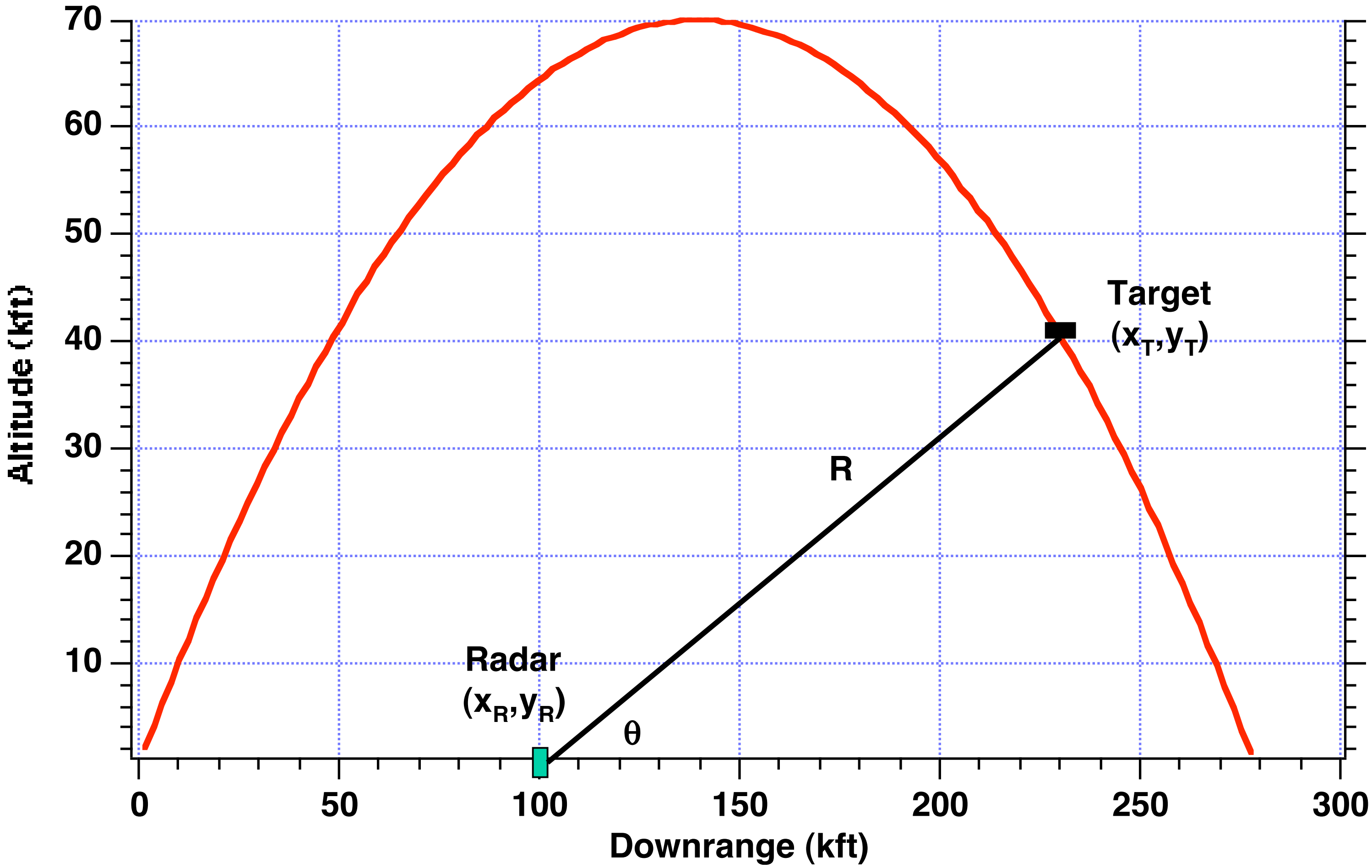
Stereo Tracking of Cannon Launched Projectile

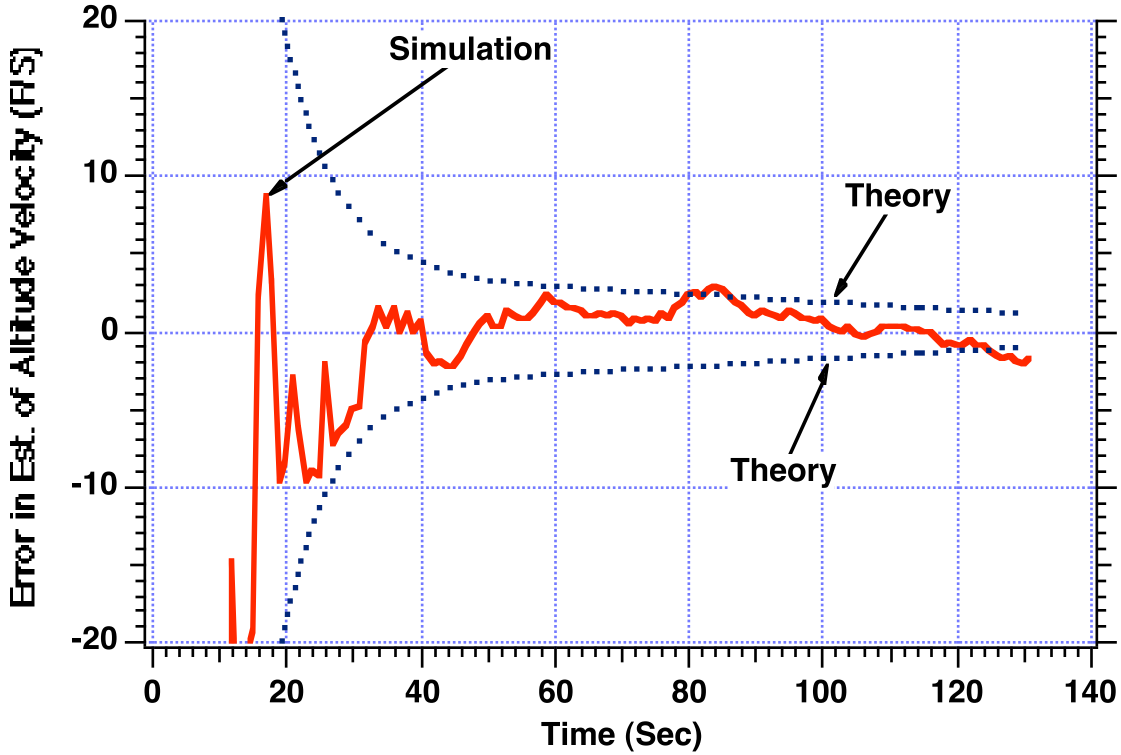
Original Radar Tracking Problem For Cannon Ball



