Homework #7

Theodor Chakhachiro
UMID: 15801216
ROB501 - Mathematics for Robotics
University of Michigan, Ann Arbor

October 20, 2021

Problem 1

Given the real and symmetric matrix A, we can apply the eigenvalue decomposition such that $A = O\Lambda O^T$, where O is an orthogonal matrix whose column vectors are the eigenvectors of A and where Λ is a diagonal matrix whose entries are the eigenvalues of A:

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

The eigenvalues and respective eigenvectors of A are computed using MATLAB:

$$\lambda_1 = 2 \quad \text{and} \quad v_1 = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}^T$$

$$\lambda_2 = 2 - \sqrt{2} \quad \text{and} \quad v_2 = \begin{bmatrix} -1 & -\sqrt{2} & 1 \end{bmatrix}^T$$

$$\lambda_3 = 2 + \sqrt{2} \quad \text{and} \quad v_3 = \begin{bmatrix} -1 & \sqrt{2} & 1 \end{bmatrix}^T$$

We can now form the orthogonal and diagonal matrices as such:

$$O = [v_1|v_2|v_3] = \begin{bmatrix} 1 & -1 & -1 \\ 0 & -\sqrt{2} & \sqrt{2} \\ 1 & 1 & 1 \end{bmatrix} \quad \text{and} \quad O^T = \begin{bmatrix} 1 & 0 & 1 \\ -1 & -\sqrt{2} & 1 \\ -1 & \sqrt{2} & 1 \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2-\sqrt{2} & 0 \\ 0 & 0 & 2+\sqrt{2} \end{bmatrix}$$

We can check the results by computing the matrix multiplication:

$$O\Lambda O^T = \begin{bmatrix} 1 & -1 & -1 \\ 0 & -\sqrt{2} & \sqrt{2} \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2-\sqrt{2} & 0 \\ 0 & 0 & 2+\sqrt{2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ -1 & -\sqrt{2} & 1 \\ -1 & \sqrt{2} & 1 \end{bmatrix} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix} = A$$

$$\implies \boxed{A = O\Lambda O^T}$$

Problem 2

Part a

$$Av^1 = \begin{bmatrix} 1 & 0 & \sqrt{2} \\ 0 & 2 & 0 \\ \sqrt{2} & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix} = 2 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = 2v^1$$

Part b

By inspection, we choose v^2 and v^3 as such:

$$v^2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad v^3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

We get:

$$V = [v^1|v^2|v^3] = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Part c

We have:

$$V = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad V^T = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = V$$

We can compute the following matrix multiplication:

$$VAV^T = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \sqrt{2} \\ 0 & 2 & 0 \\ \sqrt{2} & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 0 \\ 1 & 0 & \sqrt{2} \\ \sqrt{2} & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & \sqrt{2} \\ 0 & \sqrt{2} & 0 \end{bmatrix}$$

Now, we notice that VAV^T can be written as:

$$VAV^T = \begin{bmatrix} 2 & 0_{1\times 2} \\ 0_{2\times 1} & A_2 \end{bmatrix} \quad \text{where} \quad A_2 = \begin{bmatrix} 1 & \sqrt{2} \\ \sqrt{2} & 0 \end{bmatrix}$$

Part d

The eigenvalues and respective eigenvectors of A_2 are computed using MATLAB:

$$\lambda_1 = 2 \quad \text{and} \quad v_1 = \begin{bmatrix} \sqrt{2} & 1 \end{bmatrix}^T$$

$$\lambda_2 = -1 \quad \text{and} \quad v_2 = \begin{bmatrix} -\frac{\sqrt{2}}{2} & 1 \end{bmatrix}^T$$

We can now form the orthogonal matrix U_2, whose columns are the eigenvectors of A_2, as such:

$$U_2 = [v_1|v_2] = \begin{bmatrix} \sqrt{2} & -\frac{\sqrt{2}}{2} \\ 1 & 1 \end{bmatrix} \quad \text{and} \quad U_2^T = \begin{bmatrix} \sqrt{2} & 1 \\ -\frac{\sqrt{2}}{2} & 1 \end{bmatrix}$$

We check if the matrix multiplication results in a diagonal matrix:

$$U_2^T A_2 U_2 = \begin{bmatrix} \sqrt{2} & 1 \\ -\frac{\sqrt{2}}{2} & 1 \end{bmatrix} \begin{bmatrix} 1 & \sqrt{2} \\ \sqrt{2} & 0 \end{bmatrix} \begin{bmatrix} \sqrt{2} & -\frac{\sqrt{2}}{2} \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 2\sqrt{2} & 2 \\ \frac{\sqrt{2}}{2} & -1 \end{bmatrix} \begin{bmatrix} \sqrt{2} & -\frac{\sqrt{2}}{2} \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 0 \\ 0 & -\frac{3}{2} \end{bmatrix}$$

Part e

To check whether U is orthogonal or not, we compute its transpose and check that $UU^T = I$:

$$UU^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sqrt{2} & -\frac{\sqrt{2}}{2} \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sqrt{2} & 1 \\ 0 & -\frac{\sqrt{2}}{2} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I$$

Part f

$$O = VU = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sqrt{2} & -\frac{\sqrt{2}}{2} \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & \sqrt{2} & -\frac{\sqrt{2}}{2} \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Verifying that O is orthogonal:

$$OO^T = \begin{bmatrix} 0 & \sqrt{2} & -\frac{\sqrt{2}}{2} \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ \sqrt{2} & 0 & 1 \\ -\frac{\sqrt{2}}{2} & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I$$

Part g

We verify that the matrix multiplication results in a diagonal matrix:

$$O^T AO = \begin{bmatrix} 0 & 1 & 0 \\ \sqrt{2} & 0 & 1 \\ -\frac{\sqrt{2}}{2} & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \sqrt{2} \\ 0 & 2 & 0 \\ \sqrt{2} & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & \sqrt{2} & -\frac{\sqrt{2}}{2} \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & -\frac{3}{2} \end{bmatrix}$$

Problem 3

Part a

For $1 \leq k \leq 500$, we denote the matrices $A_k \in \mathbb{R}^{km \times n}$ and $C_k \in \mathbb{R}^{3\times 100}$ as well as the vectors $x \in \mathbb{R}^{100}$ and $y_i \in \mathbb{R}^3$:

$$A_k = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_k \end{bmatrix}$$

Thus, for each bach k, we have:

$$\dim(A_k) = km \times n = 3k \times 100$$

For A_k to have $\dim(x) = 100$ independent column vectors, we require that number of rows of A_k is greater than or equal to the number of columns of A_k:

$$3k \geq 100 \implies k \geq \mathbb{Z}(\frac{100}{3}) = 34$$

Thus, the least n is equal to 34.

