UNIVERSITY OF MICHIGAN

AERO 584, Final Exam

Huckleberry Febbo

December 15, 2017

PROBLEM #1

PART A:

GIVEN:

$$\dot{\underline{X}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \underline{X}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a \qquad X(t_0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \underline{X}(t)$$

WHERE, $\underline{X}(t) = \begin{bmatrix} r(t) \\ V(t) \end{bmatrix}$

DETERMINE:

USE METHOD OF ADJOINTS TO EVALUATE THE POSITION OF VEHICLE AT FINAL TIME

ASSUMPTIONS:

$t_0 = 0$

SOLUTION:

$$\dot{\underline{P}}(t) = -A^T(t)\,\underline{P}(t)$$

$$\underline{P}(t_f) = C^T(t_f)$$

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad -A^T = \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad C^T(t_f) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

PROB 1, PS # 2

$$\dot{P}(t) = \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix} P(t)$$

$$\dot{P}_1(t) = 0 \qquad P_1(t_f) = 1 \quad \rightarrow \quad P_1(t) = 1$$

$$\dot{P}_2(t) = -P_1(t) \qquad P_2(t_f) = 0$$

$$\frac{dP_2(t)}{dt} = -1 \quad \rightarrow \quad \int_{P_2(t_f)}^{P_2(t)} dP_2(t) = -\int_{t_f}^{t} dt$$

$$P_2(t) - \cancelto{0}{P_2(t_f)} = -[t - t_f]$$

$$P_2(t) = t_f - t$$

So, $\underline{P}(t) = \begin{bmatrix} 1 \\ t_f - t \end{bmatrix}$ & $P(\tau) = \begin{bmatrix} 1 \\ t_f - \tau \end{bmatrix}$

FORWARD INTEGRATION,

$$Y(t_f) = P^T(t_0)\cancelto{0}{X(t_0)} + \int_{t_0}^{t_f} P^T(\tau)\, B(\tau)\, u(\tau)\, d\tau$$

$$= \int_{t_0}^{t_f} \begin{bmatrix} 1 & t_f - \tau \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} a\, d\tau = a\int_{t_0}^{t_f} (t_f - \tau)\, d\tau$$

$$= a t_f^2 - \frac{a t_f^2}{2}$$ So, $\boxed{Y(t_f) = \frac{a t_f^2}{2}}$

Part B:

For an LTI system,

$$\Phi(t,\tau) = e^{A(t-\tau)} \qquad \mathcal{L}^{-1}\left\{(SI-A)^{-1}\right\} = e^{At}$$

$$(SI-A)^{-1} = \begin{bmatrix} S & -1 \\ 0 & S \end{bmatrix}^{-1} = \frac{1}{S^2}\begin{bmatrix} S & 1 \\ 0 & S \end{bmatrix} = \begin{bmatrix} \frac{1}{S} & \frac{1}{S^2} \\ 0 & \frac{1}{S} \end{bmatrix}$$

So,

$$\Phi(t,0) = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}$$

Then,

$$y(t_f) = C(t_f)\Phi(t,0)\cancel{X_0}^{\,0} + \int_{t_0}^{t_f} C(t_f)\Phi(\tau,0)\,B(\tau)u(\tau)\,d\tau$$

$$* \quad [1\ 0]\begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} a = [1\ \tau]\begin{bmatrix} 0 \\ 1 \end{bmatrix} a = \tau a$$

$$y(t_f) = \int_{t_0=0}^{t_f} \tau a\, d\tau = \left.\frac{a\tau^2}{2}\right|_0^{t_f} = \frac{a t_f^2}{2} \quad \checkmark$$

$\Rightarrow$ Matches result from Part A!

$\boxed{\text{PROBLEM \#2}}$

PART A:

$$\frac{d\psi}{dt} = \frac{V}{R}$$

ASSUME $t_0 = 0$

$$\int_{\psi(0)}^{\psi(t)} d\psi = \int_0^t \frac{V}{R}\,dt$$

$$\psi(t) = \frac{V}{R}t \qquad \psi(t_f) \Rightarrow \frac{V}{R}t_f = \pi$$

So, THE AIRCRAFT HAS A 180° AT:

$$\boxed{t_f = \frac{R}{V}\pi}$$

$$\frac{dX}{dt} = V\cos\left(\frac{V}{R}t\right)$$

$$\int_0^{X(t)} dX = \int_0^t V\cos\left(\frac{V}{R}\tau\right)d\tau$$

$$X(t) = V\int_0^t \cos(u)\,du$$

$$u = \frac{V}{R}\tau$$

LIMITS

$t = 0 \Rightarrow u = 0$

$t = t \Rightarrow u = \frac{V}{R}t$

$$du = \frac{V}{R}d\tau \Rightarrow d\tau = \frac{R}{V}du$$

So, $$X(t) = \cancel{V}\int_0^{V/R\,t} \cos(u)\frac{R}{\cancel{V}}du$$

$$= R\sin(u)\Big|_0^{\frac{V}{R}t}$$

$$= R\sin\left(\frac{V}{R}t\right)$$

THUS, AT $t_f = \frac{R}{V}\pi$:

$$X(t_f) = R\sin\left(\frac{\cancel{V}}{\cancel{R}}\frac{\cancel{R}}{\cancel{V}}\pi\right)$$

$$X(t_f) = R\sin(\pi) = 0$$

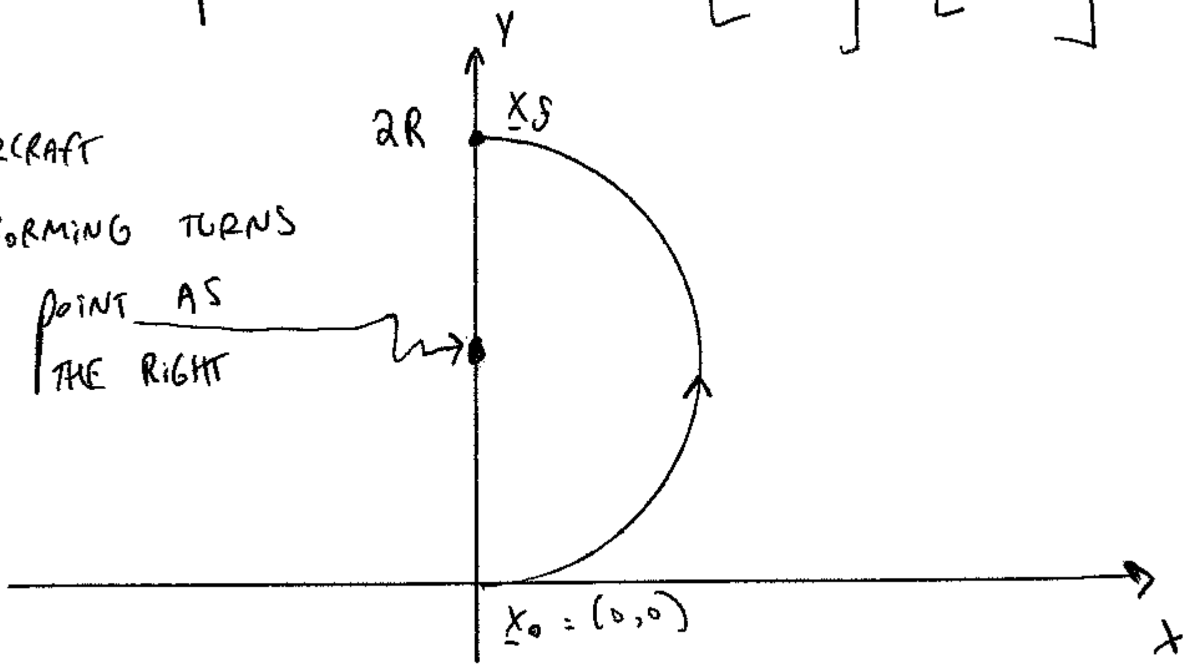
SIMILARLY FOR Y,
$$\int_0^{Y(t)} dY = \int_0^t V\sin\left(\frac{V}{R}t\right)dt \Rightarrow Y(t) = -R\left[\cos\left(\frac{V}{R}t\right) - 1\right]$$

$$Y(t_f) = -R\left[\cos(\pi) - \cos(0)\right] = -R[-1-1]$$

SO, AT t_f: $Y(t_f) = 2R$

THUS, THE POSITION AT t_f IS: $\begin{bmatrix} X(t_f) \\ Y(t_f) \end{bmatrix} = \begin{bmatrix} 0 \\ 2R \end{bmatrix}$

THE AIRCRAFT IS PERFORMING TURNS AROUND A POINT AS SHOWN TO THE RIGHT



// PART B:

GIVEN: R^0 & V^0

FIND: $\delta X(t)$ & $\delta Y(t)$ IN TERMS OF δR & δV

$$\left.\frac{\partial X(t)}{\partial R}\right|_0 = V_0 \sin\left(\frac{V_0}{R_0}t\right) V_0 t R_0^{-2}$$

$$\left.\frac{\partial X(t)}{\partial V}\right|_0 = \cos\left(\frac{V_0}{R_0}t\right) - \frac{V_0}{R_0}\sin\left(\frac{V_0}{R_0}t\right) t$$

$$\left.\frac{\partial Y(t)}{\partial R}\right|_0 = -\frac{V_0^2}{R_0^2}\cos\left(\frac{V_0}{R_0}t\right) t$$

$$\left.\frac{\partial Y(t)}{\partial V}\right|_0 = \sin\left(\frac{V_0}{R_0}t\right) + \frac{V_0}{R_0}\cos\left(\frac{V_0}{R_0}t\right) t$$

