ECEn 671
Midterm Exam # 2, Fall 2018
Due at the Beginning of Class, November 27, 2018

Name: ______________________________

Instructions:

- This exam is open book, open notes, open computer, open brain, but not open neighbor.
- You will need Matlab / Python to complete the exam.
- The exam does not have a time limit.
- If you have questions about the exam, send me an email and I will reply to the entire class.
- Be neat and thorough. Make your solutions clear, logical, and easy to follow.
- Print this page and attach it to your solutions.
- The due date will not be extended for any reason.

Problem 1	________/ 25
Problem 2	________/ 15
Problem 3	________/ 25
Problem 4	________/ 35
Total	________/ 100

I certify that the solutions to this exam represents my own work, and that I did not consult with any other individual about the exam, or seek answers to the exam on the internet or other sources.

____
Signature

1. **(25 pts)** Consider the discrete time state space equations

$$\begin{aligned} x_{k+1} &= Ax_k \\ y_k &= Cx_k, \end{aligned}$$

with initial condition x_0, where $x_k \in \mathbb{R}^n$ and y_k is a scalar, i.e., $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{1\times n}$.

   (a) **(5 pts)** Show that $y_k = CA^k x_0$.

   (b) **(10 pts)** Consider the linear operator $\mathcal{A} : \mathbb{R}^n \to \ell_2$ where

$$\mathcal{A}[x_0] = (Cx_0, CAx_0, CA^2x_0, CA^3x_0, \dots).$$

   Find the adjoint $\mathcal{A}^* : \ell_2 \to \mathbb{R}^n$ of $\mathcal{A}$.

   (c) **(10 pts)** Suppose that you collect an infinite sequence of data $e_k = y_k - Cx_k = y_k - CA^k x_0$. Find the formula for the least squares solution for the initial condition x_0 that minimizes $\sum_{k=0}^{\infty} \left\| y_k - CA^k x_0 \right\|^2$. (HINT: Use the telephone pole diagram, and assume that $\text{rank}(\mathcal{A}) = n$. HINT 2: The answer will not necessarily be computable.)

2. **(15 pts)** Suppose that $A \in \mathbb{C}^{m\times n}$ and that $\text{rank}(A) = r < \min(m, n)$. Show that the QR factorization of A can be written as

$$A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 & R_2 \\ 0_{(m-r)\times r} & 0_{(m-r)\times(n-r)} \end{bmatrix},$$

   where $Q_1 \in \mathbb{C}^{m\times r}$ and $Q_2 \in \mathbb{C}^{m\times(m-r)}$, $R_1 \in \mathbb{C}^{r\times r}$, and $R_2 \in \mathbb{C}^{r\times(n-r)}$.

   Show that $\mathcal{R}(A) = \text{span}(Q_1)$ and $\mathcal{N}(A^H) = \text{span}(Q_2)$.

3. **(25 pts)** Write a Matlab/Python function `myqr` that computes the QR factorization of a matrix $A \in \mathbb{C}^{n\times n}$ that uses only elementary Matlab/Python operations like multiply, add, transpose, etc. Generate 10 random complex matrices and check that Q is unitary, R is upper triangular, and $QR = A$. Compare `myqr` to the Matlab function `qr`. Do you get the same results? Why or why not? Is QR factorization unique?

4. **(35 pts)** Consider the linear time-invariant system that is described by the ordinary differential equation

$$\ddot{y} + a_1\dot{y} + a_0 y = b_0 u, \tag{1}$$

   where $y(t)$ is the output, $u(t)$ is the input, and a_1, a_0, and b_0 are constants. In this problem we will consider the problem of system identification which is to determine the system coefficients a_1, a_0, and b_0 given known input-output data.

   (a) **(5 pts)** Convert the ordinary differential equation (1) to a difference equation by but using the approximation

$$\begin{aligned} \dot{z}(t) &\approx \frac{z(t) - z(t - T_s)}{T_s} \\ \ddot{z}(t) &\approx \frac{z(t) - 2z(t - T_s) + z(t - 2T_s)}{T_s^2}, \end{aligned}$$

   and using the notation $z[k] \stackrel{\triangle}{=} z(kT_s)$.

   (b) **(5 pts)** Suppose that you collect N input-output samples, i.e., $(y[n], u[n])$, $n = 0, 1, \dots, N-1$, where $N > 4$. Find the equations that show how to use batch least squares to find the estimated coefficients $\hat{a}_1$, $\hat{a}_0$, and $\hat{b}_0$ so that the squared error $\sum_{n=0}^{N-1}(y[k] - \hat{y}[n])^2$ is minimized, where $\hat{y}$ satisfies

$$\ddot{\hat{y}} + \hat{a}_1\dot{\hat{y}} + \hat{a}_0\hat{y} = \hat{b}_0 u.$$

   (c) **(10 pts)** Derive the equations for a recursive least squares algorithm that solves the same problem as item (b).

(d) **(10 pts)** Implement the system and the recursive least squares system identification scheme in Matlab/Python using $T_s = 0.01$ and the real coefficients of $a_1 = 1.0$, $a_0 = 2.0$, $b_0 = 3.0$. Use the input

$$u(t) = \sin(0.01t) + \sin(0.1t) + \sin(t) + \sin(10t)$$

and initialize the estimated parameters at $\hat{a}_1 = \hat{a}_0 = \hat{b}_0 = 0$. Plot the estimated coefficients verses the real coefficients.

(e) **(5 pts)** Compare the results of batch least squares and recursive least squares.