EECS 16A Designing Information Devices and Systems I
Spring 2020 Homework 3

This homework is due Friday February 14, 2020, at 23:59.
Self-grades are due Monday February 17, 2020, at 23:59.

Submission Format
Your homework submission should consist of **one** file.

- `hw3.pdf`: A single PDF file that contains all of your answers (any handwritten answers should be scanned) as well as your IPython notebook saved as a PDF.

  If you do not attach a PDF "printout" of your IPython notebook, you will not receive credit for problems that involve coding. Make sure that your results and your plots are visible. Assign the IPython printout to the correct problem(s) on Gradescope.

- Practice problems will not be graded for homeworks, but are in scope for exams.

Submit each file to its respective assignment on Gradescope.

1. Matrix Operations Practice

$$A = \begin{bmatrix} 3 & 2 \\ 5 & 6 \\ 1 & 2 \end{bmatrix} B = \begin{bmatrix} 8 & 6 & 4 \\ 1 & 3 & 5 \end{bmatrix} C = \begin{bmatrix} 1 & 5 \\ 5 & 1 \end{bmatrix} D = \begin{bmatrix} 3 & 10 \\ 7 & 2 \end{bmatrix}$$

(a) Compute the following matrix multiplications.

i AB
ii BA
iii AD

(b) Compute the following transposes.

i A^T
ii B^T
iii C^T

(c) Show that for general matrices $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{n \times p}, and C \in \mathbb{R}^{p \times q}$

$$(AB)C = A(BC)$$

.

(d) Show that for general matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times p}$:

$$(AB)^T = B^T A^T$$

2. Mechanical Inverses

Learning Objectives: *Matrices represent linear transformations, and their inverses represent the opposite transformation. Here we practice inversion, but are also looking to develop an intuition. Visualizing the transformations might help develop this intuition.*

For each of the following matrices, state whether the inverse exists. If so, find the inverse, $\mathbf{A}^{-1}$. If not, show why no inverse exists. Solve this by hand.

For parts (a)-(d), in addition to finding the inverse (if it exists), describe how the original matrix, A, changes a vector it's applied to. For example, if $\mathbf{A}\vec{b} = \vec{c}$, then $\mathbf{A}$ could scale $\vec{b}$ by 2 to get $\vec{c}$, or $\mathbf{A}$ could reflect $\vec{b}$ across the x axis to get $\vec{c}$, etc. *Hint:* It may help to plot a few examples to recognize the pattern.

(a) $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

(b) $\mathbf{A} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$

(c) **(PRACTICE)**

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

(d) **(PRACTICE)**

$\mathbf{A} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$

Assume $\cos\theta \neq 0$. *Hint:* $\cos^2\theta + \sin^2\theta = 1$.

(e) $\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix}$

(f) $\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 2 \\ 1 & 4 & 4 \end{bmatrix}$

(g) **(PRACTICE)** $\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix}$

(h) **(PRACTICE)** $\mathbf{A} = \begin{bmatrix} -1 & 1 & -\frac{1}{2} \\ 1 & 1 & -\frac{1}{2} \\ 0 & 1 & 1 \end{bmatrix}$

(i) **(PRACTICE)**

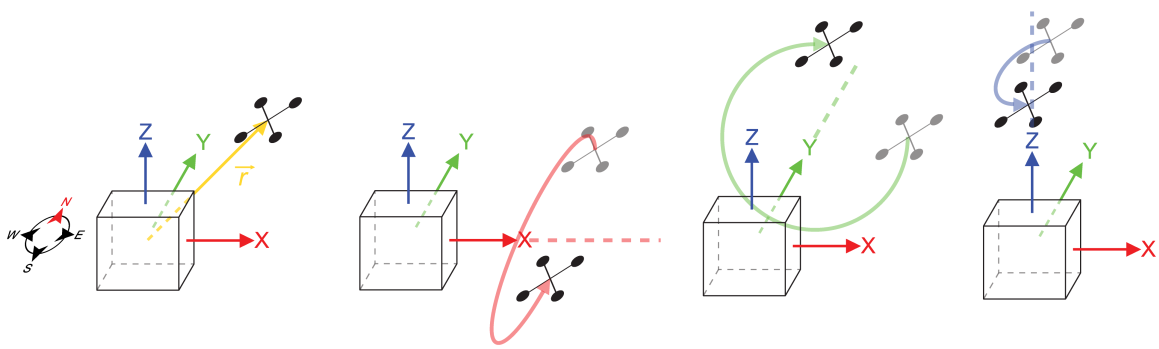
$\mathbf{A} = \begin{bmatrix} 3 & 0 & -2 & 1 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 0 & 4 \\ 1 & 0 & 0 & 1 \end{bmatrix}$

Hint 1: What do the linear (in)dependence of the rows and columns tell us about the invertibility of a matrix? Hint 2: We're reasonable people!

