K_{21_k} RES_{1_k} + K_{22_k} RES_{2_k}$$

MATLAB Simulation of Extended Kalman Filter for Locating Receiver-1

```
SIGNOISE=300.;
X=0.;
Y=0.;
```

Actual receiver location

```
XH=1000.;
YH=2000.;
```

Initial filter estimate of receiver location

```
XR1=1000000.;
YR1=20000.*3280.;
XR1D=-14600.;
XR2=50000000.;
YR2=20000.*3280.;
XR2D=-14600.;
ORDER=2;
TS=1.;
TF=100.;
PHIS=0.;
T=0.;
S=0.;
H=.01;
PHI=zeros(ORDER,ORDER);
P=zeros(ORDER,ORDER);
IDNP=eye(ORDER);
Q=zeros(ORDER,ORDER);
P(1,1)=1000.^2;
P(2,2)=2000.^2;
```

Initial covariance matrix

```
RMAT(1,1)=SIGNOISE^2;
RMAT(1,2)=0.;
RMAT(2,1)=0.;
RMAT(2,2)=SIGNOISE^2;
```

Measurement noise matrix

```
count=0;
```

MATLAB Simulation of Extended Kalman Filter for Locating Receiver-2

```
while T<=TF
          XR1OLD=XR1;
          XR2OLD=XR2;
          XR1D=-14600.;
          XR2D=-14600.;
          XR1=XR1+H*XR1D;
          XR2=XR2+H*XR2D;
          T=T+H;
          XR1D=-14600.;
          XR2D=-14600.;
          XR1=.5*(XR1OLD+XR1+H*XR1D);
          XR2=.5*(XR2OLD+XR2+H*XR2D);
```

Integrate satellite equations with second-order Runge-Kutta technique

```
          S=S+H;
          if S>=(TS-.00001)
                    S=0.;
                    R1H=sqrt((XR1-XH)^2+(YR1-YH)^2);
                    R2H=sqrt((XR2-XH)^2+(YR2-YH)^2);
                    HMAT(1,1)=-(XR1-XH)/R1H;
                    HMAT(1,2)=-(YR1-YH)/R1H;
                    HMAT(2,1)=-(XR2-XH)/R2H;
                    HMAT(2,2)=-(YR2-YH)/R2H;
```

Linearized measurement matrix

```
                    HT=HMAT';
                    PHI(1,1)=1.;
                    PHI(2,2)=1.;
```

Fundamental matrix

```
                    PHIT=PHI;
                    PHIP=PHI*P;
                    PHIPPHIT=PHIP*PHIT;
                    M=PHIPPHIT+Q;
                    HM=HMAT*M;
                    HMHT=HM*HT;
                    HMHTR=HMHT+RMAT;
                    HMHTRINV=inv(HMHTR);
                    MHT=M*HT;
                    GAIN=MHT*HMHTRINV;
                    KH=GAIN*HMAT;
                    IKH=IDNP-KH;
                    P=IKH*M;
```

Riccati equations

MATLAB Simulation of Extended Kalman Filter for Locating Receiver-3

```
        R1NOISE=SIGNOISE*randn;
        R2NOISE=SIGNOISE*randn;
        R1=sqrt((XR1-X)^2+(YR1-Y)^2);
        R2=sqrt((XR2-X)^2+(YR2-Y)^2);
        RES1=R1+R1NOISE-R1H;
        RES2=R2+R2NOISE-R2H;
        XH=XH+GAIN(1,1)*RES1+GAIN(1,2)*RES2;
        YH=YH+GAIN(2,1)*RES1+GAIN(2,2)*RES2;
```

Filter

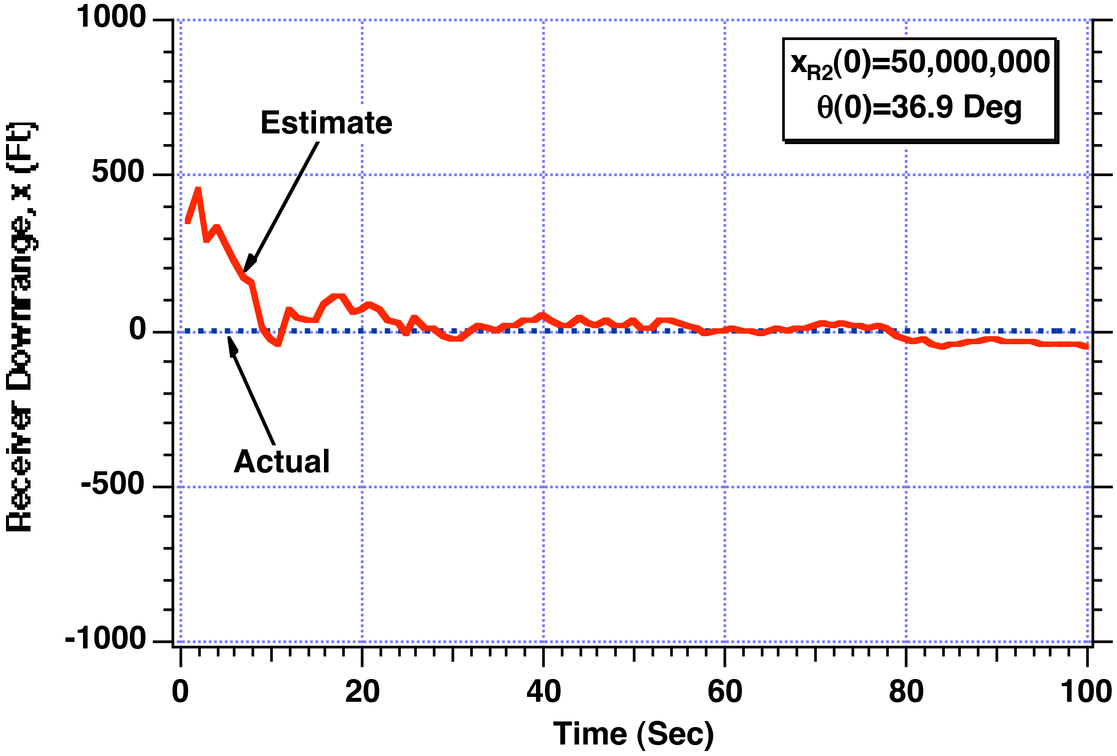
```
        ERRX=X-XH;
        SP11=sqrt(P(1,1));
        ERRY=Y-YH;
        SP22=sqrt(P(2,2));
        SP11P=-SP11;
        SP22P=-SP22;
```

Actual and theoretical errors in estimates

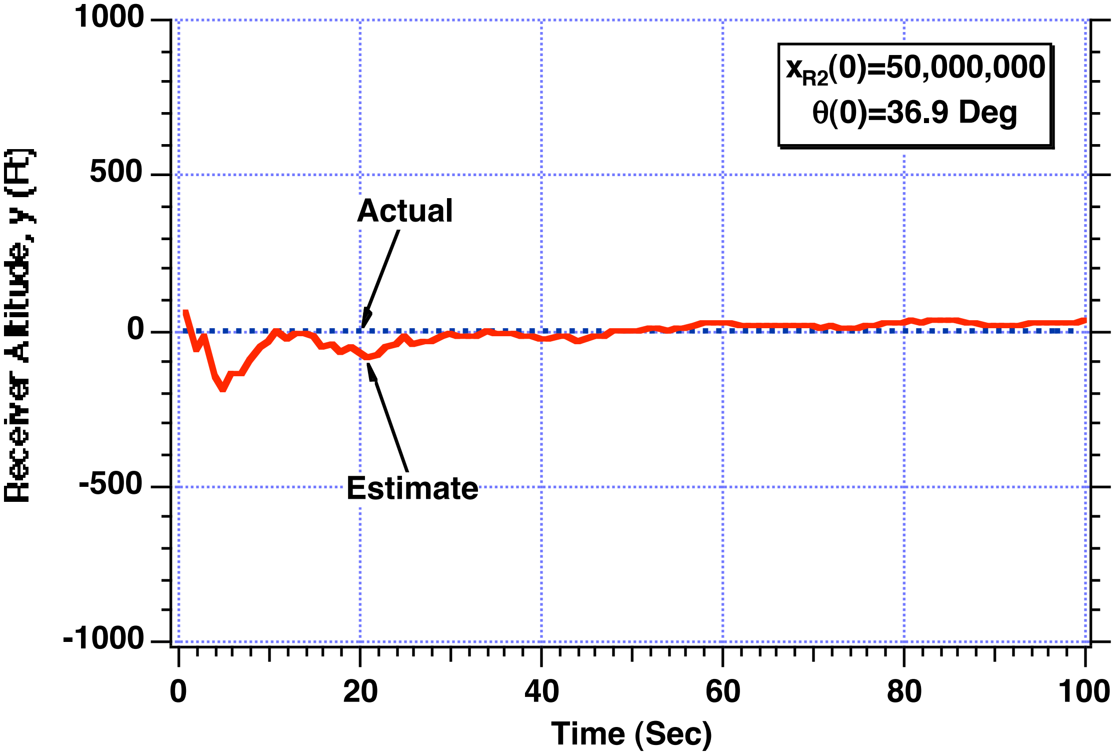
```
        count=count+1;
        ArrayT(count)=T;
        ArrayX(count)=X;
        ArrayXH(count)=XH;
        ArrayY(count)=Y;
        ArrayYH(count)=YH;
        ArrayERRX(count)=ERRX;
        ArraySP11(count)=SP11;
        ArraySP11P(count)=SP11P;
        ArrayERRY(count)=ERRY;
        ArraySP22(count)=SP22;
        ArraySP22P(count)=SP22P;
    end
end
```

Save data in arrays for plotting and writing to files

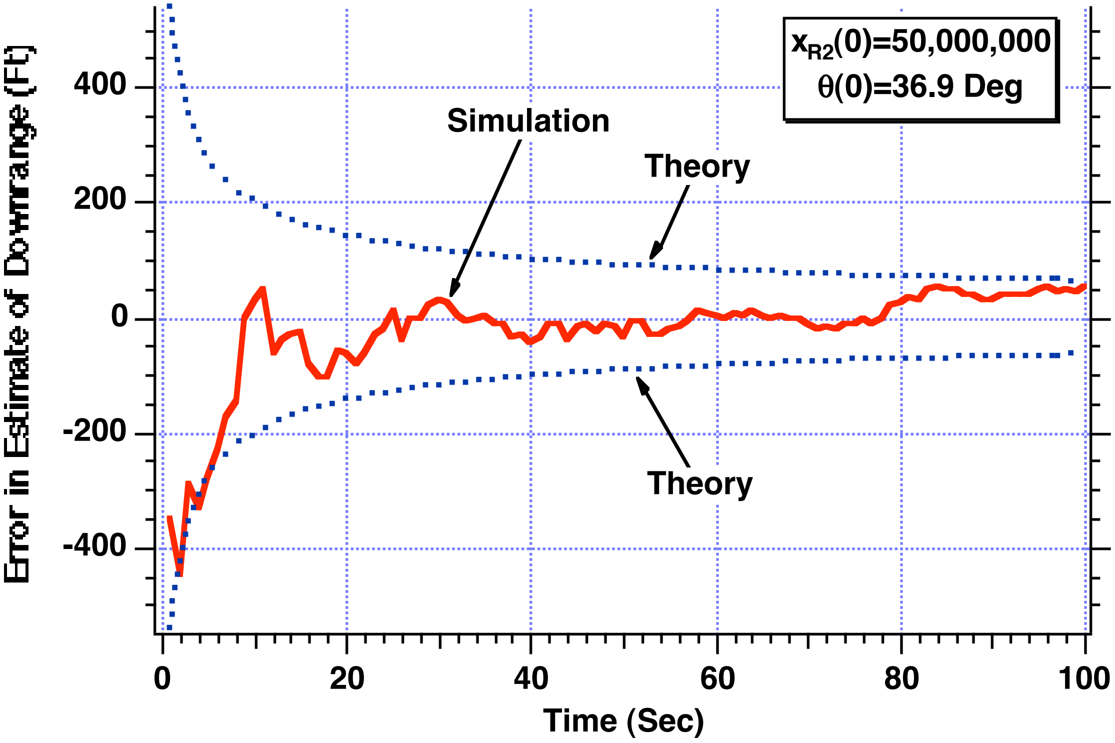
Extended Kalman Filtering Dramatically Reduces Receiver Downrange Location Errors



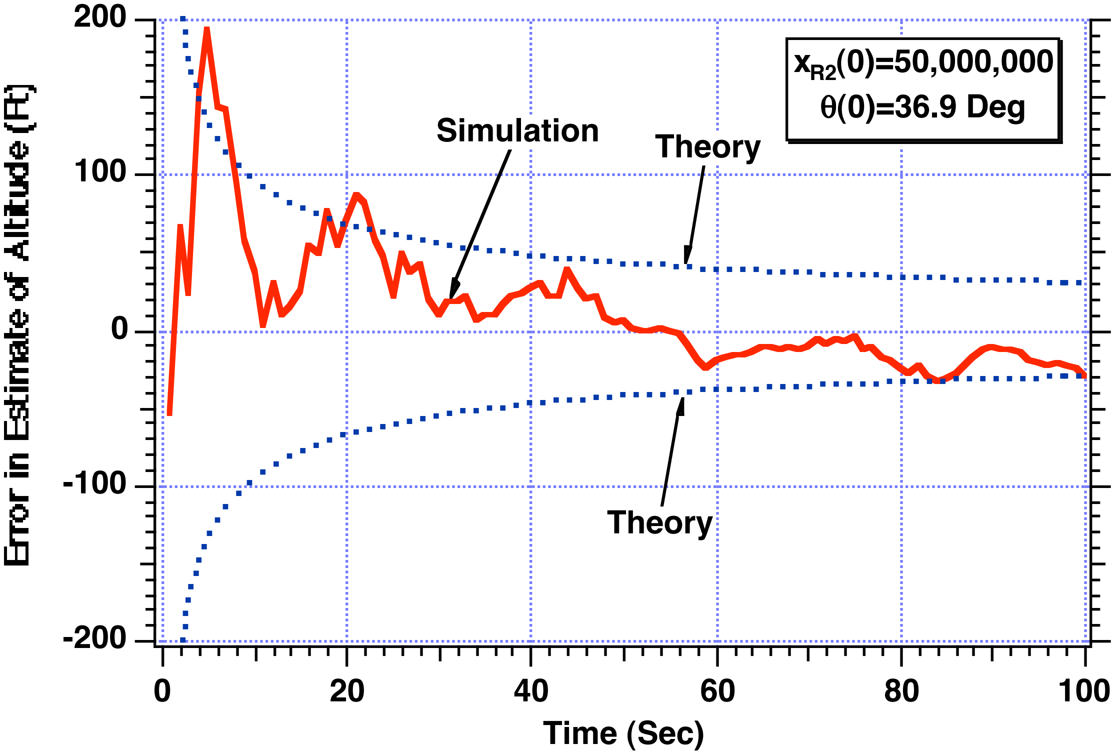
Extended Kalman Filtering Dramatically Reduces Receiver Altitude Location Errors



