Exam Number: II

ROB 501 Exam-II

You can pick any 36 hours between 4:30pm (ET) December 15, 2021 (Wednesday) and 11:59pm (ET) December 18, 2021 (Saturday) to solve this exam.

HONOR PLEDGE: Copy (NOW) and SIGN (**after the exam is completed**): I have neither given nor received aid on this exam, nor have I observed a violation of the Engineering Honor Code.

I have neither given nor recieved aid on this exam, nor have I observed a violation of the Engineering Honor Code.

TH

SIGNATURE

(Sign **after** the exam is completed)

Chakhachiro, Theodor

LAST NAME (PRINTED), *FIRST NAME*

RULES:

1. The exam is open book, open lecture handouts and slides, open recitation notes, open HW solutions, open internet (under the communication and usage restrictions mentioned below).
2. If you use MATLAB or any other scientific software to complete some parts of the exam. You are required to submit your script along with your solution in such case.
3. You are not allowed to communicate with anyone other than the Course instructor and the GSIs related to the exam during the entire period. If you have questions, you can post a private Piazza post for the instructors or email `necmiye@umich.edu` with GSIs on cc.
4. You are not allowed to use any online "course helper" sites like Chegg, Course Hero, and Slader, in any part of the exam. You are not allowed to post exam questions on the internet or discuss them online.
5. Please do not wait until the last minute to upload your solution to Gradescope and double-check to make sure you uploaded the correct pdf. If you run into problems with Gradescope, email your .pdf file as an attachment to Prof. Ozay as soon as practicable at `necmiye@umich.edu`.

SUBMISSION AND GRADING INSTRUCTIONS:

1. The maximum possible score is 80 (+3 bonus points). To maximize your own score on this exam, read the questions carefully and write legibly. For those problems that allow partial credit, show your work clearly.

2. You must submit your solutions in a single pdf. You will be asked to mark where each solution is.

3. **Honor Code:** The first page of your submitted pdf should include a hand-written and signed honor code (see the first page of this pdf). Without this, your exam will not be graded.

4. **For problems 1-5** Use this page to record your answers. We will NOT grade other pages and we do not care if you make a mistake when copying your answers to this page. Please be careful. If you are submitting handwritten (or word-processed) documents, make sure to make a similar table where you record all your True/False (and fill in the blanks for 1(a),(b),(c)) answers. There is no partial credit on these questions. You are welcome to leave some justification but we will not look at them.

5. **For problems 6-7-8** Record your final answer in the box whenever one is provided. If you are submitting hand-written (or word-processed) documents, make sure to box or highlight the final result. However, you MUST show your work to get credit. In other words, a correct result with no reasoning or wrong reasoning could lead to no points.

Answers for Problem 1	
Problem 1(a)	$(-2,3)\cup(3,4)\cup(7,\infty)$
Problem 1(b)	$[-2,4]\cup[7,\infty)$
Problem 1(c)	$\{-2,3,4,5,7\}$
Problem 1(d)	$\boxed{\text{True}}$ False

Answers for the True/False Part				
	(a)	(b)	(c)	(d)
Problem 2	True	True	False	True
Problem 3	False	False	True	True
Problem 4	True	False	False	False
Problem 5	True	True	True	True

Problem 6

Part a

$$y_0 = C_0 x + \epsilon_0$$

Where:

$$y_0 = \begin{bmatrix} 2 \\ 4 \\ 0 \end{bmatrix} \quad C_0 = \begin{bmatrix} 1 & 1 \\ 0 & 2 \\ 1 & -1 \end{bmatrix} \quad \mu_0 = \mathcal{E}\{\epsilon_0\} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad Q_0 = \mathcal{E}\{\epsilon_0 \epsilon_0^T\} = \begin{bmatrix} 0.7 & -0.4 & -0.1 \\ -0.4 & 0.8 & 0.2 \\ -0.1 & 0.2 & 0.3 \end{bmatrix}$$

The estimate $\hat{x}$ using BLUE is given by:

$$\hat{x} = (C_0^T Q_0^{-1} C_0)^{-1} C_0^T Q_0^{-1} y_0$$

Using MATLAB, we get:

$$\boxed{\hat{x} = \begin{bmatrix} 1.1875 \\ 1.5625 \end{bmatrix}}$$

Part b

$$y_1 = C_1 x + \epsilon_1$$

Where:

$$C_1 = \begin{bmatrix} 1 & 2 \end{bmatrix} \quad y_1 = 4 \quad Q_1 = \mathcal{E}\{\epsilon_1^2\} = 0.01$$