THEN,

$$\begin{bmatrix} \delta X(t) \\ \delta Y(t) \end{bmatrix} = \begin{bmatrix} \left.\frac{\partial X(t)}{\partial R}\right|_0 & \left.\frac{\partial X(t)}{\partial V}\right|_0 \\ \left.\frac{\partial Y(t)}{\partial R}\right|_0 & \left.\frac{\partial Y(t)}{\partial V}\right|_0 \end{bmatrix} \begin{bmatrix} \delta R \\ \delta V \end{bmatrix}$$

So,

$$\underbrace{\begin{bmatrix} \delta X(t) \\ \delta Y(t) \end{bmatrix}}_{\delta X} = \underbrace{\begin{bmatrix} \frac{V_0^2}{R_0^2}\sin\left(\frac{V_0}{R_0}t\right) t & \cos\left(\frac{V_0}{R_0}t\right) - \frac{V_0}{R_0}\sin\left(\frac{V_0}{R_0}t\right) t \\ -\frac{V_0^2}{R_0^2}\cos\left(\frac{V_0}{R_0}t\right) t & \sin\left(\frac{V_0}{R_0}t\right) + \frac{V_0}{R_0}\cos\left(\frac{V_0}{R_0}t\right) t \end{bmatrix}}_{A(t)} \underbrace{\begin{bmatrix} \delta R \\ \delta V \end{bmatrix}}_{\delta Q}$$

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

// PART C: GIVEN THE SYSTEM FROM PART B: P2, PB #4

$\delta X(t) = A(t)\,\delta Q$

THE COVARIANCE CAN BE DEFINED AS:

$P_X(t) = A(t)\, P\, A(t)^T$

WHERE THE DIAGONAL TERMS IN $P_X(t)$ WILL BE THE VARIANCES OF $\delta X(t)$ & $\delta Y(t)$.

$$P_X = \begin{bmatrix} P_{X11} & P_{X12} \\ P_{X21} & P_{X22} \end{bmatrix}$$

// TERM BY TERM:

$$A(t)\, P\, A^T(t) = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \sigma_{RR} & \sigma_{RV} \\ \sigma_{VR} & \sigma_{VV} \end{bmatrix} \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix}$$

$$= \begin{bmatrix} (a_{11}\sigma_{RR} + a_{12}\sigma_{VR}) & (a_{11}\sigma_{RV} + a_{12}\sigma_{VV}) \\ (a_{21}\sigma_{RR} + a_{22}\sigma_{VR}) & (a_{21}\sigma_{RV} + a_{22}\sigma_{VV}) \end{bmatrix} \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix}$$

$$= \begin{bmatrix} \underbrace{\left(a_{11}(a_{11}\sigma_{RR} + a_{12}\sigma_{VR}) + a_{12}(a_{11}\sigma_{RV} + a_{12}\sigma_{VV})\right)}_{P_{X11}} & P_{X12} \\ P_{X21} & \underbrace{\left(a_{21}(a_{21}\sigma_{RR} + a_{22}\sigma_{VR}) + a_{22}(a_{21}\sigma_{RV} + a_{22}\sigma_{VV})\right)}_{P_{X22}} \end{bmatrix}$$

$$P_{X11} = a_{11}^2\sigma_{RR} + a_{11}a_{12}\sigma_{VR} + a_{12}a_{11}\sigma_{RV} + a_{12}^2\sigma_{VV}$$

$$P_{X22} = a_{21}^2\sigma_{RR} + a_{21}a_{22}\sigma_{VR} + a_{22}a_{21}\sigma_{RV} + a_{22}^2\sigma_{VV}$$

THE VARIANCE OF $\delta x(t)$ IS:

$$P_{x11} = \left[\frac{V_o^2}{R_o^2}\sin\left(\frac{V_o}{R_o}t\right)t\right]^2 \sigma_{RR} + \left[\frac{V_o^2}{R_o^2}\sin\left(\frac{V_o}{R_o}t\right)t\right]\left[\cos\left(\frac{V_o}{R_o}t\right) - \frac{V_o}{R_o}\sin\left(\frac{V_o}{R_o}t\right)t\right]\left[\sigma_{VR}+\sigma_{RV}\right]$$

$$+\left[\cos\left(\frac{V_o}{R_o}t\right) - \frac{V_o}{R_o}\sin\left(\frac{V_o}{R_o}t\right)t\right]^2 \sigma_{VV}$$

THE VARIANCE OF $\delta y(t)$ IS:

$$P_{x22} = \left[-\frac{V_o^2}{R_o^2}\cos\left(\frac{V_o}{R_o}t\right)t\right]^2 \sigma_{RR} + \left[-\frac{V_o^2}{R_o^2}\cos\left(\frac{V_o}{R_o}t\right)t\right]\left[\sin\left(\frac{V_o}{R_o}t\right) + \frac{V_o}{R_o}\cos\left(\frac{V_o}{R_o}t\right)t\right]\left[\sigma_{VR}+\sigma_{RV}\right]$$

$$+\left[\sin\left(\frac{V_o}{R_o}t\right) + \frac{V_o}{R_o}\cos\left(\frac{V_o}{R_o}t\right)t\right]\sigma_{VV}$$

NOTE: COVARIANCE MATRICES ARE SYMETRIC SO, σ_{VR} SHOULD EQUAL σ_{RV}, BUT IT IS LEFT AS SEPERATE.

II PART D:

$$\left.\begin{aligned}\dot{X}(t) &= f(x,u,t) + w(t)\\ Y(t) &= g(x,t) + V(t)\end{aligned}\right\} \text{NONLINEAR MARKOV MODEL}$$

ASSUMPTIONS:

① $V(t)$ & $w(t)$ ARE GAUSSIAN WHITENOISE PROCESSES REPRESENTING DISTURBANCES

② OBSERVATION ERROR $\tilde{X}(t) = X(t) - \hat{X}(t)$ IS SMALL, SO OBSERVATION ERROR NONLINEAR TERMS CAN BE NEGLECTED IN OBSERVATION ERROR DYNAMICS

TO SOLVE THIS PROBLEM WE WILL USE AN EXTENDED KALMAN FILTER.

FIRST WE DEFINE THE STATE AS $X(t) = \begin{bmatrix} x(t) \\ y(t) \\ w(t) \end{bmatrix}$, WHERE

$w = \frac{V}{R}$ IS ADDED. THE SYSTEM IS THEN:

$$\dot{X}_1(t) = V\cos(wt) + w_1(t)$$

$$\dot{X}_2(t) = V\sin(wt) + w_2(t)$$

$$\dot{X}_3(t) = w + w_3(t)$$

$R = \sqrt{x^2 + y^2}$
- DON'T KNOW R IN PART (d)
- NEED TO ESTIMATE IT VIA $\hat{x}$ & $\hat{y}$

WITH OUTPUT EQUATIONS:

$$Y(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} (x^2 + y^2)^{1/2} \sin(wt) + v_1(t) \\ (x^2 + y^2)^{1/2} (1 - \cos(wt)) + v_2(t) \end{bmatrix}$$

WHERE,
$$w(t) = \begin{bmatrix} w_1(t) \\ w_2(t) \\ w_3(t) \end{bmatrix} \quad \& \quad V(t) = \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix}$$

NEXT WE NEED TO LINEARIZE OUR SYSTEM DYNAMICS AROUND $\hat{X}$ (THE ESTIMATE OF THE STATE):

$$\dot{X}(t) = f(x,u,t) + w(t) \cong f(\hat{x},u,t) + \left(\frac{\partial f}{\partial x}\right)\Bigg|_{\hat{x}} \underbrace{(X - \hat{X})}_{\tilde{X}\ (\text{ERROR})} + w(t)$$

WHERE,

$$\left.\frac{\partial f}{\partial x}\right|_{\hat{x}} = \left.\begin{bmatrix} \frac{\partial \dot{x}_1}{\partial x} & \frac{\partial \dot{x}_1}{\partial y} & \frac{\partial \dot{x}_1}{\partial w} \\ \frac{\partial \dot{x}_2}{\partial x} & \frac{\partial \dot{x}_2}{\partial y} & \frac{\partial \dot{x}_2}{\partial w} \\ \frac{\partial \dot{x}_3}{\partial x} & \frac{\partial \dot{x}_3}{\partial y} & \frac{\partial \dot{x}_3}{\partial w} \end{bmatrix}\right|_{\hat{x}} \qquad \text{WHERE, } \hat{X} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{w} \end{bmatrix}$$