3. Quadcopter Transformations

Learning Objectives: *Linear algebra is often used to represent transformations in robotics. This problem introduces some of the basic uses of transformations.*

Professor Courtade and his colleagues are interested in testing a concept to establish a communication link to a quadcopter by laser. Consider a vector $\vec{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3$ representing the location of the quadcopter relative to the origin. The quadcopter is only capable of three different maneuvers relative to the origin. The maneuvers are rotations about the x, y, and z axes. For perspective, the positive x-axis points east, the positive y-axis points north, and the positive z-axis points towards the sky. The figures below illustrate the quadcopter and these maneuvers.



We can represent each of these rotations, that are linear transformations, as matrices that operate on the location vector of the quadcopter, $\vec{r}$, to position it at it's new location. The matrices $R_x(\theta)$, $R_y(\psi)$, and $R_z(\phi)$ represent rotations about the x-axis, y-axis, and z-axis, respectively. The matrices are:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}, R_y(\psi) = \begin{bmatrix} \cos\psi & 0 & -\sin\psi \\ 0 & 1 & 0 \\ \sin\psi & 0 & \cos\psi \end{bmatrix}, R_z(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

.

(a) Professor Courtade wants to make the quadcopter to rotate first by 30° about the x-axis, and then by 60° about the z-axis. Use $R_x(\theta)$, $R_y(\psi)$, and $R_z(\phi)$ to construct a matrix that performs the operations in the specified order. You may use an ipython notebook for algebra, but show in your solutions the matrices and the operations you are doing on them by hand.

(b) Professor Courtade accidentally punched in the two commands in reverse. The rotation about the z-axis occurred before the rotation about the x-axis. Use $R_x(\theta)$, $R_y(\psi)$, and $R_z(\phi)$ to construct a matrix that performs the operations that accidentally happened. You may use an ipython notebook for algebra. Write out the matrices you are multiplying, and the computed matrix.

(c) Say the quadcopter was at $\vec{r} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$. Where did Professor Courtade intend for the quadcopter to end up? Where did it actually end up? Are they the same?

(d) Say the quadcopter starts out at distance 1 from the origin, i.e. $||\vec{r}|| = 1$. Say Professor Courtade performs an arbitrary sequence of operations (unknown to you) using the three transformation matrices $R_x(\theta)$, $R_y(\psi)$, and $R_z(\phi)$. Each transformation could be used more than once. Then the quadcopter ends up at location $\vec{s}$. What is $||\vec{s}||$?

4. Properties of Pump Systems

Learning Objectives: *This problem illustrates how matrices and vectors can be used to represent linear transformations. It also foreshadows concepts covered next week in class, where we will be exploring matrix inversion. It turns out that matrix inversion is closely related to ideas of linear dependence and independence, which you can use in this problem.*

Throughout this problem, we will consider a system of reservoirs connected to each other through pumps. An example system is shown below in Figure 1, represented as a graph. Each node in the graph is marked with a letter and represents a reservoir. Each edge in the graph represents a pump which moves a fraction of the water from one reservoir to the next at every time step. The fraction of water is written on top of the edge.

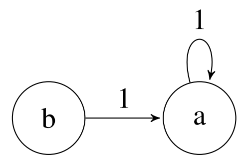


Figure 1: Pump system

(a) Consider the system of pumps shown above in Figure 1. Let $x_a[n]$ and $x_b[n]$ represent the amount of water in reservoir a and b, respectively, at time step n. Find a system of equations that represents every $x_i[n+1]$ in terms of all the different $x_i[n]$.

(b) For the system shown in Figure 1, find the associated state transition matrix. That is find the matrix $\mathbf{A}$ such that:

$$\vec{x}[n+1] = \mathbf{A}\vec{x}[n], \text{ where } \vec{x}[n] = \begin{bmatrix} x_a[n] \\ x_b[n] \end{bmatrix}$$

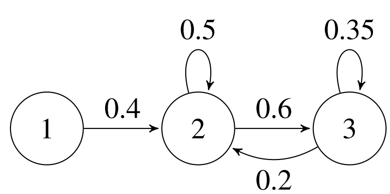
(c) Suppose that the reservoirs are initialized to the following water levels: $x_a[0] = 0.5, x_b[0] = 0.5$. In a completely alternate universe, the reservoirs are initialized to the following water levels: $x_a[0] = 0.3, x_b[0] = 0.7$. For both initial states, what are the water levels at timestep 1 ($\vec{x}[1]$)? Use your answer from part (b) to compute your solution.