Part b

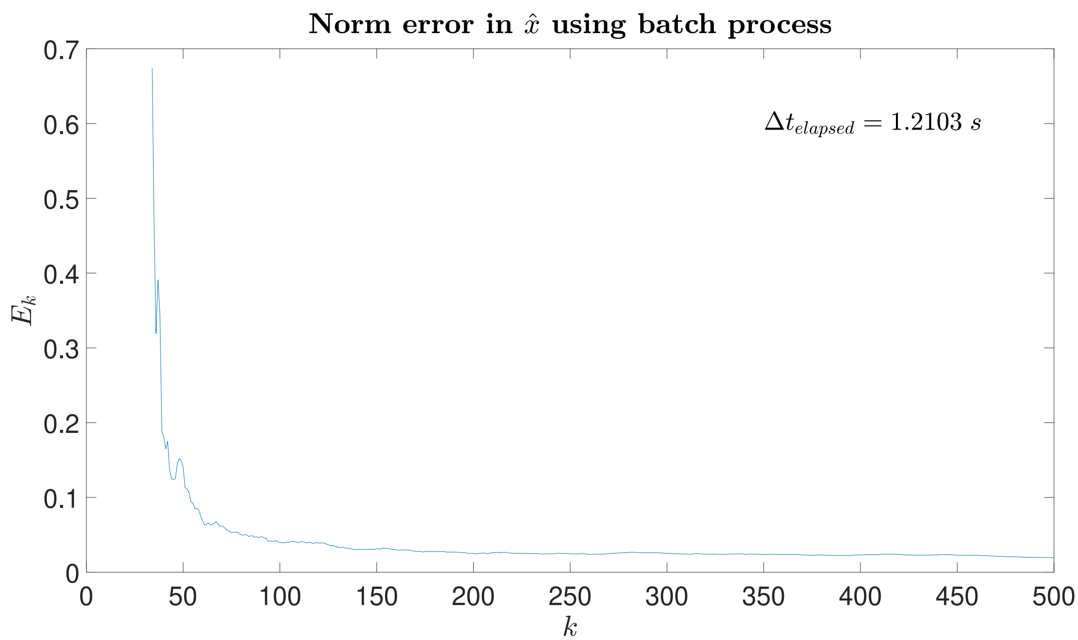

Figure 1: Error E_k Decay over time using a batch process

Part c

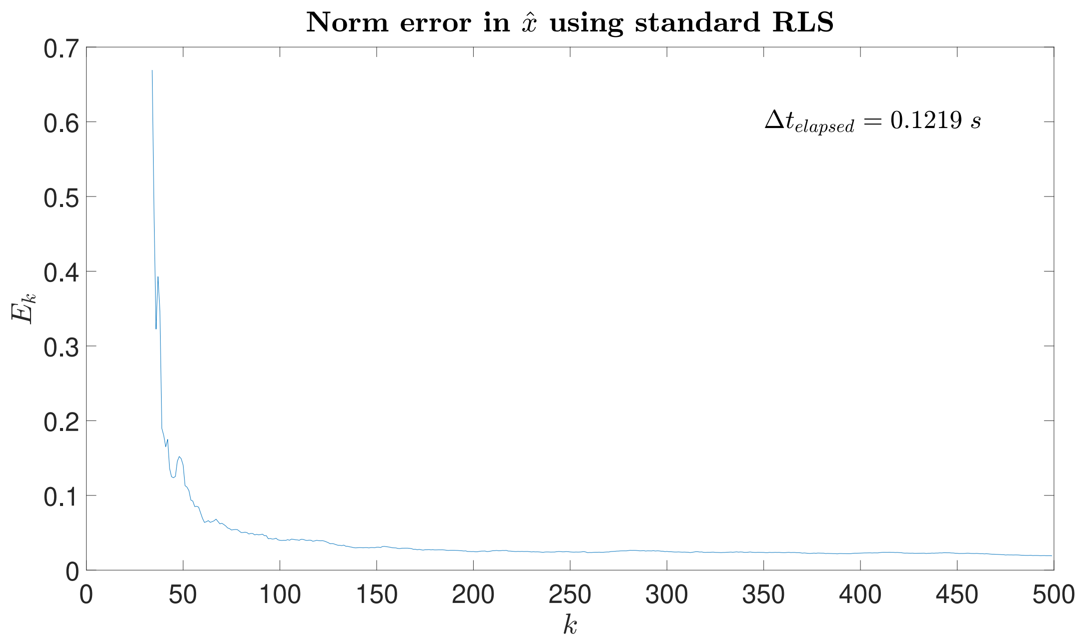

Figure 2: Error E_k Decay over time using standard Recursive Least Squares

Part d

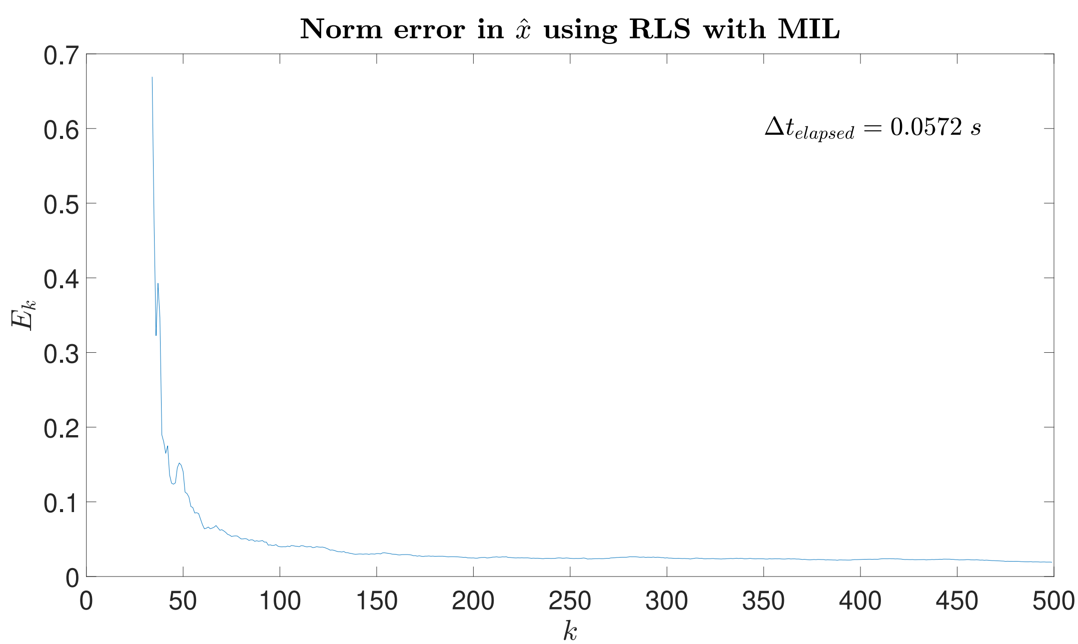

Figure 3: Error E_k Decay over time using RLS with Matrix Inversion Lemma

The matlab code used to solve this part is displayed below:

```
%% Initialize
clear all
clc
load DataHW07_Prob3.mat
%% Part b
A_k = [];
Y_k = [];
R_k = [];
E_k = [];
tic
for i = 1 : ceil(length(x_actual{1})/rank(C{1}))-1
    A_k = [A_k;C{i}];
    Y_k = [Y_k;y{i}];
end
for i = ceil(length(x_actual{1})/rank(C{1})) : length(x_actual)
    A_k = [A_k;C{i}];
    R_k = eye(3*i);
    Y_k = [Y_k;y{i}];
    x_hat = (A_k'*R_k*A_k)\A_k'*R_k*Y_k;
    E_k = [E_k norm(x_actual{i}-x_hat)];
end
toc
figure(1)
plot(ceil(length(x_actual{1})/rank(C{1})) : length(x_actual),E_k);
title('\textbf{Norm error in $\hat{x}$ using batch process}','Interpreter','latex')
xlabel('$k$','Interpreter','latex')
ylabel('$E_k$','Interpreter','latex')
text(350,0.6,'$\Delta t_{elapsed}=1.2103 \; s$','FontSize',40,'Interpreter','latex')
set(gca,'fontsize',40)
%% Part c
Q_n = [];
E_k = [];
S_k = eye(3);
K_n = [];
Q_n = zeros(100);
Gamma = zeros(100,1);
tic
for i = 1 : ceil(length(x_actual{1})/rank(C{1}))
    Q_n = Q_n + C{i}'*S_k*C{i};
    Gamma = Gamma + C{i}'*S_k*y{i};
end
x_hat = Q_n\Gamma;
for k = i : length(x_actual)-1
    Q_n = Q_n + C{k}'*S_k*C{k};
    K_n = inv(Q_n)*C{k}'*S_k;
    x_hat = x_hat + K_n*(y{k}-C{k}*x_hat);
    E_k = [E_k norm(x_actual{k}-x_hat)];
end
toc
figure(2)
plot(ceil(length(x_actual{1})/rank(C{1})) : length(x_actual)-1,E_k);