P2, PS#7

$$\frac{\partial \dot{x}_1}{\partial x} = 0 , \qquad \frac{\partial \dot{x}_1}{\partial y} = 0, \quad \frac{\partial \dot{x}_1}{\partial w} = -V\sin(wt)\,t$$

$$\frac{\partial \dot{x}_2}{\partial x} = 0 , \qquad \frac{\partial \dot{x}_2}{\partial y} = 0 , \quad \frac{\partial \dot{x}_2}{\partial \omega} = +V\cos(\omega t)\,t$$

$$\frac{\partial \dot{x}_3}{\partial x} = 0 , \qquad \frac{\partial \dot{x}_3}{\partial y} = 0 , \quad \frac{\partial \dot{x}_3}{\partial w} = 1$$

So,

$$\left(\frac{\partial f}{\partial x}\right)\Bigg|_{\hat{x}} = \begin{bmatrix} 0 & 0 & -V\sin(\hat{\omega}t)\,t \\ 0 & 0 & V\cos(\hat{\omega}t)\,t \\ 0 & 0 & 1 \end{bmatrix}$$

THEN OUR SYSTEM DYNAMICS ARE:

$$\dot{\tilde{X}}(t) = \begin{bmatrix} V\cos(\hat{\omega}t) \\ V\sin(\hat{\omega}t) \\ \hat{\omega} \end{bmatrix} + \begin{bmatrix} 0 & 0 & -V\sin(\hat{\omega}t)t \\ 0 & 0 & V\cos(\hat{\omega}t)t \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{\omega} \end{bmatrix} + \begin{bmatrix} w_1(t) \\ w_2(t) \\ w_3(t) \end{bmatrix}$$

NEXT A LINEARIZED OBSERVER WILL BE OBTAINED OF THE FORM:

$$\dot{\hat{X}}(t) = f(\hat{x},u,t) - G(t)\left(\frac{\partial g}{\partial x}\right)\Bigg|_{\hat{x}} \tilde{x} - G(t)\,V(t)$$

WHERE,

$$\left(\frac{\partial g}{\partial x}\right)\Bigg|_{\hat{x}} = \left.\begin{bmatrix} \frac{\partial g_1}{\partial x} & \frac{\partial g_1}{\partial y} & \frac{\partial g_1}{\partial w} \\ \frac{\partial g_2}{\partial x} & \frac{\partial g_2}{\partial y} & \frac{\partial g_2}{\partial w} \end{bmatrix}\right|_{\hat{x}}$$

$$\frac{\partial g_1}{\partial x} = \frac{1}{\cancel{2}}\left(x^2+y^2\right)^{-1/2}(\cancel{2})x\sin(wt)$$

$$\frac{\partial g_1}{\partial y} = \frac{1}{\cancel{2}}\left(x^2+y^2\right)^{-1/2}(\cancel{2})y\sin(wt)$$

$$\frac{\partial g_1}{\partial w} = \left(x^2+y^2\right)^{1/2}\cos(wt)\,t$$

$$\frac{\partial g_2}{\partial x} = \frac{1}{\cancel{2}}\left(x^2+y^2\right)^{-1/2}(\cancel{2})(x)\left(1-\cos(wt)\right)$$

$$\frac{\partial g_2}{\partial y} = \frac{1}{\cancel{2}}\left(x^2+y^2\right)^{-1/2}(\cancel{2})(y)\left(1-\cos(wt)\right)$$

$$\frac{\partial g_2}{\partial w} = \left(x^2+y^2\right)^{1/2}\sin(wt)\,t$$

THE <u>NON LINEAR</u> ESTIMATOR FOR ALL OF THE STATES INCLUDING w IS THEN GIVEN AS

$$\dot{\hat{X}}(t) = \begin{bmatrix} V\cos(\hat{\omega} t) \\ V\sin(\hat{\omega} t) \\ \hat{\omega} \end{bmatrix} - \quad \ldots$$

$$G(t)\begin{bmatrix} \dfrac{\hat{x}\sin(\hat{\omega}t)}{(\hat{x}^2+\hat{y}^2)^{1/2}} & \dfrac{\hat{y}\sin(\hat{\omega}t)}{(\hat{x}^2+\hat{y}^2)^{1/2}} & (\hat{x}^2+\hat{y}^2)^{1/2}\cos(\hat{\omega}t)t \\ \dfrac{\hat{x}(1-\cos(\hat{\omega}t))}{(\hat{x}^2+\hat{y}^2)^{1/2}} & \dfrac{\hat{y}(1-\cos(\hat{\omega}t))}{(\hat{x}^2+\hat{y}^2)^{1/2}} & (\hat{x}^2+\hat{y}^2)^{1/2}\sin(\hat{\omega}t)t \end{bmatrix}\tilde{X}(t) - \quad \ldots$$

$$G(t)\,V(t)$$

WHERE ~~A~~GAIN, $\tilde{X}(t)$ IS THE ERROR $\left(X(t) - \hat{X}(t)\right)$, AND THE SOLUTION FOR $X(t)$ COMES FROM THE SOLUTION FOR THE LINEARIZE SYSTEM DYNAMICS (DERIVED PREVIOUSLY). ALSO, $G(t)$ IS THE GAIN, WHICH IS CHOSEN TO BE THE KALMAN GAIN.

$$G(t) = -P_{\tilde{X}}(t)\,C^T(t)\,R_V^{-1}(t)$$

WHERE.
$$\dot{P}_{\tilde{X}}(t) = A(t)P_{\tilde{X}}(t) + P_{\tilde{X}}(t)A^T(t) - P_{\tilde{X}}(t)C^T(t)R_V^{-1}(t)C(t)P_{\tilde{X}}(t) + R_W(t)$$

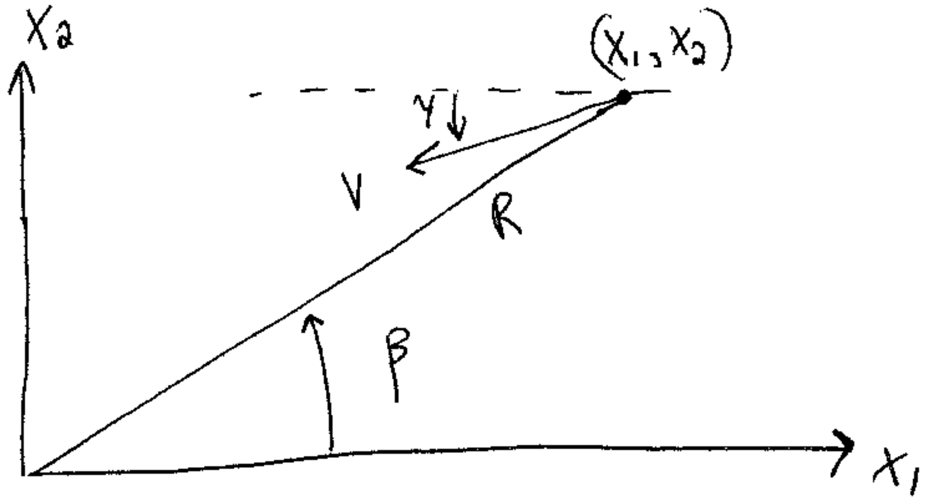
$$P_{\tilde{X}}(t_0) = P_{\tilde{X}0} \qquad // \text{ SOLVE USING COMPUTER!}$$

ESTIMATE FOR W WILL BE $\hat{\omega}$.

PROBLEM #3

$$\dot{X}_1 = -V\cos\gamma$$
$$\dot{X}_2 = -V\sin\gamma$$
$$Y = X_2$$

$$X_1 = R\cos\beta$$
$$X_2 = R\sin\beta$$



// PART A:

$$R = \sqrt{x_1^2 + x_2^2}$$

$$\frac{dR_2}{dt} = \frac{1}{\cancel{2}}(x_1^2 + x_2^2)^{-1/2}(\cancel{2})X_2\dot{X}_2 + \frac{1}{\cancel{2}}(x_1^2 + x_2^2)^{-1/2}\cancel{2}X_1\dot{X}_1$$

$$\frac{dR}{dt} = \frac{X_1\dot{X}_1 + X_2\dot{X}_2}{R}$$

$$\frac{dR}{dt} = \frac{-V\cancel{R}\cos\beta\cos\gamma - V\cancel{R}\sin\beta\sin\gamma}{\cancel{R}}$$

$$\frac{dR}{dt} = -V\cos(\beta - \gamma)$$

$$\beta = \tan^{-1}\left(\frac{X_2}{X_1}\right)$$

ASIDE:

$$\cos A\cos B + \sin A\sin B = \cos(A-B)$$

ASIDE:

$$\frac{d}{dt}\left(\tan^{-1}(x)\right) = \frac{1}{1+x^2}$$

PROB 3, pg #2

$$\frac{d\beta}{dt} = \left[\frac{1}{1+\left(\frac{x_2}{x_1}\right)^2}\right]\left[\frac{\dot{x}_2}{x_1} + x_2 x_1^{-2}(-1)\dot{x}_1\right]$$

$$= \left[\frac{1}{1+\left(\frac{x_2}{x_1}\right)^2}\right]\frac{\dot{x}_2 x_1 - \dot{x}_1 x_2}{x_1^2}$$

$$= \left[\frac{\cancel{x_1^2}}{x_1^2+x_2^2}\right]\frac{\dot{x}_2 x_1 - \dot{x}_1 x_2}{\cancel{x_1^2}} = \frac{-VR\sin\gamma\cos\beta + VR\cos\gamma\sin\beta}{R^2}$$

$$x_1^2 + x_2^2 \rightarrow R^2$$

$$\frac{d\beta}{dt} = \frac{+V\cancel{R}\sin(\beta-\gamma)}{R^{\cancel{2}}}$$

ASIDE:

$$\sin A\cos B - \sin B\cos A = \sin(A-B)$$

THE EQUATIONS OF MOTION ARE:

$$\boxed{\begin{aligned}\dot{R} &= -V\cos(\beta-\gamma)\\ \dot{\beta} &= \frac{V\sin(\beta-\gamma)}{R}\end{aligned}}$$