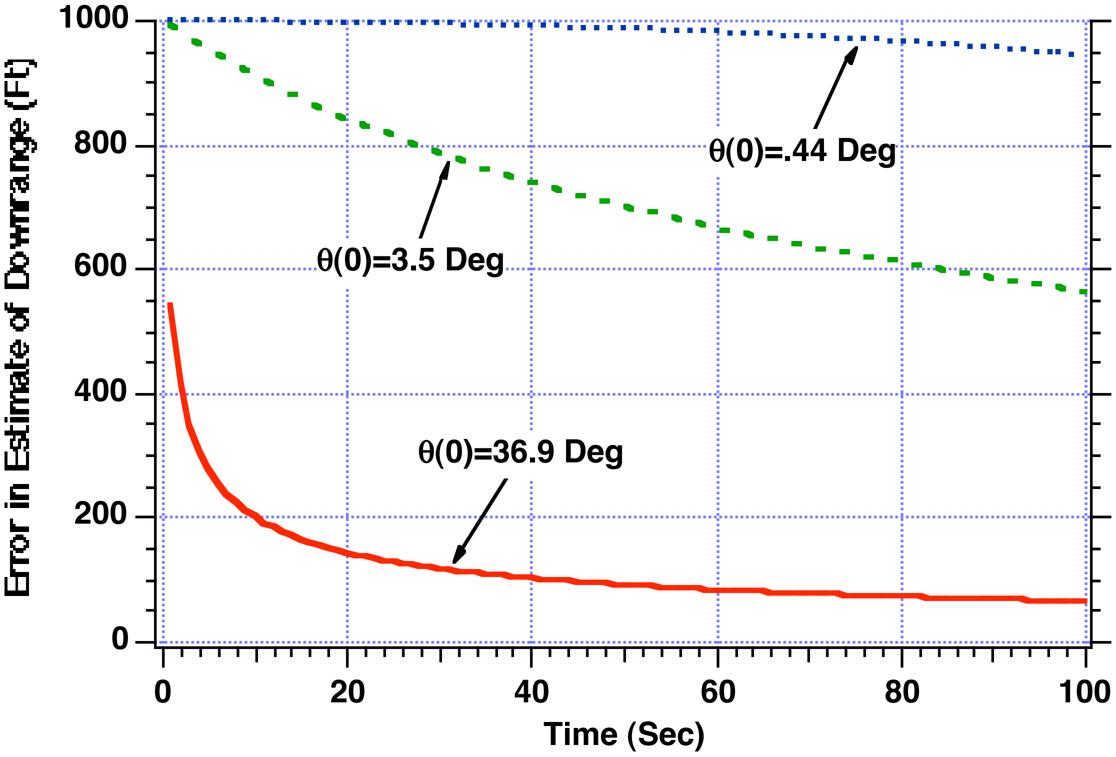
Extended Kalman Filter Appears to be Working Correctly in Downrange



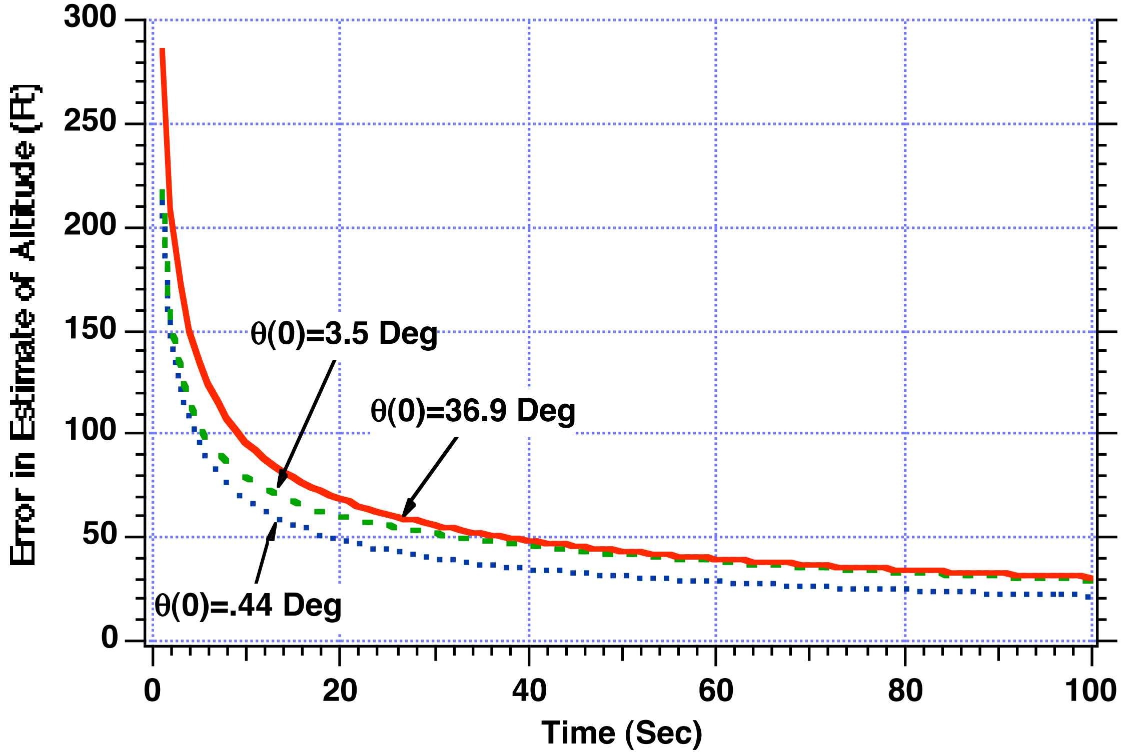
Extended Kalman Filter Appears to be Working Correctly in Altitude



Extended Kalman Filter's Downrange Estimation Errors Decrease With Increasing Angle Between Range Vectors



Extended Kalman Filter’s Altitude Estimation Errors Remain Approximately Constant With Increasing Angle Between Range Vectors



Using Extended Kalman Filtering With One Satellite

One Satellite Filter Formulation-1

Receiver is stationary

$$\dot{x} = 0$$

$$\dot{y} = 0$$

State space model of real world (no process noise)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Systems dynamics matrix is still zero

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

Fundamental matrix is still identity matrix

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

Range from satellite to receiver

$$r_1 = \sqrt{(x_{R1} - x)^2 + (y_{R1} - y)^2} \longrightarrow$$

One equation with two unknowns

Some people believe that this makes problem impossible

One Satellite Filter Formulation-2

New linearized measurement equation

$$\Delta r_1^* = \begin{bmatrix} \frac{\partial r_1}{\partial x} & \frac{\partial r_1}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} + v_{r1}$$

New measurement noise matrix is a scalar

$$\mathbf{R_k} = \sigma_{r1}^2$$

New linearized measurement matrix

$$\mathbf{H_k} = \begin{bmatrix} \frac{\partial r_1}{\partial x} & \frac{\partial r_1}{\partial y} \end{bmatrix}$$

Evaluating partial derivatives

$$r_1 = \sqrt{(x_{R1} - x)^2 + (y_{R1} - y)^2} \longrightarrow \begin{array}{l} \frac{\partial r_1}{\partial x} = .5\left[(x_{R1}-x)^2+(y_{R1}-y)^2\right]^{-.5} 2(x_{R1}-x)(-1) = \frac{-(x_{R1}-x)}{r_1} \\ \\ \frac{\partial r_1}{\partial y} = .5\left[(x_{R1}-x)^2+(y_{R1}-y)^2\right]^{-.5} 2(y_{R1}-y)(-1) = \frac{-(y_{R1}-y)}{r_1} \end{array}$$

Linearized measurement matrix

$$\mathbf{H_k} = \begin{bmatrix} \frac{-(x_{R1}-x)}{r_1} & \frac{-(y_{R1}-y)}{r_1} \end{bmatrix}$$

One Satellite Filter Formulation-3

Projected filter states

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

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

Projected range

$$\bar{r}_{1_k} = \sqrt{(x_{R1_k} - \bar{x}_k)^2 + (y_{R1_k} - \bar{y}_k)^2}$$

New extended Kalman filter

$$RES_{1_k} = r^*_{1_k} - \bar{r}_{1_k}$$

$$\hat{x}_k = \bar{x}_k + K_{11_k} RES_{1_k}$$

$$\hat{y}_k = \bar{y}_k + K_{21_k} RES_{1_k}$$

MATLAB Extended Kalman Filter for Locating Receiver Based on Measurements From 1 Satellite-1

```
SIGNOISE=300.;
X=0.;
Y=0.;
XH=1000.;
YH=2000.;
```

Initial estimate of receiver location

```
XR1=1000000.;
YR1=20000.*3280.;
ORDER=2;
TS=1.;
TF=100.;
T=0.;
S=0.;
H=.01;
PHI=zeros(ORDER,ORDER);
P=zeros(ORDER,ORDER);
IDNP=eye(ORDER);
Q=zeros(ORDER,ORDER);
P(1,1)=1000.^2;
P(2,2)=2000.^2;
```

Initial covariance matrix

```
RMAT(1,1)=SIGNOISE^2;
count=0;
while T<=TF
        XR1OLD=XR1;
        XR1D=-14600.;
        XR1=XR1+H*XR1D;
        T=T+H;
        XR1D=-14600.;
        XR1=.5*(XR1OLD+XR1+H*XR1D);
        S=S+H;
```

Integrate satellite equations using second-order Runge-Kutta technique

MATLAB Extended Kalman Filter for Locating Receiver Based on Measurements From 1 Satellite-2

```
if S>=(TS-.00001)
        S=0.;
        R1H=sqrt((XR1-XH)^2+(YR1-YH)^2);
        HMAT(1,1)=-(XR1-XH)/R1H;
        HMAT(1,2)=-(YR1-YH)/R1H;
```

Linearized measurement matrix

```
        HT=HMAT';
        PHI(1,1)=1.;
        PHI(2,2)=1.;
```

Fundamental matrix

```
        PHIT=PHI;
        PHIP=PHI*P;
        PHIPPHIT=PHIP*PHIT;
        M=PHIPPHIT+Q;
        HM=HMAT*M;
        HMHT=HM*HT;
        HMHTR=HMHT+RMAT;
        HMHTRINV=inv(HMHTR);
        MHT=M*HT;
        GAIN=MHT*HMHTRINV;
        KH=GAIN*HMAT;
        IKH=IDNP-KH;
        P=IKH*M;
```

Riccati equations

```
        R1NOISE=SIGNOISE*randn;
        R1=sqrt((XR1-X)^2+(YR1-Y)^2);
        RES1=R1+R1NOISE-R1H;
        XH=XH+GAIN(1,1)*RES1;
        YH=YH+GAIN(2,1)*RES1;
```

Filter

```
        ERRX=X-XH;
        SP11=sqrt(P(1,1));
        ERRY=Y-YH;
        SP22=sqrt(P(2,2));
        SP11P=-SP11;
        SP22P=-SP22;
```

Actual and theoretical errors in estimates

MATLAB Extended Kalman Filter for Locating Receiver Based on Measurements From 1 Satellite-3

```
        count=count+1;
        ArrayT(count)=T;
        ArrayX(count)=X;
        ArrayXH(count)=XH;
        ArrayY(count)=Y;
        ArrayYH(count)=YH;
        ArrayERRX(count)=ERRX;
        ArraySP11(count)=SP11;
        ArraySP11P(count)=SP11P;
        ArrayERRY(count)=ERRY;
        ArraySP22(count)=SP22;
        ArraySP22P(count)=SP22P;
```

Save data as arrays

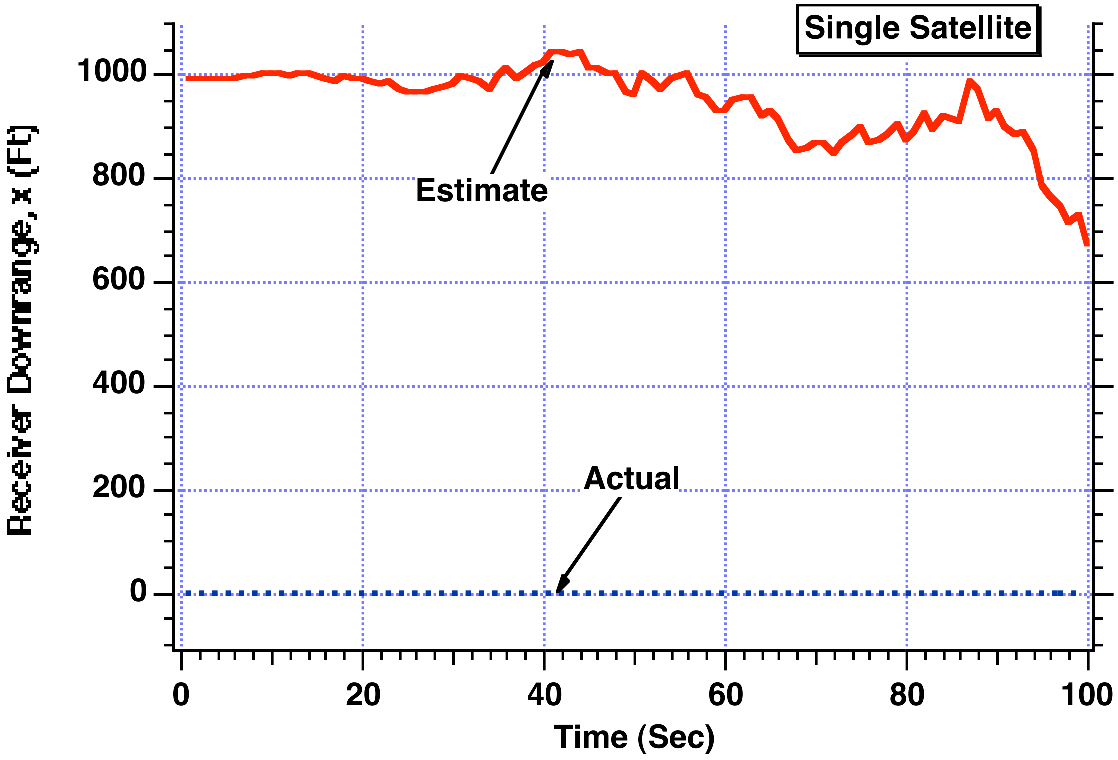
```
    end
end
figure
plot(ArrayT,ArrayX,ArrayT,ArrayXH),grid
xlabel('Time (Sec)')
ylabel('Receiver Downrange (Ft)')
axis([0 100 -100 1100])
figure
plot(ArrayT,ArrayY,ArrayT,ArrayYH),grid
xlabel('Time (Sec)')
ylabel('Receiver Altitude (Ft)')
axis([0 100 -150 150])
```

Plot some results

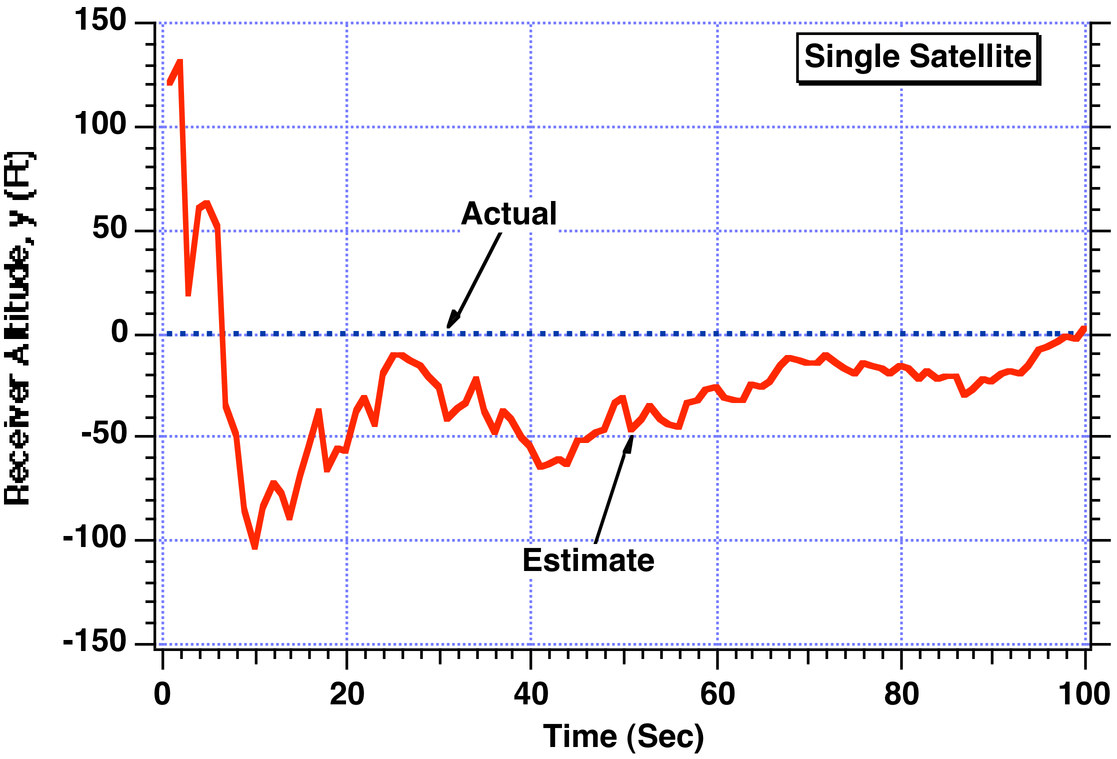
```
clc
output=[ArrayT',ArrayX',ArrayXH',ArrayY',ArrayYH'];
save datfil output  -ascii
output=[ArrayT',ArrayERRX',ArraySP11',ArraySP11P',ArrayERRY',ArraySP22',...
ArraySP22P'];
save covfil output  -ascii
disp 'simulation finished'
```

