HW - 3 Submission

AE-6505 - Kalman Filtering

Kadhir Umasankar

Spring 2022

03-02-22 - HW 3

Wednesday, March 2, 2022 4:38 PM

4.5. a. Using Eq 4.27

$$Q_{k-1} = \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)} Q_c(\tau) e^{A^T(t_k-\tau)} d\tau$$

$$A = \begin{bmatrix} -F_0/2V_0 & 0 \\ 0 & -F_0/V_0 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 1 \\ \frac{c_1-c_0}{V_0} & \frac{c_2-c_0}{V_0} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$

$$Q_c = B \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} B^T = \begin{bmatrix} 2 & 3 \\ 3 & 5 \end{bmatrix}$$

We know that matrix exponential e^M of a diagonal matrix $M = \begin{bmatrix} m_{11} & 0 \\ 0 & m_{22} \end{bmatrix}$ is of the form $e^M = \begin{bmatrix} e^{m_{11}} & 0 \\ 0 & e^{m_{22}} \end{bmatrix}$

$$\therefore e^{A(t_k-\tau)} = \begin{bmatrix} e^{\tau-t_k} & 0 \\ 0 & e^{2\tau-2t_k} \end{bmatrix} = e^{A^T(t_k-\tau)}$$

$$e^{A(t_k-\tau)} Q_c(\tau) e^{A^T(t_k-\tau)} = \begin{bmatrix} e^{\tau-t_k} & 0 \\ 0 & e^{2\tau-2t_k} \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 3 & 5 \end{bmatrix} \begin{bmatrix} e^{\tau-t_k} & 0 \\ 0 & e^{2\tau-2t_k} \end{bmatrix}$$

$$= \begin{bmatrix} 2e^{\tau-t_k} & 3e^{\tau-t_k} \\ 3e^{2\tau-2t_k} & 5e^{2\tau-2t_k} \end{bmatrix} \begin{bmatrix} e^{\tau-t_k} & 0 \\ 0 & e^{2\tau-2t_k} \end{bmatrix}$$

$$= \begin{bmatrix} 2e^{2\tau-2t_k} & 3e^{3\tau-3t_k} \\ 3e^{3\tau-3t_k} & 5e^{4\tau-4t_k} \end{bmatrix}$$

$$\therefore Q_{k-1} = \int_{t_{k-1}}^{t_k} \begin{bmatrix} 2e^{2\tau-2t_k} & 3e^{3\tau-3t_k} \\ 3e^{3\tau-3t_k} & 5e^{4\tau-4t_k} \end{bmatrix} d\tau = \begin{bmatrix} 1-e^{-2t_k+2t_{k-1}} & 1-e^{-3t_k+3t_{k-1}} \\ 1-e^{-3t_k+3t_{k-1}} & \frac{5}{4}\left(1-e^{4t_k-4t_{k-1}}\right) \end{bmatrix}$$

b. Using Eq 4.28

$$Q_{k-1} = Q_c(t_k)\Delta t = \begin{bmatrix} 2 & 3 \\ 3 & 5 \end{bmatrix}(t_k - t_{k-1}) = \begin{bmatrix} 2(t_k - t_{k-1}) & 3(t_k - t_{k-1}) \\ 3(t_k - t_{k-1}) & 5(t_k - t_{k-1}) \end{bmatrix}$$

c. We can expand the solution of (a) using a Taylor series as follows:

$$Q_{k-1} = \begin{bmatrix} 1-(1-2\Delta t + (2\Delta t)^2/2! + (2\Delta t)^3/3! + \dots) & 1-(1-3\Delta t + (3\Delta t)^2/2! + (3\Delta t)^3/3! + \dots) \\ 1-(1-3\Delta t + (3\Delta t)^2/2! + (3\Delta t)^3/3! + \dots) & \frac{5}{4} - \frac{5}{4}(1-4\Delta t + (4\Delta t)^2/2! + \dots) \end{bmatrix}$$

where $\Delta t = t_k - t_{k-1}$

From this representation, we can see that for small Δt higher order terms can be ignored, making our result approximately equal to the result from part (b).

$\therefore$ Answer to part (a) $\approx$ Answer to part (b)

4.8. $F = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix}$

$Q = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$

$i=0 \Rightarrow F^0 Q (F^T)^0 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$

$i=1 \Rightarrow \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} = \begin{bmatrix} 1/4 & 1/4 \\ 0 & 1/4 \end{bmatrix}$

$i=2 \Rightarrow \begin{bmatrix} 1/2^2 & 0 \\ 0 & 1/2^2 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1/2^2 & 0 \\ 0 & 1/2^2 \end{bmatrix} = \begin{bmatrix} 1/16 & 1/16 \\ 0 & 1/16 \end{bmatrix}$

$i=3 \Rightarrow \begin{bmatrix} 1/2^3 & 0 \\ 0 & 1/2^3 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1/2^3 & 0 \\ 0 & 1/2^3 \end{bmatrix} = \begin{bmatrix} 1/64 & 1/64 \\ 0 & 1/64 \end{bmatrix}$

$i=4 \Rightarrow \begin{bmatrix} 1/2^4 & 0 \\ 0 & 1/2^4 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1/2^4 & 0 \\ 0 & 1/2^4 \end{bmatrix} = \begin{bmatrix} 1/256 & 1/256 \\ 0 & 1/256 \end{bmatrix}$