PART B:

$$\delta\dot{x}(t) = \begin{bmatrix}\delta\dot{R}(t)\\ \delta\dot{\beta}(t)\end{bmatrix} = \begin{bmatrix}\left.\frac{\partial\dot{R}(t)}{\partial R}\right|_0 & \left.\frac{\partial\dot{R}(t)}{\partial\beta}\right|_0\\ \left.\frac{\partial\dot{\beta}(t)}{\partial R}\right|_0 & \left.\frac{\partial\dot{\beta}(t)}{\partial\beta}\right|_0\end{bmatrix}\begin{bmatrix}\delta R\\ \delta\beta\end{bmatrix}$$

$$\left.\frac{\partial\dot{R}}{\partial R}\right|_0 = 0 \qquad \left.\frac{\partial\dot{R}}{\partial\beta}\right|_0 = V_0\sin(\beta_0-\gamma_0)$$

$$\left.\frac{\partial\dot{\beta}}{\partial R}\right|_0 = V_0\sin(\beta_0-\gamma_0)R_0^{-2}(-1) = -\frac{V_0\sin(\beta_0-\gamma_0)}{R_0^2}$$

$$\left.\frac{\partial \dot{\beta}}{\partial \beta}\right|_0 = \frac{V_0}{R_0}\cos(\beta_0 - \gamma_0)$$

THUS, THE LINEARIZED SYSTEM IS:

$$\boxed{\begin{bmatrix} \delta \dot{R}(t) \\ \delta \dot{\beta}(t) \end{bmatrix} = \begin{bmatrix} 0 & V_0 \sin(\beta_0 - \gamma_0) \\ \dfrac{-V_0 \sin(\beta_0 - \gamma_0)}{R_0^2} & \dfrac{V_0 \cos(\beta_0 - \gamma_0)}{R_0} \end{bmatrix} \begin{bmatrix} \delta R \\ \delta \beta \end{bmatrix} \qquad \delta y(t) = [\sin\beta_0 \quad R_0 \cos\beta_0] \begin{bmatrix} \delta R \\ \delta \beta \end{bmatrix}}$$

(4)

PART C:

FIRST LINEARIZE THE OUPUT EQUATION $y = x_2$ AROUND NOMINAL TRAJECTORY (MAY BE PART OF PART B)

$$\delta y(t) = \left[\left.\frac{\partial y}{\partial R}\right|_0 \quad \left.\frac{\partial y}{\partial \beta}\right|_0 \right] \begin{bmatrix} \delta R \\ \delta \beta \end{bmatrix} \qquad y = R \sin\beta$$

$$\left.\frac{\partial y}{\partial R}\right|_0 = \sin\beta_0 \qquad \left.\frac{\partial y}{\partial \beta}\right|_0 = R_0 \cos\beta_0$$

So, $\delta y(t) = [\sin\beta_0 \quad R_0 \cos\beta_0] \begin{bmatrix} \delta R \\ \delta \beta \end{bmatrix}$

NEXT DEFINE A STATE OBSERVER SYSTEM AS:

$$\dot{\hat{x}}(t) = A(t)\hat{x}(t) + B(t)u(t) + G(t)\big(\underbrace{C(t)\hat{x}(t)}_{\text{ESTIMATE}} - \underbrace{y(t)}_{\text{ACTUAL}}\big)$$

$$\hat{x}(t_0) = \hat{x}_0$$

THEN THE ESTIMATION ERROR CAN BE DEFINED AS:

$$\tilde{X}(t) = X(t) - \hat{X}(t)$$

GOVERNED BY THE DYNAMIC EQUATION:

$$G = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$$

$$\dot{\tilde{X}}(t) = \big(A(t) + G(t)\,C(t)\big)\,\tilde{X}(t)$$

IN THIS PROBLEM, THE ESTIMATION ERROR EQUATIONS ARE:

$$\boxed{\dot{\tilde{X}}(t) = \left[\underbrace{\begin{bmatrix} 0 & V_0 \sin(\beta_0 - \gamma_0) \\ -\dfrac{V_0 \sin(\beta_0 - \gamma_0)}{R_0^2} & \dfrac{V_0 \cos(\beta_0 - \gamma_0)}{R_0} \end{bmatrix}}_{A} + \underbrace{\begin{bmatrix} g_1 \\ g_2 \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} \sin\beta_0 & R_0 \cos\beta_0 \end{bmatrix}}_{C} \right] \tilde{X}(t)}$$

THE GAINS OF THE G MATRIX SHOULD BE SELECTED SUCH THAT THE EIGENVALUES λ_i OF THE $(A + GC)$ MATRIX HAVE STRICTLY NEGATIVE REAL PARTS,

$Re(\lambda_i) < 0 \quad \forall i = 1, 2$; THIS WILL MAKE THE ESTIMATION ERROR BE ASYMPTOTICALLY STABLE.

PROBLEM 4, PART A

First, for a sanity check, in Fig. 0.1-0.2, the position of the MAV using the Vicon data is plotted against the position of the MAV calculated using the data given (assumed) for the GV, the quaternion data transformed to a rotation matrix and the given data for eB. It can be seen that there is an offset for the two trajectories in the x direction. [1]

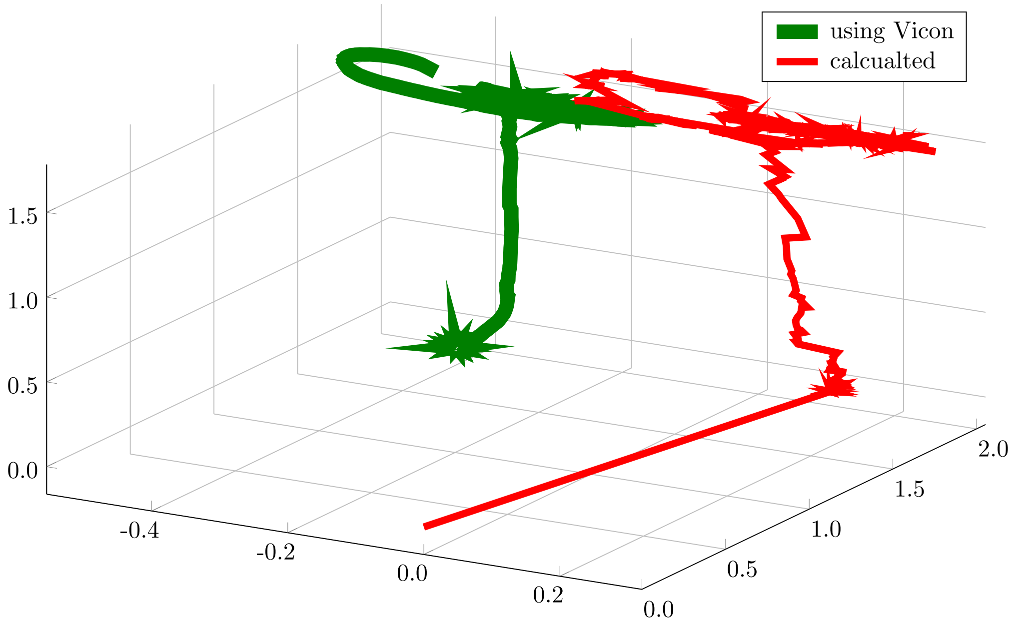


Figure 0.1

Next, after formulating a Kalman Filter for the closed loop system, the resulting trajectories for eB are compared to the given data for eB in Fig. **??**. It can be seen that the trajectory determined using the Kalman Filter is offset from the data that was given for eB for both the x and the y, but the z matches fairly closely.

Finally, the trajectory for xW is plotted for both the Vicon system as well as the trajectory transformed from the eB trajectory determined using the Kalman Filter (shown in Fig. **??**) to the W frame is shown in Fig. 0.4. Again there is an offset for both the x and the y, but the z matches fairly closely.