(d) If you observe the reservoirs at timestep 1, can you figure out what the initial ($\vec{x}[0]$) water levels were? Why or why not?

(e) Now let us generalize what we observed. Say there is a transition matrix $\mathbf{A}$ representing a pump system. Say there exist two distinct initial state vectors $\vec{x}[0]$ and $\vec{y}[0]$ (i.e. water levels) that lead to the same state vector $\vec{x}[1]$ after $\mathbf{A}$ acts on them.You do not know which of the two initial state vectors you started in. Can you decide which initial state you started in by observing $\vec{x}[1]$? What does this say about the matrix $\mathbf{A}$?

(f) Set up the state transition matrix $\mathbf{A}$ for the system of pumps shown below. Compute the sum of the entries of the columns of the state transition matrix. Is it greater than/less than/equal to 1? Explain what this $\mathbf{A}$ matrix physically implies about the total amount of water in this system.

Note: If there is no "self-arrow/self-loop," you can interpret it as a self-loop with weight 0, i.e. no water returns..



5. Segway Tours

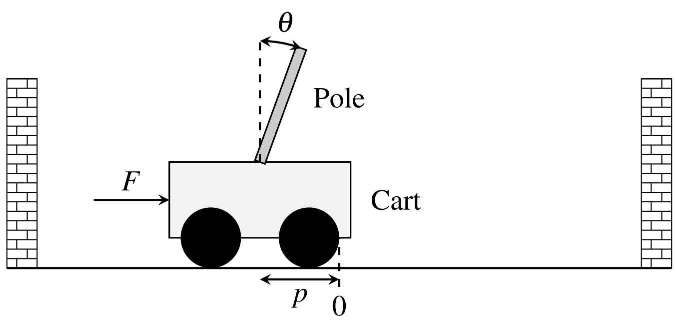
Learning Objective: *The learning objective of this problem is to see how the concept of span can be applied to control problems. If a desired state vector of a linear control problem is in a span of a particular set of vectors, then the system may be steered to reach that particular vector using the available inputs.*

Your friend has decided to start a new SF tour business, and you suggest he use segways.

A segway is essentially a stand on two wheels. He becomes intrigued by your idea and asks you how a segway works.

The segway works by applying a force (through the spinning wheels) to the base of the segway. This controls both the position on the segway and the angle of the stand. As the driver pushes on the stand, the segway tries to bring itself back to the upright position, and it can only do this by moving the base.

Is it possible for the segway to be brought upright and to a stop from any initial configuration? There is only one input (force) used to control two outputs (position and angle). You both talk to a third friend who is GSIing EE128, and she tells you that a segway can be modeled as a cart-pole system.



A cart-pole system can be fully described by its position p, velocity $\dot{p}$, angle θ, and angular velocity $\dot{\theta}$. We write this as a "state vector", $\vec{x}$:

$$\vec{x} = \begin{bmatrix} p \\ \dot{p} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

The input to this system is a scalar quantity $u[n]$ at time n that is the force applied to the cart (or base of the segway).[1]

The cart-pole system can be represented by a linear model:

$$\vec{x}[n+1] = \mathbf{A}\vec{x}[n] + \vec{b}u[n], \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{4\times 4}$ and $\vec{b} \in \mathbb{R}^{4\times 1}$.

The control allows us to move the state by $u[n]$ in direction $\vec{b}$. So, if $u[n] = 2$, we move the state by $2\vec{b}$ at time n, and so on. We can choose different controls at different times.

[1] You might note that velocity and angular velocity are derivatives of position and angle respectively. Differential equations are used to describe continuous time systems, which you will learn more about in EECS 16B. But even without these techniques, we can still approximate the solution to be a continuous time system by modeling it as a discrete time system where we take very small steps in time. We think about applying a force constantly for a given finite duration and we see how the system responds after that finite duration.

The model tells us how the the state vector will evolve over time as a function of the current state vector and control inputs.

You look at this general linear system and try to answer the following question: Starting from some initial state $\vec{x}_0$, can we reach a final desired state, $\vec{x}_f$, in N steps?

The challenge seems to be that the state is four-dimensional and keeps evolving and that we can only apply a one-dimensional (scalar) control at each time. Typically, to set the values of four variables to desired quantities, you would need four inputs. Can you do this with just one input?

We will solve this problem by walking through several steps.