$i=5 \Rightarrow \begin{bmatrix} 1/2^5 & 0 \\ 0 & 1/2^5 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1/2^5 & 0 \\ 0 & 1/2^5 \end{bmatrix} = \begin{bmatrix} 1/1024 & 1/1024 \\ 0 & 1/1024 \end{bmatrix}$

$i=6 \Rightarrow \begin{bmatrix} 1/2^6 & 0 \\ \dots \end{bmatrix}\begin{bmatrix} 1 & 1 \\ \dots \end{bmatrix}\begin{bmatrix} 1/2^6 & 0 \\ \dots \end{bmatrix} = \begin{bmatrix} 1/4096 & 1/4096 \\ \dots \end{bmatrix}$

$$i=6 \Rightarrow \begin{bmatrix} 1/2^6 & 0 \\ 0 & 1/2^6 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1/2^6 & 0 \\ 0 & 1/2^6 \end{bmatrix} = \begin{bmatrix} 1/4096 & 1/4096 \\ 0 & 1/4096 \end{bmatrix}$$

$$i=7 \Rightarrow \begin{bmatrix} 1/2^7 & 0 \\ 0 & 1/2^7 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1/2^7 & 0 \\ 0 & 1/2^7 \end{bmatrix} = \begin{bmatrix} 1/16384 & 1/16384 \\ 0 & 1/16384 \end{bmatrix}$$

$$\vdots$$

P begins to converge by $i=7$.

Adding the matrices, we find that

$$P = \sum_{i=0}^{\infty} F^i Q (F^T)^i \approx \underline{\underline{\begin{bmatrix} 1.3333 & 0 \\ 0 & 1.3333 \end{bmatrix}}}$$

4.11. Using $\bar{x}_0 = 1$

$P_0 = 2$

$f = -0.5$

$q_c = 1$

$\Delta t = 0.01s$

$t_f = 5s$

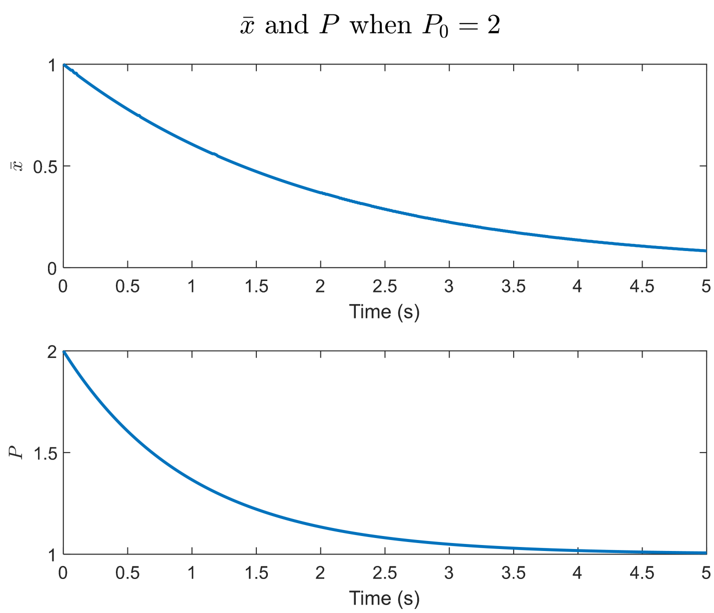
and using Eqn 4.32

$\bar{x}_k = \exp(kf\Delta t)\,\bar{x}_0$

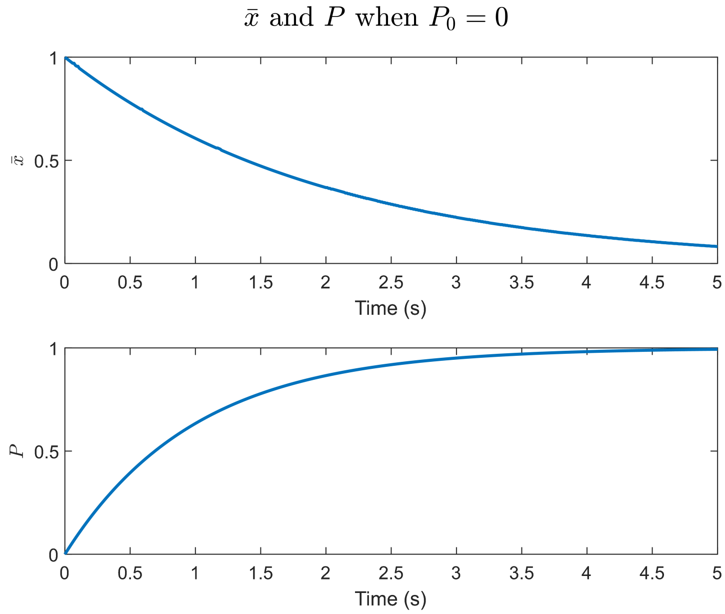
and Eqn 4.33

$P_k = (2fP_{k-1} + q_c)\Delta t + P_{k-1}$

the following plots were found for $\bar{x}$ and P:



With $P_0 = 0$ the following plots were found:



In both plots, the steady-state value of the variance appears to be 1, which is consistent with the result of the analytically determined steady-state variance using the equation from the text:

$$P_{k-1} = -\frac{q_c}{2f} = -\frac{1}{2 * -0.5} = 1$$

4.13.a. Using Eq 4.49

$$\dot{P} = AP + PA^T + Q_c$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1/C \\ -1/L & -R/L \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/L \end{bmatrix} u$$

Using $R=3$, $L=1$, $C=0.5$

$$A = \begin{bmatrix} 0 & 2 \\ -1 & -3 \end{bmatrix}$$

$$Q_c = B q_c B^T = \begin{bmatrix} 0 \\ 1 \end{bmatrix} (1) \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$AP + PA^T + Q_c = 0$$

$$\begin{bmatrix} 0 & 2 \\ -1 & -3 \end{bmatrix} P + P \begin{bmatrix} 0 & -1 \\ 2 & -3 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}$$

*Solved using MATLAB

$$P = \begin{bmatrix} 1/3 & 0 \\ 0 & 1/6 \end{bmatrix}$$

b. $A = \begin{bmatrix} 0 & 2 \\ -1 & -3 \end{bmatrix}$

$$\det(A - \lambda I) = \det\begin{bmatrix} -\lambda & 2 \\ -1 & -3-\lambda \end{bmatrix}$$

$$= (-\lambda)(-3-\lambda) + 2$$

$$= \lambda^2 + 3\lambda + 2$$

$$= (\lambda+1)(\lambda+2)$$

$$\ker(A + I) = \ker\begin{bmatrix} 1 & 2 \\ -1 & -2 \end{bmatrix} = \begin{bmatrix} -0.8944 \\ 0.4472 \end{bmatrix}$$

$$\ker(A + 2I) = \ker\begin{bmatrix} 2 & 2 \\ -1 & -1 \end{bmatrix} = \begin{bmatrix} -0.7071 \\ 0.7071 \end{bmatrix}$$

$$\therefore S = \begin{bmatrix} -0.8944 & -0.7071 \\ 0.4472 & 0.7071 \end{bmatrix}$$

$$\hat{A} = S^{-1} A S = \begin{bmatrix} -0.8944 & -0.7071 \\ 0.4472 & 0.7071 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 2 \\ -1 & -3 \end{bmatrix} \begin{bmatrix} -0.8944 & -0.7071 \\ 0.4472 & 0.7071 \end{bmatrix}$$

$$= \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}$$

Using Eq 1.71

$$e^{AT} = Q e^{\hat{A}T} Q^{-1}$$

$$= \begin{bmatrix} -0.8944 & -0.7071 \\ 0.4472 & -0.7071 \end{bmatrix} \begin{bmatrix} e^{-T} & 0 \\ 0 & e^{-2T} \end{bmatrix} \begin{bmatrix} -0.8944 & -0.7071 \\ 0.4472 & -0.7071 \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} 2e^{-T} - e^{-2T} & 2e^{-T} - 2e^{-2T} \\ e^{-2T} - e^{-T} & 2e^{-2T} - e^{-T} \end{bmatrix}$$

$\therefore$ Proven

2. Estimate X_0 and V_0

$X^* = [3.9$

2. Estimate X_0 and V_0

$$X_0^* = \begin{bmatrix} 3.9 \\ 0.5 \end{bmatrix}$$

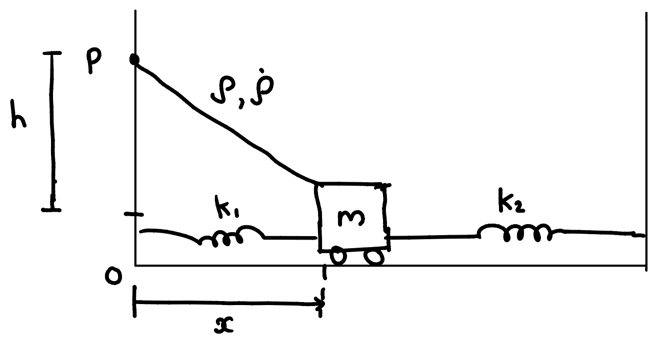
$$\bar{P}_0 = \begin{bmatrix} 1000 & 0 \\ 0 & 100 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.0625 & 0 \\ 0 & 0.01 \end{bmatrix}$$

$k_1 = 2.6$, $k_2 = 3.5$, $m = 1.8$, $h = 6.4$

$t_{span} = 0:1:11$

Let the setup be as follows



$x(t)$ = position $\qquad \bar{x}$ = static equilibrium position

$v(t)$ = velocity

$\dot{x}(t) = v(t)$

$$\rho(t) = (x(t)^2 + h^2)^{1/2}$$

$$\dot{\rho}(t) = \frac{d}{dt}\left(x(t)^2 + h^2\right)^{1/2} = \frac{1}{2}(x^2+h^2)^{-1/2} 2x\dot{x} = \frac{xv}{\rho}$$

We know that for a mass spring system

$$\ddot{x} = -(k_1 + k_2)(x - \bar{x})/m = \dot{v}$$

Let $\omega^2 = (k_1 + k_2)/m$

Assembling state vector $\underline{X}$:

$$\underline{X} = \begin{bmatrix} x \\ v \end{bmatrix}$$

$$\dot{\underline{X}} = \begin{bmatrix} \dot{x} \\ \end{bmatrix} = \begin{bmatrix} v \\ \end{bmatrix} = F$$