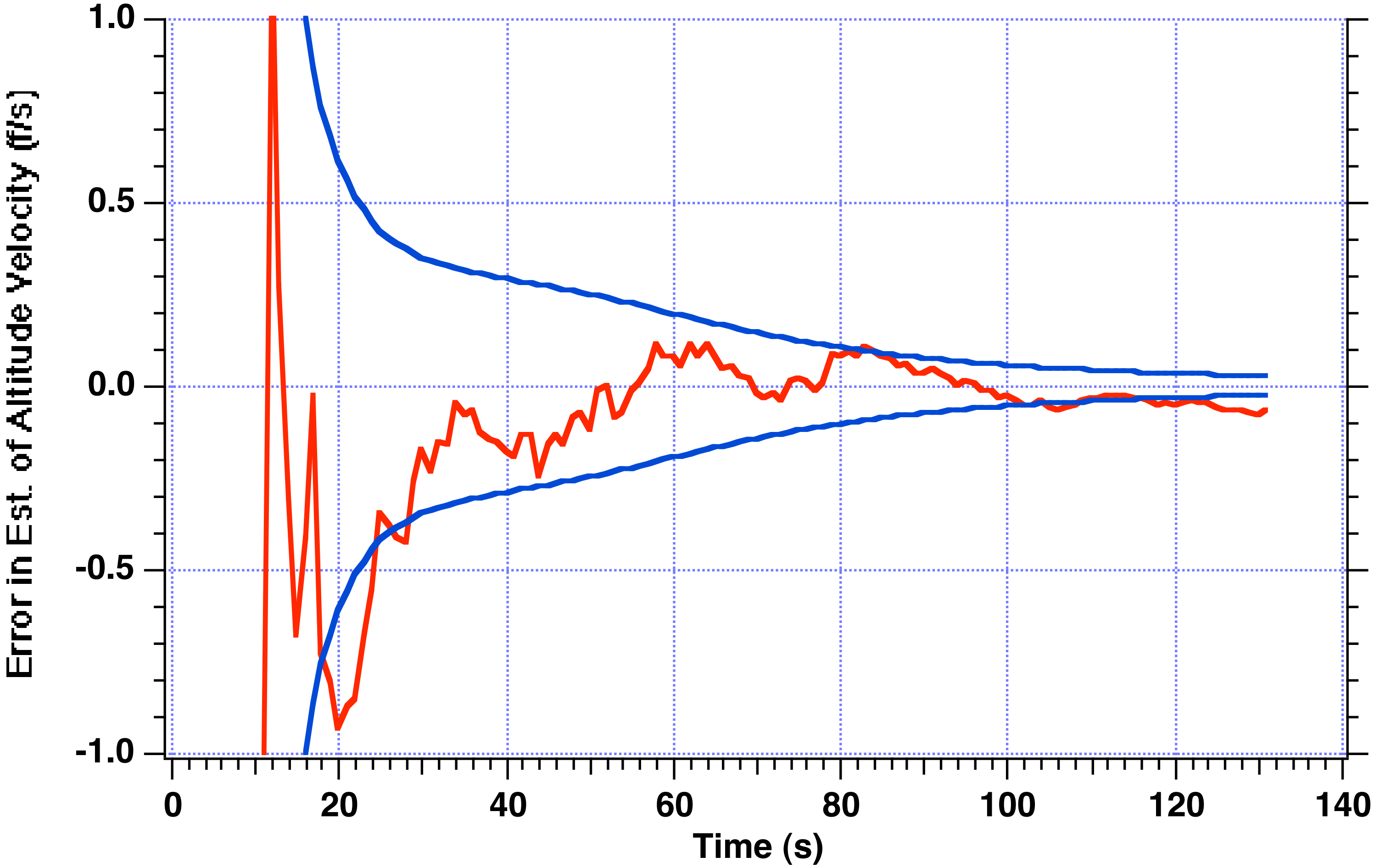
Radar measures range and angle to target

Previous Results From Extended Cartesian Kalman Filter For Radar Tracking Problem With Great Initialization

(T_s=1 s, σ_R=100 ft, σ_θ=.01 r)



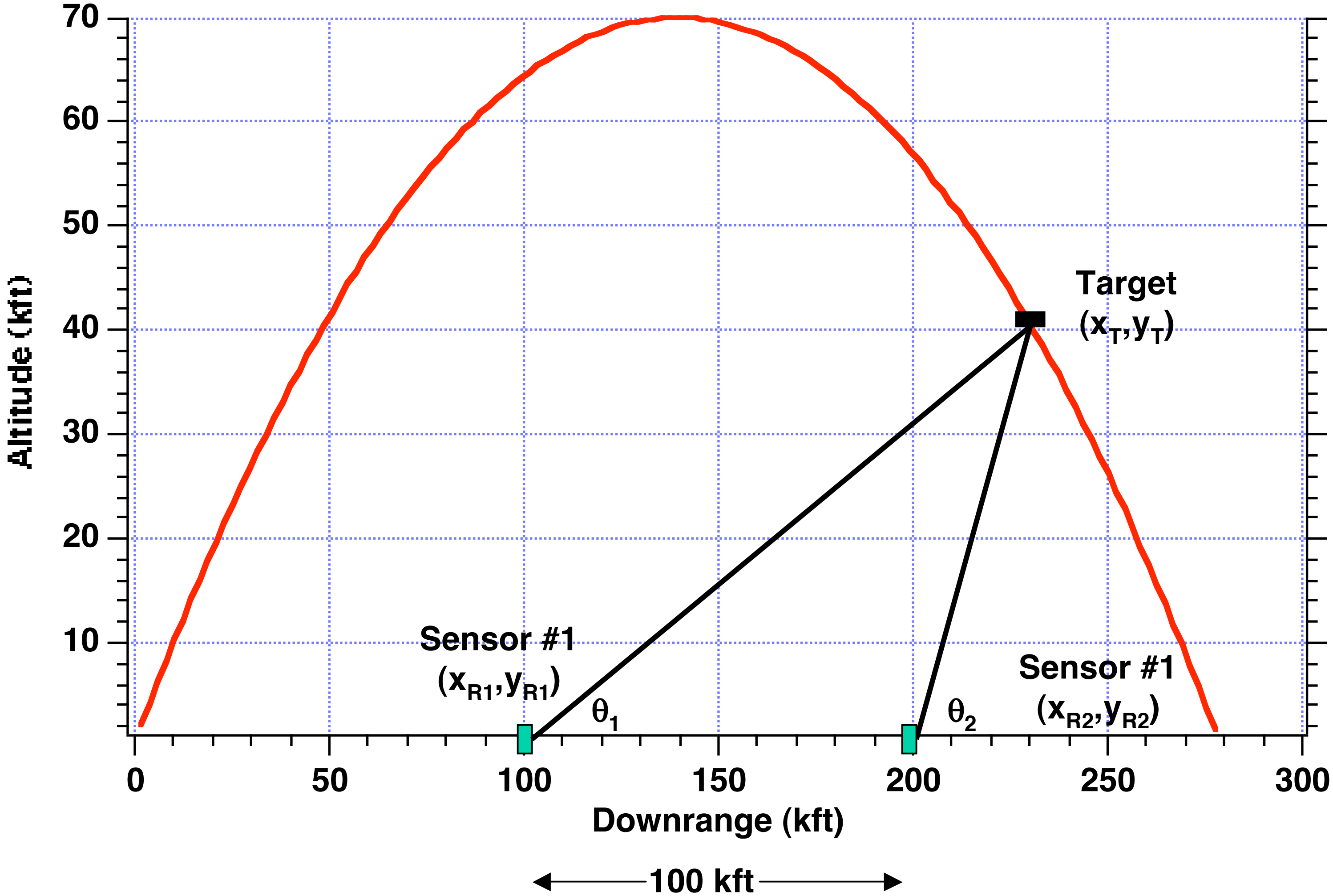
Same Extended Cartesian Kalman Filter For Radar Tracking Problem With Different Inputs (T_s=1 s, σ_R=2 ft, σ_θ=.001 r)