[1] For posterity, the code that produced the results for Problem 4 is included in the appendix

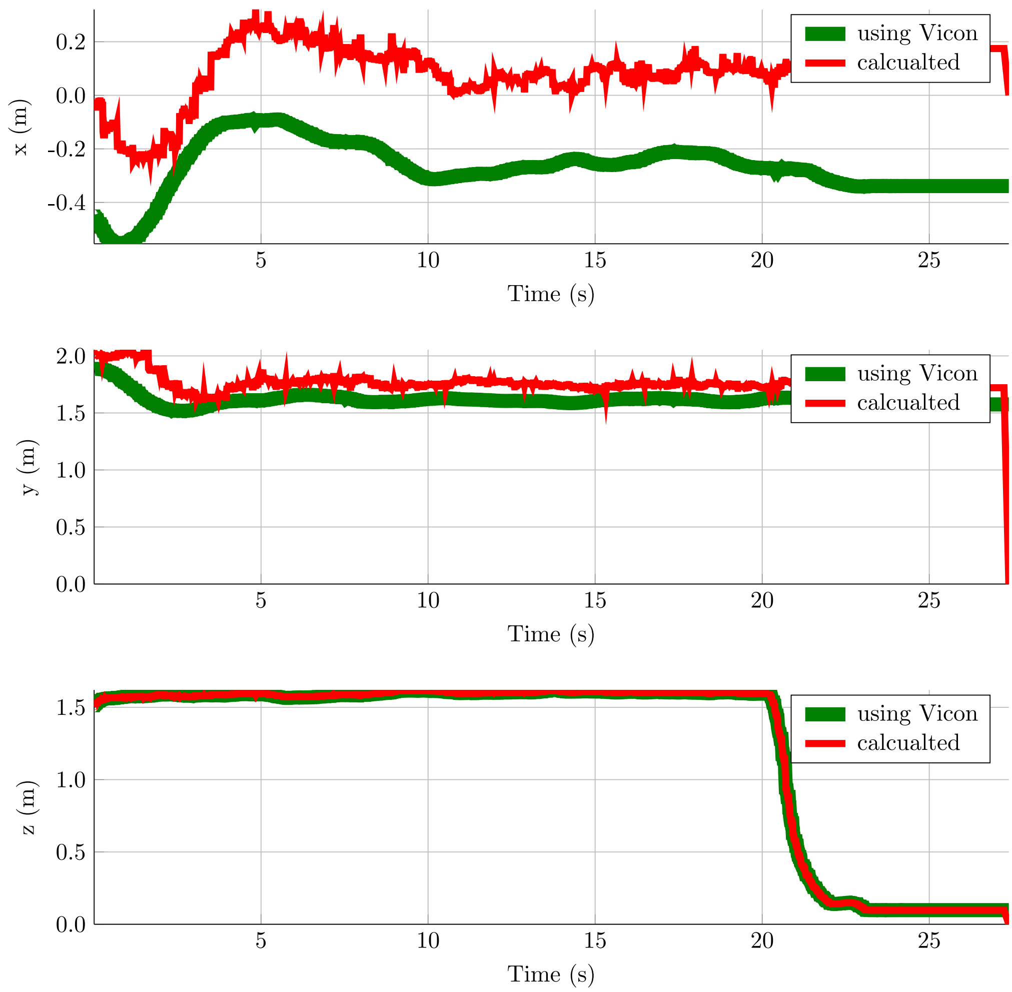

x (m)
0.2
0.0
-0.2
-0.4
using Vicon
calcualted
Time (s)
y (m)
2.0
1.5
1.0
0.5
0.0
z (m)
5
10
15
20
25

Figure 0.2

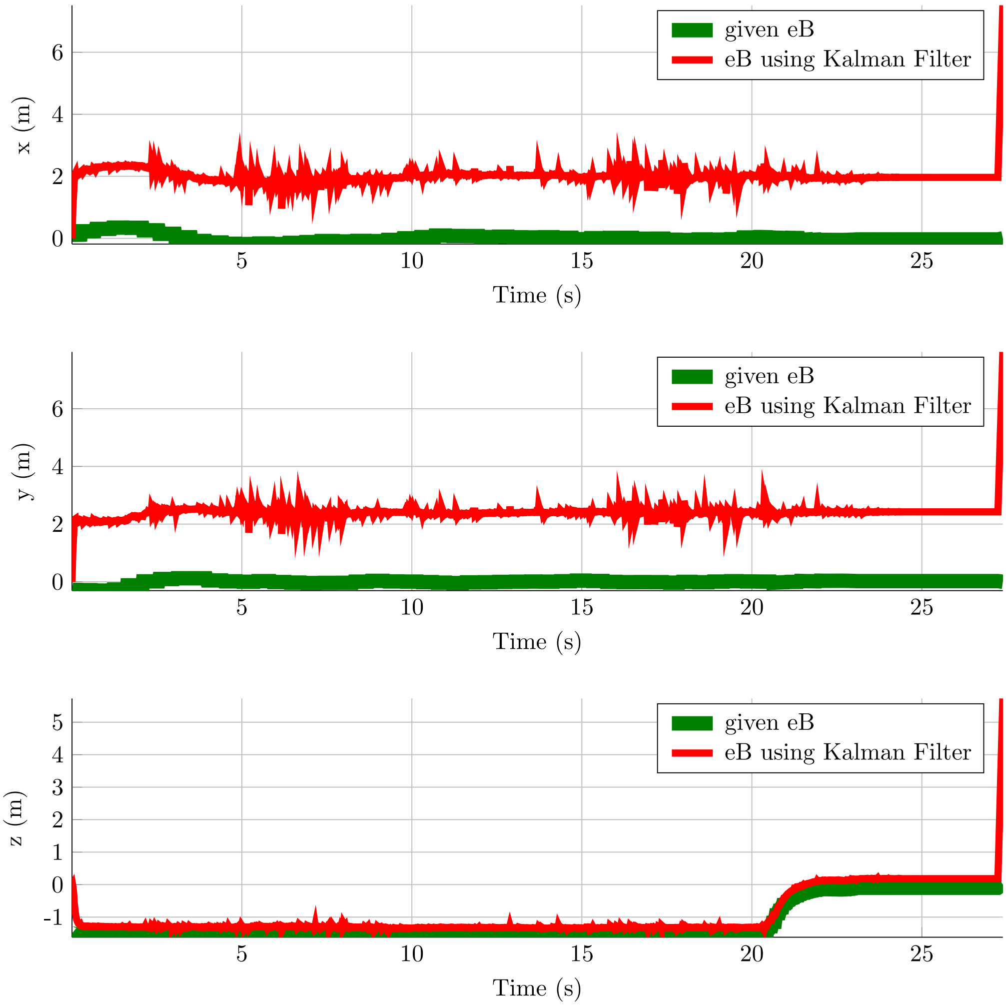
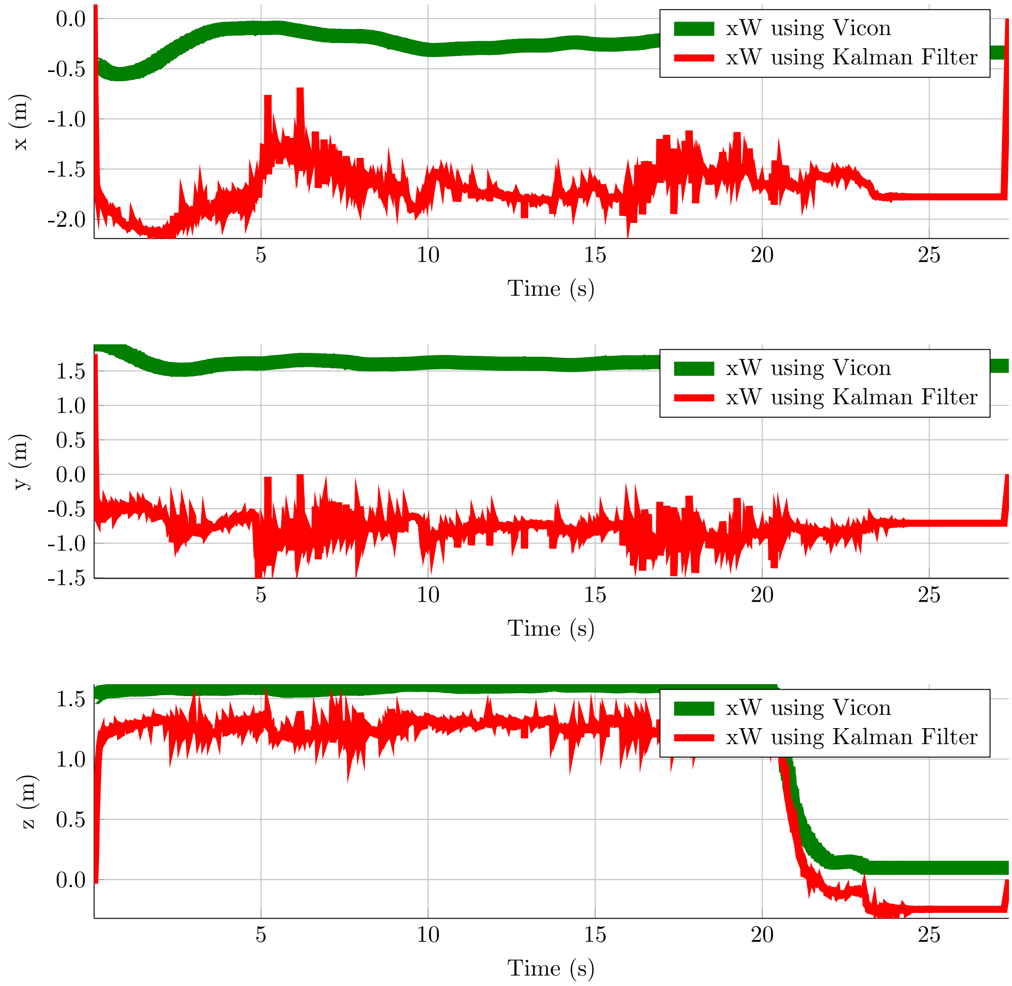


Figure 0.3



0.0
-0.5
-1.0
-1.5
-2.0
x (m)
5
10
15
20
25
Time (s)
xW using Vicon
xW using Kalman Filter
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
y (m)
5
10
15
20
25
Time (s)
xW using Vicon
xW using Kalman Filter
1.5
1.0
0.5
0.0
z (m)
5
10
15
20
25
Time (s)
xW using Vicon
xW using Kalman Filter

Figure 0.4

PROBLEM 4, PART B

For this set of results, the covariance matrix for the noise is increased as mentioned in the problem statement. All of the same results where collected and are shown in Figs. 0.4-0.5. Again, we see an offset for both the x and the y, but now the z has an offset as well. When compared to the results in Part a, there is a degradation in the estimate that the Kalman Filter is performing, this is especially evident when considering that there is now an offsest in z as well and that the offsets are larger for x and y as well.