To compute the new BLUE estimate $\hat{x}_{new}$ using recursion, we first calculate the covariance of the previous estimate:

$$P_1 = C_0^T Q_0^{-1} C_0$$

Then, we can compute the gain K_1 as follows:

$$K_1 = P_1 C_1^T (C_1 P_1 C_1^T + Q_1)^{-1}$$

The new estimate is thus computed as follows:

$$\hat{x}_{new} = \hat{x} + K_1 (y_1 - C_1 \hat{x})$$

Using MATLAB, we can compute the new estimate and find:

$$\boxed{\hat{x}_{new} = \begin{bmatrix} 1.0405 \\ 1.4834 \end{bmatrix}}$$

The MATLAB code used to solve this problem is show below:

```
%% FINAL PROBLEM 6
clear all
clc
%% Initialize
C_0 = [1 1;0 2;1 -1];
Q_0 = [0.7 -0.4 -0.1;-0.4 0.8 0.2; -0.1 0.2 0.3];
y_0 = [2;4;0];
```

```
%% Part a
x_hat = inv(C_0'*inv(Q_0)*C_0)*C_0'*inv(Q_0)*y_0;
%% Part b Method 1: without recursion
% Initializing new measurement
C_1 = [1 2];
y_1 = 4;
Q_1 = 0.01;
% Augmenting matrices
C = [C_0;C_1];
Q = [Q_0 zeros(3,1); zeros(1,3) Q_1];
y = [y_0;y_1];
% Estimating
x_new_m1 = inv(C'*inv(Q)*C)*C'*inv(Q)*y;
%% Part b Method 2: using recursion
% Initializing new measurement
C_1 = [1 2];
y_1 = 4;
Q_1 = 0.01;
P_1 = inv(C_0'*inv(Q_0)*C_0);
% Estimating
x_new_m2 = x_hat + (P_1*C_1'*inv(C_1*P_1*C_1'+Q_1))*(y_1-C_1*x_hat);
%% Comparison
% Comparing both results using each method, we realize that they are
% similar.
```

Problem 7

$$\begin{aligned} x_{k+1} &= x_k + 0.1s_k + 0.1w_k \\ s_{k+1} &= s_k + 0.2w_k \end{aligned}$$

Part a

Writing the model equations in state-space form, we get:

$$z_{k+1} = A_k z_k + G_k w_k$$

Where:

$$z_k = \begin{bmatrix} x_k \\ s_k \end{bmatrix} \quad A_k = A = \begin{bmatrix} 1 & 0.1 \\ 1 & 0 \end{bmatrix} \quad G_k = G = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$$

The estimate of the conditional dependency of z_3 given z_2 is found as follows. First, using the dynamics of the system, we have:

$$z_{3|2} = A_2 z_{2|2} + G_2 w_{2|2}$$

Now, the estimate $\hat{z}_{3|2}$ is given by:

$$\begin{aligned} \hat{z}_{3|2} &= \mathcal{E}\{z_{3|2}\} \\ &= \mathcal{E}\{A_2 z_{2|2} + G_2 w_{2|2}\} \\ &= A\mathcal{E}\{z_{2|2}\} + G\mathcal{E}\{w_{2|2}\} \\ &= \hat{z}_{3|2} + G\mathcal{E}\{w_2\} \end{aligned}$$

Since w_k is a zero mean white noise then $\mathcal{E}\{w_2\} = 0$. Given, $\hat{z}_{2|2} = \mathcal{E}\{z_{2|2}\} = \begin{bmatrix} 2 & 0.5 \end{bmatrix}^T$, we can calculate the value of estimate $\hat{z}_{3|2}$ as follows:

$$\hat{z}_{3|2} = A\hat{z}_{2|2}$$

Using MATLAB, we obtain:

$$\boxed{\hat{z}_{3|2} = \begin{bmatrix} 2.05 \\ 0.5 \end{bmatrix}}$$

Part b

To obtain the Kalman gain $K_3^{(1)}$, we first need to compute the conditional covariance matrix $P_{3|2}$. The latter is found using the following formula since w_k is zero mean white noise ($R_{2|2} = R_2 = R$):

$$P_{3|2} = AP_{2|2}A^T + GR_2G^T$$

Writing the first sensor model in state-space format, we obtain:

$$y_k^{(1)} = x_k + s_k + v_k^{(1)} := C^{(1)}z_k + v_k^{(1)}$$

Where:

$$C^{(1)} = \begin{bmatrix} 1 & 1 \end{bmatrix} \quad \mathcal{E}\{v_k^{(1)2}\} = Q^{(1)} = 2$$