Write data to files

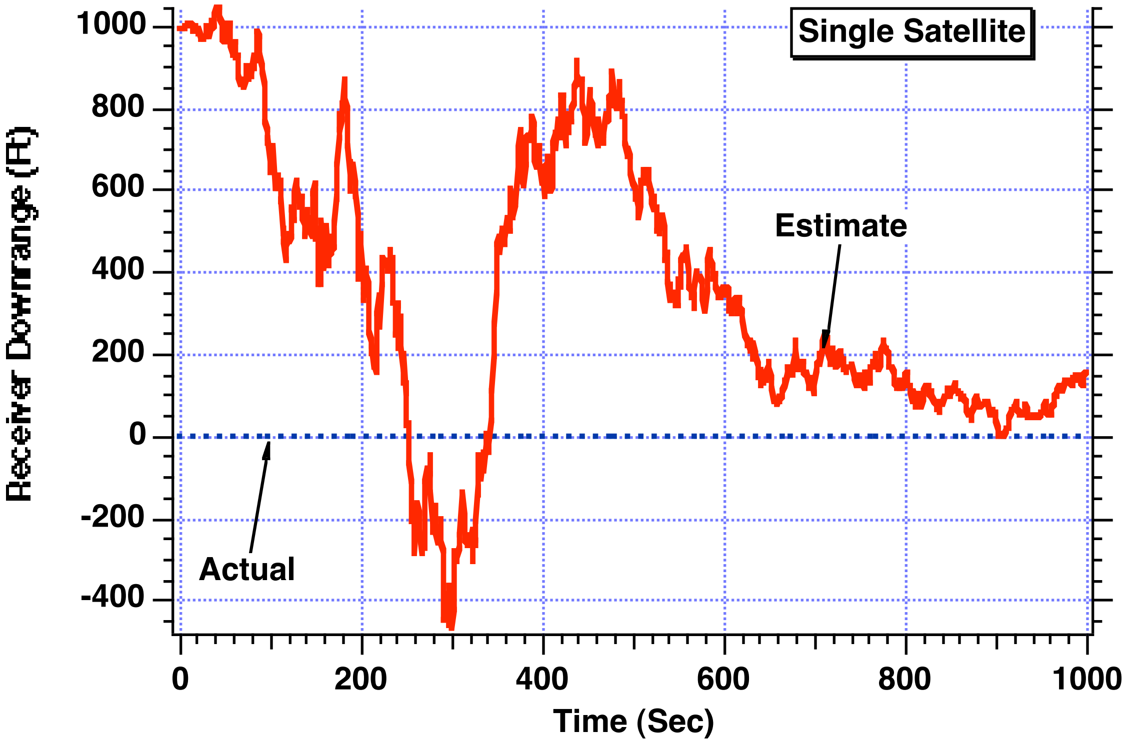
It is Not Clear if Filter Can Estimate Receiver Downrange Location if Only One Satellite is Used



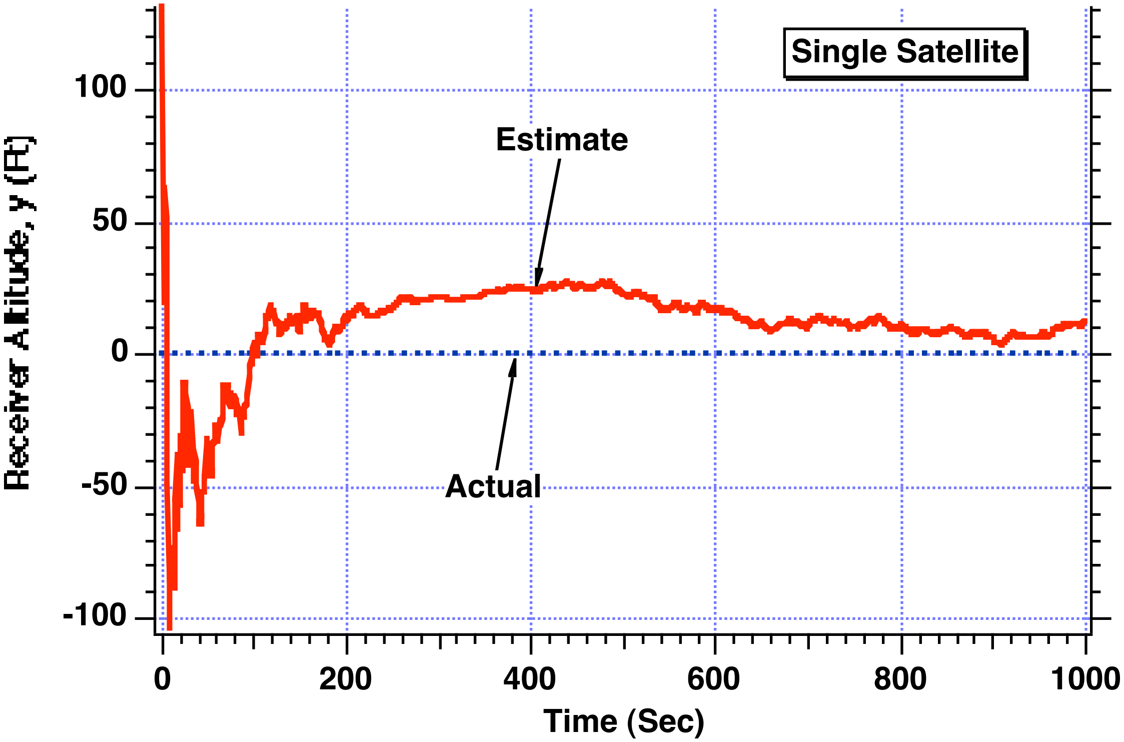
Filter Appears to be Able to Estimate Receiver Altitude if Only One Satellite is Used



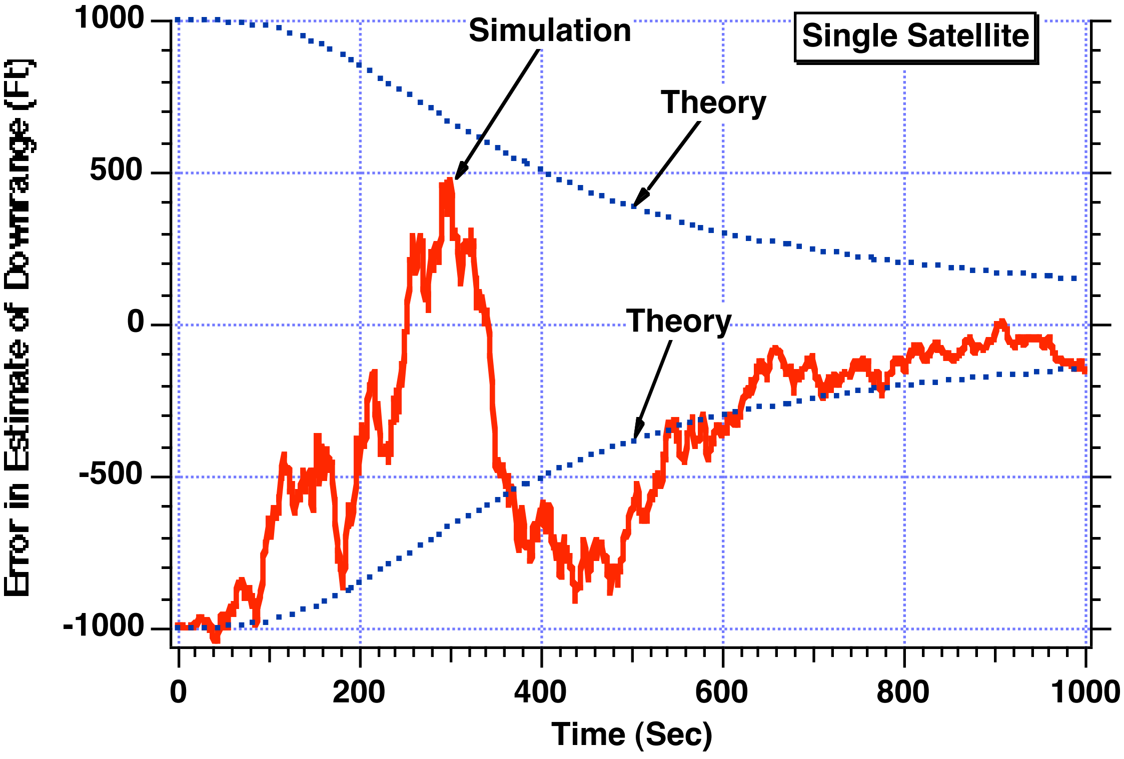
Filter Appears to be Able to Estimate Receiver Downrange After Approximately 600 Sec if Only One Satellite is Used



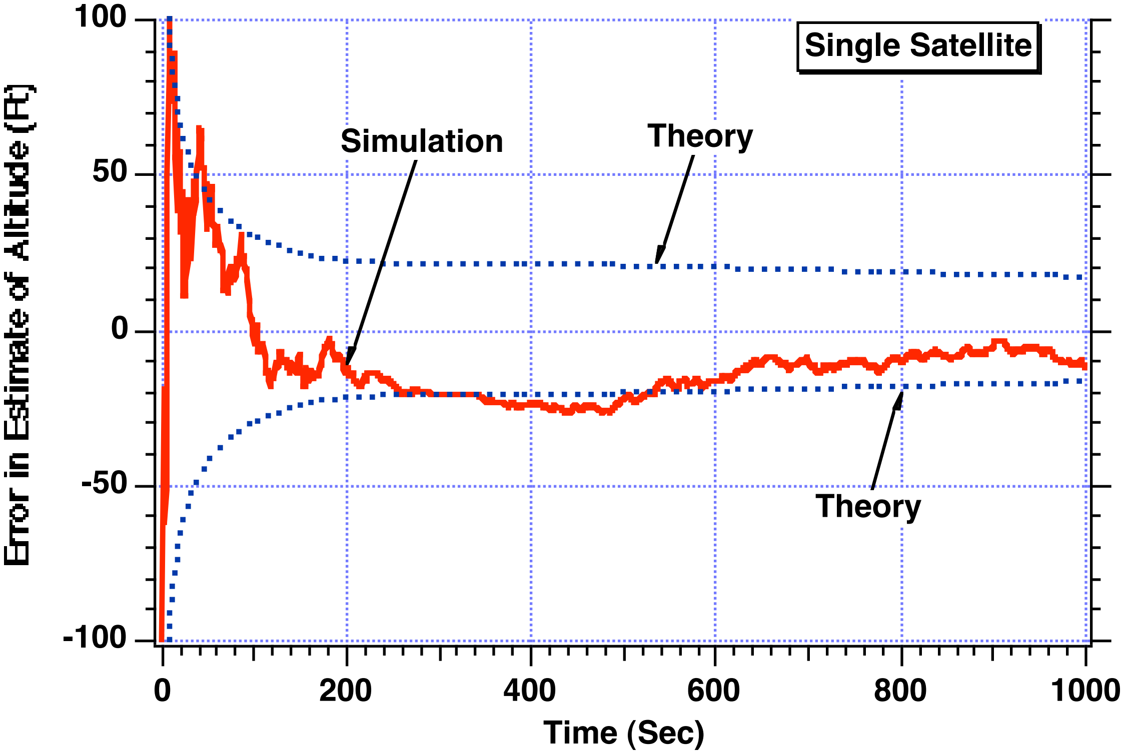
Filter Appears to be Able to Estimate Receiver Altitude if Only One Satellite is Used



Extended Kalman Filter Appears to be Operating Properly in Downrange



Extended Kalman Filter Appears to be Operating Properly in Altitude



Using Extended Kalman Filtering With Constant Velocity Receiver

Developing New Extended Kalman Filter-1

Model of real world for moving receiver

$$\ddot{x} = u_s$$

$$\ddot{y} = u_s$$

Put model in state space form

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

Continuous process noise matrix

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

Systems dynamics matrix

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

Developing New Extended Kalman Filter-2

Since F squared is zero

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

Therefore the discrete fundamental matrix is

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

Range from each satellite to receiver is

$$r_1 = \sqrt{(x_{R1} - x)^2 + (y_{R1} - y)^2}$$

$$r_2 = \sqrt{(x_{R2} - x)^2 + (y_{R2} - y)^2}$$

Linearized measurement equation

$$\begin{bmatrix} \Delta r_1^* \\ \Delta r_2^* \end{bmatrix} = \begin{bmatrix} \frac{\partial r_1}{\partial x} & \frac{\partial r_1}{\partial \dot{x}} & \frac{\partial r_1}{\partial y} & \frac{\partial r_1}{\partial \dot{y}} \\ \frac{\partial r_2}{\partial x} & \frac{\partial r_2}{\partial \dot{x}} & \frac{\partial r_2}{\partial y} & \frac{\partial r_1}{\partial \dot{y}} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \dot{x} \\ \Delta y \\ \Delta \dot{y} \end{bmatrix} + \begin{bmatrix} v_{r1} \\ v_{r2} \end{bmatrix}$$

Developing New Extended Kalman Filter-3

Discrete measurement noise matrix

$$\mathbf{R_k} = \begin{bmatrix} \sigma_{r1}^2 & 0 \\ 0 & \sigma_{r2}^2 \end{bmatrix}$$

Linearized measurement equation

$$\mathbf{H_k} = \begin{bmatrix} \frac{\partial r_1}{\partial x} & \frac{\partial r_1}{\partial \dot{x}} & \frac{\partial r_1}{\partial y} & \frac{\partial r_1}{\partial \dot{y}} \\ \frac{\partial r_2}{\partial x} & \frac{\partial r_2}{\partial \dot{x}} & \frac{\partial r_2}{\partial y} & \frac{\partial r_1}{\partial \dot{y}} \end{bmatrix}$$

Where partial derivatives evaluated at projected state estimates

Evaluation of first row of partial derivatives

$$r_1 = \sqrt{(x_{R1} - x)^2 + (y_{R1} - y)^2} \longrightarrow$$

$$\frac{\partial r_1}{\partial x} = .5\left[(x_{R1}-x)^2+(y_{R1}-y)^2\right]^{-.5} 2(x_{R1}-x)(-1) = \frac{-(x_{R1}-x)}{r_1}$$

$$\frac{\partial r_1}{\partial \dot{x}} = 0$$

$$\frac{\partial r_1}{\partial y} = .5\left[(x_{R1}-x)^2+(y_{R1}-y)^2\right]^{-.5} 2(y_{R1}-y)(-1) = \frac{-(y_{R1}-y)}{r_1}$$

$$\frac{\partial r_1}{\partial \dot{y}} = 0$$

Developing New Extended Kalman Filter-4

Evaluation of second row of partial derivatives

$$r_2 = \sqrt{(x_{R2} - x)^2 + (y_{R2} - y)^2} \longrightarrow$$

$$\frac{\partial r_2}{\partial x} = .5\left[(x_{R1}-x)^2+(y_{R1}-y)^2\right]^{-.5} 2(x_{R2}-x)(-1) = \frac{-(x_{R2}-x)}{r_2}$$

$$\frac{\partial r_2}{\partial \dot{x}} = 0$$

$$\frac{\partial r_2}{\partial y} = .5\left[(x_{R1}-x)^2+(y_{R1}-y)^2\right]^{-.5} 2(y_{R2}-y)(-1) = \frac{-(y_{R2}-y)}{r_2}$$

$$\frac{\partial r_2}{\partial \dot{y}} = 0$$

Linearized measurement matrix

$$\mathbf{H_k} = \begin{bmatrix} \frac{-(x_{R1}-x)}{r_1} & 0 & \frac{-(y_{R1}-y)}{r_1} & 0 \\ \frac{-(x_{R2}-x)}{r_2} & 0 & \frac{-(y_{R2}-y)}{r_2} & 0 \end{bmatrix}$$