title('\textbf{Norm error in $\hat{x}$ using standard RLS}','Interpreter','latex')
xlabel('$k$','Interpreter','latex')
ylabel('$E_k$','Interpreter','latex')
text(350,0.6,'$\Delta t_{elapsed}=0.1219 \; s$','FontSize',40,'Interpreter','latex')
set(gca,'fontsize',40)
%% Part d
Q_n = [];
P_n = [];
E_k = [];
S_k = eye(3);
K_n = [];
Q_n = zeros(100);
Gamma = zeros(100,1);
tic
```

```
for i = 1 : ceil(length(x_actual{1})/rank(C{1}))
    Q_n = Q_n + C{i}'*S_k*C{i};
    Gamma = Gamma + C{i}'*S_k*y{i};
end
P_n = inv(Q_n);
x_hat = P_n*Gamma;
for k = i : length(x_actual)-1
    P_n = P_n-P_n*C{k}'*inv((inv(S_k)+C{k}*P_n*C{k}'))*C{k}*P_n;
    K_n = P_n*C{k}'*S_k;
    x_hat = x_hat + K_n*(y{k}-C{k}*x_hat);
    E_k = [E_k norm(x_actual{k}-x_hat)];
end
toc
figure(3)
plot(ceil(length(x_actual{1})/rank(C{1})) : length(x_actual)-1,E_k);
title('\textbf{Norm error in $\hat{x}$ using RLS with MIL}','Interpreter','latex')
xlabel('$k$','Interpreter','latex')
ylabel('$E_k$','Interpreter','latex')
text(350,0.6,'$\Delta t_{elapsed}=0.0572 \; s$','FontSize',40,'Interpreter','latex')
set(gca,'fontsize',40)
```

Problem 4

Part a

For $1 \le k \le 500$, we denote the matrices $A_k \in \mathbb{R}^{km \times n}$ and $C_k \in \mathbb{R}^{3\times 20}$ as well as the vectors $x_i \in \mathbb{R}^{20}$ and $y_i \in \mathbb{R}^3$:

$$A_k = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_k \end{bmatrix}$$

Thus, for each bach k, we have:

$$\dim(A_k) = km \times n = 3k \times 20$$

For A_k to have $\dim(x) = 20$ independent column vectors, we require that number of rows of A_k is greater than or equal to the number of columns of A_k:

$$3k \ge 20 \implies k \ge \mathbb{Z}(\frac{20}{3}) = 7$$

Thus, the least n is equal to 7.