The Kalman gain $K_3^{(1)}$ is then computed as follows:

$$K_3^{(1)} = P_{3|2}C^{(1)T}(C^{(1)}P_{3|2}C^{(1)T} + Q^{(1)})^{-1}$$

Now, we can compute the estimate $\hat{z}_{3|3}$ as follows:

$$\hat{z}_{3|3}^{(1)} = \hat{z}_{3|2} + K_3^{(1)}(y_3^{(1)} - C^{(1)}\hat{z}_{3|2})$$

Using MATLAB for $y_3^{(1)} = 2.5$, we obtain:

$$\boxed{K_3^{(1)} = \begin{bmatrix} 0.3298 \\ 0.4920 \end{bmatrix} \quad \hat{z}_{3|3}^{(1)} = \begin{bmatrix} 2.0335 \\ 0.4754 \end{bmatrix}}$$

Part c

Repeating the same procedure for sensor 2 using the same $P_{3|2}$ computed in the previous part. We first write the second sensor model in state-space format, we obtain:

$$y_k^{(2)} = 0.1x_k + s_k + v_k^{(2)} := C^{(2)}z_k + v_k^{(2)}$$

Where:

$$C^{(2)} = \begin{bmatrix} 0.1 & 1 \end{bmatrix} \quad \mathcal{E}\{v_k^{(2)2}\} = Q^{(2)} = 1$$

The Kalman gain $K_3^{(2)}$ is then computed as follows:

$$K_3^{(2)} = P_{3|2}C^{(2)T}(C^{(2)}P_{3|2}C^{(2)T} + Q^{(2)})^{-1}$$

Now, we can compute the estimate $\hat{z}_{3|3}$ as follows:

$$\hat{z}_{3|3}^{(2)} = \hat{z}_{3|2} + K_3^{(2)}(y_3^{(2)} - C^{(2)}\hat{z}_{3|2})$$

Using MATLAB for $y_3^{(2)} = 0.7$, we obtain:

$$\boxed{K_3^{(2)} = \begin{bmatrix} 0.3091 \\ 0.7836 \end{bmatrix} \quad \hat{z}_{3|3}^{(2)} = \begin{bmatrix} 2.0485 \\ 0.4961 \end{bmatrix}}$$

Part d

Computing the covariance matrix for each estimate using the following formula:

$$P_{3|3}^{(i)} = P_{3|2} - P_{3|2}C^{(i)T}(C^{(i)}P_{3|2}C^{(i)T} + Q^{(i)})^{-1}C^{(i)}P_{3|2} \quad \forall i = 1, 2$$

Using MATLAB, we find the covariance matrix of each sensor model to be:

$$P_{3|3}^{(1)} = \begin{bmatrix} 1.0399 & -0.3803 \\ -0.3803 & 1.3643 \end{bmatrix} \quad P_{3|3}^{(2)} = \begin{bmatrix} 1.7451 & 0.1345 \\ 0.1345 & 0.7701 \end{bmatrix}$$

Now, since we only care about the accuracy of the position x_3 at time $k = 3$, we only look at the first element of each of the covariance matrices which indicates the variance on the position x_3. Thus, we can get the standard deviation of each estimate as follows:

$$\sigma_{x_3}^{(1)} = \sqrt{1.0399} = 1.019$$
$$\sigma_{x_3}^{(2)} = \sqrt{1.7451} = 1.321$$

Since the standard deviation of the position x_3 using the first sensor model is less than the standard deviation given by the second sensor mode, sensor model 1 is more accurate. The standard deviation is a useful parameter to look at when checking accuracy, since large values indicate that the values are more spread out while while low values imply they are narrowly bounded.

Thus, **sensor 1** gives a more accurate (less uncertain) estimate of the position x_3 at time $k = 3$.