Discrete process noise matrix can be derived from continuous Q

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

Developing New Extended Kalman Filter-5

Substitution yields

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

After multiplication and integration

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

Since fundamental matrix exact projected states are $\bar{\mathbf{x}}_\mathbf{k} = \Phi\, \hat{\mathbf{x}}_{\mathbf{k}-1}$ or

$$\bar{x}_k = \hat{x}_{k-1} + T_s\hat{\dot{x}}_{k-1} \qquad \bar{y}_k = \hat{y}_{k-1} + T_s\hat{\dot{y}}_{k-1}$$

$$\bar{\dot{x}}_k = \hat{\dot{x}}_{k-1} \qquad \bar{\dot{y}}_k = \hat{\dot{y}}_{k-1}$$

Developing New Extended Kalman Filter-6

Projected range from each satellite to receiver

$$\bar{r}_{1_k} = \sqrt{(x_{R1_k} - \bar{x}_k)^2 + (y_{R1_k} - \bar{y}_k)^2}$$

$$\bar{r}_{2_k} = \sqrt{(x_{R2_k} - \bar{x}_k)^2 + (y_{R2_k} - \bar{y}_k)^2}$$

Residual

$$RES_{1_k} = r^*_{1_k} - \bar{r}_{1_k}$$

$$RES_{2_k} = r^*_{2_k} - \bar{r}_{2_k}$$

Filtering equations

$$\hat{x}_k = \bar{x}_k + K_{11_k} RES_{1_k} + K_{12_k} RES_{2_k}$$

$$\hat{\dot{x}}_k = \bar{\dot{x}}_k + K_{21_k} RES_{1_k} + K_{22_k} RES_{2_k}$$

$$\hat{y}_k = \bar{y}_k + K_{31_k} RES_{1_k} + K_{32_k} RES_{2_k}$$

$$\hat{\dot{y}}_k = \bar{\dot{y}}_k + K_{41_k} RES_{1_k} + K_{42_k} RES_{2_k}$$

MATLAB Extended Kalman Filter for Estimating the States of a Receiver Moving at Constant Velocity-1

```
SIGNOISE=300.;
X=0.;
Y=0.;
XH=1000.;
YH=2000.;
XDH=0.;
YDH=0.;
XR1=1000000.;
YR1=20000.*3280.;
XR2=50000000.;
YR2=20000.*3280.;
ORDER=4;
TS=1.;
TF=200.;
PHIS=0.;
T=0.;
S=0.;
H=.01;
PHI=zeros(ORDER,ORDER);
P=zeros(ORDER,ORDER);
IDNP=eye(ORDER);
Q=zeros(ORDER,ORDER);
```

```
P(1,1)=1000.^2;
P(2,2)=100.^2;
P(3,3)=2000.^2;
P(4,4)=100.^2;
```

Initial covariance matrix

```
RMAT(1,1)=SIGNOISE^2;
RMAT(1,2)=0.;
RMAT(2,1)=0.;
RMAT(2,2)=SIGNOISE^2;
```

Measurement noise matrix

```
TS2=TS*TS;
TS3=TS2*TS;
```

MATLAB Extended Kalman Filter for Estimating the States of a Receiver Moving at Constant Velocity-2

```
Q(1,1)=PHIS*TS3/3.;
Q(1,2)=PHIS*TS2/2.;
Q(2,1)=Q(1,2);
Q(2,2)=PHIS*TS;
Q(3,3)=PHIS*TS3/3.;
Q(3,4)=PHIS*TS2/2.;
Q(4,3)=Q(3,4);
Q(4,4)=PHIS*TS;
```

Process noise matrix

```
count=0;
while T<=TF
          XR1OLD=XR1;
          XR2OLD=XR2;
          XOLD=X;
          YOLD=Y;
          XR1D=-14600.;
          XR2D=-14600.;
          XD=100.;
          YD=0.;
          XR1=XR1+H*XR1D;
          XR2=XR2+H*XR2D;
          X=X+H*XD;
          Y=Y+H*YD;
          T=T+H;
          XR1D=-14600.;
          XR2D=-14600.;
          XD=100.;
          YD=0.;
          XR1=.5*(XR1OLD+XR1+H*XR1D);
          XR2=.5*(XR2OLD+XR2+H*XR2D);
          X=.5*(XOLD+X+H*XD);
          Y=.5*(YOLD+Y+H*YD);
```

Integrating satellite and receiver equations using second-order Runge-Kutta numerical integration

```
          S=S+H;
```

MATLAB Extended Kalman Filter for Estimating the States of a Receiver Moving at Constant Velocity-3

```
if S>=(TS-.00001)
          S=0.;
```

```
          XB=XH+XDH*TS;
          YB=YH+YDH*TS;
          R1B=sqrt((XR1-XB)^2+(YR1-YB)^2);
          R2B=sqrt((XR2-XB)^2+(YR2-YB)^2);
          HMAT(1,1)=-(XR1-XB)/R1B;
          HMAT(1,2)=0.;
          HMAT(1,3)=-(YR1-YB)/R1B;
          HMAT(1,4)=0.;
          HMAT(2,1)=-(XR2-XB)/R2B;
          HMAT(2,2)=0.;
          HMAT(2,3)=-(YR2-YB)/R2B;
          HMAT(2,4)=0.;
```

Linearized measurement matrix

```
          HT=HMAT';
          PHI(1,1)=1.;
          PHI(1,2)=TS;
          PHI(2,2)=1.;
          PHI(3,3)=1.;
          PHI(3,4)=TS;
          PHI(4,4)=1.;
```

Fundamental matrix

```
          PHIT=PHI';
          PHIP=PHI*P;
          PHIPPHIT=PHIP*PHIT;
          M=PHIPPHIT+Q;
          HM=HMAT*M;
          HMHT=HM*HT;
          HMHTR=HMHT+RMAT;
          HMHTRINV=inv(HMHTR);
          MHT=M*HT;
          GAIN=MHT*HMHTRINV;
          KH=GAIN*HMAT;
          IKH=IDNP-KH;
          P=IKH*M;
```

Riccati equations

```
          R1NOISE=SIGNOISE*randn;
          R2NOISE=SIGNOISE*randn;
          R1=sqrt((XR1-X)^2+(YR1-Y)^2);
          R2=sqrt((XR2-X)^2+(YR2-Y)^2);
```

MATLAB Extended Kalman Filter for Estimating the States of a Receiver Moving at Constant Velocity-4

```
        RES1=R1+R1NOISE-R1B;
        RES2=R2+R2NOISE-R2B;
        XH=XB+GAIN(1,1)*RES1+GAIN(1,2)*RES2;
        XDH=XDH+GAIN(2,1)*RES1+GAIN(2,2)*RES2;
        YH=YB+GAIN(3,1)*RES1+GAIN(3,2)*RES2;
        YDH=YDH+GAIN(4,1)*RES1+GAIN(4,2)*RES2;
```

Filter

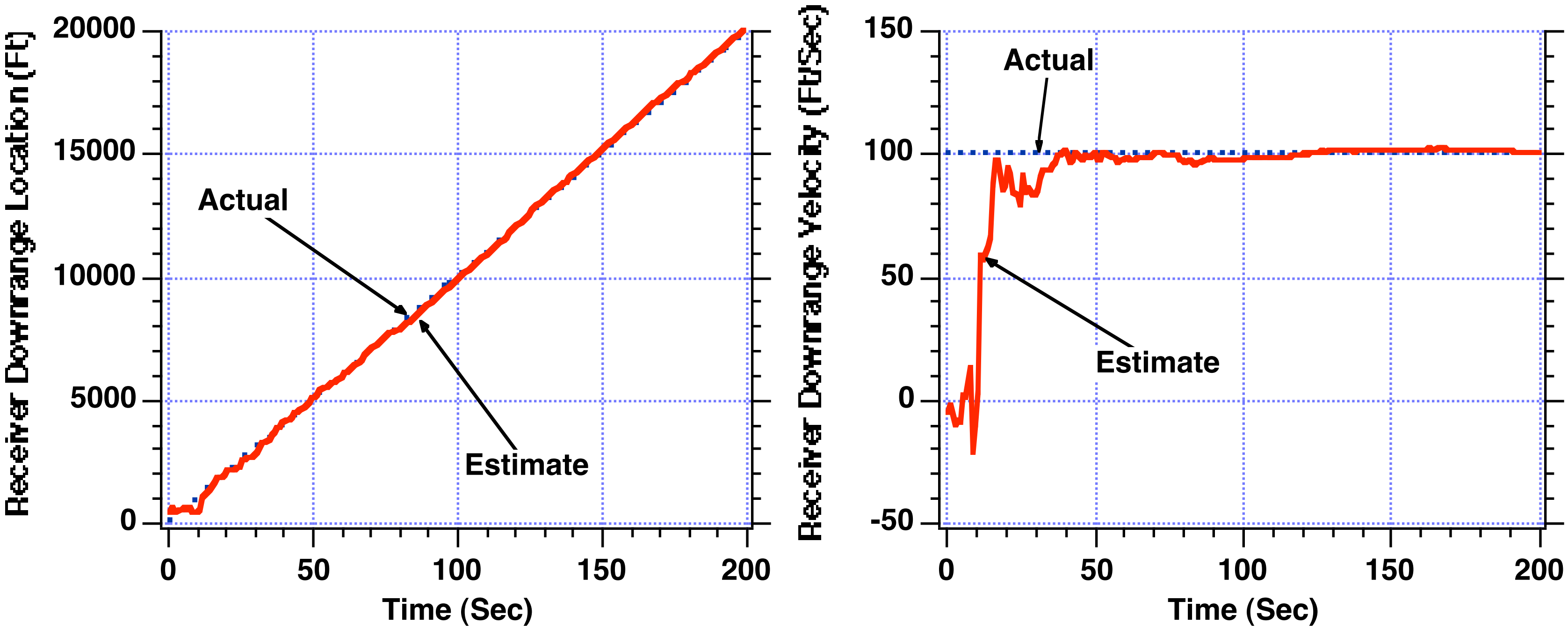
```
        ERRX=X-XH;
        SP11=sqrt(P(1,1));
        ERRXD=XD-XDH;
        SP22=sqrt(P(2,2));
        ERRY=Y-YH;
        SP33=sqrt(P(3,3));
        ERRYD=YD-YDH;
        SP44=sqrt(P(4,4));
        SP11P=-SP11;
        SP22P=-SP22;
        SP33P=-SP33;
        SP44P=-SP44;
```

Actual and theoretical errors in estimates

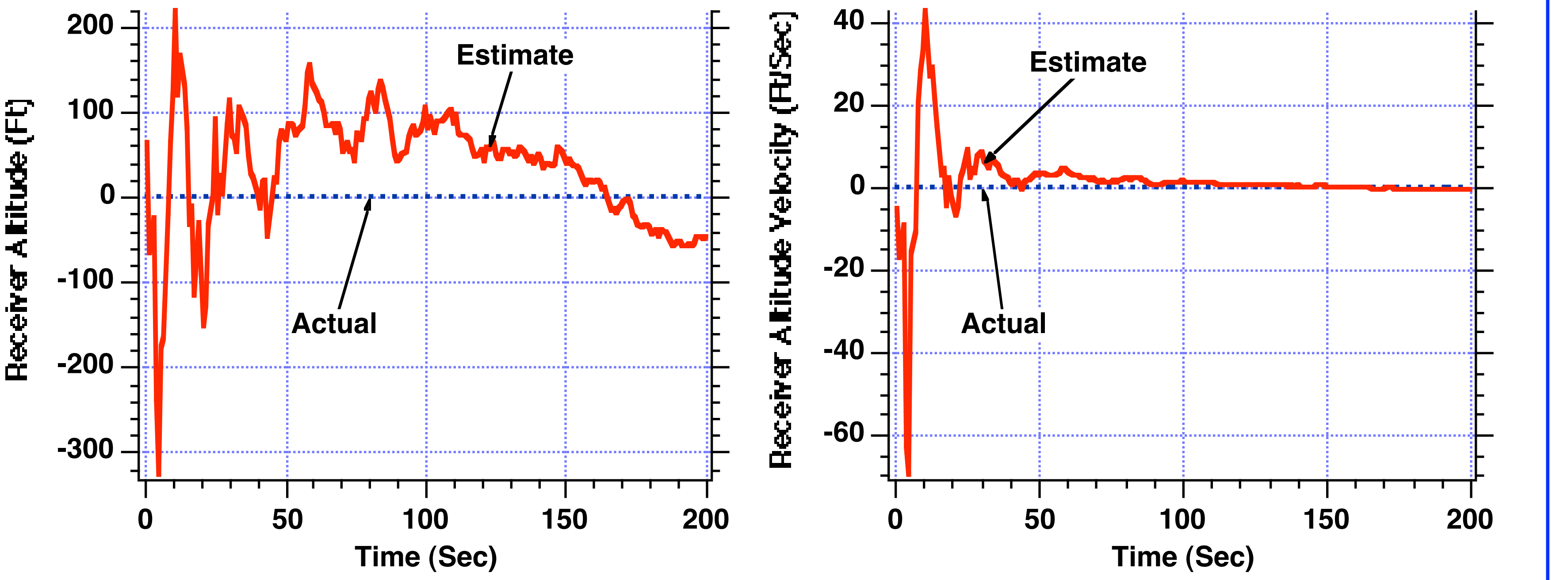
```
        count=count+1;
        ArrayT(count)=T;
        ArrayX(count)=X;
        ArrayXH(count)=XH;
        ArrayXD(count)=XD;
        ArrayXDH(count)=XDH;
        ArrayY(count)=Y;
        ArrayYH(count)=YH;
        ArrayYD(count)=YD;
        ArrayYDH(count)=YDH;
        ArrayERRX(count)=ERRX;
        ArraySP11(count)=SP11;
        ArraySP11P(count)=SP11P;
        ArrayERRXD(count)=ERRXD;
        ArraySP22(count)=SP22;
        ArraySP22P(count)=SP22P;
        ArrayERRY(count)=ERRY;
        ArraySP33(count)=SP33;
        ArraySP33P(count)=SP33P;
        ArrayERRYD(count)=ERRYD;
        ArraySP44(count)=SP44;
        ArraySP44P(count)=SP44P;
        end
end
```

Save data in arrays for plotting and writing to files

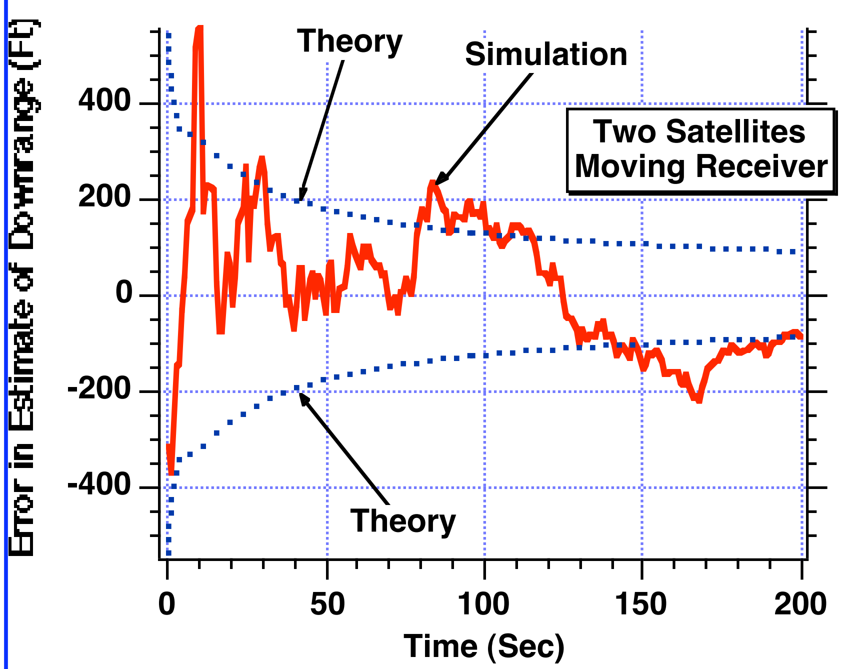
Extended Kalman Filter is Able to Estimate the Location and Velocity of the Receiver Quite Well in Downrange