Part b

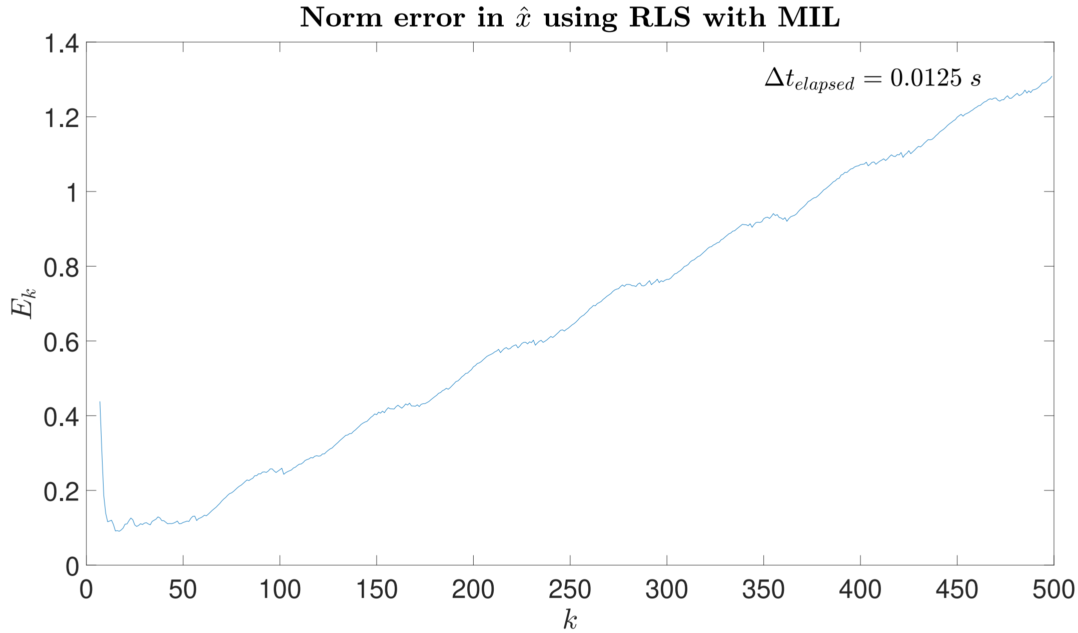

Figure 4: Error E_k Decay over time using RLS with Matrix Inversion Lemma

Part c

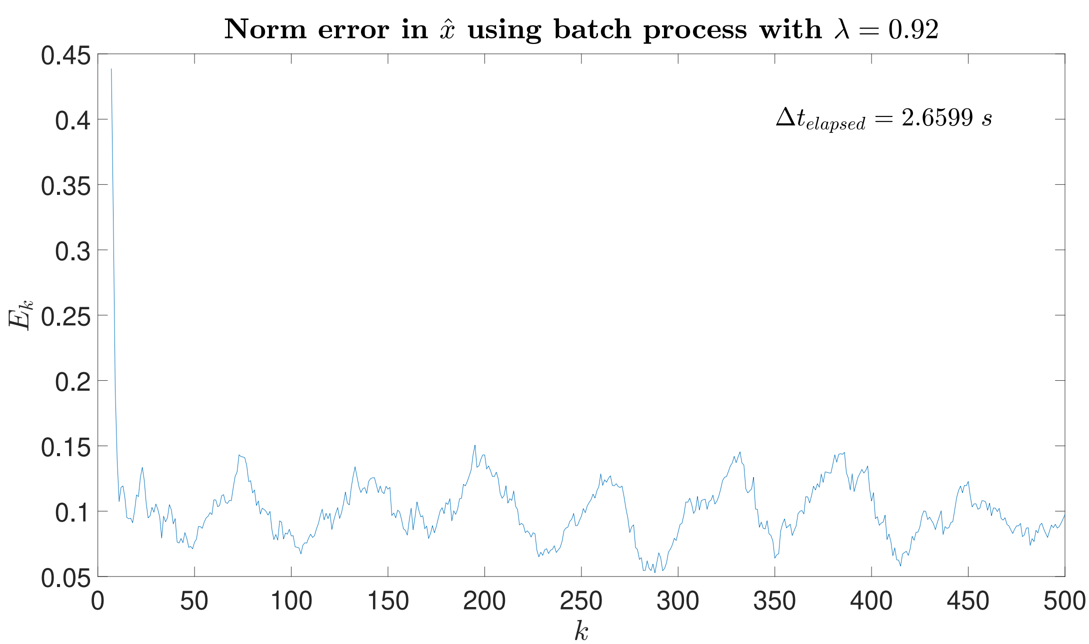

Figure 5: Error E_k Decay over time using a batch process with a forgetting factor $\lambda = 0.92$

Part d

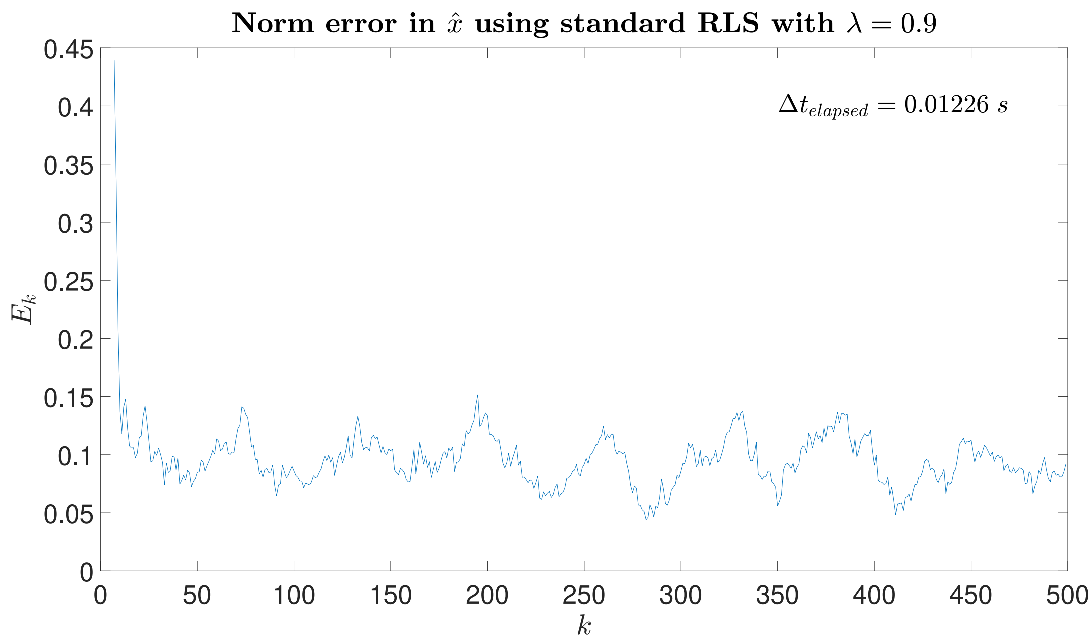

Figure 6: Error E_k Decay over time using Recursive Least Squares with a forgetting factor $\lambda = 0.9$

