Introduction to Linear Dynamical Systems EE263, Summer 2020
August 14-15 2020

Final Exam

This is a 12 hour take-home exam with 4 problems. Please turn it in on Gradescope 12 hours after it is emailed to you.

• You may use any books, notes, or computer programs (e.g., matlab), but you may not discuss the exam with others until August 15, after everyone has taken the exam. The only exception is that you can ask the course staff for clarification, by emailing to the staff email address {gorish, ntragus} @stanford.edu. We’ve tried pretty hard to make the exam unambiguous and clear, so we’re unlikely to say much. Please do not post any exam related questions on Piazza.

• Since you have 12 hours, we expect your solutions to be legible, neat, and clear. Do not hand in your rough notes, and please try to simplify your solutions as much as you can. We will deduct points from solutions that are technically correct, but much more complicated than they need to be.

• Please check your email and canvas a few times during the exam, just in case we need to send out a clarification or other announcement. It’s unlikely we’ll need to do this, but you never know.

• Assemble your solutions to the problems in order, i.e., problem 1, problem 2, problem 3, problem 4. Start each problem on a new page.

• Please make a copy of your exam before handing it in. We have never lost one, but it might occur.

• If a problem asks for some specific answers, make sure they are obvious in your solutions. You might put a box around the answers, so they stand out from the surrounding discussion, justification, plots, etc.

• When a problem involves some computation (say, using matlab), we do not want just the final answers. We want a clear discussion and justification of exactly what you did, the matlab source code that produces the result, and the final numerical result. Be sure to show us your verification that your computed solution satisfies whatever properties it is supposed to, at least up to numerical precision. For example, if you compute a vector $x$ that is supposed to satisfy $Ax = b$ (say), show us the matlab code that checks this, and the result. (This might be done by the matlab code $\text{norm}(A*x-b)$; be sure to show us the result, which should be very small.) \textit{We will not check your numerical solutions for you, in cases where there is more than one solution.}
• In the portion of your solutions where you explain the mathematical approach, you cannot refer to matlab operators, such as the backslash operator. (You can, of course, refer to inverses of matrices, or any other standard mathematical construct.)

• Some of the problems are described in a practical setting, such as Robotics or Mechanical systems. You do not need to understand anything about the application area to solve these problems. We’ve taken special care to make sure all the information and math needed to solve the problem is given in the problem description.

• The zip file for the datasets required for the exam have been emailed to you alongside the exam pdf.

• Please respect the honor code. Although we encourage you to work on homework assignments in small groups, you cannot discuss the final exam with anyone, with the exception of EE263 course staff, until August 15 when everyone has taken it and the solutions are posted online.

• Finally, a few hints:
  – Problems may be easier (or harder) than they might at first appear.
  – None of the problems require long calculations or any serious programming.
1. And away we go... [10 marks]

(a) (3 marks) $AA^T$ and $A^TA$ have the same eigenvalues except possible zeros? Prove or give a counter example.

(b) (3 marks) If $A \in \mathbb{R}^{n \times m}$ is full rank and $m > n$ does $(AA^T)^{-1}$ exist? Prove or give a counter example.

(c) (4 marks) Let $a, b \in \mathbb{R}^n$ and $H \in S_+$ where $S_+$ is the set of positive semidefinite matrices, that is $x^T H x \geq 0 \ \forall x \in \mathbb{R}^n$. Show that if $Ha = b$ and $Hb = a$ then $a = b$. Hint: use a norm
2. They grow up so fast [30 marks].

Space. The final frontier. Where there is no food in the cold vacuum.

You are the chief mad scientist on a new space colony charged with growing a new type of food. We have \( n \) dishes where we can grow the food (in this case a colony of cells). We observe all the dishes for \( K \) days. Let \( x(k) \in \mathbb{R}^{n+1} \) for \( k = 1, \ldots, K \) be the vector of cell populations at the start of the day where the first \( n \) elements are the populations of each dish and the last element of \( x \) represents a cellular reservoir of frozen cells. Although populations are normally positive values, for this problem we will let each element of \( x \) be either positive or negative, meaning that a negative population is okay. On day 1 we start with initial non-zero populations in each dish and the reservoir given by \( x(1) \in \mathbb{R}^{n+1} \). Each day we have two events that affect our colony.