(a) Express $\vec{x}[1]$ in terms of $\vec{x}[0]$ and the input $u[0]$. (*Hint:* This is easy.)

(b) Express $\vec{x}[2]$ in terms of *only* $\vec{x}[0]$ and the inputs, $u[0]$ and $u[1]$. Then express $\vec{x}[3]$ in terms of *only* $\vec{x}[0]$ and the inputs, $u[0]$, $u[1]$, and $u[2]$, and express $\vec{x}[4]$ in terms of *only* $\vec{x}[0]$ and the inputs, $u[0]$, $u[1]$, $u[2]$, and $u[3]$.

(c) Now, generalize the pattern you saw in the earlier part to write an expression for $\vec{x}[N]$ in terms of $\vec{x}[0]$ and the inputs from $u[0], \ldots, u[N-1]$.

For the next four parts of the problem, you are given the matrix $\mathbf{A}$ and the vector $\vec{b}$:

$$\mathbf{A} = \begin{bmatrix} 1 & 0.05 & -0.01 & 0 \\ 0 & 0.22 & -0.17 & -0.01 \\ 0 & 0.10 & 1.14 & 0.10 \\ 0 & 1.66 & 2.85 & 1.14 \end{bmatrix}$$

$$\vec{b} = \begin{bmatrix} 0.01 \\ 0.21 \\ -0.03 \\ -0.44 \end{bmatrix}$$

Assume the cart-pole starts in an initial state $\vec{x}[0] = \begin{bmatrix} -0.3853493 \\ 6.1032227 \\ 0.8120005 \\ -14 \end{bmatrix}$, and you want to reach the desired state $\vec{x}_f = \vec{0}$ using the control inputs $u[0], u[1], \ldots$. The state vector $\vec{x}_f = \vec{0}$ corresponds to the cart-pole (or segway) being upright and stopped at the origin. (Reaching $\vec{x}_f = \vec{0}$ in N steps means that, given that we start at $\vec{x}[0]$, we can find control inputs, such that we get $\vec{x}[N]$, the state vector at the Nth time step, equal to $\vec{x}_f$.)

Note: You may use IPython to solve the next three parts of the problem. You may use the function we provided (`gauss_elim(matrix)`) to help you find the upper triangular form of matrices. An example of Gaussian Elimination using this code is provide in the iPython notebook. You may also use the function (`np.linalg.solve`) to solve the equations.

(d) Can you reach $\vec{x}_f$ in *two* time steps? (*Hint: Express* $\vec{x}[2] - \mathbf{A}^2\vec{x}[0]$ *in terms of the inputs* $u[0]$ *and* $u[1]$. *Then determine if the system of equations can be solved to obtain* $u[0]$ *and* $u[1]$. *If we obtain valid solutions for* $u[0]$ *and* $u[1]$, *then we can say we will reach* $\vec{x}_f$ *in two time steps.*)

(e) Can you reach $\vec{x}_f$ in *three* time steps?

(f) Can you reach $\vec{x}_f$ in *four* time steps?

(g) If you have found that you can get to the final state in 4 time steps, find the required correct control inputs using IPython and verify the answer by entering these control inputs into the *Plug in your controller* section of the code in the IPython notebook. The code has been written to simulate this system. *Suggestion: See what happens if you enter all four control inputs equal to 0. This gives you an idea of how the system naturally evolves!*

(h) Let us reflect on what we just did. Recall the system we have:

$$\vec{x}[n+1] = \mathbf{A}\vec{x}[n] + \vec{b}u[n].$$

The control allows us to move the state by $u[n]$ in direction $\vec{b}$. We know from part (c) that:

$$\vec{x}[2] = \mathbf{A}^2\vec{x}[0] + \mathbf{A}\vec{b}u[0] + \vec{b}u[1].$$

What are the vectors that the combination of controls $u[0]$ and $u[1]$ allow us to move in? Can you express the possible positions you can arrive at in two time steps using the span of these vectors?

(i) Can you generalize the idea in the previous part? Express the positions you can reach in N timesteps as a span of some vectors.

(j) **(Challenge, optional)** Now say you wanted to reach anywhere in $\mathbb{R}^4$, i.e. $\vec{x}_f$ is an unspecified vector in $\mathbb{R}^4$. Under what conditions can you guarantee that you can "reach" $\vec{x}_f$ from any $\vec{x}_0$?
Wouldn't this be cool?

6. Homework Process and Study Group

Who else did you work with on this homework? List names and student ID's. (In case of homework party, you can also just describe the group.) How did you work on this homework?