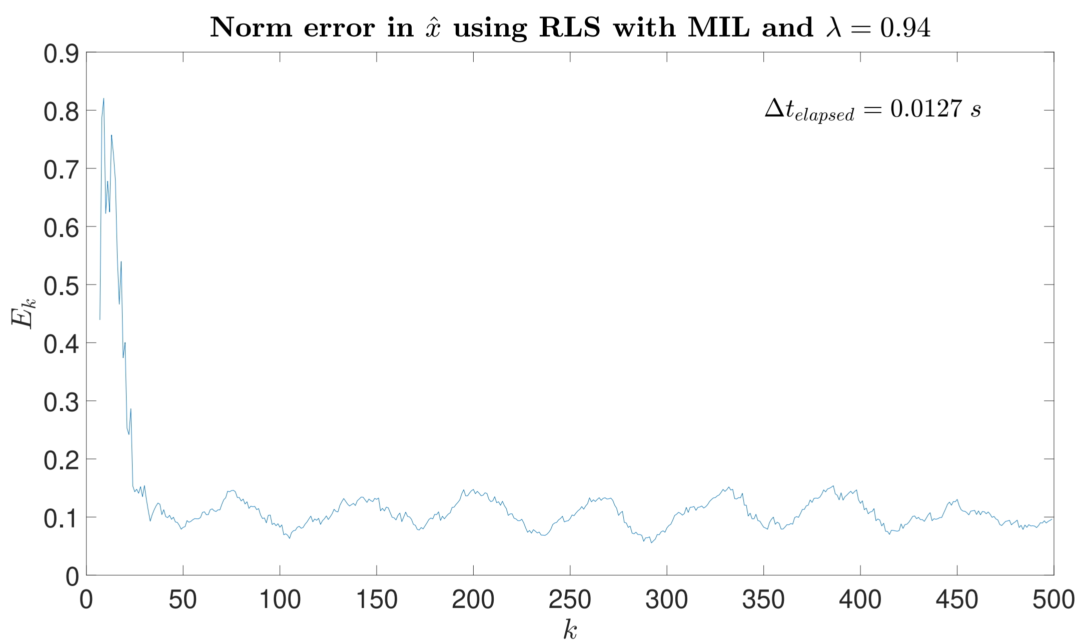

Figure 7: Error E_k Decay over time using RLS and MIL with a forgetting factor $\lambda = 0.94$

The matlab code used to solve this part is displayed below:

```matlab
%% Initialize
clear all
clc
load DataHW07_Prob4.mat
%% Part b
Q_n = [];
P_n = [];
E_k = [];
S_k = eye(3);
K_n = [];
Q_n = zeros(20);
Gamma = zeros(20,1);
tic
for i = 1 : ceil(length(x_actual{1})/rank(C{1}))
    Q_n = Q_n + C{i}'*S_k*C{i};
    Gamma = Gamma + C{i}'*S_k*y{i};
end
P_n = inv(Q_n);
x_hat = P_n*Gamma;
for k = i : length(x_actual)-1
    P_n = P_n-P_n*C{k}'*inv((inv(S_k)+C{k}*P_n*C{k}'))*C{k}*P_n;
    K_n = P_n*C{k}'*S_k;
    x_hat = x_hat + K_n*(y{k}-C{k}*x_hat);
    E_k = [E_k norm(x_actual{k}-x_hat)];
end
toc
figure(1)
plot(ceil(length(x_actual{1})/rank(C{1})) : length(x_actual)-1,E_k);
title('\textbf{Norm error in $\hat{x}$ using RLS with MIL}','Interpreter','latex')
xlabel('$k$','Interpreter','latex')
ylabel('$E_k$','Interpreter','latex')
text(350,1.3,'$\Delta t_{elapsed}=0.0125 \; s$','FontSize',40,'Interpreter','latex')
set(gca,'fontsize',40)
%% Part c
A_k = [];
Y_k = [];
R_k = eye(3);
E_k = [];
lambda = 0.92;
A_k = [A_k;C{1}];
Y_k = [Y_k;y{1}];
tic
for i = 2 : ceil(length(x_actual{1})/rank(C{1}))-1
    A_k = [A_k;C{i}];
    Y_k = [Y_k;y{i}];
    R_k = [lambda*R_k zeros(3*(i-1),3);zeros(3,3*(i-1)) eye(3)];
end
for i = ceil(length(x_actual{1})/rank(C{1})) : length(x_actual)
    A_k = [A_k;C{i}];
    R_k = [lambda*R_k zeros(3*(i-1),3);zeros(3,3*(i-1)) eye(3)];
    Y_k = [Y_k;y{i}];
    x_hat = inv(A_k'*R_k*A_k)*A_k'*R_k*Y_k;
    E_k = [E_k norm(x_actual{i}-x_hat)];
end
toc
figure(2)
plot(ceil(length(x_actual{1})/rank(C{1})) : length(x_actual),E_k);
title('\textbf{Norm error in $\hat{x}$ using batch process with $\lambda=0.92$}','Interpreter','latex')
xlabel('$k$','Interpreter','latex')
ylabel('$E_k$','Interpreter','latex')
text(350,0.4,'$\Delta t_{elapsed}=2.6599 \; s$','FontSize',40,'Interpreter','latex')
set(gca,'fontsize',40)
%% Part d RLS with MIL
Q_n = [];
P_n = [];
E_k = [];
```

```matlab
lambda = 0.94;
Q_n = zeros(20);
Gamma = zeros(20,1);
tic
for i = 1 : ceil(length(x_actual{1})/rank(C{1}))
    Q_n = Q_n + C{i}'*lambda^(ceil(length(x_actual{1})/rank(C{1}))-1)*C{i};
    Gamma = Gamma + C{i}'*lambda^(ceil(length(x_actual{1})/rank(C{1}))-1)*y{i};
end
P_n = inv(Q_n);
x_hat = P_n*Gamma;
for k = i : length(x_actual)-1
    P_n = P_n/lambda-P_n*C{k}'*inv(lambda*eye(size(C{k}*P_n*C{k}'))+C{k}*P_n*C{k}')*C{k}*P_n
        ;
%     P_n = MIL(P_n,C{k}',1/lambda,C{k});
    K_n = P_n*C{k}';
    x_hat = x_hat + K_n*(y{k}-C{k}*x_hat);
    E_k = [E_k norm(x_actual{k}-x_hat)];
end
toc
figure(3)
plot(ceil(length(x_actual{1})/rank(C{1})) : length(x_actual)-1,E_k);
title('\textbf{Norm error in $\hat{x}$ using RLS with MIL and $\lambda=0.94$}','Interpreter'
    ,'latex')
xlabel('$k$','Interpreter','latex')
ylabel('$E_k$','Interpreter','latex')
text(350,0.8,'$\Delta t_{elapsed}=0.0127 \; s$','FontSize',40,'Interpreter','latex')
set(gca,'fontsize',40)
%% Part d RLS
Q_n = [];
P_n = [];
E_k = [];
K_n = [];
lambda = 0.9;
Q_n = zeros(20);
Gamma = zeros(20,1);
tic
for i = 1 : ceil(length(x_actual{1})/rank(C{1}))
    Q_n = Q_n + C{i}'*lambda^(ceil(length(x_actual{1})/rank(C{1}))-1)*C{i};
    Gamma = Gamma + C{i}'*lambda^(ceil(length(x_actual{1})/rank(C{1}))-1)*y{i};
end
P_n = inv(Q_n);
x_hat = P_n*Gamma;
for k = i : length(x_actual)-1
    Q_n = lambda*Q_n+C{k}'*C{k};
    K_n = inv(Q_n)*C{k}';
    x_hat = x_hat + K_n*(y{k}-C{k}*x_hat);
    E_k = [E_k norm(x_actual{k}-x_hat)];
end
toc
figure(3)
plot(ceil(length(x_actual{1})/rank(C{1})) : length(x_actual)-1,E_k);
title('\textbf{Norm error in $\hat{x}$ using standard RLS with $\lambda=0.9$}','Interpreter'
    ,'latex')
xlabel('$k$','Interpreter','latex')
ylabel('$E_k$','Interpreter','latex')
text(350,0.4,'$\Delta t_{elapsed}=0.01226 \; s$','FontSize',40,'Interpreter','latex')
set(gca,'fontsize',40)
```