First, cells grow by a certain percent. Let \( \alpha_i \in \mathbb{R}_+ \) be the daily growth rate associated with dish \( i \). Simply put, at the end of day \( k \) the colony population in dish \( i \) is \( \alpha_i x_i(k) \). Since the cells in the reservoir are frozen they do not grow thus the end-of-day population of \( x_{n+1}(k) \) is unchanged by the growth step.

Next, we transfer (thaw or freeze) the cells from or to the reservoir. We do not wish to move cells directly from one dish to another as that might contaminate the growth material in each dish. Suppose \( u_i \in \mathbb{R} \) is the amount of cells from dish \( i \) we thaw from (or freeze and move to) the reservoir, note that we must remove or add that amount from the reservoir, depending on if \( u_i \) is positive (thaw) or negative (freeze). If we wanted to move cells between dishes, it would take 2 days because they would have to be frozen first (i.e moved to the reservoir).

(a) (3 marks) Formulate the growth and transfer steps as a linear dynamical system

\[
x(k + 1) = Ax(k) + Bu(k),
\]

where the vector \( u(k) \) should have dimension \( n \).

(b) (4 marks) Transferring cells to and from the reservoir can damage them, causing a loss in cell fitness. Since each of the dishes are different growth materials, they "shock" the cells different amounts. Let

\[
\rho_i u_i(k)^2
\]

model the lost cellular fitness incurred due to transferring \( u_i \) cells to or from dish \( i \) at day \( k \), the cells receive the same damage regardless of being added or removed. The number cells in each dish also contributes to the overall cellular fitness of the colony as the total number of cells times a constant, the health is

\[
\gamma 1^T x.
\]

Thus the overall fitness of the colony at start of the day \( K \) is the health minus the total damage.
\[ J(K) = \gamma x(K)^T - \sum_{k=1}^{K-1} \sum_{i=1}^{n} \rho_i u_i(k)^2 \]

Represent the overall fitness of the colony at start of the day \( K \), \( J(K) \) as a function of cell populations at day 1 \( x(1) \) and the sequence of transfers \( U \in \mathbb{R}^{n \times (K-1) \times 1} \) given by

\[
U = \begin{bmatrix} u(1) \\ \vdots \\ u(K-1) \end{bmatrix}
\]

(c) (5 marks) Derive the closed form solution for the optimal sequence of transfers \( U^* \) such that the cellular fitness at the start of day \( K \), \( J(K) \) is maximized.

(d) (8 marks) Apply your method to data below

\[
\begin{align*}
n &= 5 \\
gamma &= 2 \\
alpha &= [1.02, 1.03, 1.04, 1.05, 1.01] \\
K &= 21 \\
rho &= [.2, .3, .4, .5, .7] \\
x1 &= [10, 15, 20, 5, 10, 0]
\end{align*}
\]

Report the optimal fitness value and a plot overtime of the cell population in each dish as well as the reservoir. What do you observe?

(e) (10 marks) Now consider another situation. Having cells in the dishes at the start of day \( K \) is useless as we plan to ship the cells to our planetary base, thus we require \( x_1(K), \ldots, x_n(K) = 0 \). Formulate another optimization problem that again maximizes cell fitness subject to this new requirement. Again, report the new optimal fitness value and a plot overtime of the cell population in each dish as well as the reservoir. (Note that now this has becomes a constraint optimization problem.)
3. **Attack position of Fighter Jets** [30 marks].

Consider you are commanding a fleet of MiG-17 fighter jets, labeled $1, \ldots, n$, which move along a line with (scalar) positions $y_1, \ldots, y_n$. We let $v_1, \ldots, v_n$ denote the velocities of the jets, and $u_1, \ldots, u_n$ the net forces applied to the jets. The motions of these fighter jets are governed by the equations

$$
\dot{y}_i = v_i, \quad \dot{v}_i = u_i - v_i.
$$

(Here we take mass of each jet to be one, and include a damping term in the equations.)