What if We Had a Sensor With No Range Measurement But Much Better Angle Measurement

- Usually one angle only sensor is not sufficient for estimating target position and velocity in a timely fashion
- Two angle only sensors are required to triangulate on the target to get target position and velocity quickly. This is known as stereo tracking

New Tracking Problem For Cannon Ball With Two Angle Only Sensors



Sensors measure angle to target

Expressing Target Location in Terms of Sensor Angles

From previous slide

$$\tan\theta_1 = \frac{y_T}{x_T - x_{R1}} \quad and \quad \tan\theta_2 = \frac{y_T}{x_T - x_{R2}}$$

After some algebraic manipulation we find that

$$x_T = \frac{x_{R1}\tan\theta_1 - x_{R2}\tan\theta_2}{\tan\theta_1 - \tan\theta_2}$$

$$y_T = \frac{\tan\theta_1 \tan\theta_2 (x_{R1} - x_{R2})}{\tan\theta_1 - \tan\theta_2}$$

Although we are actually measuring θ_1 and θ_2 we can pretend we are measuring x_T and y_T

We will build two linear polynomial Kalman filters in x and y. We need To find the variance of the pseudo noise in x and y.

From the chain rule we can say that

$$\Delta x_T = \frac{\partial x_T}{\partial \theta_1}\Delta\theta_1 + \frac{\partial x_T}{\partial \theta_2}\Delta\theta_2$$

$$\Delta y_T = \frac{\partial y_T}{\partial \theta_1}\Delta\theta_1 + \frac{\partial y_T}{\partial \theta_2}\Delta\theta_2$$

Deriving Pseudo Measurement Variances - 1

From

$$x_T = \frac{x_{R1}\tan\theta_1 - x_{R2}\tan\theta_2}{\tan\theta_1 - \tan\theta_2}$$

the partial derivatives required for the first equation of the chain rule are

$$\frac{\partial x_T}{\partial \theta_1} = \frac{\tan\theta_2 (x_{R2} - x_{R1})}{\cos^2\theta_1 (\tan\theta_1 - \tan\theta_2)^2}$$

$$\frac{\partial x_T}{\partial \theta_2} = \frac{\tan\theta_1 (x_{R1} - x_{R2})}{\cos^2\theta_1 (\tan\theta_1 - \tan\theta_2)^2}$$

Since

$$\Delta x_T = \frac{\partial x_T}{\partial \theta_1}\Delta\theta_1 + \frac{\partial x_T}{\partial \theta_2}\Delta\theta_2$$

The variance of the measurement noise in downrange can be found by squaring and taking expectations of the above equation yielding

$$\sigma_{x_T}^2 = \left(\frac{\partial x_T}{\partial \theta_1}\sigma_{\theta_1}\right)^2 + \left(\frac{\partial x_T}{\partial \theta_2}\sigma_{\theta_2}\right)^2$$

Deriving Pseudo Measurement Variances - 2

From

$$y_T = \frac{\tan\theta_1 \tan\theta_2 (x_{R1} - x_{R2})}{\tan\theta_1 - \tan\theta_2}$$

the partial derivatives required for the second equation of the chain rule are

$$\frac{\partial y_T}{\partial \theta_1} = \frac{-\tan^2\theta_2 \left(x_{R1} - x_{R2}\right)}{\cos^2\theta_1 \left(\tan\theta_1 - \tan\theta_2\right)^2}$$

$$\frac{\partial y_T}{\partial \theta_2} = \frac{\tan^2\theta_1 \left(x_{R1} - x_{R2}\right)}{\cos^2\theta_1 \left(\tan\theta_1 - \tan\theta_2\right)^2}$$

Since

$$\Delta y_T = \frac{\partial y_T}{\partial \theta_1}\Delta\theta_1 + \frac{\partial y_T}{\partial \theta_2}\Delta\theta_2$$

The variance of the measurement noise in altitude can be found by squaring and taking expectations of the above equation yielding

$$\sigma_{y_T}^2 = \left(\frac{\partial y_T}{\partial \theta_1}\sigma_{\theta_1}\right)^2 + \left(\frac{\partial y_T}{\partial \theta_2}\sigma_{\theta_2}\right)^2$$

Decoupled Stereo Tracking Kalman Filters-1

```
      GLOBAL DEFINE
              INCLUDE 'quickdraw.inc'
      END
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 P(2,2),Q(2,2),M(2,2),PHI(2,2),HMAT(1,2),HT(2,1),PHIT(2,2)
      REAL*8 RMAT(1,1),IDN(2,2),PHIP(2,2),PHIPPHIT(2,2),HM(1,2)
      REAL*8 HMHT(1,1),HMHTR(1,1),HMHTRINV(1,1),MHT(2,1),K(2,1)
      REAL*8 KH(2,2),IKH(2,2)
      REAL*8 RMATY(1,1),PY(2,2),PHIPY(2,2),PHIPPHITY(2,2),MY(2,2)
      REAL*8 HMY(1,2),HMHTY(1,1),HMHTRY(1,1),HMHTRINVY(1,1)
      REAL*8 MHTY(2,1),KY(2,1),KHY(2,2),IKHY(2,2)
      INTEGER ORDER
      ORDER=2
      PHIS=0.
      TS=1.
      SIGTH1=.0001
      SIGTH2=.0001
      VT=3000.
      GAMDEG=45.
      G=32.2
      XT=0.
      YT=0.
      XTD=VT*COS(GAMDEG/57.3)
      YTD=VT*SIN(GAMDEG/57.3)
      XR1=100000.
      YR1=0.
      XR2=200000.
      YR2=0.
      OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
      OPEN(2,STATUS='UNKNOWN',FILE='COVFIL')
      T=0.
      S=0.
      H=.001
      DO 14 I=1,ORDER
      DO 14 J=1,ORDER
      PHI(I,J)=0.
      P(I,J)=0.
      Q(I,J)=0.
      IDN(I,J)=0.
      PY(I,J)=0.
 14   CONTINUE
```