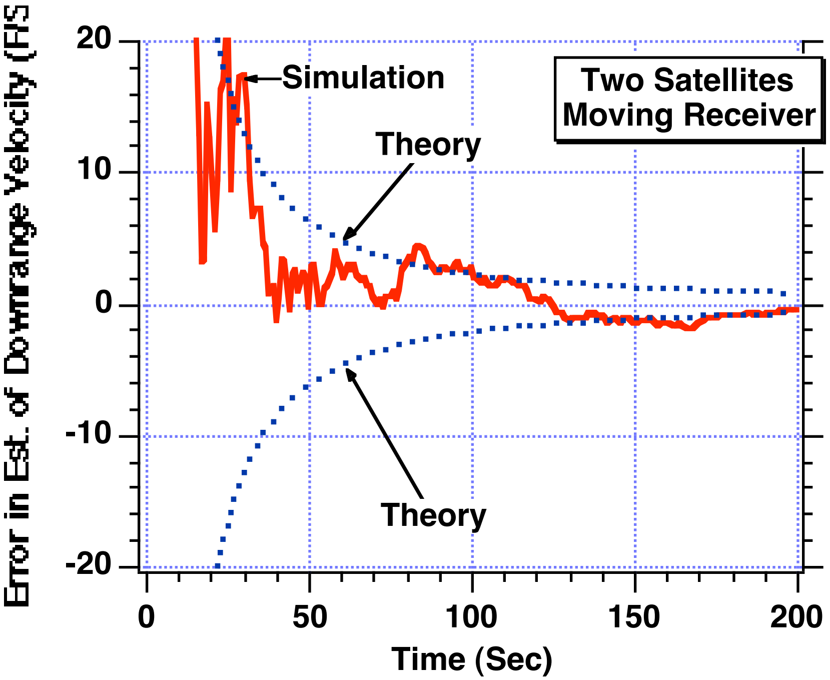


Extended Kalman Filter is Able to Estimate the Location and Velocity of the Receiver Quite Well in Altitude

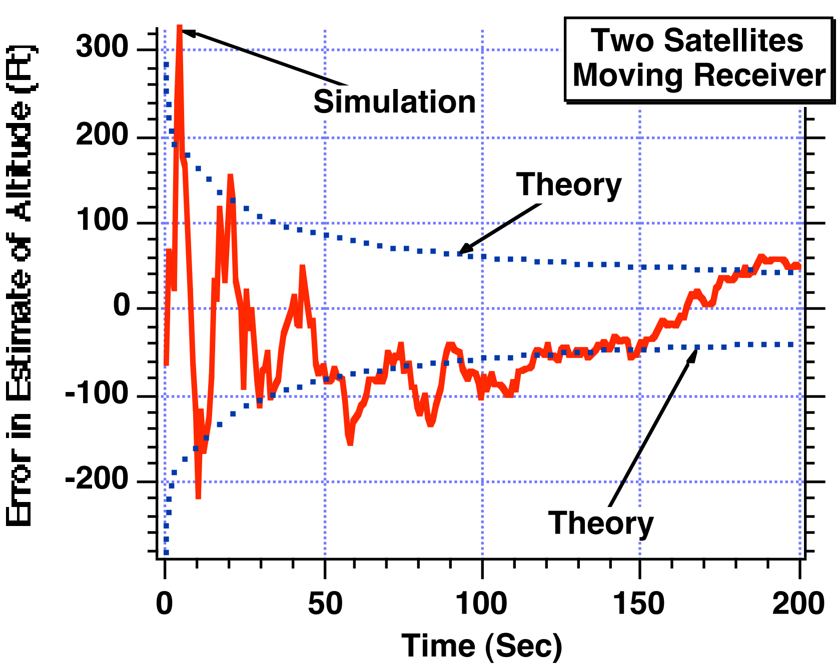


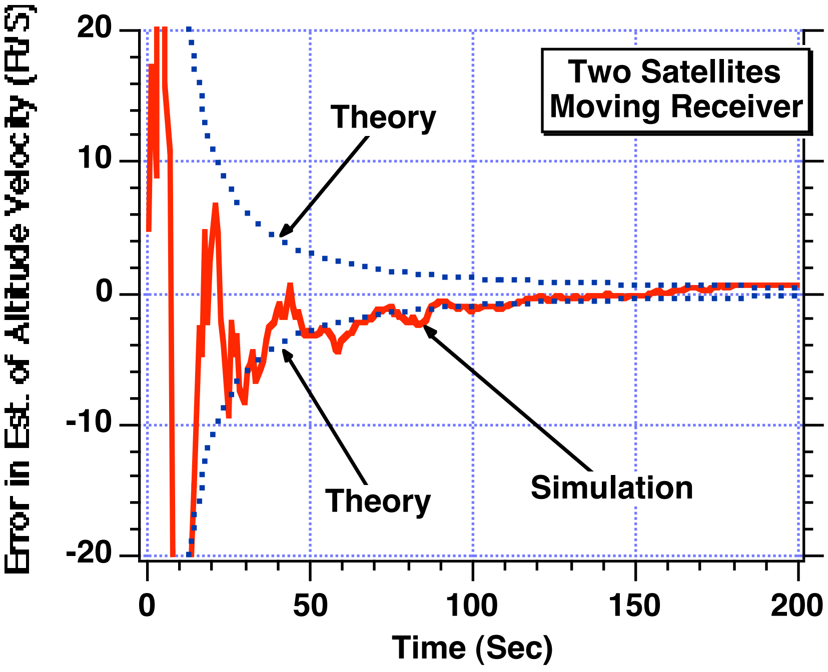
New Extended Kalman Filter Appears to be Operating Properly in Downrange





New Extended Kalman Filter Appears to be Operating Properly in Altitude





Single Satellite With Constant Velocity Receiver

Single Satellite Extended Kalman Filter-1

Model of real world for moving receiver

$$\ddot{x} = u_s$$

$$\ddot{y} = u_s$$

Put model in state space form

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

Continuous process noise matrix

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

Systems dynamics matrix

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

Single Satellite Extended Kalman Filter-2

Since F squared is zero

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

Therefore the discrete fundamental matrix is

$$\boldsymbol{\Phi}_{\mathbf{k}} = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Range from single satellite to receiver is

$$r_1 = \sqrt{(x_{R1} - x)^2 + (y_{R1} - y)^2}$$

Linearized measurement equation

$$\Delta r_1^* = \begin{bmatrix} \frac{\partial r_1}{\partial x} & \frac{\partial r_1}{\partial \dot{x}} & \frac{\partial r_1}{\partial y} & \frac{\partial r_1}{\partial \dot{y}} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \dot{x} \\ \Delta y \\ \Delta \dot{y} \end{bmatrix} + v_{r1}$$

Single Satellite Extended Kalman Filter-3

Discrete measurement noise matrix is now a scalar

$$\mathbf{R_k} = \sigma_{r1}^2$$

Linearized measurement equation

$$\mathbf{H_k} = \begin{bmatrix} \frac{\partial r_1}{\partial x} & \frac{\partial r_1}{\partial \dot{x}} & \frac{\partial r_1}{\partial y} & \frac{\partial r_1}{\partial \dot{y}} \end{bmatrix}$$

Where partial derivatives evaluated at projected state estimates

Evaluation of partial derivatives

$$r_1 = \sqrt{(x_{R1} - x)^2 + (y_{R1} - y)^2} \longrightarrow$$

$$\frac{\partial r_1}{\partial x} = .5\left[(x_{R1}-x)^2+(y_{R1}-y)^2\right]^{-.5} 2(x_{R1}-x)(-1) = \frac{-(x_{R1}-x)}{r_1}$$

$$\frac{\partial r_1}{\partial \dot{x}} = 0$$

$$\frac{\partial r_1}{\partial y} = .5\left[(x_{R1}-x)^2+(y_{R1}-y)^2\right]^{-.5} 2(y_{R1}-y)(-1) = \frac{-(y_{R1}-y)}{r_1}$$

$$\frac{\partial r_1}{\partial \dot{y}} = 0$$

Single Satellite Extended Kalman Filter-4

Linearized measurement matrix

$$\mathbf{H_k} = \begin{bmatrix} \frac{-(x_{R1}-x)}{r_1} & 0 & \frac{-(y_{R1}-y)}{r_1} & 0 \end{bmatrix}$$

Discrete process noise matrix can be derived from continuous Q

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

Substitution yields

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

After multiplication and integration

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

Single Satellite Extended Kalman Filter-5

Project states ahead with exact fundamental matrix $\bar{\mathbf{x}}_{\mathbf{k}} = \Phi \hat{\mathbf{x}}_{\mathbf{k}-1}$ or

$$\bar{x}_k = \hat{x}_{k-1} + T_s\hat{\dot{x}}_{k-1}$$

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

$$\bar{y}_k = \hat{y}_{k-1} + T_s\hat{\dot{y}}_{k-1}$$

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

Projected range from satellite to receiver

$$\bar{r}_{1_k} = \sqrt{(x_{R1_k} - \bar{x}_k)^2 + (y_{R1_k} - \bar{y}_k)^2}$$

Extended Kalman filtering equations

$$RES_{1_k} = r^*_{1_k} - \bar{r}_{1_k}$$

$$\hat{x}_k = \bar{x}_k + K_{11_k}RES_{1_k}$$

$$\hat{\dot{x}}_k = \bar{\dot{x}}_k + K_{21_k}RES_{1_k}$$

$$\hat{y}_k = \bar{y}_k + K_{31_k}RES_{1_k}$$

$$\hat{\dot{y}}_k = \bar{\dot{y}}_k + K_{41_k}RES_{1_k}$$

MATLAB Single Satellite Filter for Estimating the States of a Receiver Moving at Constant Velocity-1

```
SIGNOISE=300.;
PHIS=0.;
X=0.;
Y=0.;
XH=1000.;
YH=2000.;
XDH=0.;
YDH=0.;
XR1=1000000.;
YR1=20000.*3280.;
ORDER=4;
TS=1.;
TF=1000.;
T=0.;
S=0.;
H=.01;
PHI=zeros(ORDER,ORDER);
P=zeros(ORDER,ORDER);
IDNP=eye(ORDER);
Q=zeros(ORDER,ORDER);
```

```
P(1,1)=1000.^2;
P(2,2)=100.^2;
P(3,3)=2000.^2;
P(4,4)=100.^2;
```

Initial covariance matrix

```
RMAT(1,1)=SIGNOISE^2;
TS2=TS*TS;
TS3=TS2*TS;
```

```
Q(1,1)=PHIS*TS3/3.;
Q(1,2)=PHIS*TS2/2.;
Q(2,1)=Q(1,2);
Q(2,2)=PHIS*TS;
Q(3,3)=PHIS*TS3/3.;
Q(3,4)=PHIS*TS2/2.;
Q(4,3)=Q(3,4);
Q(4,4)=PHIS*TS;
```

Process noise matrix

```
count=0;
```

MATLAB Single Satellite Filter for Estimating the States of a Receiver Moving at Constant Velocity-2

```
while T<=TF
        XR1OLD=XR1;
        XOLD=X;
        YOLD=Y;
        XR1D=-14600.;
        XD=100.;
        YD=0.;
        XR1=XR1+H*XR1D;
        X=X+H*XD;
        Y=Y+H*YD;
        T=T+H;
        XR1D=-14600.;
        XD=100.;
        YD=0.;
        XR1=.5*(XR1OLD+XR1+H*XR1D);
        X=.5*(XOLD+X+H*XD);
        Y=.5*(YOLD+Y+H*YD);
        S=S+H;
        if S>=(TS-.00001)
                S=0.;
                XB=XH+XDH*TS;
                YB=YH+YDH*TS;
                R1B=sqrt((XR1-XB)^2+(YR1-YB)^2);
                HMAT(1,1)=-(XR1-XB)/R1B;
                HMAT(1,2)=0.;
                HMAT(1,3)=-(YR1-YB)/R1B;
                HMAT(1,4)=0.;
                HT=HMAT';
                PHI(1,1)=1.;
                PHI(1,2)=TS;
                PHI(2,2)=1.;
                PHI(3,3)=1.;
                PHI(3,4)=TS;
                PHI(4,4)=1.;
                PHIT=PHI';
```

Integrating satellite and receiver equations Using second-order Runge-Kutta technique

Linearized measurement matrix

Fundamental matrix

MATLAB Single Satellite Filter for Estimating the States of a Receiver Moving at Constant Velocity-3

```
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;
GAIN=MHT*HMHTRINV;
KH=GAIN*HMAT;
IKH=IDNP-KH;
P=IKH*M;
```

Riccati equations

```
R1NOISE=SIGNOISE*randn;
```

```
R1=sqrt((XR1-X)^2+(YR1-Y)^2);
RES1=R1+R1NOISE-R1B;
XH=XB+GAIN(1,1)*RES1;
XDH=XDH+GAIN(2,1)*RES1;
YH=YB+GAIN(3,1)*RES1;
YDH=YDH+GAIN(4,1)*RES1;
```

Filter

```
ERRX=X-XH;
SP11=sqrt(P(1,1));
ERRXD=XD-XDH;
SP22=sqrt(P(2,2));
ERRY=Y-YH;
SP33=sqrt(P(3,3));
ERRYD=YD-YDH;
SP44=sqrt(P(4,4));
SP11P=-SP11;
SP22P=-SP22;
SP33P=-SP33;
SP44P=-SP44;
```

Actual and theoretical errors in estimates

```
count=count+1;
```

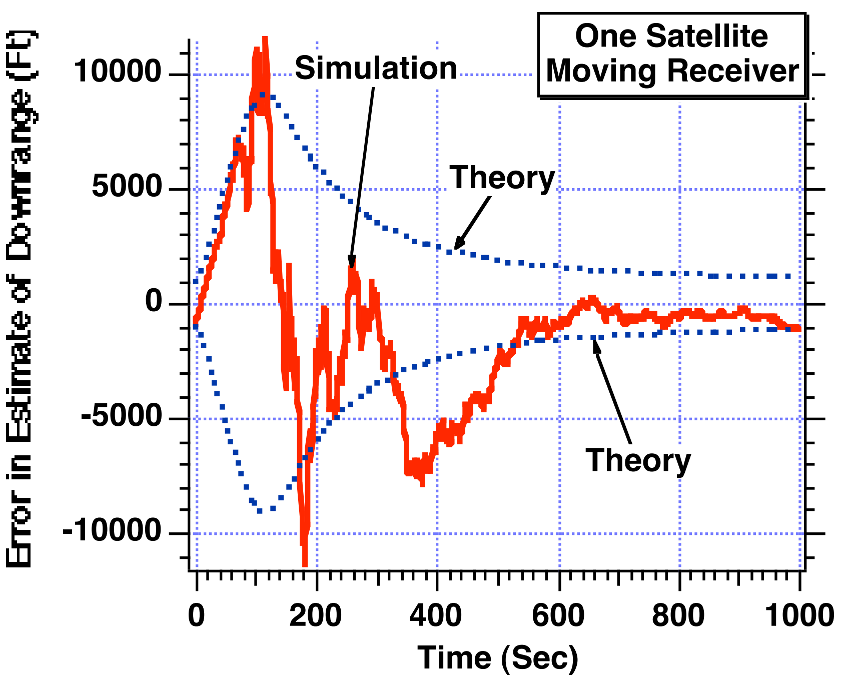
MATLAB Single Satellite Filter for Estimating the States of a Receiver Moving at Constant Velocity-4