Part e

We want to minimize the following problem:

$$\begin{aligned} \min \quad & ||u||_\infty \\ \text{s.t.} \quad & x_4 \in [-0.1, 0.1] \\ & x_2 = 2 \\ & x_3 = x_2 + 0.1u_2 \\ & x_4 = x_3 + 0.1u_3 \\ & u := [u_2, u_3]^T \end{aligned}$$

We can write write an equivalent Linear Program as follows:

$$\begin{aligned} \min \quad & c^T X \\ \text{s.t.} \quad & AX \leq b \\ & A_{eq}X = b_{eq} \end{aligned}$$

Where:

$$X = \begin{bmatrix} x_2 \\ x_3 \\ x_4 \\ u_2 \\ u_3 \\ s \end{bmatrix} \in \mathbb{R}^{n+1} \quad c = \begin{bmatrix} 0_{1\times5} & 1 \end{bmatrix} \quad n = 5$$

Constructing the equality constraint matrices, we have:

$$\begin{aligned} `x_2 &= 2 \\ x_2 + 0.1u_2 - x_3 &= 0 \\ x_3 + 0.1u_3 - x_4 &= 0 \end{aligned}$$

Thus:

$$A_{eq} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0.1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0.1 & 0 \end{bmatrix} \quad b_{eq} = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$$

Now, constructing the inequality constraint matrices, we have:

$$||u||_\infty \leq s \implies \max(|u_2|, |u_3|) \leq s \tag{1}$$

Thus:

$$\begin{aligned} -s \leq & u_2 \leq s \\ -s \leq & u_3 \leq s \end{aligned}$$

$$x_4 \in [-0.1, 0.1] \implies ||x_4||_\infty \leq 0.1 \implies |x_4| \leq 0.1 \tag{2}$$

Thus:

$$-0.1 \leq x_4 \leq 0.1$$

From equations (1) and (2), we have:

$$\begin{aligned} u_2 - s &\leq 0 \\ -u_2 - s &\leq 0 \\ u_3 - s &\leq 0 \\ -u_3 - s &\leq 0 \\ x_4 &\leq 0.1 \\ -x_4 &\leq 0.1 \end{aligned}$$

Thus:

$$A_in = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \quad b_in = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.1 \\ 0.1 \end{bmatrix}$$

Solving the LP, we get:

$$\boxed{u^* = \begin{bmatrix} -9.5 \\ -9.5 \end{bmatrix}}$$

The MATLAB code used to solve this problem is displayed below:

```
%% FINAL PROBLEM 7
clear all
clc
%% Model + Initializing
% z_k+1 = A_k*z_k + G_k*w_k
% y_1k = C_1k*z_k + v_1k;
% y_2k = C_2k*z_k +v_2k;
A_k = [1 0.1;0 1];
G_k = [0.1;0.2];
C_1k = [1 1];
C_2k = [0.1 1];
R_k = 2;
Q_1k = 2;
Q_2k = 1;
z_2_2 = [2;0.5];
P_2_2 = [2 1;1 4];
%% Part a
z_3_2 = A_k*z_2_2;
%% Part b
y_3_1 = 2.5;
P_3_2 = A_k*P_2_2*A_k'+G_k*R_k*G_k';
K_3_1 = P_3_2*C_1k'*inv(C_1k*P_3_2*C_1k'+Q_1k);
z_3_3_1 = z_3_2 + K_3_1*(y_3_1 - C_1k*z_3_2);
%% Part c
y_3_2 = 0.7;
P_3_2 = A_k*P_2_2*A_k'+G_k*R_k*G_k';
K_3_2 = P_3_2*C_2k'*inv(C_2k*P_3_2*C_2k'+Q_2k);
z_3_3_2 = z_3_2 + K_3_2*(y_3_2 - C_2k*z_3_2);
%% Part d
P_3_3_1 = P_3_2 - P_3_2*C_1k'*inv(C_1k*P_3_2*C_1k'+Q_1k)*C_1k*P_3_2;
P_3_3_2 = P_3_2 - P_3_2*C_2k'*inv(C_2k*P_3_2*C_2k'+Q_2k)*C_2k*P_3_2;
%% Part e
A_eq = [1 0 0 0 0 0;1 -1 0 0.1 0 0;0 1 -1 0 0.1 0];
```

```
B_eq = [2;0;0];
A_in = [0   0   0   1   0   -1 ;0   0   0   -1   0   -1 ;0   0   0   0   1   -1 ; ...
        0   0   0   0   -1   -1 ;0   0   1   0   0   0 ;0   0   -1   0   0   0] ;
B_in = [0;0;0;0;0.1;0.1];
% Inifinity Norm
X_inf_norm = linprog([zeros(1,5) 1],A_in,B_in,A_eq,B_eq);
```

Problem 8

Part a: FALSE

Since M is one dimensional, then $M \cup M^{\perp}$ is the set of the points contained on the one dimensional line of M and its respective orthogonal line in $M^{\perp}$. Thus, for $x \in M$ and $y \in M^{\perp}$, a line connecting x to y is not in the set $M \cup M^{\perp}$ which implies that it is not a convex set.