Process Noise, Sampling Time And Sensor Accuracy

Sensor Location

Decoupled Stereo Tracking Kalman Filters -2

```
PHI(1,1)=1.
PHI(1,2)=TS
PHI(2,2)=1.
HMAT(1,1)=1.
HMAT(1,2)=0.
CALL MATTRN(PHI,ORDER,ORDER,PHIT)
CALL MATTRN(HMAT,1,ORDER,HT)
IDN(1,1)=1.
IDN(2,2)=1.
Q(1,1)=PHIS*TS*TS*TS/3.
Q(1,2)=PHIS*TS*TS/2.
Q(2,1)=Q(1,2)
Q(2,2)=PHIS*TS
P(1,1)=99999999.
P(2,2)=99999999.
PY(1,1)=9999999.
PY(2,2)=9999999.
XTH=0.
XTDH=0.
YTH=0.
YTDH=0.
WHILE(YT>=0.)
          XTOLD=XT
          XTDOLD=XTD
          YTOLD=YT
          YTDOLD=YTD
          XTDD=0.
          YTDD=-G
          XT=XT+H*XTD
          XTD=XTD+H*XTDD
          YT=YT+H*YTD
          YTD=YTD+H*YTDD
          T=T+H
          XTDD=0.
          YTDD=-G
          XT=.5*(XTOLD+XT+H*XTD)
          XTD=.5*(XTDOLD+XTD+H*XTDD)
          YT=.5*(YTOLD+YT+H*YTD)
          YTD=.5*(YTDOLD+YTD+H*YTDD)
          S=S+H
```

Fundamental, Measurement, Identity, Process Noise and Infinite Initial Covariance Matrices

Bad Initial State Estimates

Integrating Cannon Ball Equations Using Second-Order Runge-Kutta Integration

Decoupled Stereo Tracking Kalman Filters -3

```
        IF(S>=(TS-.00001))THEN
                S=0.
                THET1=ATAN2(YT,(XT-XR1))
                THET2=ATAN2(YT,(XT-XR2))
```

True Sensor Angles

```
                BOT=TAN(THET1)-TAN(THET2)
                DXDT1=(TAN(THET2)*(XR2-XR1))/((COS(THET1)*
     1          (TAN(THET1)-TAN(THET2)))**2)
                DXDT2=(TAN(THET1)*(XR1-XR2))/((COS(THET2)*
     1          (TAN(THET1)-TAN(THET2)))**2)
                SIGX=SQRT((DXDT1*SIGTH1)**2+(DXDT2*SIGTH2)**2)
                DYDT1=-(XR1-XR2)*TAN(THET2)*TAN(THET2)/
     1          ((COS(THET1)*(TAN(THET1)-TAN(THET2)))**2)
                DYDT2=(XR1-XR2)*TAN(THET1)*TAN(THET1)/
     1          ((COS(THET2)*(TAN(THET1)-TAN(THET2)))**2)
                SIGY=SQRT((DYDT1*SIGTH1)**2+(DYDT2*SIGTH2)**2)
```

Pseudo Measurement Angle Noise Standard Deviation

```
                CALL GAUSS(THET1NOISE,SIGTH1)
                CALL GAUSS(THET2NOISE,SIGTH2)
                THET1S=THET1+THET1NOISE
                THET2S=THET2+THET2NOISE
```

Actual Angle Measurements

```
                BOTS=TAN(THET1S)-TAN(THET2S)
                XTS=(XR1*TAN(THET1S)-XR2*TAN(THET2S))/BOTS
                YTS=(TAN(THET1S)*TAN(THET2S)*(XR1-XR2))/BOTS
```

Pseudo Position Measurements

```
                XTNOISE=XT-XTS
                YTNOISE=YT-YTS
                RMAT(1,1)=SIGX**2
                CALL MATMUL(PHI,ORDER,ORDER,P,ORDER,ORDER,PHIP)
                CALL MATMUL(PHIP,ORDER,ORDER,PHIT,ORDER,ORDER,
     1                  PHIPPHIT)
                CALL MATADD(PHIPPHIT,ORDER,ORDER,Q,M)
                CALL MATMUL(HMAT,1,ORDER,M,ORDER,ORDER,HM)
                CALL MATMUL(HM,1,ORDER,HT,ORDER,1,HMHT)
                CALL MATADD(HMHT,1,1,RMAT,HMHTR)
                HMHTRINV(1,1)=1./HMHTR(1,1)
                CALL MATMUL(M,ORDER,ORDER,HT,ORDER,1,MHT)
                CALL MATMUL(MHT,ORDER,1,HMHTRINV,1,1,K)
                CALL MATMUL(K,ORDER,1,HMAT,1,ORDER,KH)
                CALL MATSUB(IDN,ORDER,ORDER,KH,IKH)
                CALL MATMUL(IKH,ORDER,ORDER,M,ORDER,ORDER,P)
```

Ricatti Equations For Downrange Filter

Decoupled Stereo Tracking Kalman Filters -4

```
            RES1=XTS-XTH-TS*XTDH
            XTH=XTH+TS*XTDH+K(1,1)*RES1
            XTDH=XTDH+K(2,1)*RES1
```

Two-State Kalman Filter Equations For Downrange Filter

```
            RMATY(1,1)=SIGY**2
            CALL MATMUL(PHI,ORDER,ORDER,PY,ORDER,ORDER,
1                       PHIPY)
            CALL MATMUL(PHIPY,ORDER,ORDER,PHIT,ORDER,ORDER
1                       ,PHIPPHITY)
            CALL MATADD(PHIPPHITY,ORDER,ORDER,Q,MY)
            CALL MATMUL(HMAT,1,ORDER,MY,ORDER,ORDER,HMY)
            CALL MATMUL(HMY,1,ORDER,HT,ORDER,1,HMHTY)
            CALL MATADD(HMHTY,1,1,RMATY,HMHTRY)
            HMHTRINVY(1,1)=1./HMHTRY(1,1)
            CALL MATMUL(MY,ORDER,ORDER,HT,ORDER,1,MHTY)
            CALL MATMUL(MHTY,ORDER,1,HMHTRINVY,1,1,KY)
            CALL MATMUL(KY,ORDER,1,HMAT,1,ORDER,KHY)
            CALL MATSUB(IDN,ORDER,ORDER,KHY,IKHY)
            CALL MATMUL(IKHY,ORDER,ORDER,MY,ORDER,ORDER,PY)
```

Ricatti Equations For Altitude Filter

```
            RES2=YTS-YTH-TS*YTDH+.5*TS*TS*G
            YTH=YTH+TS*YTDH-.5*TS*TS*G+KY(1,1)*RES2
            YTDH=YTDH-TS*G+KY(2,1)*RES2
```