We assume that $y_1(0) < \cdots < y_n(0)$, i.e., the jets start out with jet 1 in the leftmost position, followed by the jet 2 to its right, and so on, with jet $n$ in the rightmost position. Your fleet has been called to action to bomb an enemy base. In order to achieve this, you must arrange your vehicles in the following attack configuration after a sufficiently large time:

$$
\lim_{t \to \infty} y_i(t) = i, \quad \lim_{t \to \infty} v_i(t) = 0, \quad i = 1, \ldots, n,
$$

i.e., first jet at position 1, with unit spacing between adjacent jets, and all stationary. We call this configuration *aligned*, and the goal is to drive the jets to this configuration, i.e., to align the jets. We define the spacing between jet $i$ and $i+1$ as $s_i(t) = y_{i+1}(t) - y_i(t)$, for $i = 1, \ldots, n - 1$. (When the jets are aligned, these spacings are all one.)

Back at the command center, you have at your disposal the following three control schemes to achieve the given fleet configuration.

- **Right looking control** is based on the spacing to the jet to the right. We use the control law

$$
u_i(t) = s_i(t) - 1, \quad i = 1, \ldots, n - 1,
$$

for jets $i = 1, \ldots, n - 1$. In other words, we apply a force on jet $i$ proportional to its spacing error with respect to the jet to the right (i.e., jet $i+1$). The rightmost jet uses the control law

$$
u_n(t) = -(y_n(t) - n),
$$

which applies a force proportional to its position error, in the opposite direction. This control law has the advantage that only the rightmost jet needs an absolute measurement sensor; the others only need a measurement of the distance to their righthand neighbor.

- **Left and right looking control** adjusts the input force based on the spacing errors to the jet to the left and the jet to the right:

$$
u_i(t) = \frac{s_i(t) - 1}{2} - \frac{s_{i-1}(t) - 1}{2}, \quad i = 2, \ldots, n - 1,
$$

The rightmost jet uses the same absolute method as in right looking control, i.e.,

$$
u_n(t) = -(y_n(t) - n),
$$
and the first jet, which has no jet to its left, uses a right looking control scheme,

\[ u_1(t) = s_1(t) - 1. \]

This scheme requires jet \( n \) to have an absolute position sensor, but the other jets only need to measure the distance to their neighbors.

- **Independent alignment** is based on each jet independently adjusting its position with respect to its required position:

\[ u_i(t) = -(y_i(t) - i), \quad i = 1, \ldots, n. \]

This scheme requires all jets to have absolute position sensors.

In the questions below, we consider the case where you have \( n = 7 \) jets in your squadron.

(a) (4*3+3 marks) Which of the three schemes work? By ‘work’ we mean that the jets converge to the alignment configuration, no matter what the initial positions and velocities are. Among the schemes that do work, which one gives the fastest asymptotic convergence to alignment? (If there is a tie between two or three schemes, say so.) In this part of the problem you can ignore the issue of collisions of the jets, i.e., spacings that pass through zero.

Hint: Try expressing each of the control schemes as a Linear Dynamical System.

(b) (5*3 marks) **Collisions.** In this problem we analyze jet collisions, which occur when any spacing between jets is equal to zero. (For example, \( s_3(5.7) = 0 \) means that jets 3 and 4 collide at \( t = 5.7 \).) We take the particular starting configuration

\[ y = (0, 1, 2, 4, 6, 7, 8), \quad v = (0, 0, 0, 0, 0, 0), \]

which corresponds to the jets with zero initial velocity, but not in the aligned positions. For each of the three schemes above (whether or not they work), determine if a collision occurs. If a collision does occur, find the earliest collision, giving the time and the jets involved in that collision. (For example, ‘jets 3 and 4 collide at \( t = 7.7 \).’) If there is a tie, i.e., two pairs of jets collide at the same time, say so. If the jets do not collide, find the point of closest approach, i.e., the minimum spacing that occurs, between any pair of jets, for \( t \geq 0 \). (Give the time, the jets involved, and the minimum spacing.) If there is a tie, i.e., two or more pairs of jets have the same distance of closest approach, say so. Be sure to give us times of collisions or closest approach with an absolute precision of at least 0.1.
4. UN General Assembly voting [30 marks].

CNN News network has decided to analyse the voting patterns of countries in the UN General Assembly which comprises of both capitalist and communist nations along with socialist nations which essentially follow a stance between the two hard-core ideologies. As the journalist assigned to this study, you collect the results of votes on the key economic issues raised in the Assembly for the year 2018-19. This data is then compiled by your team into the file `un_voting_patterns.m`, which defines the following variables.