Problem 5

A symmetric matrix is positive definite if and only if all its eigenvalues are positive. It is positive semi-definite if and only if all eigenvalues are non-negative. Thus, we compute the eigenvalues for each matrix and classify them accordingly:

Part a

Define the pumpkin matrix below:

$$🎃 = \begin{bmatrix} 1 & 3 \\ 3 & 9 \end{bmatrix}$$

We compute the eigenvalues of 🎃 as follows:

$$\det(🎃 - \lambda I) = \begin{vmatrix} 1-\lambda & 3 \\ 3 & 9-\lambda \end{vmatrix} = (1-\lambda)(9-\lambda) - 9 = 0$$

$$\det(🎃 - \lambda I) = \lambda^2 - 10\lambda = 0 \quad \Rightarrow \quad \lambda_1 = 0 \quad \text{and} \quad \lambda_2 = 10$$

Thus 🎃 is symmetric positive semi-definite since all its eigenvalues are non-negative, i.e. $\lambda_{1,2} \geq 0$. Therefore, 🎃 is invertible and we can split it as such:

$$🎃 = (🎃^{\frac{1}{2}})^T 🎃^{\frac{1}{2}} = 🎃^{\frac{1}{2}} 🎃^{\frac{1}{2}}$$

Applying the eigenvalue decomposition where O is and orthogonal matrix whose column vectors are the eigenvectors of 🎃 and Λ is a diagonal matrix whose entries are the eigenvalues of 🎃:

$$🎃 = O\Lambda O^T$$

From MATLAB, we obtain:

$$O = \begin{bmatrix} -3 & \frac{1}{3} \\ 1 & 1 \end{bmatrix} \quad \text{and} \quad \Lambda = \begin{bmatrix} 0 & 0 \\ 0 & 10 \end{bmatrix} = (\Lambda^{\frac{1}{2}})^T \Lambda^{\frac{1}{2}} = \Lambda^{\frac{1}{2}} \Lambda^{\frac{1}{2}} \quad \text{where} \quad \Lambda^{\frac{1}{2}} = \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{10} \end{bmatrix}$$

Thus, we can write:

$$🎃 = O\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}O^T = N^T N \quad \text{where} \quad N = \Lambda^{\frac{1}{2}} O^T$$

Finally:

$$N = \Lambda^{\frac{1}{2}} O^T = \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{10} \end{bmatrix} \begin{bmatrix} -3 & 1 \\ \frac{1}{3} & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 3 \end{bmatrix}$$

$$N^T N = \begin{bmatrix} 0 & 1 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 3 & 9 \end{bmatrix} = 🎃$$

Part b

Define the skull matrix below:

$$💀 = \begin{bmatrix} 6 & 10 & 11 \\ 10 & 19 & 19 \\ 11 & 19 & 21 \end{bmatrix}$$

We compute the eigenvalues of 💀 as follows:

$$\det(\text{💀} - \lambda I) = \begin{vmatrix} 6-\lambda & 10 & 11 \\ 10 & 19-\lambda & 19 \\ 11 & 19 & 21-\lambda \end{vmatrix} = 0$$

Then:

$$(6-\lambda)((19-\lambda)(21-\lambda) - 19^2) - 10(10(21-\lambda) - 19*11) + 11(10*19 - 11(19-\lambda)) = 0$$

$$(6-\lambda)(\lambda^2 - 40\lambda + 38) - 10(1 - 10\lambda) + 11(11\lambda - 19) = 0$$

$$\lambda^3 - 46\lambda^2 + 57\lambda - 9 = 0$$

$$\lambda_1 \approx 0.186 \quad , \quad \lambda_2 \approx 1.084 \quad \text{and} \quad \lambda_3 \approx 44.73$$

Thus 💀 is positive definite since all its eigenvalues are positive, i.e. $\lambda_{1,2,3} > 0$.Therefore, 💀 is invertible and we can split it as such:

$$\text{💀} = (\text{💀}^{\frac{1}{2}})^T \text{💀}^{\frac{1}{2}} = \text{💀}^{\frac{1}{2}} \text{💀}^{\frac{1}{2}}$$