Two-State Kalman Filter Equations For Altitude Filter

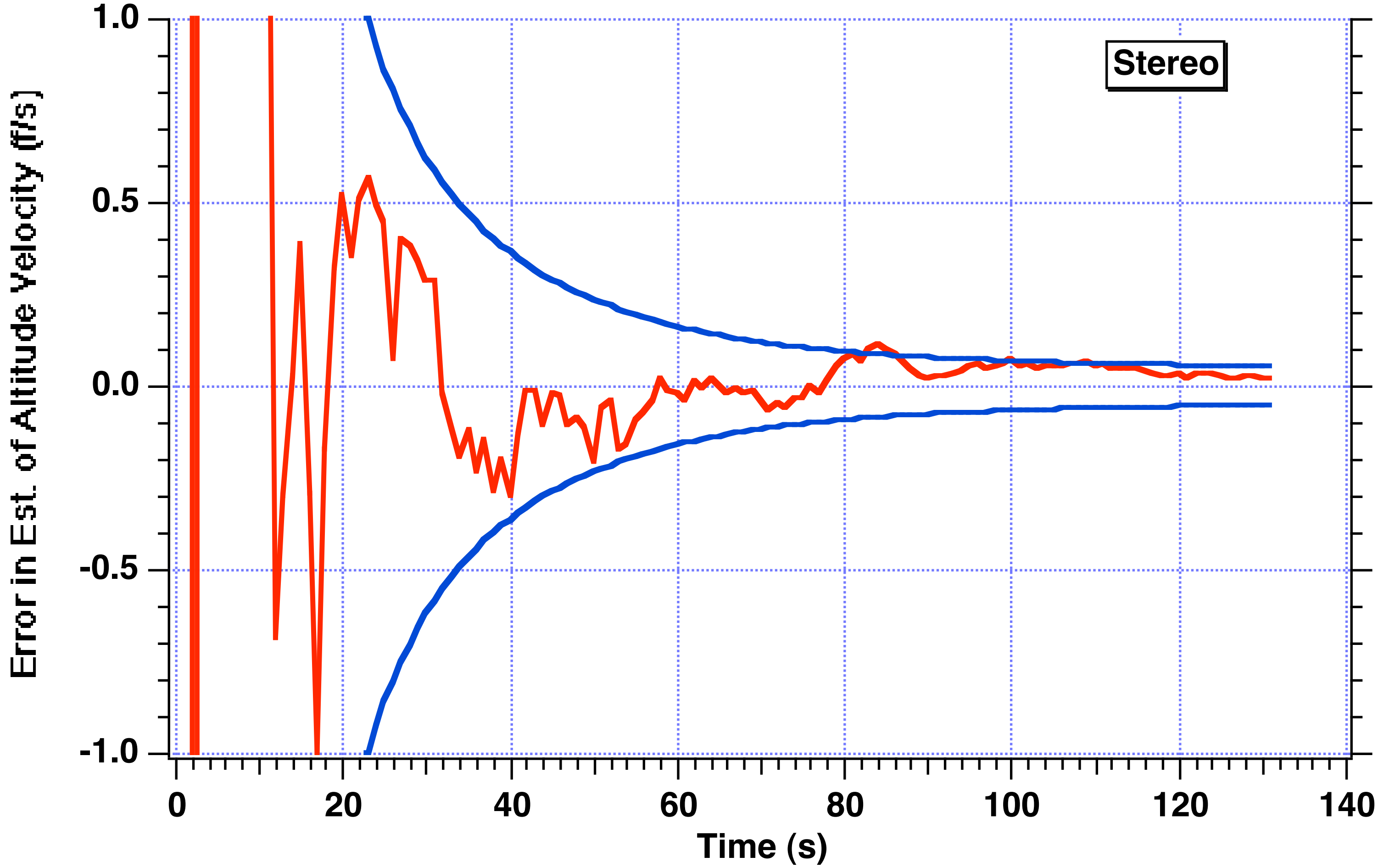
```
            ERRX=XT-XTH
            SP11=SQRT(P(1,1))
            ERRXD=XTD-XTDH
            SP22=SQRT(P(2,2))
            ERRY=YT-YTH
            SP11Y=SQRT(PY(1,1))
            ERRYD=YTD-YTDH
            SP22Y=SQRT(PY(2,2))
```

Comparing Actual and Theoretical Errors in the Estimates

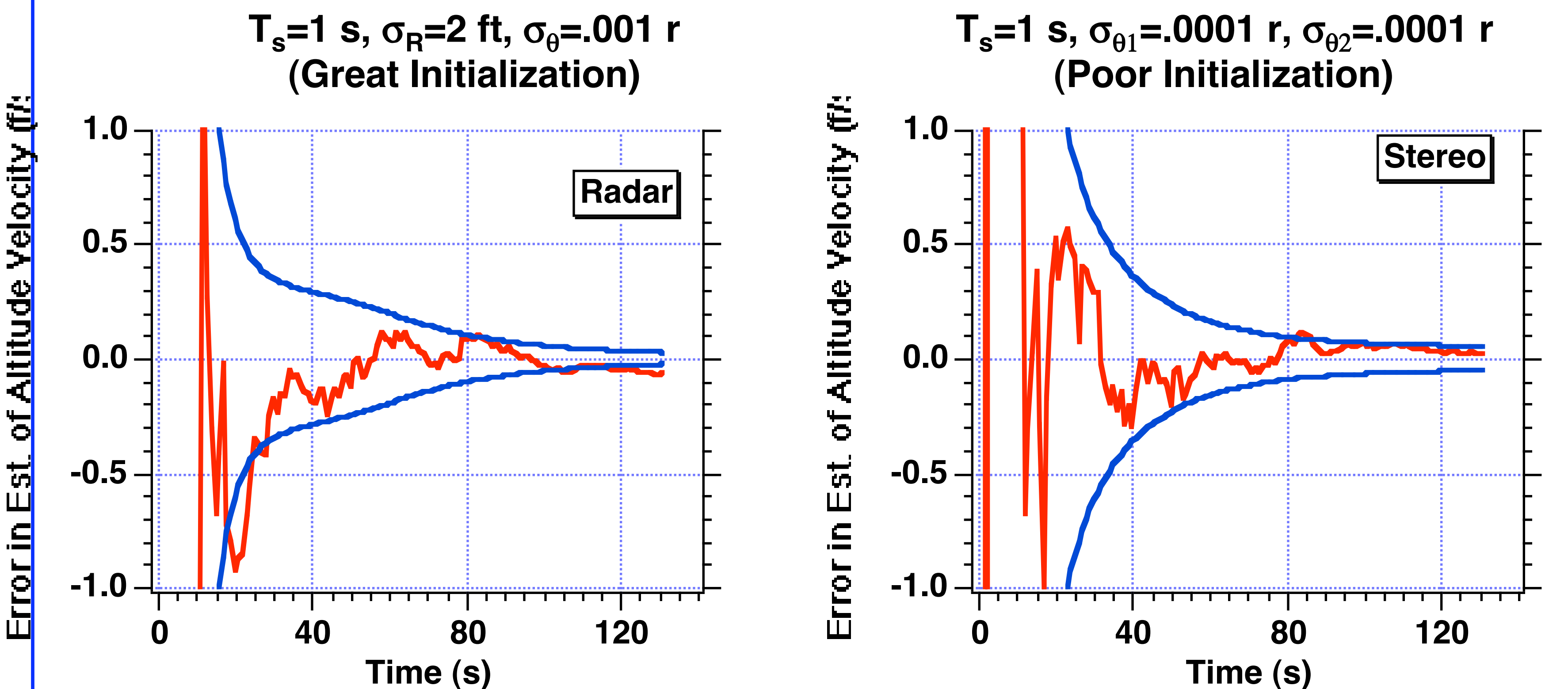
```
            WRITE(9,*)T,XT,YT
            WRITE(1,*)T,XT,YT
            WRITE(2,*)T,ERRX,SP11,-SP11,ERRXD,SP22,-SP22,
1                       ERRY,SP11Y,-SP11Y,ERRYD,SP22Y,-SP22Y

        ENDIF
     END DO
     PAUSE
     CLOSE(1)
     CLOSE(2)
     END
```