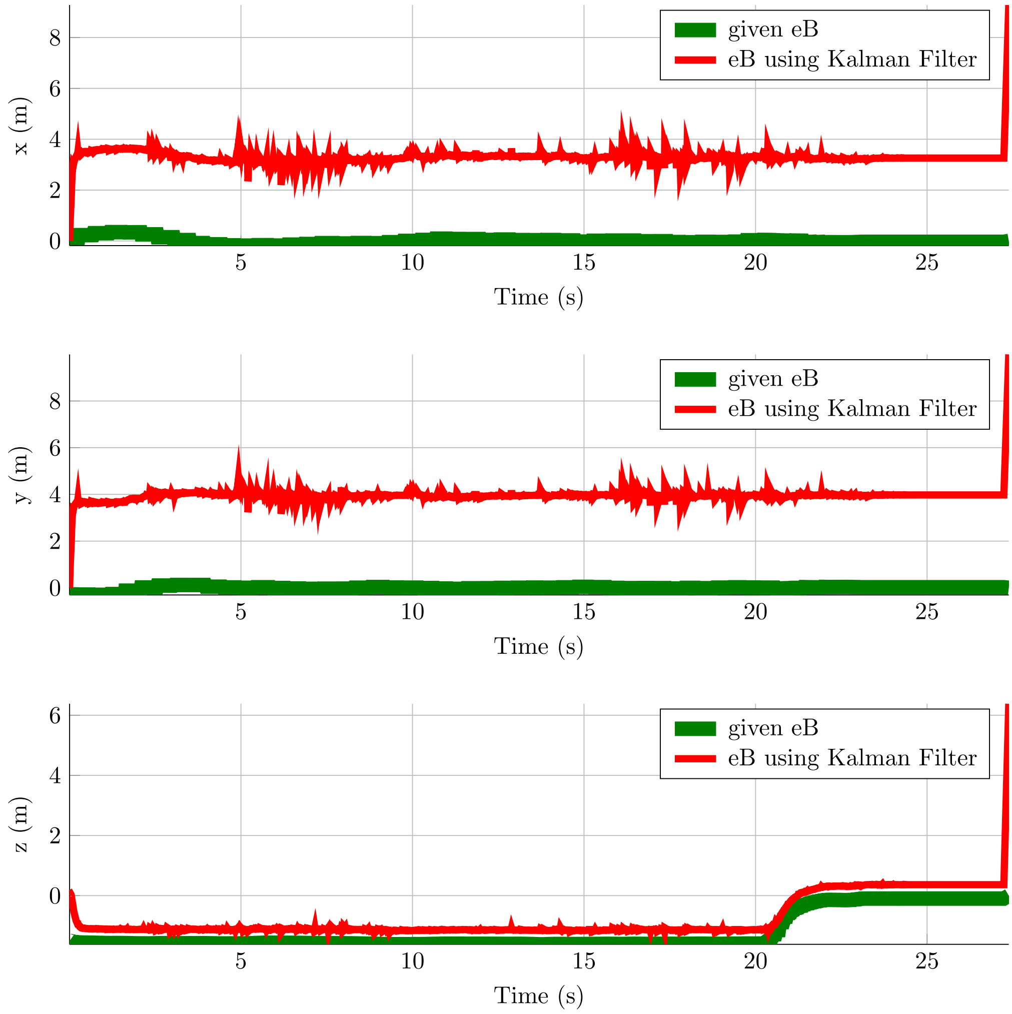


Figure 0.5

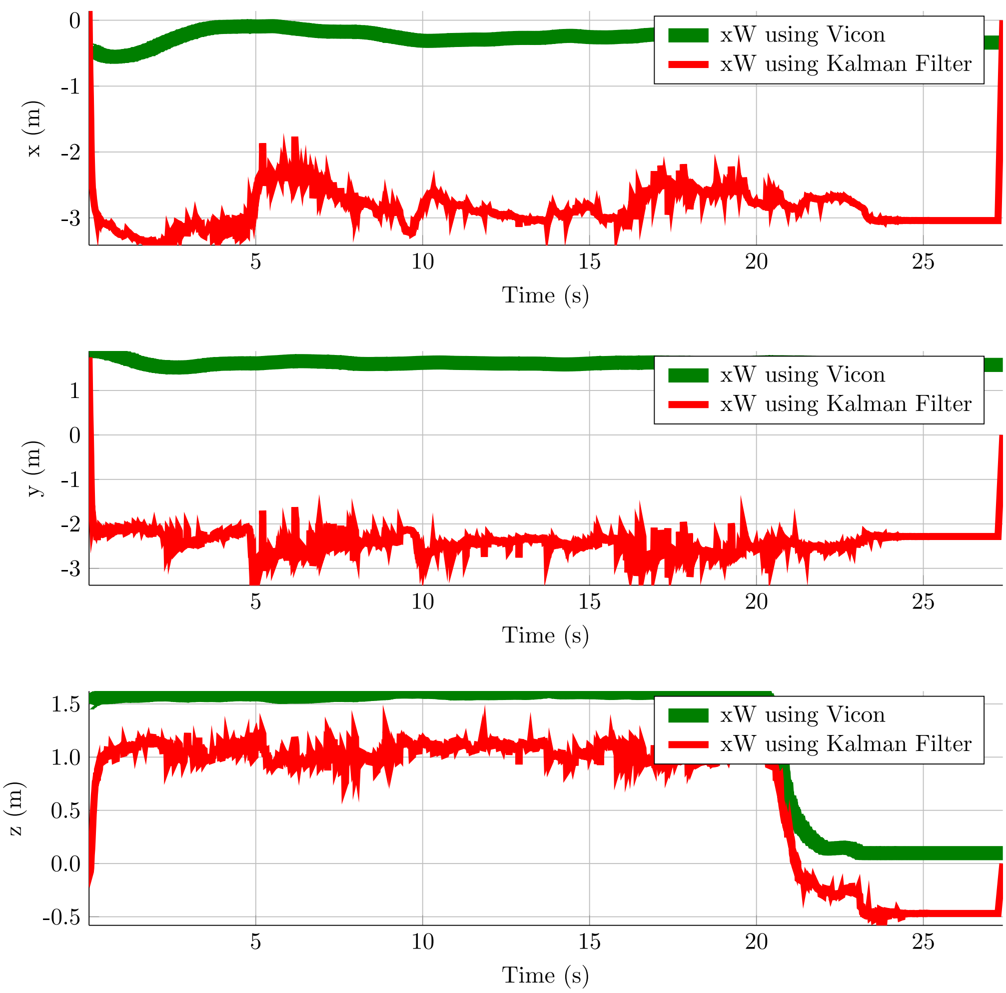


Figure 0.6

PROBLEM 5, PART A

In Fig. 0.7 it can be seen that Iron Man is able to hit the target.

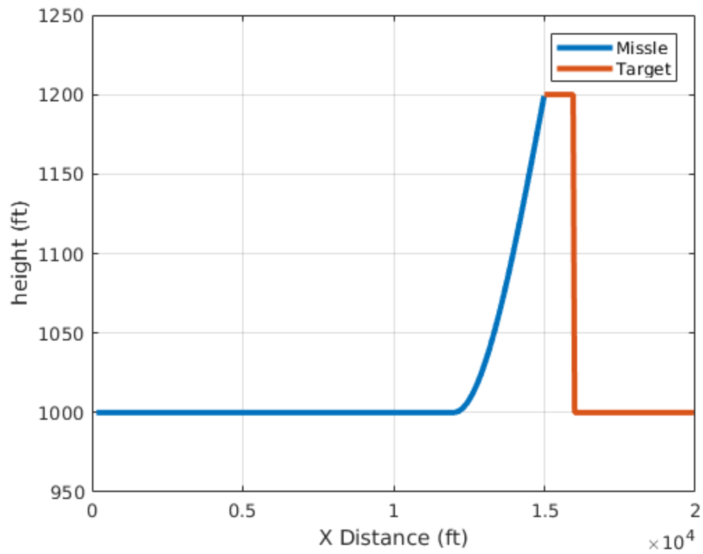


Figure 0.7: Position tracjectories of Iron Man and his target, when Time-to-Go = 1s

PROBLEM 5, PART B

In Fig. 0.8 it can be seen that the acceleration is very large for Iron Man to perform the maneuver in Fig. 0.7.