Applying the eigenvalue decomposition where O is and orthogonal matrix whose column vectors are the eigenvectors of 💀 and Λ is a diagonal matrix whose entries are the eigenvalues of 💀:

$$\text{💀} = O \Lambda O^T$$

From MATLAB, we obtain:

$$O = \begin{bmatrix} -0.8745 & 0.3270 & 0.3583 \\ -0.0244 & -0.7674 & 0.6407 \\ 0.4845 & 0.5515 & 0.6791 \end{bmatrix} \quad \text{and} \quad \Lambda = \begin{bmatrix} 0.1856 & 0 & 0 \\ 0 & 1.0842 & 0 \\ 0 & 0 & 44.7302 \end{bmatrix} = (\Lambda^{\frac{1}{2}})^T \Lambda^{\frac{1}{2}} = \Lambda^{\frac{1}{2}} \Lambda^{\frac{1}{2}}$$

Where:

$$\Lambda^{\frac{1}{2}} = \begin{bmatrix} 0.4308 & 0 & 0 \\ 0 & 1.0413 & 0 \\ 0 & 0 & 6.6881 \end{bmatrix}$$

Thus, we can write:

$$\text{💀} = O \Lambda^{\frac{1}{2}} \Lambda^{\frac{1}{2}} O^T = N^T N \quad \text{where} \quad N = \Lambda^{\frac{1}{2}} O^T$$

Finally:

$$N = \Lambda^{\frac{1}{2}} O^T = \begin{bmatrix} 0.4308 & 0 & 0 \\ 0 & 1.0413 & 0 \\ 0 & 0 & 6.6881 \end{bmatrix} \begin{bmatrix} -0.8745 & -0.0244 & 0.4845 \\ 0.3270 & -0.7674 & 0.5515 \\ 0.3583 & 0.6407 & 0.6791 \end{bmatrix} = \begin{bmatrix} -0.3767 & -0.0105 & 0.2087 \\ 0.3405 & -0.7991 & 0.5743 \\ 2.3963 & 4.2850 & 4.5417 \end{bmatrix}$$

$$N^T N = \begin{bmatrix} -0.3767 & 0.3405 & 2.3963 \\ -0.0105 & -0.7991 & 4.2850 \\ 0.2087 & 0.5743 & 4.5417 \end{bmatrix} \begin{bmatrix} -0.3767 & -0.0105 & 0.2087 \\ 0.3405 & -0.7991 & 0.5743 \\ 2.3963 & 4.2850 & 4.5417 \end{bmatrix} = \begin{bmatrix} 6 & 10 & 11 \\ 10 & 19 & 19 \\ 11 & 19 & 21 \end{bmatrix} = \text{💀}$$

Part c

Define the ghost matrix below:

$$\text{👻} = \begin{bmatrix} 2 & 6 & 10 \\ 6 & 10 & 14 \\ 10 & 14 & 18 \end{bmatrix}$$

We compute the eigenvalues of 👻 as follows:

$$\det(\text{👻} - \lambda I) = \begin{vmatrix} 2-\lambda & 6 & 10 \\ 6 & 10-\lambda & 14 \\ 10 & 14 & 18-\lambda \end{vmatrix} = 0$$

Then:

$$(2-\lambda)((10-\lambda)(18-\lambda) - 14^2) - 6(6(18-\lambda) - 14*10) + 10(6*14 - 10(10-\lambda)) = 0$$

$$(2-\lambda)(\lambda^2 - 28\lambda - 16) + 6(6\lambda + 32) + 10(10\lambda - 16) = 0$$

$$\lambda^3 - 30\lambda^2 + 96\lambda = 0$$

$$\lambda_1 = 0 \quad , \quad \lambda_2 \approx -2.91 \quad \text{and} \quad \lambda_3 \approx 32.91$$

Thus 👻 is neither positive definite nor positive semi-definite since $\lambda_2 < 0$.

Problem 6

Part a

Define the pumpkin matrix below:

$$\text{🎃} = \begin{bmatrix} 1 & 3 \\ 3 & 8 \end{bmatrix}$$

We can write 🎃 in the following form:

$$\text{🎃} = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \quad \text{where} \quad A, B, C \in \mathbb{R}$$

In this form, we can determine if 🎃 is PD using Schur's complement. In fact, the following two conditions should be satisfied:

1. $A \succ 0$
2. $C - B^T A^{-1} B \succ 0$

Then:

$$A = 1 > 0$$

$$8 - 3*1*3 = -1 < 0$$

Thus, 🎃 is not positive definite.

Part b

Define the skull matrix below:

$$💀 = \begin{bmatrix} 1 & 0 & 6 \\ 0 & 4 & 7 \\ 6 & 7 & 10 \end{bmatrix}$$

We can write 💀 in the following form:

$$💀 = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \quad \text{where} \quad A \in \mathbb{R} \quad , \quad B \in \mathbb{R}^{1\times 2} \quad \text{and} \quad C \in \mathbb{R}^{2\times 2}$$

In this form, we can determine if 💀 is PD using Schur's complement. In fact, the following two conditions should be satisfied:

1. $A \succ 0$
2. $C - B^T A^{-1} B \succ 0$

Then, we have:

$$A = 1 > 0$$

$$C - B^T A^{-1} B = \begin{bmatrix} 4 & 7 \\ 7 & 10 \end{bmatrix} - \begin{bmatrix} 0 \\ 6 \end{bmatrix} \begin{bmatrix} 0 & 6 \end{bmatrix} = \begin{bmatrix} 4 & 7 \\ 7 & -26 \end{bmatrix}$$

Recursively for the schur complement of 💀, we check the above conditions for the following form:

$$C - B^T A^{-1} B = \begin{bmatrix} m & n \\ n & p \end{bmatrix} \quad \text{where} \quad m = 4, \quad n = 7 \quad \text{and} \quad p = -26$$

Then:

1. $m = 4 > 0$
2. $p - \frac{n^2}{m} = -26 - \frac{49}{4} < 0$

Thus, 💀 is not positive definite since its Schur complement of is not PD.