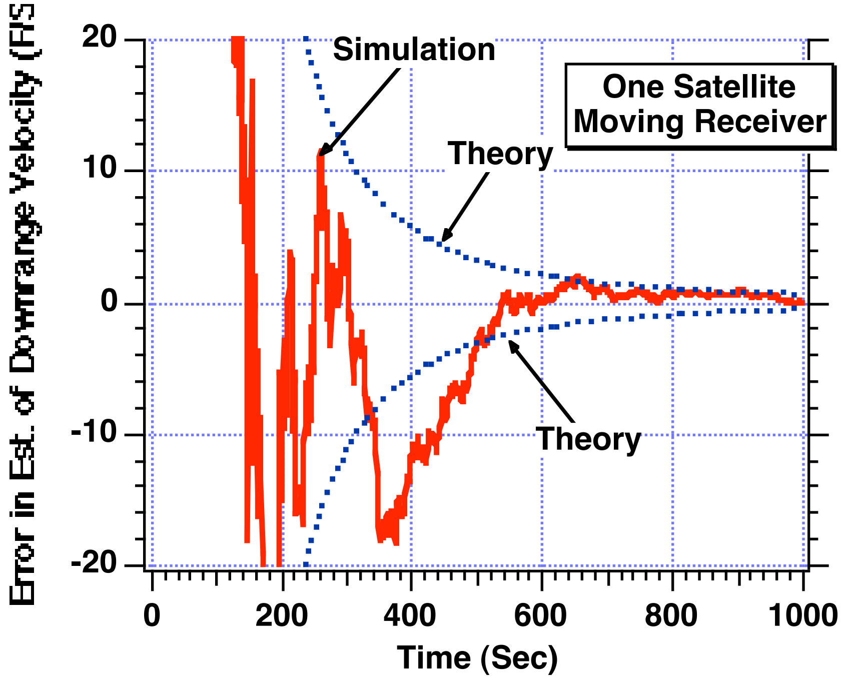
```
                    ArrayT(count)=T;
                    ArrayX(count)=X;
                    ArrayXH(count)=XH;
                    ArrayXD(count)=XD;
                    ArrayXDH(count)=XDH;
                    ArrayY(count)=Y;
                    ArrayYH(count)=YH;
                    ArrayYD(count)=YD;
                    ArrayYDH(count)=YDH;
                    ArrayERRX(count)=ERRX;
                    ArraySP11(count)=SP11;
                    ArraySP11P(count)=SP11P;
                    ArrayERRXD(count)=ERRXD;
                    ArraySP22(count)=SP22;
                    ArraySP22P(count)=SP22P;
                    ArrayERRY(count)=ERRY;
                    ArraySP33(count)=SP33;
                    ArraySP33P(count)=SP33P;
                    ArrayERRYD(count)=ERRYD;
                    ArraySP44(count)=SP44;
                    ArraySP44P(count)=SP44P;
          end
end
figure
plot(ArrayT,ArrayERRX,ArrayT,ArraySP11,ArrayT,ArraySP11P),grid
xlabel('Time (Sec)')
ylabel('Error in Estimate of Downrange (Ft)')
axis([0 1000 -11000 11000])
figure
plot(ArrayT,ArrayERRXD,ArrayT,ArraySP22,ArrayT,ArraySP22P),grid
xlabel('Time (Sec)')
ylabel('Error in Estimate of Downrange Velocity (Ft/Sec)')
axis([0 1000 -20 20])
```

Save data as arrays for plotting and writing to files

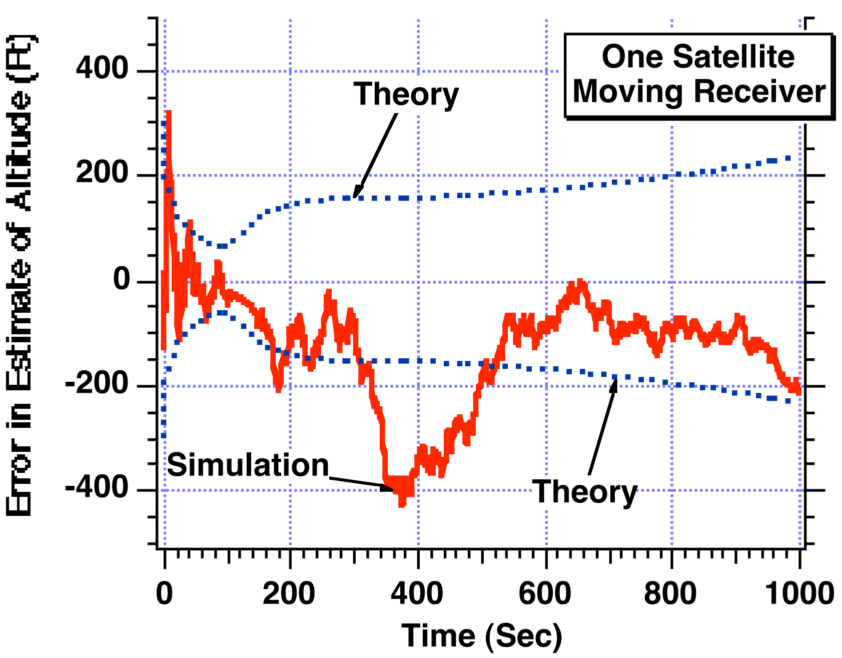
Sample plots

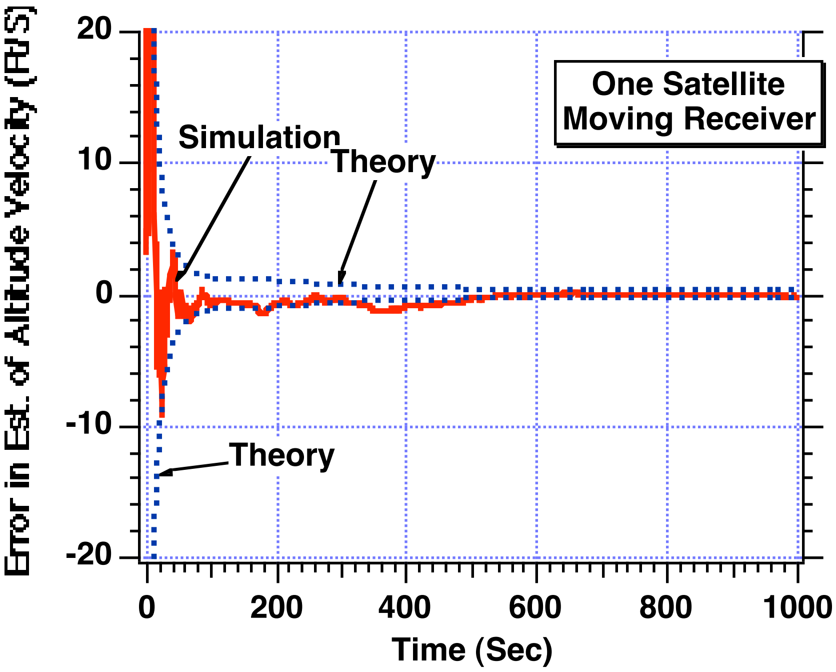
New Extended Kalman Filter Appears to be Operating Properly in Downrange





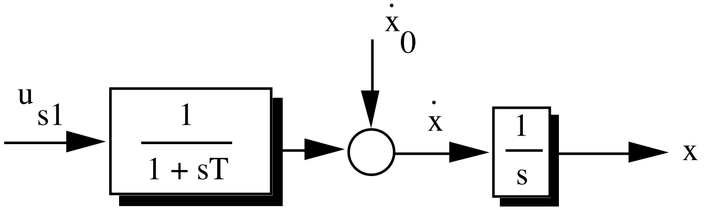
New Extended Kalman Filter Appears to be Operating Properly in Altitude





Using Extended Kalman Filtering With Variable Velocity Receiver

Gauss-Markov Model for Downrange Velocity and Location of the Receiver



Variance of low pass filter output

$$\sigma^2 = \frac{\Phi_{s1}}{2T}$$

100 ft/s average speed with Gauss-Markov 30 ft/s and 5 s correlation time

$$\Phi_{s1} = 2T\sigma^2 = 2*5*30^2 = 9000$$

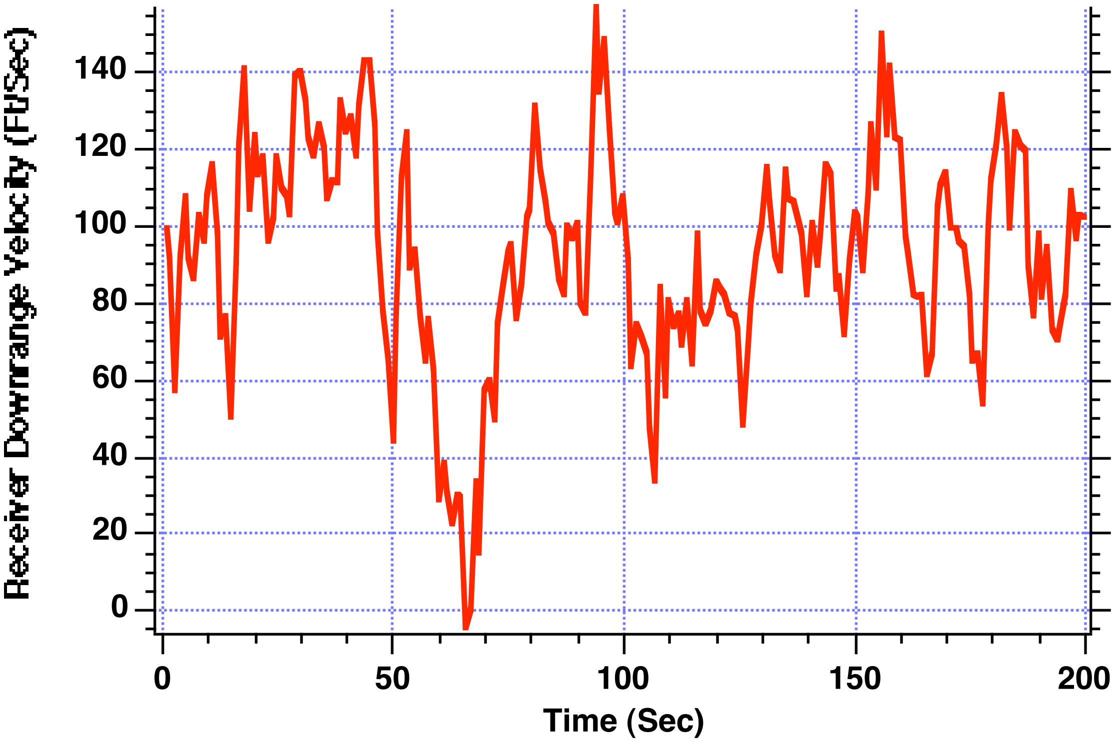
Add zero-mean Gaussian noise every integration interval with sigma

$$\sigma_{Noise} = \sqrt{\frac{\Phi_{s1}}{H}}$$

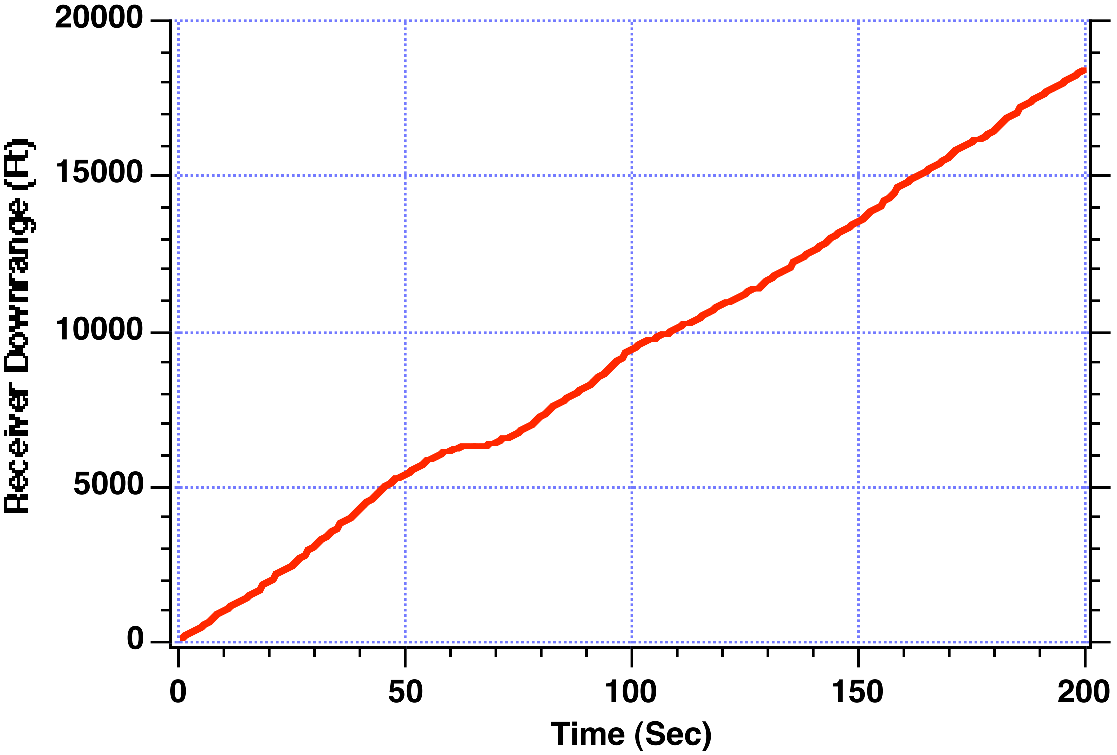
Using a Gauss-Markov Model to Represent Receiver Velocity in FORTRAN

```
GLOBAL DEFINE
          INCLUDE 'quickdraw.inc'
END
IMPLICIT REAL*8(A-H)
IMPLICIT REAL*8(O-Z)
TAU=5.
PHI=9000.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
T=0.
S=0.
H=.01
SIG=SQRT(PHI/H)
X=0.
Y1=0.
XDP=100.
TS=1.
TF=200.
WHILE(T<=TF)
          CALL GAUSS(X1,SIG)
          XOLD=X
          Y1OLD=Y1
          Y1D=(X1-Y1)/TAU
          XD=XDP+Y1
          X=X+H*XD
          Y1=Y1+H*Y1D
          T=T+H
          Y1D=(X1-Y1)/TAU
          XD=XDP+Y1
          X=.5*(XOLD+X+H*XD)
          Y1=.5*(Y1OLD+Y1+H*Y1D)
          S=S+H
          IF(S>=(TS-.00001))THEN
                    S=0.
                    WRITE(9,*)T,XD,X
                    WRITE(1,*)T,XD,X
          ENDIF
END DO
PAUSE
CLOSE(1)
END
```

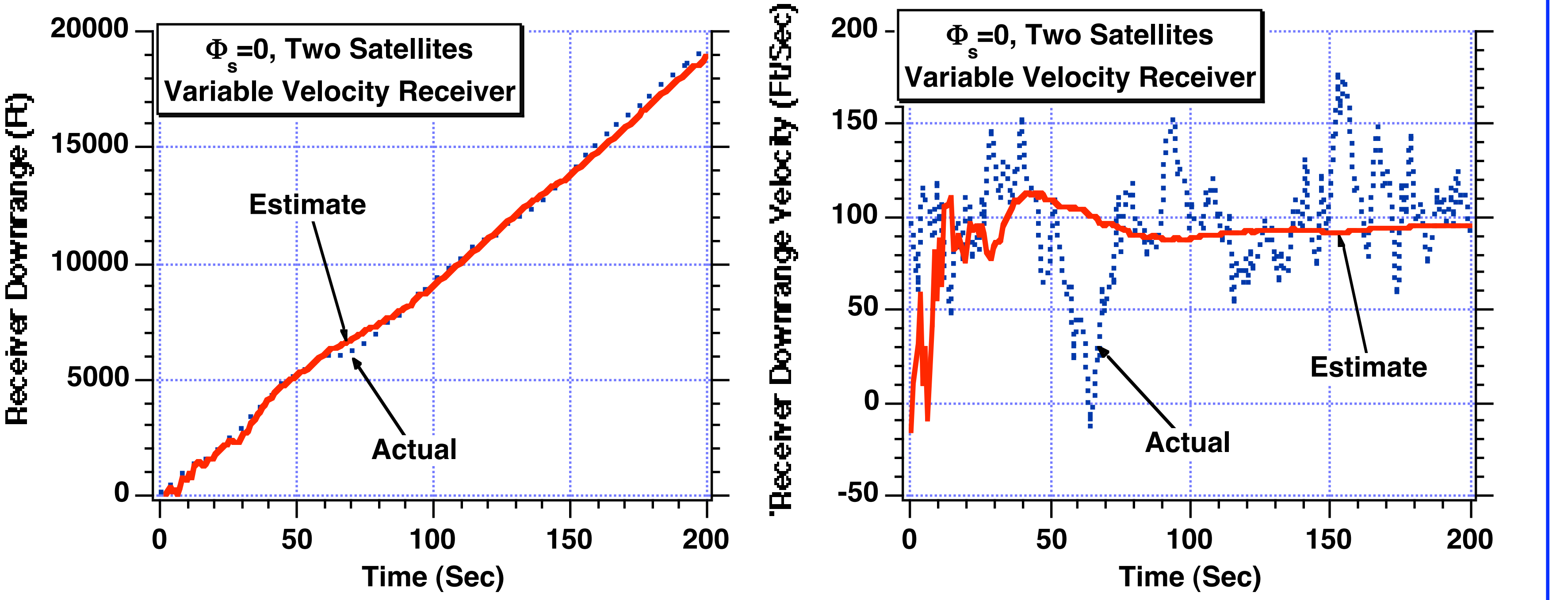
Downrange Velocity of Receiver Varies Quite a Bit