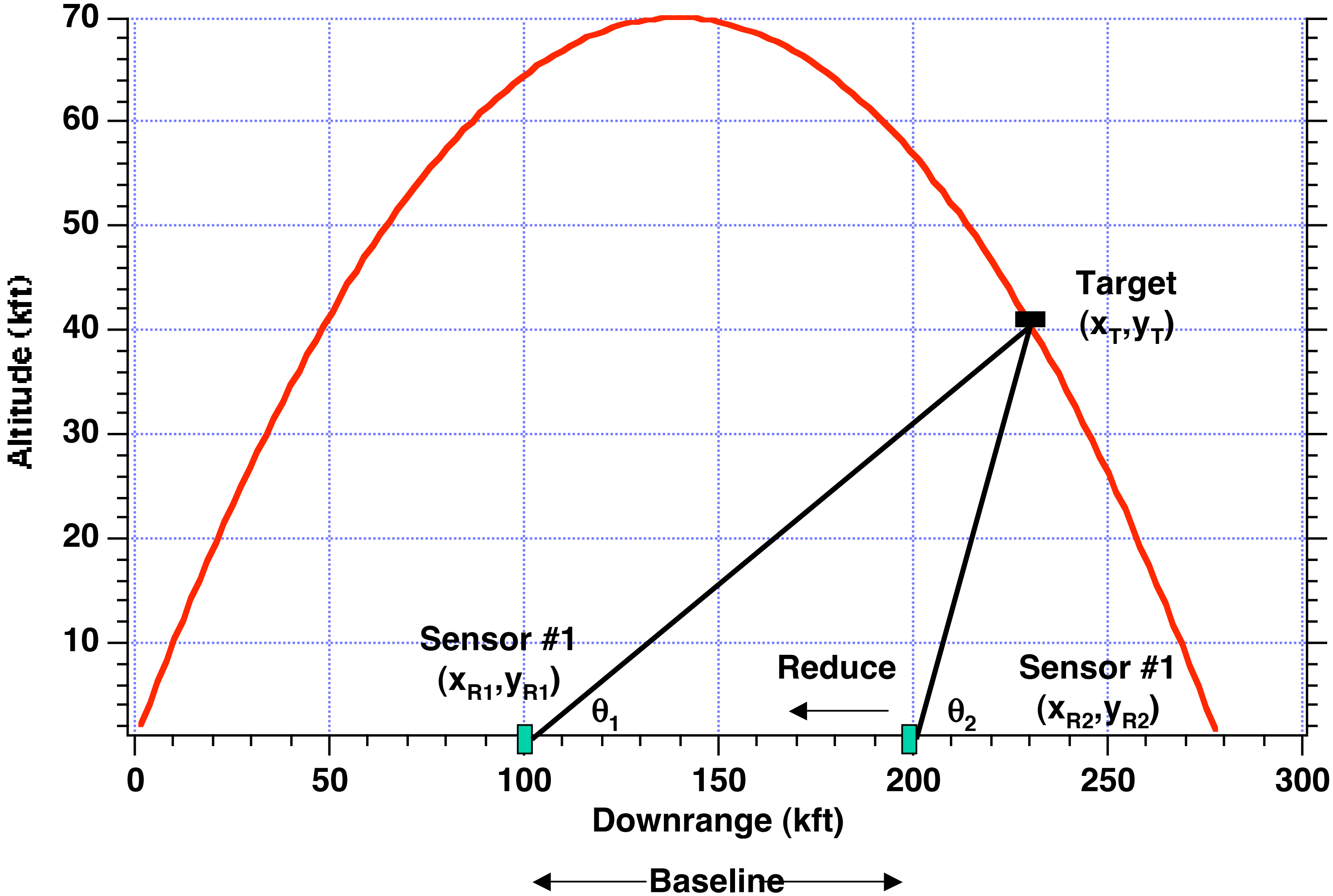
Stereo Tracking With 2 Decoupled Linear Polynomial Kalman Filters - Poor Initialization (T_s=1 s, $\sigma_{\theta 1}$=.0001 r, $\sigma_{\theta 2}$=.0001 r)