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ -\omega^2 (x - \bar{x}) \end{bmatrix} = F$$

Assembling observations Y:

$$Y = \begin{bmatrix} \rho \\ \dot{\rho} \end{bmatrix} + \begin{bmatrix} \varepsilon_\rho \\ \varepsilon_{\dot{\rho}} \end{bmatrix} = G(X(t), t) + \varepsilon(t)$$

where $G = \begin{bmatrix} \rho \\ \dot{\rho} \end{bmatrix} = \begin{bmatrix} [x^2 + h^2]^{1/2} \\ xv/\rho \end{bmatrix}$

Linearizing dynamics:

$$A = \frac{\partial}{\partial X} F = \begin{bmatrix} \partial f_1/\partial X_1 & \partial f_1/\partial X_2 \\ \partial f_2/\partial X_1 & \partial f_2/\partial X_2 \end{bmatrix} = \begin{bmatrix} \partial v/\partial x & \partial v/\partial v \\ \frac{\partial}{\partial x}(-\omega^2(x-\bar{x})) & \frac{\partial}{\partial v}(-\omega^2(x-\bar{x})) \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix}$$

Linearizing measurement model:

$$\tilde{H} = \frac{\partial G}{\partial X} = \begin{bmatrix} \partial\rho/\partial X_1 & \partial\rho/\partial X_2 \\ \partial\dot{\rho}/\partial X_1 & \partial\dot{\rho}/\partial X_2 \end{bmatrix} = \begin{bmatrix} \partial\rho/\partial x & \partial\rho/\partial v \\ \partial\dot{\rho}/\partial x & \partial\dot{\rho}/\partial v \end{bmatrix}$$

* Computing partials in MATLAB

$$\tilde{H} = \begin{bmatrix} x/\rho & 0 \\ v/\rho - x^2 v/\rho^3 & x/\rho \end{bmatrix}$$

State transition matrix Φ:

$$\dot{\Phi} = A\,\Phi(t, t_0) \qquad \Phi(t_0, t_0) = I$$

$$\begin{bmatrix} \dot{\Phi}_{11} & \dot{\Phi}_{12} \\ \dot{\Phi}_{21} & \dot{\Phi}_{22} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix} = \begin{bmatrix} \Phi_{21} & \Phi_{22} \\ -\omega^2 \Phi_{11} & -\omega^2 \Phi_{12} \end{bmatrix}$$

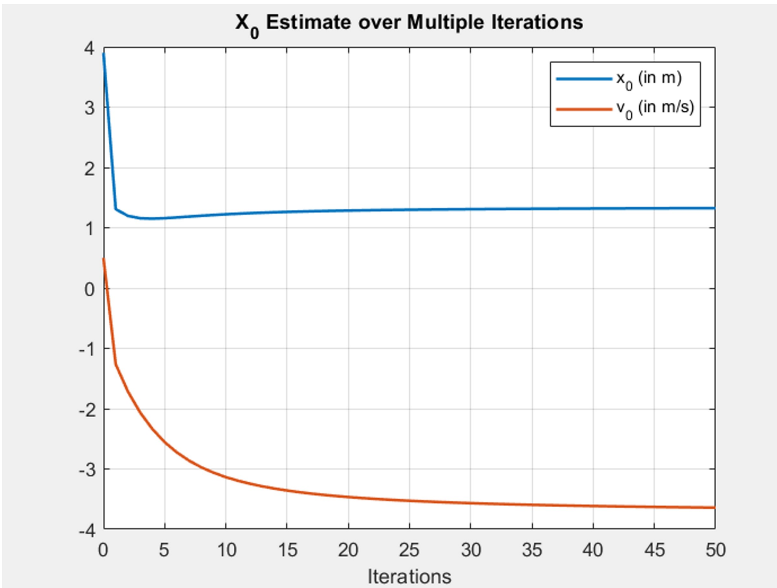
* Solving using dsolve in MATLAB

$$\Phi(t, t_0) = \begin{bmatrix} \cos(\omega t) & \sin(\omega t)/\omega \\ -\omega \sin(\omega t) & \cos(\omega t) \end{bmatrix}$$

After implementing this setup in MATLAB, X_0 after 3 iterations is

$$\begin{bmatrix} 1.157 \\ \underline{\underline{-2.065}} \end{bmatrix}$$

When running more iterations, the values for X_0 changed as follows:



3. Assembling state vector X:

$$X = \begin{bmatrix} x & y & z & v_x & v_y & v_z \end{bmatrix}'$$

$$\dot{X} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \dot{v}_x & \dot{v}_y & \dot{v}_z \end{bmatrix}'$$

$$= \begin{bmatrix} v_x & v_y & v_z - gt & 0 & 0 & -g \end{bmatrix} = F$$

Assembling observations Y:

$$Y = \begin{bmatrix} \rho \\ \alpha \\ \beta \end{bmatrix} + \begin{bmatrix} \varepsilon_\rho \\ \varepsilon_\alpha \\ \varepsilon_\beta \end{bmatrix} = G(X(t), t) + \varepsilon(t)$$

where $G = \begin{bmatrix} \rho \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} (x^2 + y^2 + z^2)^{1/2} \\ \tan^{-1}(y/x) \\ \tan^{-1}(z/\sqrt{x^2+y^2}) \end{bmatrix}$