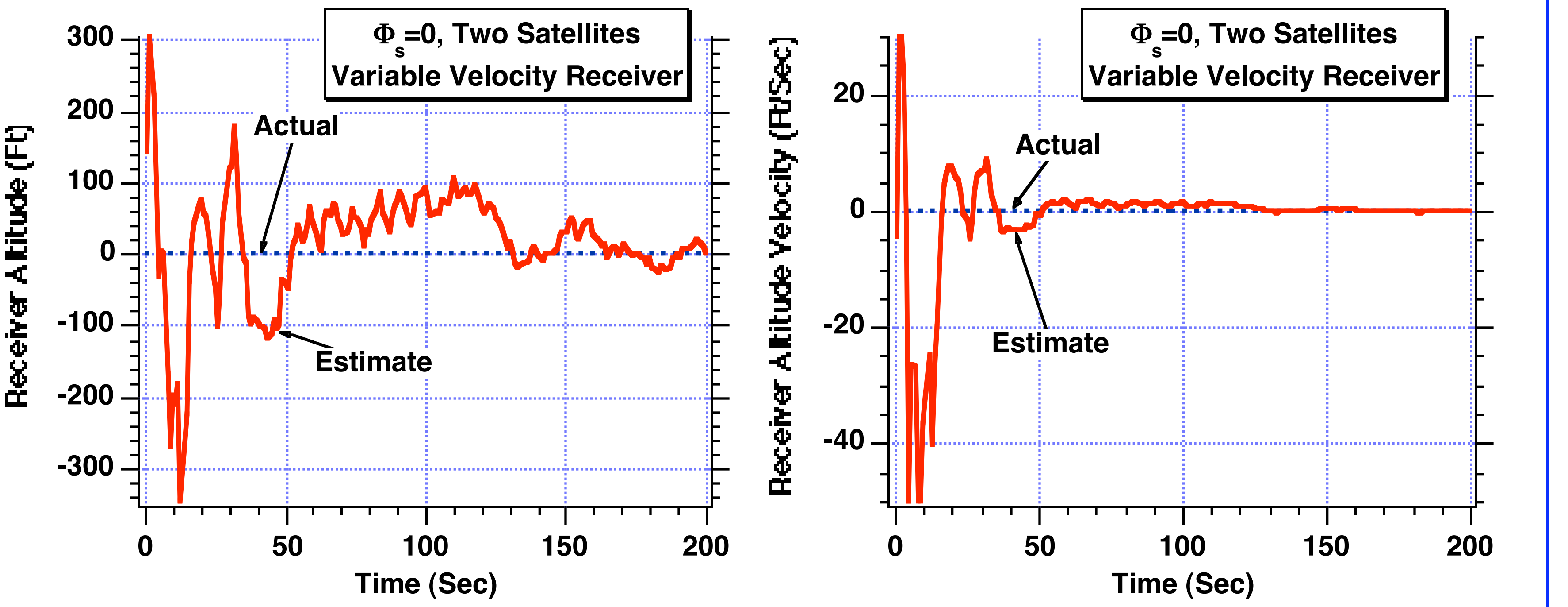
Downrange Location of the Receiver as a Function of Time is Nearly a Straight Line



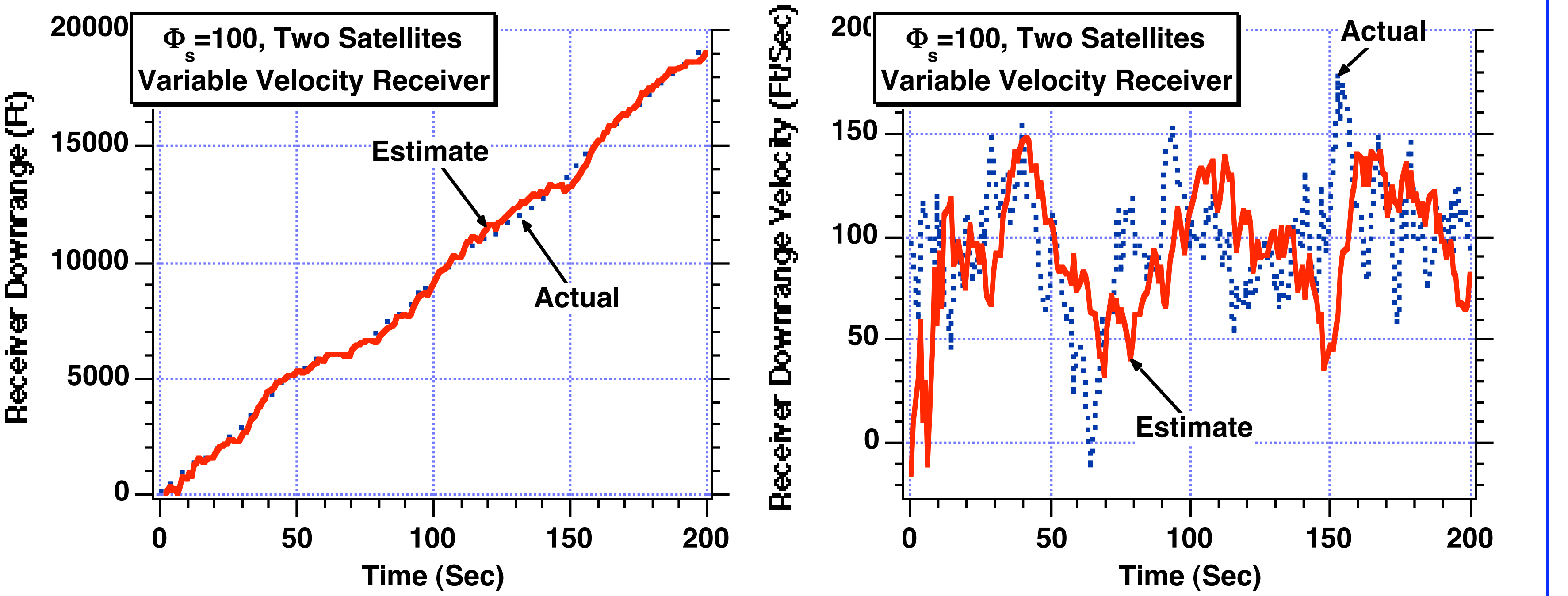
Filter is Able to Track Receiver Downrange But Not Velocity When Filter Does Not Have Process Noise



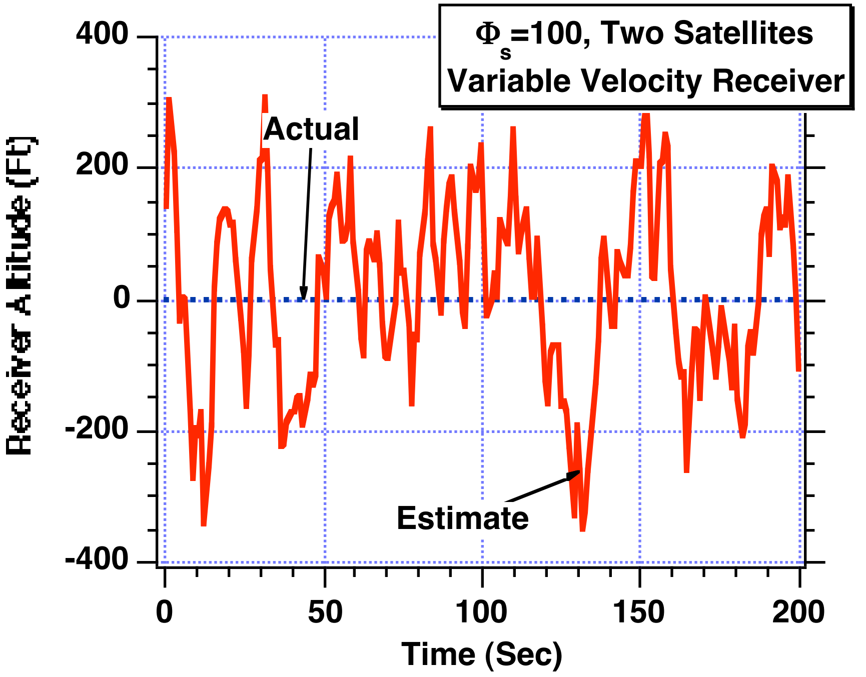
Filter is Able to Track Receiver Altitude and Zero Velocity When Filter Does Not Have Process Noise

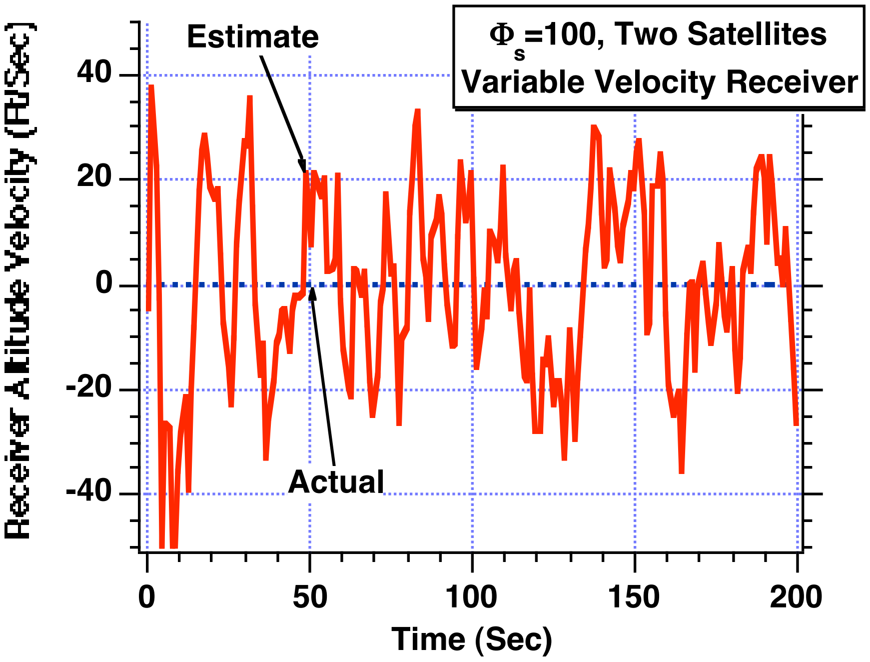


Filter is Able to Track Receiver Downrange Velocity When Filter Has Process Noise

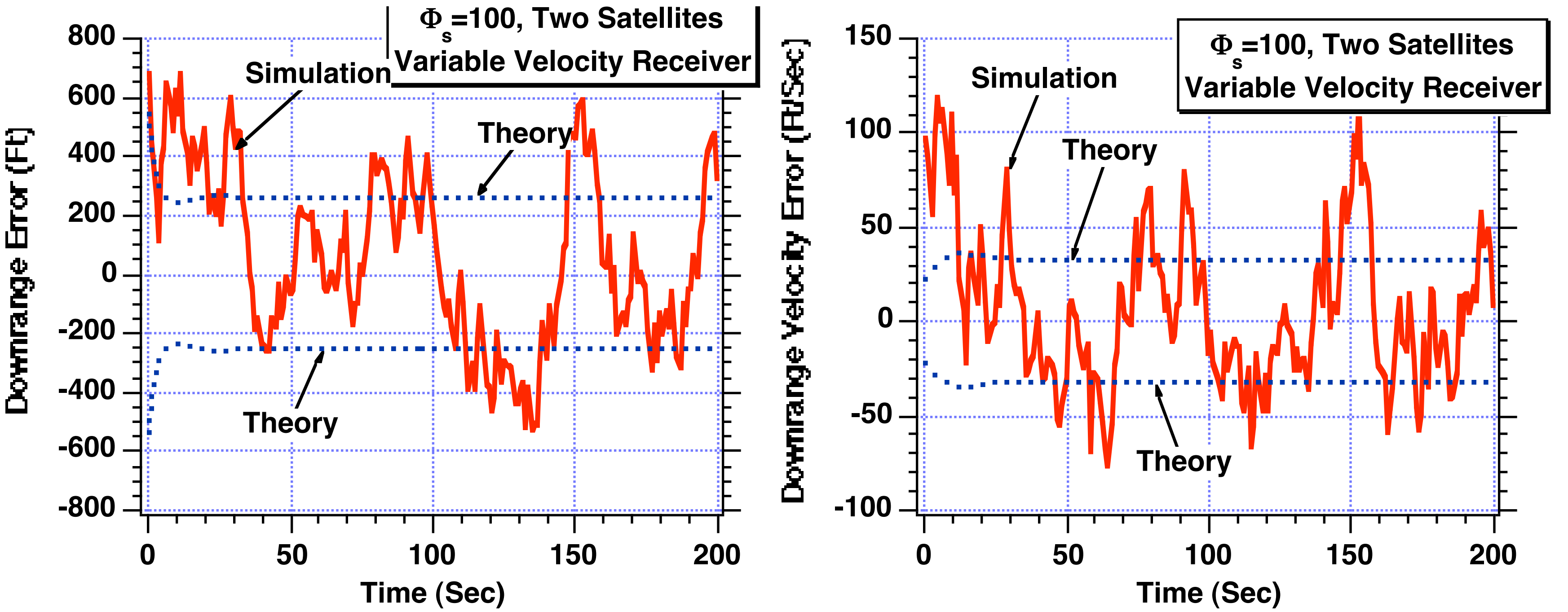


Estimates of Altitude and Velocity are Noisier When There is Process Noise

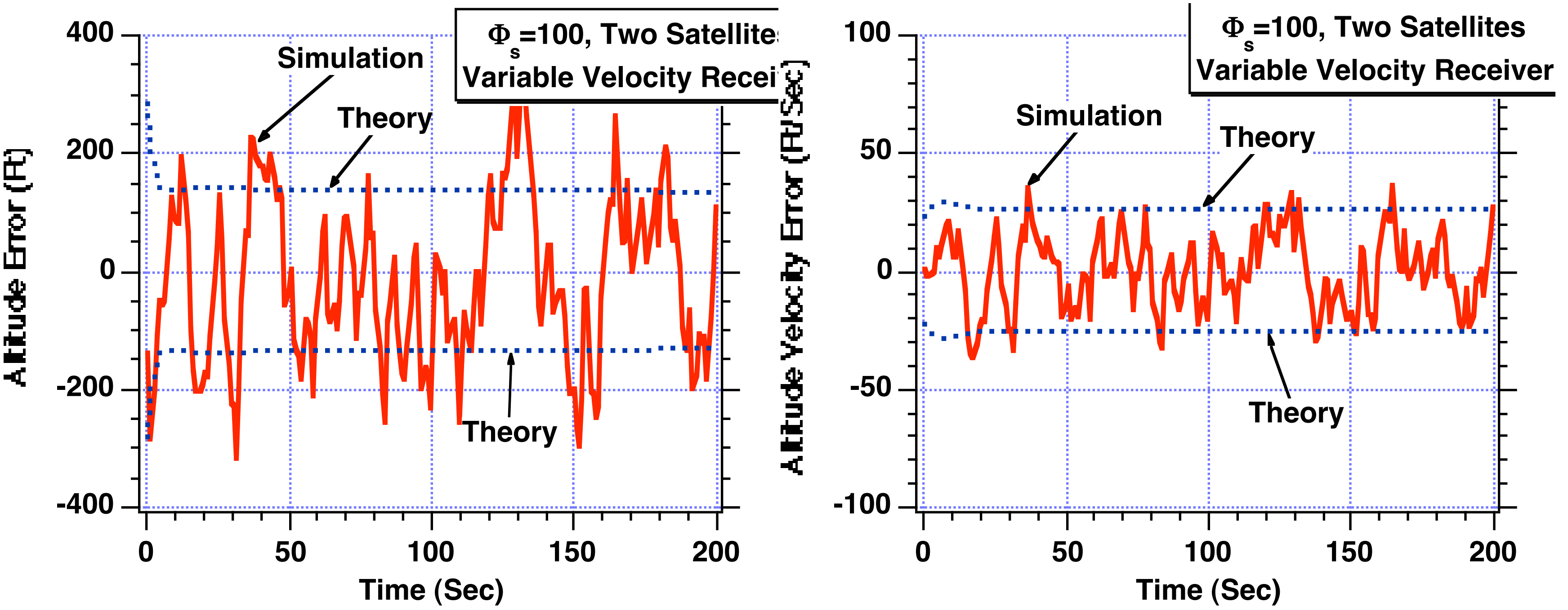




Error in Estimate of Receiver Downrange and Velocity are Within Theoretical Bounds For Variable Velocity