- **countries**, a $102 \times 2$ cell array. There are 102 rows because Tanzania exited during its term, and was replaced by Belgium. Also Albania, has been switching its stance through the year, recording both pro-capitalism and socialists votes
  - `countries{i,1}` is the name of the $i$th country
  - `countries{i,2}` is the economy type of the $i$th country
- **votes**, a $102 \times 633$ matrix where `votes(i,j)` is the vote cast by country $i$ in vote $j$ (+1 if country $i$ voted “Yea” on vote $j$, −1 if country $i$ voted “Nay” on vote $j$, and 0 if country $i$ did not participate in vote $j$)

(a) (5 marks) Make a stem plot of the singular values of `votes`. How many significant singular values are there? The fraction of the variation in the voting data associated with the first two singular values is

\[ f_2 = \frac{\sigma_1^2 + \sigma_2^2}{\sum_{i=1}^{r} \sigma_i^2}. \]

What is $f_2$ for the Assembly voting data?

(b) (9 marks) Missing votes will complicate our subsequent analysis, so we use a low-rank model of the voting matrix to guess the missing votes. Initialize $A$ to the observed voting matrix; repeat the following steps until convergence:

- replace $A$ with its best rank-two approximation;
- replace the entries of $A$ corresponding to the known votes with their true values;
- replace the entries of $A$ corresponding to the unknown votes by setting non-negative entries to +1, and negative entries to −1.

Report the numbers of +1’s and −1’s in your final matrix $A$.

(c) (3*2+2 marks) Let $A = U \Sigma V^T$ be the singular-value decomposition of $A$. We want to find interpretations for the first two left singular vectors, $u_1$ and $u_2$.

i. The function `spatial_plot`, which is available in the data zip file, can be used to generate color-coded scatter plots. Use this function to make a scatter plot of $(u_1)_i$ and $(u_2)_i$ where Capitalists, Communists and Socialists are represented by red, blue, and green markers, respectively. You can make such a plot using the command
spatial_plot(u1, u2, z, 3, eye(3));

where $u_1$ and $u_2$ are the first two left singular vectors of $A$, and $z$ is a vector where $z(i)$ is 0 if country $i$ is a Capitalist economy, 1 if country $i$ is a Socialist economy, and 2 if country $i$ is a Communist economy.

ii. Make a scatter plot showing the fraction of votes in which each country voted with the majority of the other countries. You can make such a plot using the command

spatial_plot(u1, u2, z, 10);

where $u_1$ and $u_2$ are the first two left singular vectors of $A$, and $z$ is a vector where $z(i)$ is the fraction of votes $j$ for which

$$A_{ij} = sgn\left(\sum_{i=1}^{102} A_{ij}\right).$$

(By default, spatial_plot uses the cool color map, which is pink for large values, and blue for small values. Also, sgn refers to the sign function.)

Based on these plots, give intuitive interpretations of $(u_1)_i$ and $(u_2)_i$.

(d) (3*2+2 marks) We can also find interpretations for the first two right singular vectors.

i. Make a scatter plot of $(v_1)_j$ and $(v_2)_j$ where the colors indicate the total support received by vote $j$. In particular, the color should correspond to

$$z_j = \sum_{i=1}^{102} A_{ij}.$$ 

ii. Make a scatter plot of $(v_1)_j$ and $(v_2)_j$ where the colors indicate the partisan support received by vote $j$. In particular, let $Cap \subset \{1, \ldots, 102\}$ be the set of Capitalists, and let $Com \subset \{1, \ldots, 102\}$ be the set of Communists. The color in your plot should correspond to

$$z_j = \frac{1}{|Cap|} \sum_{i \in Cap} A_{ij} - \frac{1}{|Com|} \sum_{i \in Com} A_{ij}.$$ 

Based on these plots, give intuitive interpretations of $(v_1)_j$ and $(v_2)_j$.

The data given in the above question is artificially generated and should not be taken as an indicator for the world reality in any way whatsoever.

Congratulations! You’ve finished EE263! We hope this was a great learning experience for you all and we wish you the best in all of your future endeavors.