Answer for part (b)

Answer for part (b)

Linearizing dynamics:

$$A = \frac{\partial F}{\partial \bar{X}} = \begin{bmatrix} \partial f_1/\partial \bar{X}_1 & \cdots & \partial f_1/\partial \bar{X}_6 \\ \partial f_2/\partial \bar{X}_1 & & \vdots \\ \partial f_3/\partial \bar{X}_1 & \ddots & \vdots \\ \partial f_4/\partial \bar{X}_1 & & \vdots \\ \partial f_5/\partial \bar{X}_1 & & \vdots \\ \partial f_6 \, \partial \bar{X}_1 & \cdots & \partial f_6/\partial \bar{X}_6 \end{bmatrix}$$

* Computing in MATLAB

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Linearizing measurement model:

$$\tilde{H} = \frac{\partial G}{\partial \bar{X}} = \begin{bmatrix} \partial \rho/\partial \bar{X}_1 & \cdots & \partial \rho/\partial \bar{X}_6 \\ \partial \alpha/\partial \bar{X}_1 & \cdots & \partial \alpha/\partial \bar{X}_6 \\ \partial \beta/\partial \bar{X}_1 & \cdots & \partial \beta/\partial \bar{X}_6 \end{bmatrix}$$

* Computing partials in MATLAB:

$$= \begin{bmatrix} x/\rho & y/\rho & z/\rho & 0 & 0 & 0 \\ -y/(y^2+x^2) & x/(x^2+y^2) & 0 & 0 & 0 & 0 \\ -xz/\rho^2\sqrt{x^2+y^2} & -yz/\rho^2\sqrt{x^2+y^2} & \sqrt{x^2+y^2}/\rho^2 & 0 & 0 & 0 \end{bmatrix}$$

Answer for part (c)

State transition matrix Φ:

$$\dot{\Phi}(t,t_0) = A(t,t_0) \qquad \Phi(t_0,t_0) = I$$

$$\begin{bmatrix} \dot{\Phi}_{11} & \cdots & \dot{\Phi}_{16} \\ \vdots & \ddots & \vdots \\ \dot{\Phi}_{61} & \cdots & \dot{\Phi}_{66} \end{bmatrix} = \begin{bmatrix} A_{11} & \cdots & A_{16} \\ \vdots & \ddots & \vdots \\ A_{61} & \cdots & A_{66} \end{bmatrix} \begin{bmatrix} \Phi_{11} & \cdots & \Phi_{16} \\ \vdots & \ddots & \vdots \\ \Phi_{61} & \cdots & \Phi_{66} \end{bmatrix}$$

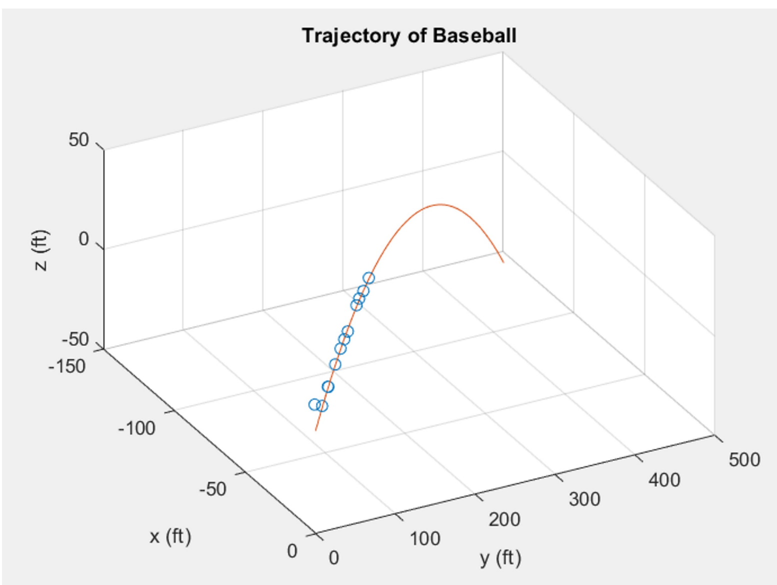
$$\begin{bmatrix} \vdots & \ddots & \vdots \\ \dot{\Phi}_{61} & \cdots & \dot{\Phi}_{66} \end{bmatrix} = \begin{bmatrix} \vdots & \ddots & \vdots \\ A_{61} & \cdots & A_{66} \end{bmatrix} \begin{bmatrix} \vdots & \ddots & \vdots \\ \Phi_{61} & \cdots & \Phi_{66} \end{bmatrix}$$

* Solving using dsolve in MATLAB

$$\Phi(t, t_0) = \begin{bmatrix} 1 & 0 & 0 & t & 0 & 0 \\ 0 & 1 & 0 & 0 & t & 0 \\ 0 & 0 & 1 & 0 & 0 & t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Answer for part (a)

d. The process was repeated 4 times (as gone over in class), and the following trajectory was obtained:



Using the distance formula, it was found that the baseball landed 405.71 ft from the origin.

e. To find the uncertainty of the estimate, the last P_0 term was propagated to the final time using $\Phi(t, 0)$.