Part b: TRUE

For a sequence $f_k(x)$ to be Cauchy, then $\forall \epsilon > 0$, $\exists N(\epsilon) < \infty$ s.t. $\forall n, m \geq N$, $||f_n - f_m|| < \epsilon$. Thus, assuming $n < m$, we have:

$$f_m - f_n = \begin{cases} 1 & x \in [\frac{n}{n+1}, \frac{m}{m+1}) \\ 0 & \text{otherwise} \end{cases}$$

Calculating the norm, we have:

$$N = \int_{\frac{n}{n+1}}^{\frac{m}{m+1}} |f_m - f_n| dx = \frac{m-n}{(m+1)(n+1)}$$

Taking the limit as $n, m \to \infty$, we have:

$$\lim_{n,m \to \infty} \frac{m-n}{(m+1)(n+1)} = \frac{1}{n} - \frac{1}{m} = 0 < \epsilon \implies f_k(x) \text{ is Cauchy}$$

Part c: FALSE

Method 1:

A counter-example is playing around with the values of the matrix. Indeed, switching the elements of the diagonal of the matrix A^* will yield the same results. Thus, for some $\tilde{A}^*$:

$$\tilde{A}^* = \begin{bmatrix} 0 & 3 \\ 3 & 1 \end{bmatrix} \neq A^*$$

We have:

$$\sqrt{A^{*T} A^*} = \sqrt{\tilde{A}^{*T} \tilde{A}^*} = \sqrt{19}$$

And the constraints are still satisfied **Method 2:**
Let M be the subspace spanned by:

$$M := \text{span}(\{Y_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, Y_2 = \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix}\})$$

Using the projection theorem, we can find the solution to the following minimization problem:

$$\begin{aligned} A^* = \arg\min \quad & \sqrt{\langle A, A \rangle} \\ \text{s.t.} \quad & \langle A, Y_1 \rangle = 4 \\ & \langle A, Y_2 \rangle = 2 \end{aligned}$$

Constructing the Gram matrix, we obtain:

$$G = \begin{bmatrix} \langle Y_1, Y_1 \rangle & \langle Y_1, Y_2 \rangle \\ \langle Y_2, Y_1 \rangle & \langle Y_2, Y_2 \rangle \end{bmatrix} = \begin{bmatrix} 3 & -2 \\ -2 & 3 \end{bmatrix}$$

$$\beta = \begin{bmatrix} \langle A, Y_1 \rangle \\ \langle A, Y_2 \rangle \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

Thus:

$$\alpha = G^{-1}\beta = \begin{bmatrix} 3.2 \\ 2.8 \end{bmatrix}$$

Finally:

$$A^* = 3.2Y_1 + 2.8Y_2 = \begin{bmatrix} 0.4 & 3.2 \\ 2.8 & 0.4 \end{bmatrix} \neq \begin{bmatrix} 1 & 3 \\ 3 & 0 \end{bmatrix}$$

The MATLAB code used to solve this problem is displayed below:

```
%% FINAL PROBLEM 8
clear all
close all
clc
%% Initialize
Y_1 = [1 1;0 1];
Y_2 = [-1 0;1 -1];
Y_1_A = 4;
Y_2_A = 2;
%% Part c Method 1
A_s = [1 3;3 0];
A_s2 = [0 3;3 1];
sqrt(trace(A_s'*A_s)) - sqrt(trace(A_s2'*A_s2))
sqrt(trace(A_s'*Y_1)) - sqrt(trace(A_s2'*Y_1))
sqrt(trace(A_s'*Y_2)) - sqrt(trace(A_s2'*Y_2))
%% Part c Method 2
G = [trace(Y_1'*Y_1) trace(Y_1'*Y_2); trace(Y_2'*Y_1) trace(Y_2'*Y_2)];
Beta = [Y_1_A; Y_2_A];
alpha = inv(G)'*Beta;
A_star = alpha(1)*Y_1 + alpha(2)*Y_2;
```