Part c

Define the ghost matrix below:

$$👻 = \begin{bmatrix} 1 & 2 & 6 \\ 2 & 5 & 7 \\ 6 & 7 & a \end{bmatrix}$$

We can write 👻 in the following form:

$$👻 = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \quad \text{where} \quad A \in \mathbb{R} \quad , \quad B \in \mathbb{R}^{1\times 2} \quad \text{and} \quad C \in \mathbb{R}^{2\times 2}$$

In this form, we can determine if 👻 is PD using Schur's complement. In fact, the following two conditions should be satisfied:

1. $A \succ 0$
2. $C - B^T A^{-1} B \succ 0$

Then, we have:

$$A = 1 > 0$$

$$C - B^T A^{-1} B = \begin{bmatrix} 5 & 7 \\ 7 & a \end{bmatrix} - \begin{bmatrix} 2 \\ 6 \end{bmatrix} \begin{bmatrix} 2 & 6 \end{bmatrix} = \begin{bmatrix} 1 & -5 \\ -5 & a-36 \end{bmatrix}$$

Recursively for the schur complement of , we check the above conditions for the following form:

$$C - B^T A^{-1} B = \begin{bmatrix} m & n \\ n & p \end{bmatrix} \quad \text{where} \quad m = 1, \quad n = -5 \quad \text{and} \quad p = a - 36$$

Then:

1. $m = 1 > 0$
2. $p - \frac{n^2}{m} = a - 36 - 25 = a - 61 > 0 \qquad a > 61$

Thus, for to be PD, we require the following bounds on a:

$$\boxed{a \in \,]61, +\infty[}$$

Problem 7

Part a

Since we have an under-determined system, we can find an approximate solution as such:

$$\begin{aligned}
\hat{x} &= A^T (AA^T)^{-1} b \\
&= \begin{bmatrix} 1 & 3 \\ 3 & 8 \\ 2 & 4 \end{bmatrix} \left(\begin{bmatrix} 1 & 3 & 2 \\ 3 & 8 & 4 \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 3 & 8 \\ 2 & 4 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 3 \\ 3 & 8 \\ 2 & 4 \end{bmatrix} \left(\begin{bmatrix} 14 & 35 \\ 35 & 89 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 3 \\ 3 & 8 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 4.23 & -1.67 \\ -1.67 & 0.67 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 3 \\ 3 & 8 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 0.905 \\ -0.33 \end{bmatrix} \\
\hat{x} &= \begin{bmatrix} -0.095 \\ 0.048 \\ 0.4762 \end{bmatrix}
\end{aligned}$$

Part b

Denote the symmetric matrix Q:

$$Q = \begin{bmatrix} 5 & 1 & 9 \\ 1 & 2 & 1 \\ 9 & 1 & 17 \end{bmatrix}$$

From MATLAB, we compute the eigenvalues of Q and obtain:

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \text{eig}(Q) = \begin{bmatrix} 0.0913 \\ 2 \\ 21.9087 \end{bmatrix}$$

Since the eigenvalues of Q are positive and real, then Q is a hermitian matrix and we can utilize the following property:

$$Q^T = Q \quad \text{and} \quad y^T Q = y^T Q^T = (Qy)^T \tag{1}$$

We define the orthogonal nullspace of A as follows:

$$\mathcal{N}(A)^\perp = \{y \mid y^T Q x = 0 \;,\; \forall x \in \mathcal{N}(A)\}$$

Thus, from (1):

$$\mathcal{N}(A)^\perp = \{y \mid (Qy)^T x = 0 \;,\; \forall x \in \mathcal{N}(A)\}$$

Thus, Qy is orthogonal to all x in the $\mathcal{N}(A)$ and we can write:

$$\mathcal{N}(A)^\perp = \{y \mid Qy = A^T\alpha \;,\; \alpha \in \mathbb{R}^m\}$$

Rearranging, we obtain:

$$\begin{aligned} Qy &= A^T\alpha \\ y &= Q^{-1}A^T\alpha \\ y &= (AQ^{-1})^T\alpha \end{aligned}$$

We define the new matrix $\tilde{A}$ as follows:

$$\tilde{A} = AQ^{-1}$$

Since $\hat{x} \in \mathcal{R}(\tilde{A})$, we can write:

$$\begin{aligned} \hat{x} &= \tilde{A}^T\alpha \\ A\hat{x} &= A\tilde{A}^T\alpha \\ \alpha &= (A\tilde{A}^T)^{-1}A\hat{x} \\ \hat{x} &= \tilde{A}(A\tilde{A}^T)^{-1}A\hat{x} \\ \implies \hat{x} &= \tilde{A}(A\tilde{A}^T)^{-1}b \end{aligned}$$

For the above expression to be valid, we need to prove that $A\tilde{A}^T$ is invertible:

$$\tilde{A} = AQ^{-1} = \begin{bmatrix} 1 & 3 & 2 \\ 3 & 8 & 4 \end{bmatrix} \begin{bmatrix} 5 & 1 & 9 \\ 1 & 2 & 1 \\ 9 & 1 & 17 \end{bmatrix}^{-1} = \begin{bmatrix} -6.25 & 3 & 3.25 \\ -8.25 & 6 & 4.25 \end{bmatrix}$$

And:

$$\det(A\tilde{A}^T) = \left| \begin{bmatrix} 1 & 3 & 2 \\ 3 & 8 & 4 \end{bmatrix} \begin{bmatrix} -6.25 & -8.25 \\ 3 & 6 \\ 3.25 & 4.25 \end{bmatrix} \right| = \begin{vmatrix} 9.25 & 18.25 \\ 18.25 & 40.25 \end{vmatrix} = 39.25 > 0$$

Thus $A\tilde{A}^T$ is invertible and we can find the minimum norm solution as such:

$$\begin{aligned} \hat{x} &= \tilde{A}^T (A\tilde{A}^T)^{-1} b \\ &= \begin{bmatrix} -6.25 & -8.25 \\ 3 & 6 \\ 3.25 & 4.25 \end{bmatrix} \left(\begin{bmatrix} 1 & 3 & 2 \\ 3 & 8 & 4 \end{bmatrix} \begin{bmatrix} -6.25 & -8.25 \\ 3 & 6 \\ 3.25 & 4.25 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \\ &= \begin{bmatrix} -6.25 & -8.25 \\ 3 & 6 \\ 3.25 & 4.25 \end{bmatrix} \left(\begin{bmatrix} 9.25 & 18.25 \\ 18.25 & 40.25 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \\ &= \begin{bmatrix} -6.25 & -8.25 \\ 3 & 6 \\ 3.25 & 4.25 \end{bmatrix} \begin{bmatrix} 1.0255 & -0.4650 \\ -0.4650 & 0.2357 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \\ &= \begin{bmatrix} -6.25 & -8.25 \\ 3 & 6 \\ 3.25 & 4.25 \end{bmatrix} \begin{bmatrix} 0.0955 \\ 0.0064 \end{bmatrix} \\ \hat{x} &= \begin{bmatrix} -0.6497 \\ 0.3249 \\ 0.3376 \end{bmatrix} \end{aligned}$$