$t_f = 3.41s$

$P = \Phi P_0 \Phi^T$ = * Evaluated in MATLAB

The Frobenius norm of this matrix was found

$$\sqrt{\sum_{j=1}^{6} \sum_{i=1}^{6} a_{ij}^2} = \underline{\underline{1.3289}}$$

1 Appendix: MATLAB Code

```
%% 4.5
clc; clear all; close all;
syms tk tau tk1
% tk = 1e-5;
% tk1 = 0;
A = [-1 0; 0 -2]*(tk-tau)
eye(2)*(tk-tk1)
% integral(@(tau)(expm(A)*eye(2)*expm(A)), tk1, tk)
B = [1 1; 1 2];
simplify(expm(A)*B*eye(2)*B')
simplify(expm(A)*B*eye(2)*B'*expm(A))
%% 4.8
clc; clear all; close all;
F = eye(2)*1/2;
Q = [1 1; 0 1];
P = 0;
for i=0:100
    i
    F^i*Q*F'^i
    P = P + F^i*Q*F'^i
end

%% 4.11
clc; clear all; close all;

m0 = 1;
P0 = 2;
f = -0.5;
qc = 1;
tf = 5;
deltat = 0.01;
Q = [];
xbar = [m0];
P = [P0];

for k = deltat:deltat:tf
    Q(int16((k-deltat)/deltat + 1)) = qc/(2*f)*(exp(2*f*deltat) - 1);
    xbar(int16(k/deltat)+1) = exp(floor(k/deltat)*f*deltat)*xbar(1);
    P(int16(k/deltat)+1) = (2*f*P(int16(k/deltat)) + qc)*deltat + P(int16(k/deltat));
end

figure()
subplot(2,1,1)
plot(0:deltat:tf, xbar, 'LineWidth', 1.5)
hold on
sgtitle('$\bar{x}$ and $P$ when $P_0 = 2$', 'Interpreter', 'latex')
ylabel('$\bar{x}$', 'Interpreter', 'latex')
xlabel('Time (s)')
subplot(2,1,2)
plot([0 deltat:deltat:tf], P, 'LineWidth', 1.5)
ylabel('$P$', 'Interpreter', 'latex')
xlabel('Time (s)')
hold off

P0 = 0;
Q = [];
xbar = [m0];
P = [P0]

for k = deltat:deltat:tf
    Q(int16((k-deltat)/deltat + 1)) = qc/(2*f)*(exp(2*f*deltat) - 1);
    xbar(int16(k/deltat)+1) = exp(floor(k/deltat)*f*deltat)*xbar(1);
```

```
    P(int16(k/deltat)+1) = (2*f*P(int16(k/deltat)) + qc)*deltat + P(int16(k/deltat));
end

figure()
subplot(2,1,1)
plot(0:deltat:tf, xbar, 'LineWidth', 1.5)
hold on
sgtitle('$\bar{x}$ and $P$ when $P_0 = 0$', 'Interpreter', 'latex')
ylabel('$\bar{x}$', 'Interpreter', 'latex')
xlabel('Time (s)')
subplot(2,1,2)
plot([0 deltat:deltat:tf], P, 'LineWidth', 1.5)
ylabel('$P$', 'Interpreter', 'latex')
xlabel('Time (s)')
hold off

%% 4.13
clc; clear all; close all;
syms T
AHat = [-1 0; 0 -2];
Q = [-0.8944 -0.7071; 0.4472 0.7071];
A = [0 2; -1 -3];
Q*expm(AHat*T)*inv(Q)

%%
clc; clear all; close all;
A = [0 2; -1 -3];
syms p11 p12 p21 p22
P = [p11 p12; p21 p22];
A*P + P*A'
solve(A*P + P*A' + [0 0; 0 1] == 0)

%%
clc; clear all; close all;
syms P1(t) P2(t)
A = [0 2; -1 -3];
Qc = eye(2);
P = [P1 0;
     0 P2];
dsolve(diff(P, t) == A*P + P*A' + Qc);
%%
clc; clear all; close all;

% Given info
h = 6.4;
k1 = 2.6;
k2 = 3.5;
m = 1.8;
w = sqrt((k1+k2)/m);
% Measurement noise covariance
R0 = [0.0625 0;
     0 0.01];
% Initial state
X0 = [3.9;
      0.5];
tspan = 0:10;
P0_bar = [1000 0;
          0 100];
X0_bar = [0; 0];

% Initial measurements
Y_obs = [];
rho = [6.3118 6.0217 5.1295 6.3685 5.5422 5.5095 5.9740 5.4930 6.8653 6.6661 5.0692]';
rhoDot = [0.3035 1.3854 -1.5512 0.6064 0.8642 -1.5736 1.1588 0.4580 -1.2328 1.4348 -0.4229]';
for i = 1:length(rho)
```

```
    Y_obs = vertcat(Y_obs, rho(i));
    Y_obs = vertcat(Y_obs, rhoDot(i));
end

x_hist = [X0(1)];
v_hist = [X0(2)];
for iter = 1:50
    [~, X_computed] = ode45(@spring_mass_dynamics, tspan, X0);

    Y_computed = [];
    for i = 1:length(X_computed)
        c_i = X_computed(i, 1);
        d_i = X_computed(i, 2);
        rho_i = sqrt(c_i^2 + h^2);
        Y_computed = vertcat(Y_computed, [rho_i; c_i*d_i/rho_i]);
    end

    y_dev = Y_obs - Y_computed;

    H = [];
    R = [];
    for t = 0:10
        Phi = [cos(w*t) 1/w*sin(w*t);
               -w*sin(w*t) cos(w*t)];
        X_i = Phi * X0;