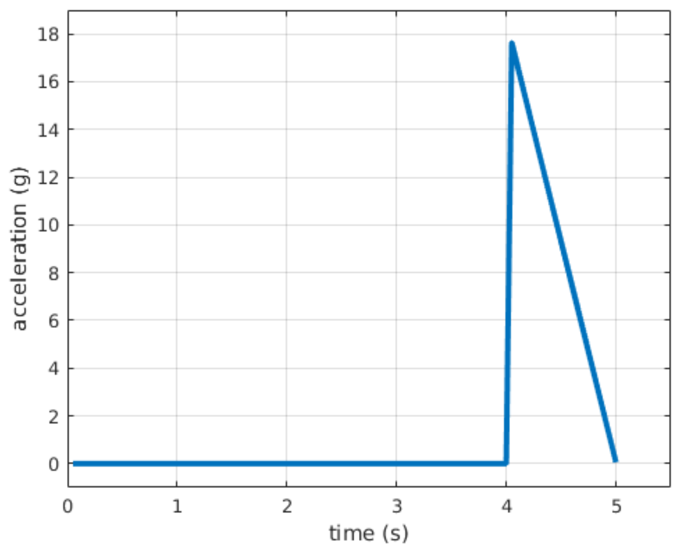


Figure 0.8: Acceleration required by Iron Man so the he catches the destructive square

PROBLEM 5, PART C

It can be seen in Fig. 0.10 that as the Time-To-Go is increased the acceleration required by Iron Man to hit the target is decreased. This can also be seen in Fig. 0.11, where Iron Man is allowed to start moving at the start of the simulation.

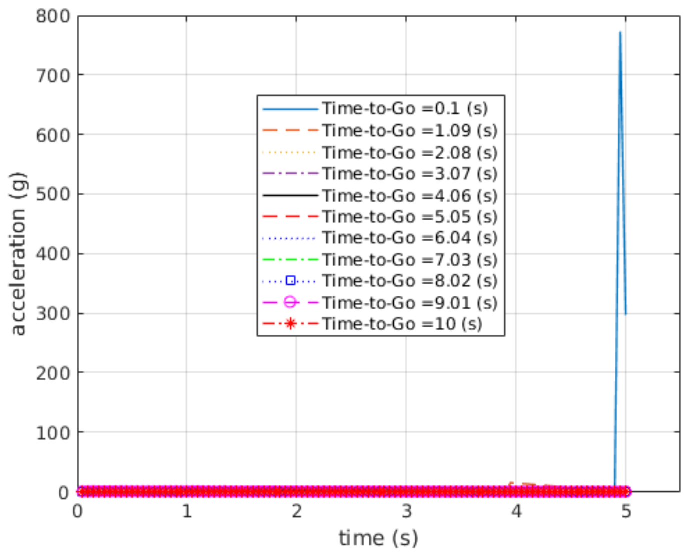


Figure 0.9: Acceleration required by Iron Man for various Time-to-Go's form $0.1sec \rightarrow 10sec$

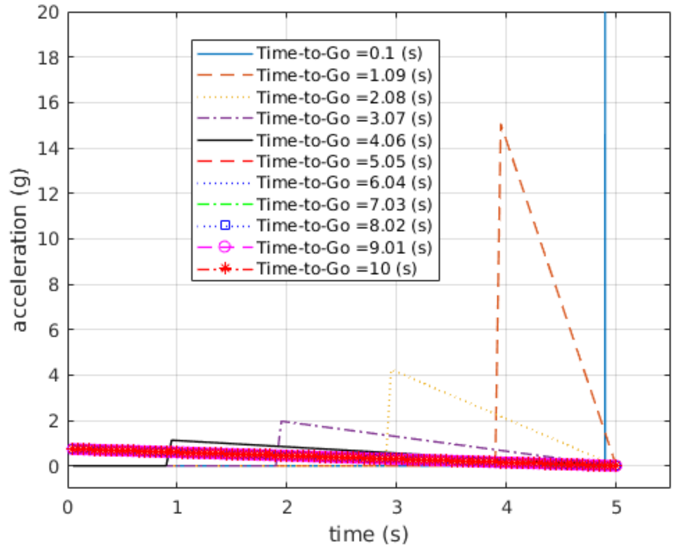


Figure 0.10: Zoomed in plot of acceleration required by Iron Man for various Time-to-Go's from $0.1 sec \rightarrow 10 sec$

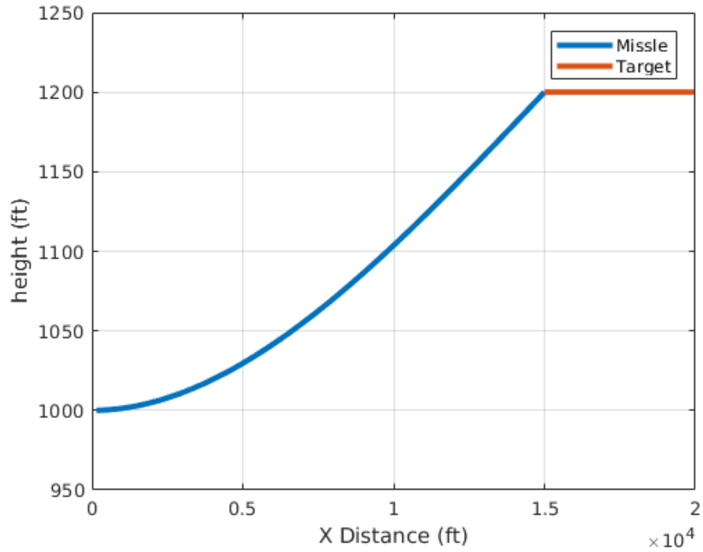


Figure 0.11: Just for fun, a look at the position plot when Time-to-go = $10 sec$

PROBLEM 5, PART A.2

In Fig. 0.12, when Iron Man cannot instantaneously track the target and there is a first-order dynamics with $T = 1$. In this case, it can be seen that Iron Man misses the target.

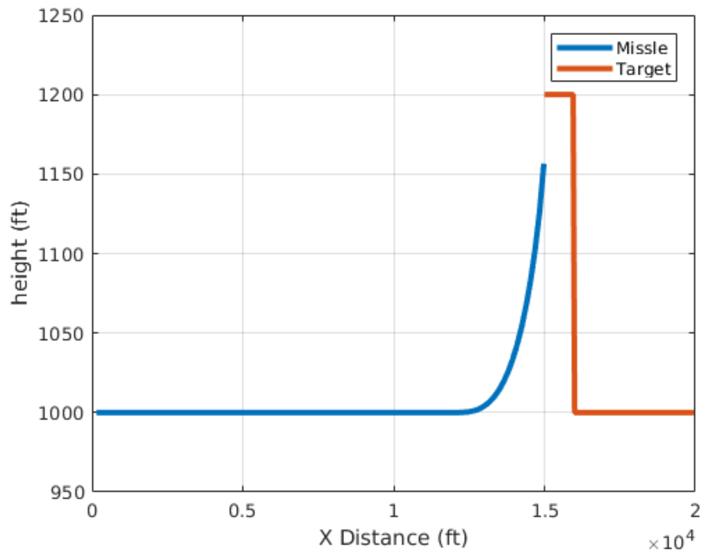


Figure 0.12: Position tracjectories of Iron Man and his target, when Time-to-Go = $1s$

PROBLEM 5, PART B.2

In Fig. 0.13 the miss distance for the Time-to-Go = 1s [2] is shown by the red dot. It can be seen that that the miss distance is $= 38.94(ft)$.

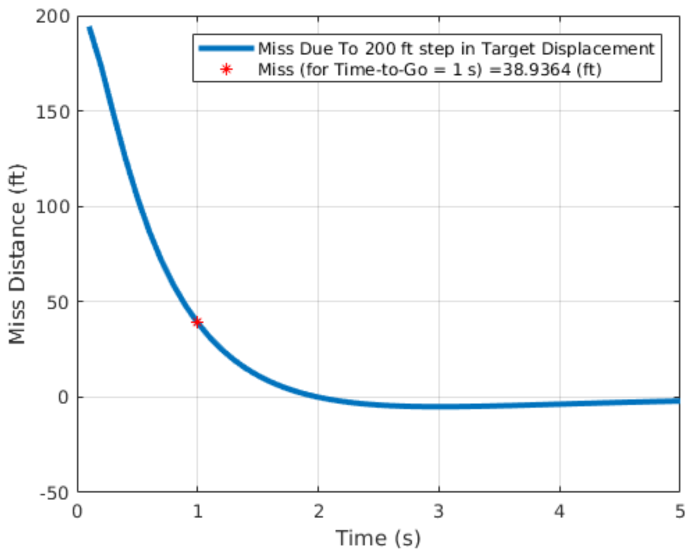


Figure 0.13: Position tracjectories of Iron Man and his target, when Time-to-Go = $1s$

[2]assuming that the Time-To-Go is still = $1s$, as given in the original problem statement