Stereo Tracking Can Yield Similar Results to Radar Tracking

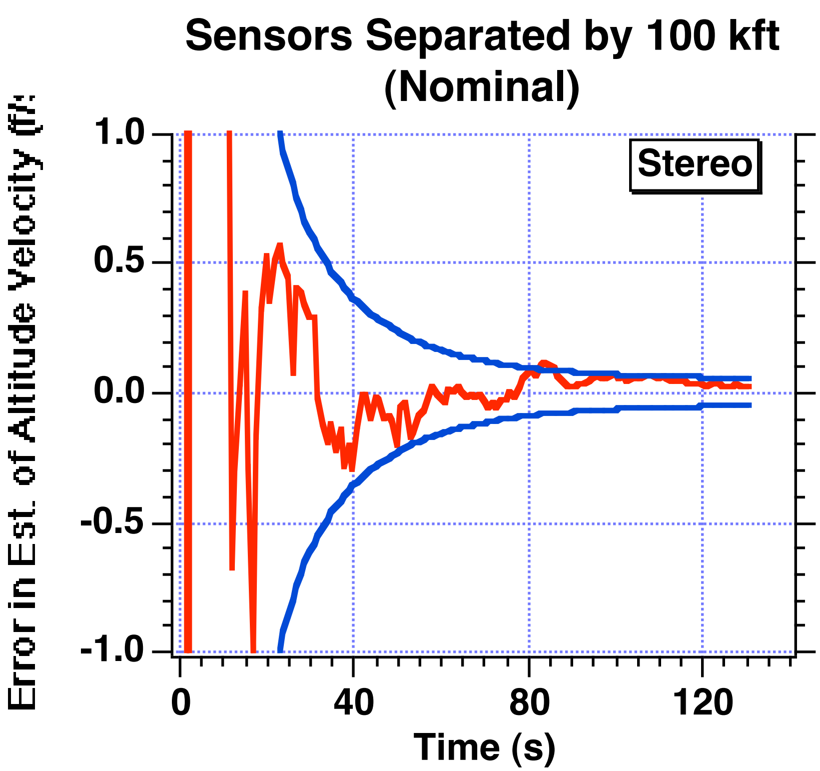


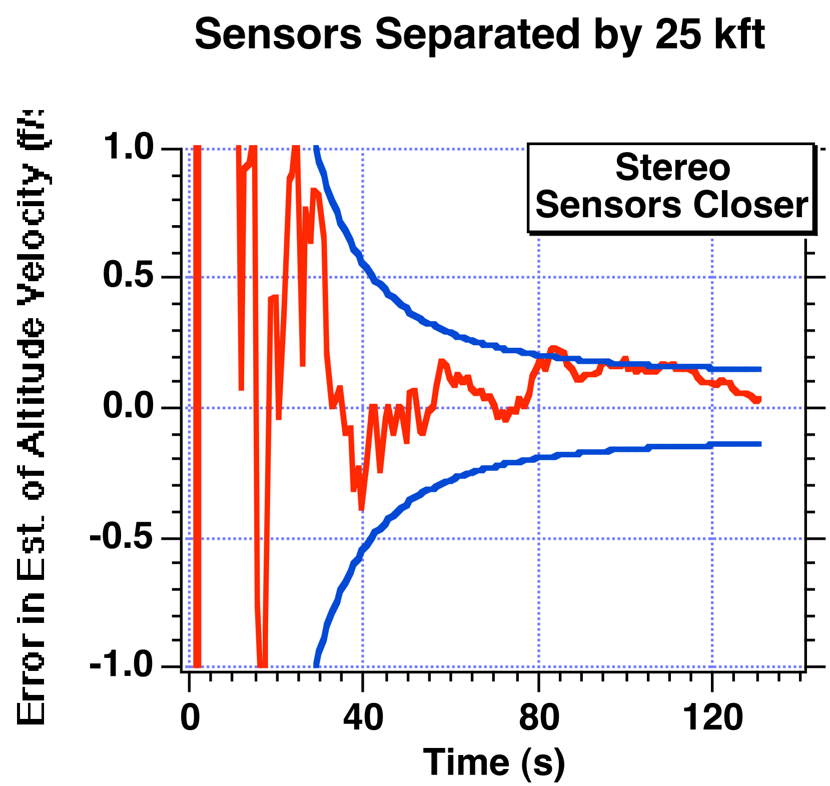
We Will Now See How Estimation Accuracy For Stereo Tracking Changes if We Reduce Sensor Baseline



Sensors measure angle to target

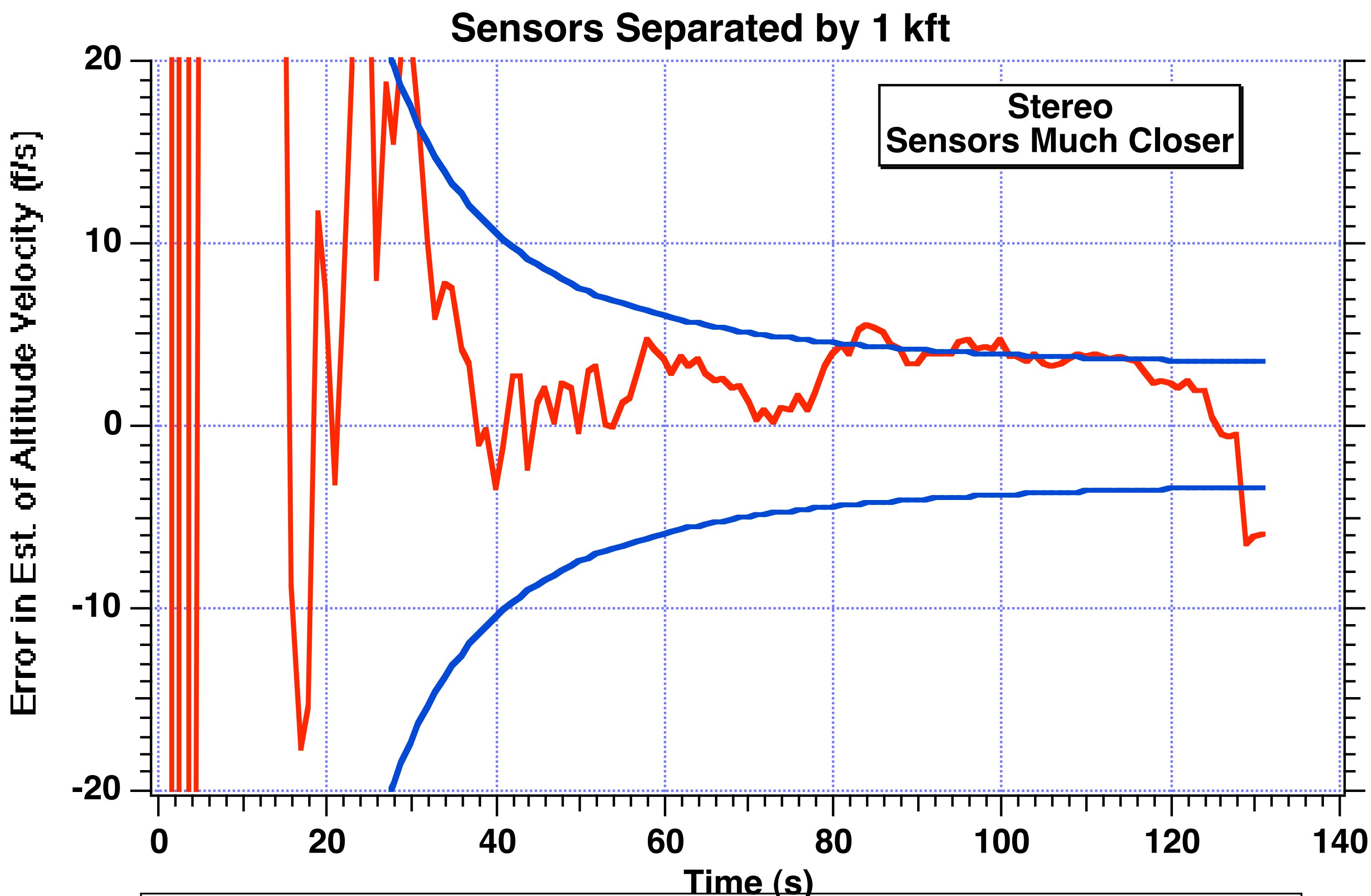
Estimation With Stereo Tracking Depends on Sensor Geometry