        c_i = X_i(1);
        d_i = X_i(2);
        rho_i = sqrt(c_i^2 + h^2);
        Htilda = [c_i/rho_i 0;
                  d_i/rho_i - c_i^2*d_i/rho_i^3 c_i/rho_i];
        H = vertcat(H, Htilda * Phi);
        R = blkdiag(R, R0);
    end
    X0_hat = inv(H'*inv(R)*H + inv(P0_bar)) * (H'*inv(R)*y_dev + inv(P0_bar)*X0_bar);
    P0_hat = inv(H'*inv(R)*H + inv(P0_bar));

    X0 = X0 + X0_hat;
    P0_bar = P0_hat
    x_hist = vertcat(x_hist, X0(1));
    v_hist = vertcat(v_hist, X0(2));
end

plot(0:length(x_hist)-1, x_hist, 'LineWidth', 1.5)
hold on
plot(0:length(v_hist)-1, v_hist, 'LineWidth', 1.5)
title('X_0 Estimate over Multiple Iterations')
legend('x_0 (in m)', 'v_0 (in m/s)')
xlabel('Iterations')
grid on
box on
hold off

%% 2 math
clc; clear all; close all;
syms phi11(t) phi12(t) phi21(t) phi22(t) w real
phi_mat = [phi11 phi12; phi21 phi22];
conds = [phi11(0) == 1; phi12(0) == 0; phi21(0) == 0; phi22(0) == 1];
A = [0 1; -w^2 0];
sol = dsolve(diff(phi_mat, t) == A*phi_mat, conds)

%% 3 math
clc; clear all; close all;
syms x y z vx vy vz g rho alpha beta real
X = [x; y; z; vx; vy; vz];
F = [vx; vy; vz; 0; 0; -g];
```

```
A = jacobian(F, X)
rho = sqrt(x^2 + y^2 + z^2);
alpha = atan(y/x);
beta = atan(z/sqrt(x^2 + y^2));
G = [rho; alpha; beta];
Htilda = jacobian(G, X)

syms phi11(t) phi12(t) phi13(t) phi14(t) phi15(t) phi16(t)
syms phi21(t) phi22(t) phi23(t) phi24(t) phi25(t) phi26(t)
syms phi31(t) phi32(t) phi33(t) phi34(t) phi35(t) phi36(t)
syms phi41(t) phi42(t) phi43(t) phi44(t) phi45(t) phi46(t)
syms phi51(t) phi52(t) phi53(t) phi54(t) phi55(t) phi56(t)
syms phi61(t) phi62(t) phi63(t) phi64(t) phi65(t) phi66(t)

phi_mat = [phi11, phi12, phi13, phi14, phi15, phi16;
           phi21, phi22, phi23, phi24, phi25, phi26;
           phi31, phi32, phi33, phi34, phi35, phi36;
           phi41, phi42, phi43, phi44, phi45, phi46;
           phi51, phi52, phi53, phi54, phi55, phi56;
           phi61, phi62, phi63, phi64, phi65, phi66];
conds = [phi11(0) == 1;
         phi12(0) == 0;
         phi13(0) == 0;
         phi14(0) == 0;
         phi15(0) == 0;
         phi16(0) == 0;
         phi21(0) == 0;
         phi22(0) == 1;
         phi23(0) == 0;
         phi24(0) == 0;
         phi25(0) == 0;
         phi26(0) == 0;
         phi31(0) == 0;
         phi32(0) == 0;
         phi33(0) == 1;
         phi34(0) == 0;
         phi35(0) == 0;
         phi36(0) == 0;
         phi41(0) == 0;
         phi42(0) == 0;
         phi43(0) == 0;
         phi44(0) == 1;
         phi45(0) == 0;
         phi46(0) == 0;
         phi51(0) == 0;
         phi52(0) == 0;
         phi53(0) == 0;
         phi54(0) == 0;
         phi55(0) == 1;
         phi56(0) == 0;
         phi61(0) == 0;
         phi62(0) == 0;
         phi63(0) == 0;
         phi64(0) == 0;
         phi65(0) == 0;
         phi66(0) == 1];
sol = dsolve(diff(phi_mat, t) == A*phi_mat, conds);

sol_phi_mat = [sol.phi11, sol.phi12, sol.phi13, sol.phi14, sol.phi15, sol.phi16;
               sol.phi21, sol.phi22, sol.phi23, sol.phi24, sol.phi25, sol.phi26;
               sol.phi31, sol.phi32, sol.phi33, sol.phi34, sol.phi35, sol.phi36;
               sol.phi41, sol.phi42, sol.phi43, sol.phi44, sol.phi45, sol.phi46;
               sol.phi51, sol.phi52, sol.phi53, sol.phi54, sol.phi55, sol.phi56;
               sol.phi61, sol.phi62, sol.phi63, sol.phi64, sol.phi65, sol.phi66]
```

```matlab
%% 3
clc; clear all; close all;

% Given info

% Measurement noise covariance
R0 = [5^2 0 0;
      0 (0.1*pi/180)^2 0;