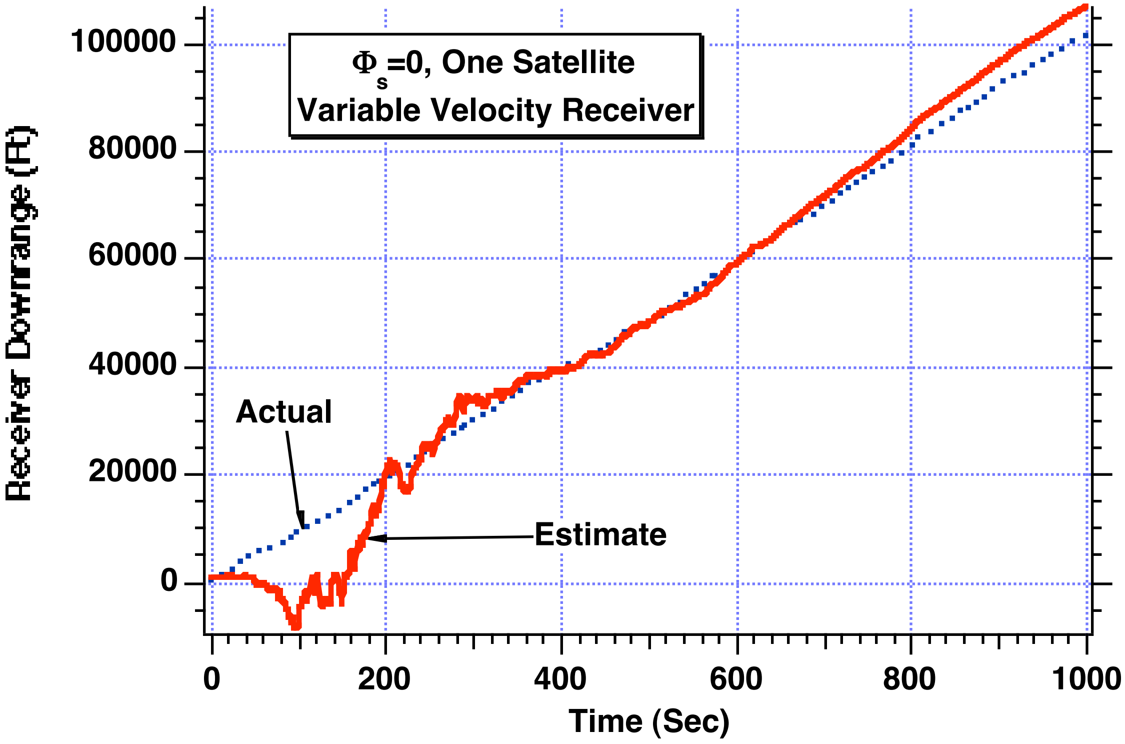


Error in Estimate of Receiver Altitude and Velocity are Within Theoretical Bounds For Variable Velocity

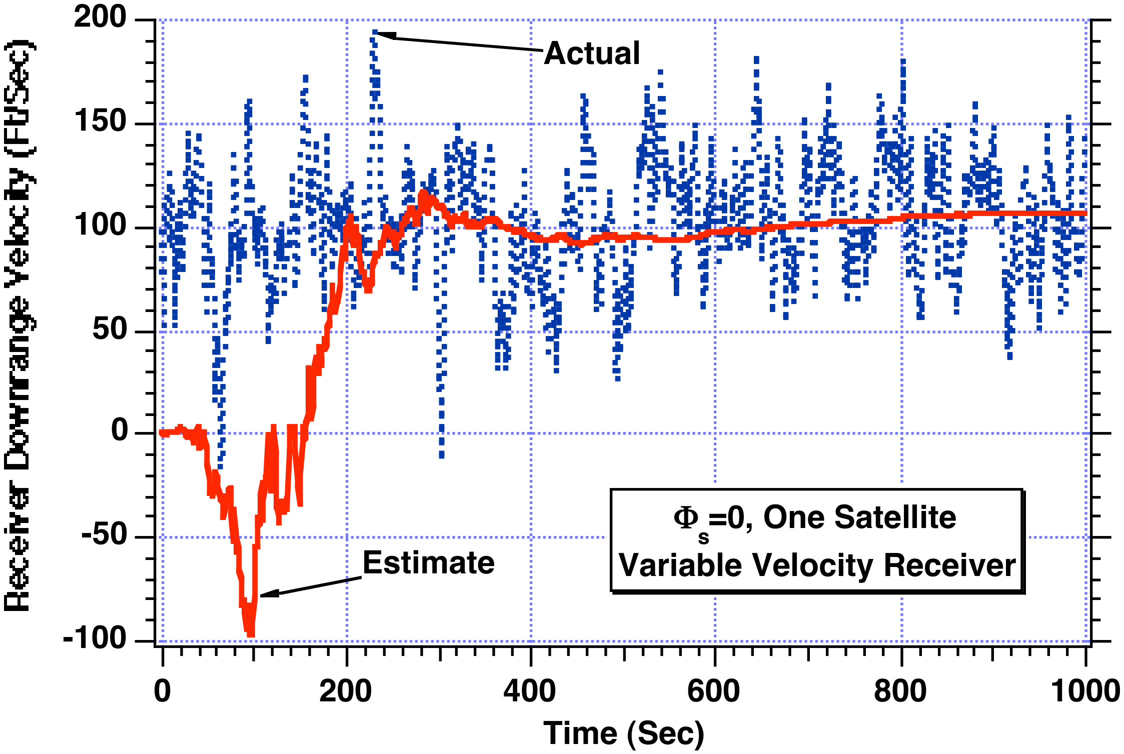


Variable Velocity Receiver and Single Satellite

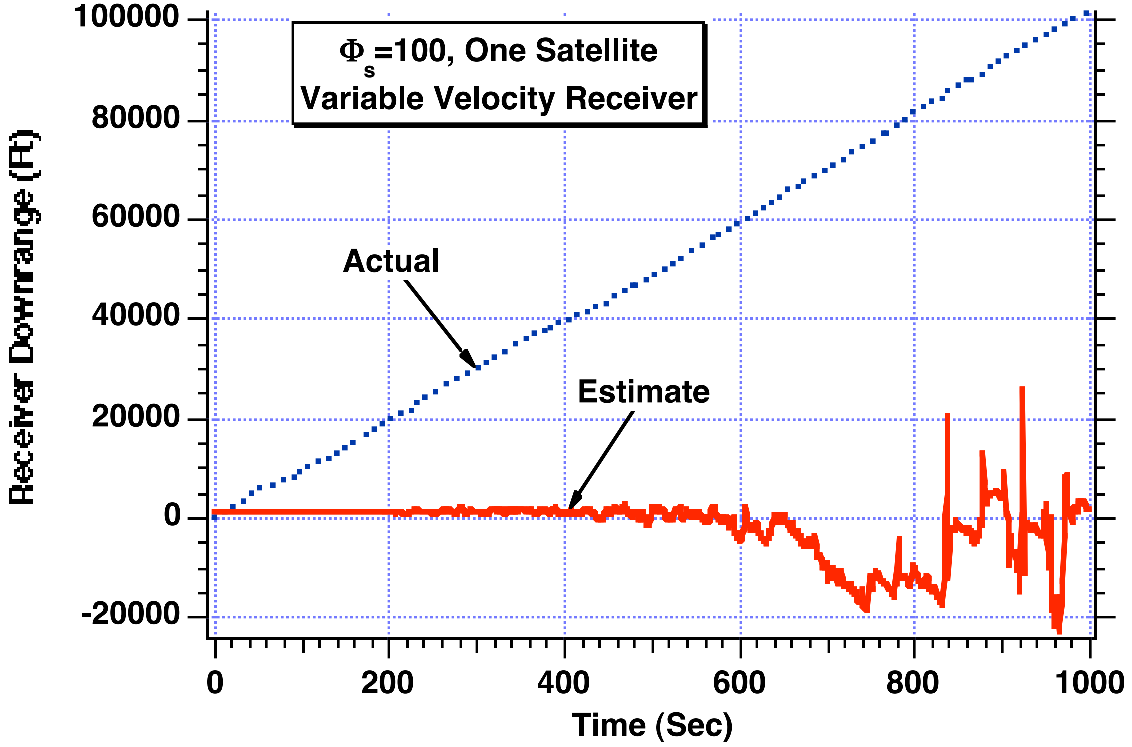
After a While Filter Appears to be Able to Track Receiver Downrange Location When Filter Does Not Have Process Noise



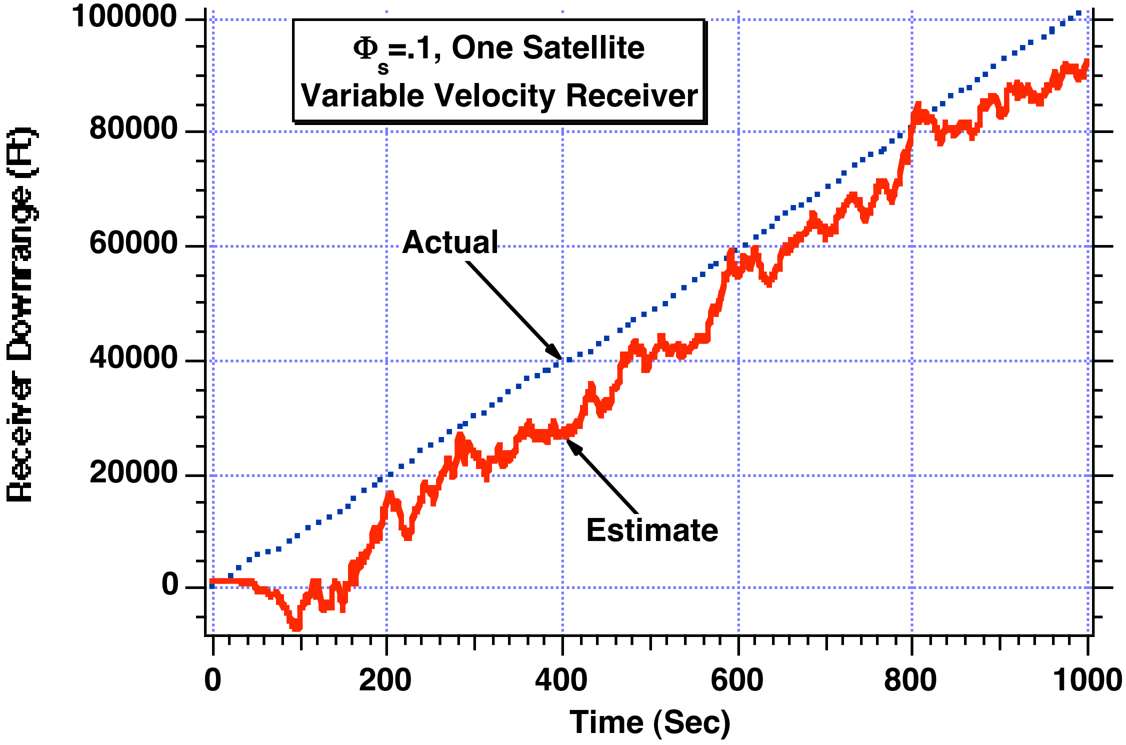
Filter is Unable to Follow Receiver Downrange Velocity Variations When Filter Does Not Have Process Noise



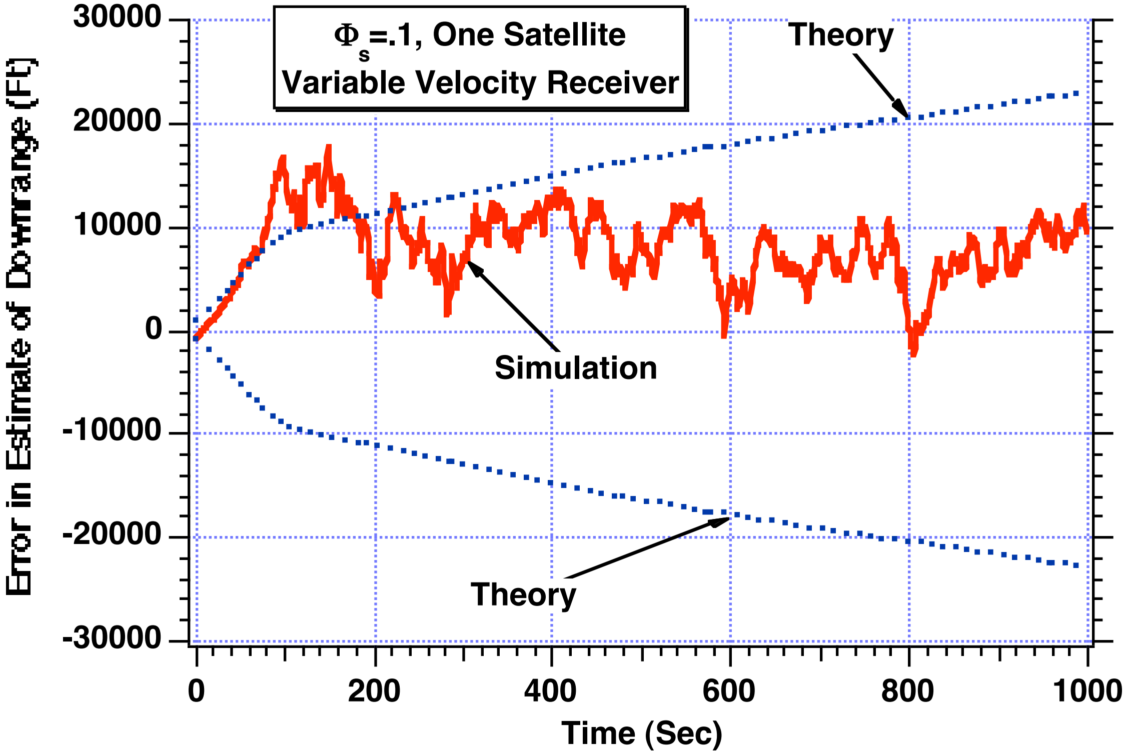
The Addition of Process Noise has Ruined the Tracking Ability of the Extended Kalman Filter When Only One Satellite is Available



Reducing the Process Noise Has Enabled the Extended Kalman Filter to Track the Receiver With a Very Significant Lag



It is Not Possible to Track the Variable Velocity Receiver With Only a Single Satellite



Satellite Navigation
Summary

- Various options for deriving stationary receiver location based on noisy range measurements from two satellites
  - Linear filtering of range better than no filtering at all
  - Extended Kalman filter even better
- Satellite geometry is important
  - Larger angle between range vectors yield better estimates
- Can track stationary receiver with single satellite
  - Have problems with variable velocity receiver
- Can track variable velocity receiver with two satellites