Effectiveness of stereo tracking depends on sensor baseline

Estimation With Stereo Tracking Degrades Severely When Sensors Are Very Close

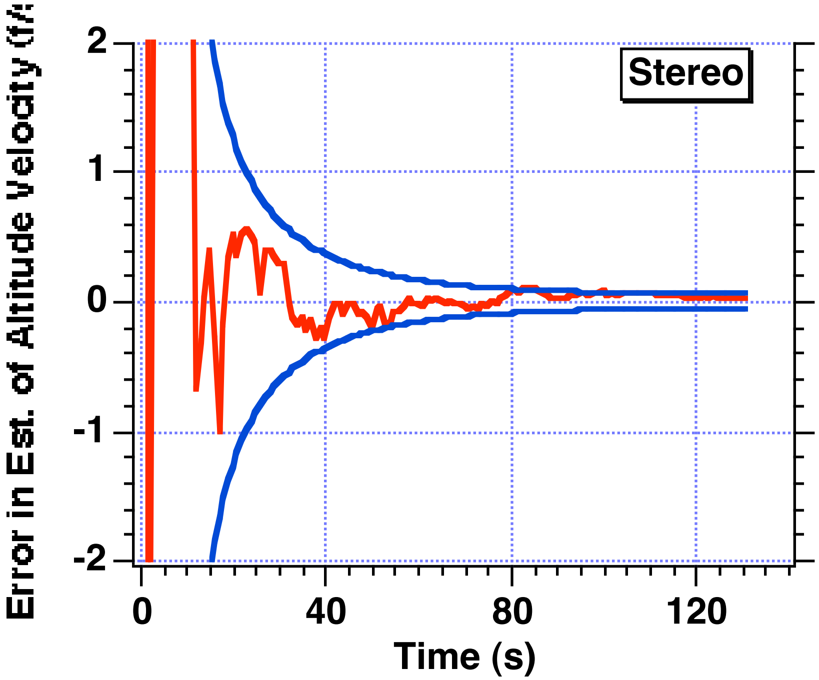


Effectiveness of stereo tracking depends on sensor baseline

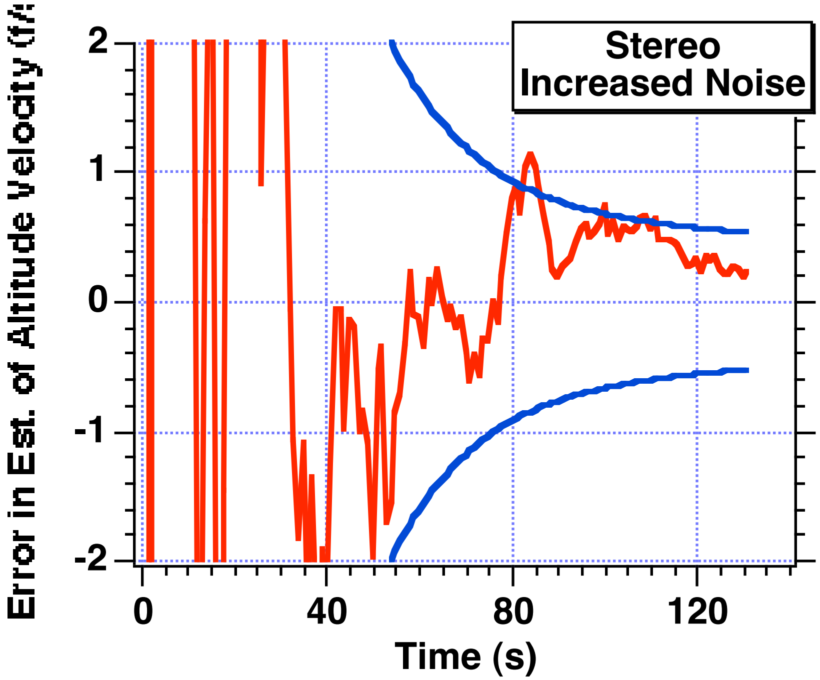
*Note that y-axis scale is more than an order of magnitude greater than on previous slides

Increasing Sensor Noise By Order of Magnitude Degrades Stereo Estimates (100 kft Baseline)

T_s=1 s, $\sigma_{\theta 1}$=.0001 r, $\sigma_{\theta 2}$=.0001 r
(Nominal)

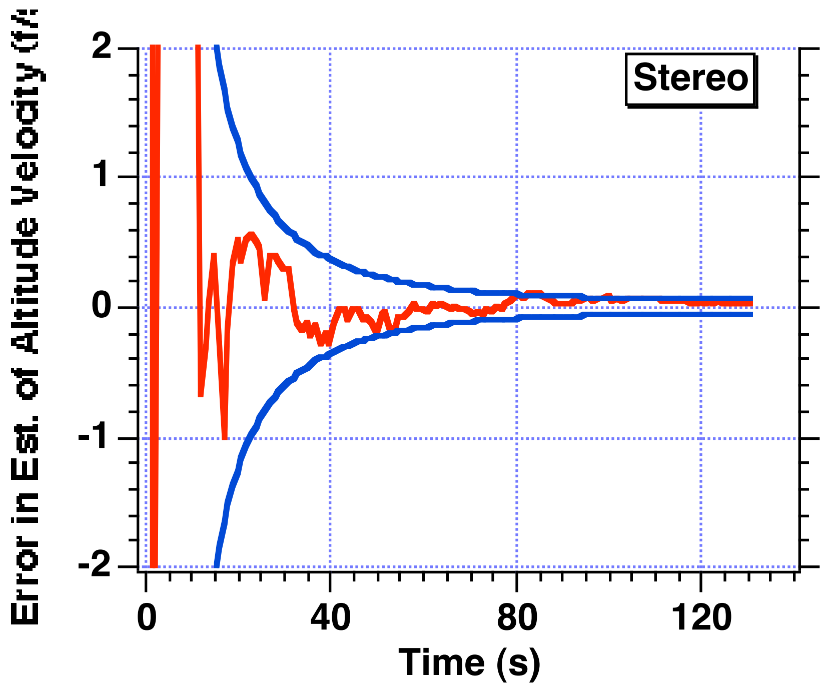


T_s=1 s, $\sigma_{\theta 1}$=.001 r, $\sigma_{\theta 2}$=.001 r



Increasing Data By Order of Magnitude Significantly Improves Stereo Estimates (100 kft Baseline)

T_s=1 s, $\sigma_{\theta 1}$=.0001 r, $\sigma_{\theta 2}$=.0001 r
(Nominal)



T_s=0.1 s, $\sigma_{\theta 1}$=.001 r, $\sigma_{\theta 2}$=.001 r