PROBLEM 4, JULIA CODE

```
using Plots
pgfplots()
using LaTeXStrings
PGFPlots.pushPGFPlotsPreamble("\\usepackage{amssymb}")
using Interpolations
using OrdinaryDiffEq
using DiffEqBase
using DataFrames

d = readtable("data.csv")

# extract data
t = d[:t]; # seconds
XxV = d[:XxV]
XyV = d[:XyV]
XzV = d[:XzV]
qx = d[:qx]
qy = d[:qy]
qz = d[:qz]
qw = d[:qw]
ZxW = d[:ZxW]
ZyW = d[:ZyW]
ZzW = d[:ZzW]
ExB = d[:ExB]
EyB = d[:EyB]
EzB = d[:EzB]
eta = d[:eta]
E1I = d[:E1I]
E2I = d[:E2I]
E3I = ones(length(E2I))

lambda = abs(EzB)

# misc variables
L = length(t)
l1 = (6,:green,:solid)
l2 = (3,:red,:solid)
l3 = (2.2,:black,:dash)
l4 = (1.5,:blue,:dot)

# rotation matrix, https://en.wikipedia.org/wiki/Conversion_between_quaternions_and_Eul
```

```
function ROT(q_x,q_y,q_z,q_w)
   [1 - 2*(q_y^2 + q_z^2)        2*(q_x*q_y - q_w*q_z)      2*(q_w*q_y + q_x*q_z);
    2*(q_x*q_y + q_w*q_z)        1-2*(q_x^2 + q_z^2)        2*(q_z*q_y - q_w*q_x);
    2*(q_x*q_z - q_w*q_y)        2*(q_z*q_y + q_w*q_x)      1-2*(q_y^2 + q_x^2)];
end

# direction cosine matrices, 3-2-1 sequence (psi,theta,phi)
# [x,y,z] = Rz(psi)*Ry(theta)*Rz(phi)[X;Y;Z]
function Rz(psi)
 [cos(psi) -sin(psi) 0;
  sin(psi)  cos(psi) 0;
     0         0     1]
end

function Ry(theta)
 [cos(theta)      0    sin(theta);
      0           1        0;
  -sin(theta)     0     cos(theta)]
end

function Rx(phi)
 [1         0         0;
  0       cos(phi) -sin(phi);
  0       sin(phi)  cos(phi)]
end

##################
# part a)
K_intr = [1030.597415177913            0             361.451236121491;
                 0             1030.358516382353   246.1238464630347;
                 0                     0                    1]
C_intr = [1   0   0;
          0  -1   0;
          0   0  -1]

KK = [  0.5    0      0;
        0     0.6     0;
        0      0    0.07]

rotM = zeros(L,3,3)
eB = zeros(L,3)
eBM = zeros(L,3)
xW = zeros(L,3)
```

```
for i in 1:L−1
 # calculate rotation matrix using quaternion data from Vicon system
 rotM[i,:,:] = ROT(qx[i],qy[i],qz[i],qw[i])
 # put eB into a matrix
 eB[i,:] = [ExB[i];EyB[i];EzB[i]]
 # calculate eB from the measured data
 eI = [E1I[i];E2I[i];E3I[i]]
 eBM[i,:] = inv(C_intr)*inv(KK)*lambda[i]*eI
 # estimate xW from vectors
 xW[i,:] = [−ZxW[i];ZyW[i];ZzW[i]] − rotM[i,:,:]*eB[i,:,:]
end
wB = zeros(L,3,3)
for i in 1:L−1
 dt = d[:t][i+1] − d[:t][i]
 wB[i,:,:] = (rotM[i+1,:,:] − rotM[i,:,:])/dt
end
# xW) plots
s1 = "using Vicon"
s2 = "calcualted "

plot(XxV,XyV,XzV,line=l1,label=s1)
plot!(xW[:,1],xW[:,2],xW[:,3],line=l2,label=s2,cbar=false)
savefig(string("figs/p4a",".",:svg));

# position
p1 = plot(t,XxV,line=l1,label=s1)
plot!(t,xW[:,1],line=l2,label=s2)
yaxis!("x (m)")
xaxis!("Time (s)")

p2 = plot(t,XyV,line=l1,label=s1)
plot!(t,xW[:,2],line=l2,label=s2)
yaxis!("y (m)")
xaxis!("Time (s)")

p3 = plot(t,XzV,line=l1,label=s1)
plot!(t,xW[:,3],line=l2,label=s2)
yaxis!("z (m)")
xaxis!("Time (s)")
plot(p1,p2,p3,layout=@layout([a;b;c]),size=[600,600])
savefig(string("figs/p4b",".",:svg));

Rw = zeros(6,6)
Rw[1:3,1:3] = [0.1    0    0;
```

```
                 0    0.1    0;
                 0      0   0.1]
Rv= [2  0  0;
     0  2  0;
     0  0  2]
x_k = zeros(L); x_k[1] = 0;
y_k = zeros(L); y_k[1] = 0;
z_k = zeros(L); z_k[1] = 0;
vx_k = zeros(L); vx_k[1] = 0;
vy_k = zeros(L); vy_k[1] = 0;
vz_k = zeros(L); vz_k[1] = 0;
x = zeros(L,6); x[1,1:6] = [0,0,0,0,0,0];
P = zeros(L,6,6)
for i in 1:L-1
 dt = d[:t][i+1] - d[:t][i]
 A = [1   0   0   dt  0    0;
      0   1   0   0   dt   0;
      0   0   1   0   0   dt;
      0   0   0   1   0    0;
      0   0   0   0   1    0;
      0   0   0   0   0    1]

 B = [1    0   0;
      0    1   0;
      0    0   1;
      0    0   0;
      0    0   0;
      0    0   0]

 C = [1  0   0  dt  0    0;
      0  1   0  0   dt   0;
      0  0   1  0   0   dt]

 # time update (predict)
 u = (-wB[i,:,:] + KK)
 x[i+1,:] = A*x[i,:] + B*[u[1,1];u[2,2];u[3,3]]
 P[i+1,:,:] = A*P[i,:,:]*A' + Rw

 # measurment update
 eI = [E1I[i];E2I[i];E3I[i]]
 y = inv(C_intr)*inv(K_intr)*lambda[i]*eI
 K = P[i+1,:,:]*C'*inv((C*P[i+1,:,:]*C' + Rv))
 x[i+1,:] = x[i+1,:] + K*(y-C*x[i+1,:])
 P[i+1,:,:] = (eye(6) - K*C)*P[i+1,:,:]
```

```
 # save results
 x_k[i+1] = x[i+1,1]
 vx_k[i+1] = x[i+1,4]

 y_k[i+1] = x[i+1,2]
 vy_k[i+1] = x[i+1,5]

 z_k[i+1] = x[i+1,3]
 vz_k[i+1] = x[i+1,6]
end

# part a) plot eB
s1 = "given eB"
s2 = "eB using Kalman Filter"

# position
p1 = plot(t,eB[:,1],line=l1,label=s1)
plot!(t,x_k,line=l2,label=s2)
yaxis!("x (m)")
xaxis!("Time (s)")

p2 = plot(t,eB[:,2],line=l1,label=s1)
plot!(t,y_k,line=l2,label=s2)
yaxis!("y (m)")
xaxis!("Time (s)")

p3 = plot(t,eB[:,3],line=l1,label=s1)
plot!(t,z_k,line=l2,label=s2)
yaxis!("z (m)")
xaxis!("Time (s)")
plot(p1,p2,p3,layout=@layout([a;b;c]),size=[600,600])
savefig(string("figs/p4c",".",:svg));

plot(eB[:,1],eB[:,2],eB[:,3],line=l1,label=s1,cbar=false)
plot!(x_k,y_k,z_k,line=l2,label=s2,cbar=false,size=[800,800])
xaxis!("x (m)")
yaxis!("y (m)")
savefig(string("figs/p4d",".",:svg));

# xWK)
xWK = zeros(L,3)
for i in 1:L-1
 # estimate xWK from vectors
```

```
 xWK[i,:] = [−ZxW[i];ZyW[i];ZzW[i]] − rotM[i,:,:]*[x_k[i];y_k[i];z_k[i]]
end

s1 = "xW using Vicon"
s2 = "xW using Kalman Filter"

# position
p1 = plot(t,XxV,line=l1,label=s1)
plot!(t,xWK[:,1],line=l2,label=s2)
yaxis!("x (m)")
xaxis!("Time (s)")

p2 = plot(t,XyV,line=l1,label=s1)
plot!(t,xWK[:,2],line=l2,label=s2)
yaxis!("y (m)")
xaxis!("Time (s)")

p3 = plot(t,XzV,line=l1,label=s1)
plot!(t,xWK[:,3],line=l2,label=s2)
yaxis!("z (m)")
xaxis!("Time (s)")
plot(p1,p2,p3,layout=@layout([a;b;c]),size=[600,600])
savefig(string("figs/p4e",".",:svg));

plot(XxV,XyV,XzV,line=l1,label=s1,cbar=false)
plot!(xWK[:,1],xWK[:,2],xWK[:,3],line=l2,label=s2,cbar=false,size=[800,800])
xaxis!("x (m)")
yaxis!("y (m)")
savefig(string("figs/p4f",".",:svg));
```