      0 0 (0.1*pi/180)^2];
% Initial state
X0 = [0.4921; 0.4921; 2.0013; -26.2467; 114.3051; 65.9941];
P0_bar = [inf 0 0 0 0 0;
          0 inf 0 0 0 0;
          0 0 inf 0 0 0;
          0 0 0 inf 0 0;
          0 0 0 0 inf 0;
          0 0 0 0 0 inf];
X0_bar = [0; 0; 0; 0; 0; 0];

% Initial measurements
data = importdata('homerun_data.txt');
tspan = data(:, 1);
rho = data(:, 2);
alpha = data(:, 3)*pi/180;
beta = data(:, 4)*pi/180;

Y_obs = [];
for i = 1:length(rho)
    Y_obs = vertcat(Y_obs, rho(i));
    Y_obs = vertcat(Y_obs, alpha(i));
    Y_obs = vertcat(Y_obs, beta(i));
end

x_hist = [X0(1)];
y_hist = [X0(2)];
z_hist = [X0(3)];
all_X = [];
all_Y = [];
all_Z = [];
for iter = 1:10
    [~, X_computed] = ode45(@baseball_dynamics, tspan, X0);


    for i = 1:length(X_computed)
        x_i = X_computed(i, 1);
        y_i = X_computed(i, 2);
        z_i = X_computed(i, 3);
    end


    Y_computed = [];
    X_computed = [];


    H = [];
    R = [];
    for t = 0:0.1:1.1
        Phi = [1 0 0 t 0 0;
               0 1 0 0 t 0;
               0 0 1 0 0 t;
               0 0 0 1 0 0;
               0 0 0 0 1 0;
               0 0 0 0 0 1];
        B = [0; 0; -0.5*t^2; 0; 0; -t];
        X_i = Phi * X0 + B * 32.185;
```

```
        x = X_i(1);
        y = X_i(2);
        z = X_i(3);
        Htilda = [x/sqrt(x^2+y^2+z^2), y/sqrt(x^2+y^2+z^2), z/sqrt(x^2+y^2+z^2), 0, 0, 0;
                  -y/(x^2*(y^2/x^2 + 1)), 1/(x*(y^2/x^2 + 1)), 0, 0, 0, 0;
                  -(x*z)/((z^2/(x^2 + y^2) + 1)*(x^2 + y^2)^(3/2)), -(y*z)/((z^2/(x^2 + y^2) + 1)*(x^2 + y^2)^(3/2)), 1/((z^2/(x^2 + y^2) + 1)*(x^2 + y^2)^(1/2)), 0, 0, 0];
        H = vertcat(H, Htilda * Phi);
        R = blkdiag(R, R0);
        Y_computed = vertcat(Y_computed, [sqrt(x^2 + y^2 + z^2); atan2(y,x); atan2(z,sqrt(x^2 + y^2))]);
%         X_computed = vertcat(X_computed,
        [temp_Y temp_X] = ode45(@baseball_dynamics, linspace(0,2), X0);
        all_X = horzcat(all_X, temp_X(:, 1));
        all_Y = horzcat(all_Y, temp_X(:, 2));
        all_Z = horzcat(all_Z, temp_X(:, 3));
    end
    y_dev = Y_obs - Y_computed;
    X0_hat = inv(H'*inv(R)*H + inv(P0_bar)) * (H'*inv(R)*y_dev + inv(P0_bar)*X0_bar);
%     X0_hat = inv(H'*inv(R)*H) * (H'*inv(R)*y_dev);
%     P0_hat = inv(H'*inv(R)*H);
    P0_hat = inv(H'*inv(R)*H + inv(P0_bar));

    X0 = X0 + X0_hat;
    P0_bar = P0_hat;
    x_hist = vertcat(x_hist, X0(1));
    y_hist = vertcat(y_hist, X0(2));
    z_hist = vertcat(z_hist, X0(3));
end

figure()
plot(x_hist)
hold on
plot(y_hist)
plot(z_hist)
hold off

figure()
[x y z] = sph2cart(alpha, beta, rho)
scatter3(x, y, z)
hold on
% for i=1:size(all_X, 2)
%     plot3(all_X(:, i), all_Y(:, i), all_Z(:, i))
% end
plot3(all_X(:, end), all_Y(:, end), all_Z(:, end))
xlabel('x (ft)')
ylabel('y (ft)')
zlabel('z (ft)')
title('Trajectory of Baseball')
% plot3(all_X(:, 2), all_Y(:, 2), all_Z(:, 2))
% plot3(all_X(:, 3), all_Y(:, 3), all_Z(:, 3))
% plot3(all_X(:, 6), all_Y(:, 4), all_Z(:, 4))
% legend('s', '1', '2', '3', '4', '5')
hold off

[~, idx] = min(all_Z(:, end).^2);
sqrt(all_X(idx, end)^2 + all_Y(idx, end)^2 + all_Z(idx, end)^2)
norm(P0_bar, 'fro')

%% Functions
function Xdot = spring_mass_dynamics(t, state)
    k1 = 2.6;
    k2 = 3.5;
    m = 1.8;
    w = sqrt((k1+k2)/m);
    Xdot = [state(2);
```

```
            -w^2*state(1)];
end

function Xdot = baseball_dynamics(t, state)
    g = 32.185;
    Xdot = [state(4);
            state(5);
            state(6) - g*t;
            0;
            0;
            -g];
end
```