EECS 16A Designing Information Devices and Systems I
Summer 2020 Homework 1B

This homework is due Monday July 6, 2020, at 23:59 PT.
Self-grades are due Wednesday July 8, 2020, at 23:59 PT.

Submission Format
Your homework submission should consist of 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.

Please attach a PDF of your Jupyter notebook for all the problems that involve coding. Make sure the results of your plots (if any) are visible. Please assign the PDF of the notebook to the correct problems on Gradescope — we will be unable to grade the problems without this assignment or submission.

Homework Learning Goals: *The objective of this homework is to introduce the concept of linear transformation, commutativity and invertibility.*

1. 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) and (b), 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.

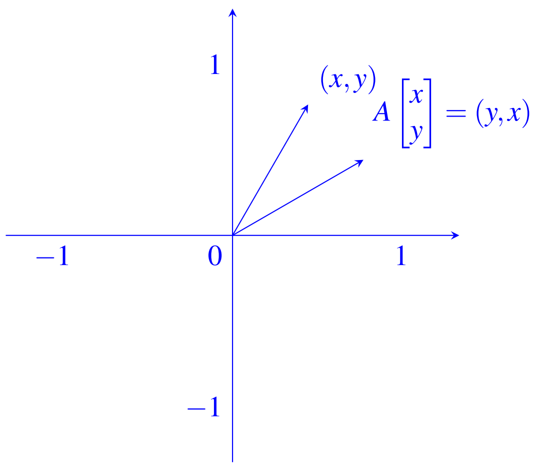
(a) $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

Solution:

$$\left[\begin{array}{cc|cc} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{array}\right] \sim \left[\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right]$$

The inverse does exist.

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



The original matrix $\mathbf{A}$ flips the x and y components of the vector. Any correct equivalent sequence of operations warrants full credit. Notice how the inverse does the exact same thing–that is, it switches the x and y components of the vector it's applied to. This makes sense–switching x and y twice on a vector $\begin{bmatrix}x\\y\end{bmatrix}$ gives us $\begin{bmatrix}x\\y\end{bmatrix}$.

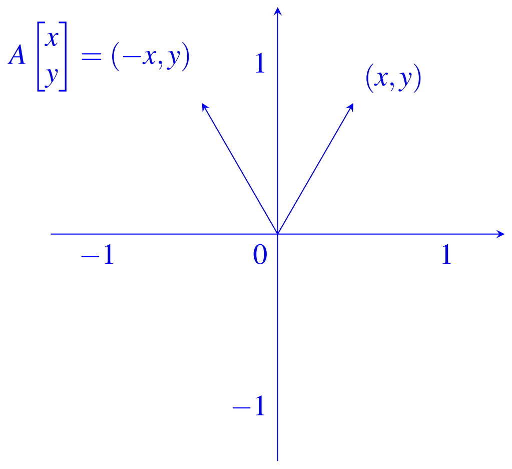
(b) $\mathbf{A}=\begin{bmatrix}-1 & 0\\0 & 1\end{bmatrix}$

Solution:

$$\left[\begin{array}{cc|cc}-1 & 0 & 1 & 0\\0 & 1 & 0 & 1\end{array}\right]\sim\left[\begin{array}{cc|cc}1 & 0 & -1 & 0\\0 & 1 & 0 & 1\end{array}\right]$$

The inverse does exist.

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



The original matrix $\mathbf{A}$ reflects the vector across the y-axis, i.e. it multiplies the vector's x-component by a factor of -1. Reflecting the vector across the y-axis again with $\mathbf{A}^{-1}=\begin{bmatrix}-1 & 0\\0 & 1\end{bmatrix}$ will give you the original vector.

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

Solution:

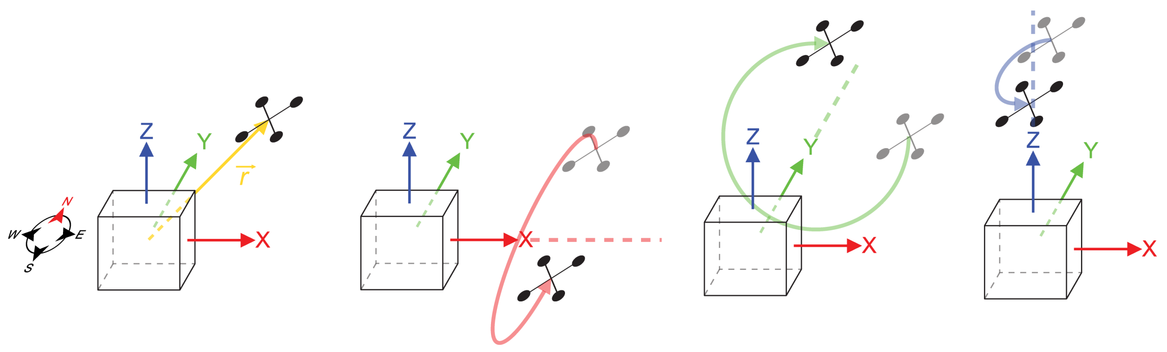
$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 2 & 0 & 1 & 0 \\ 1 & 4 & 4 & 0 & 0 & 1 \end{array}\right] \sim \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 2 & 0 & 1 & 0 \\ 0 & 4 & 4 & -1 & 0 & 1 \end{array}\right] \sim \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -2 & 1 \end{array}\right]$$

The inverse does not exist because the last equation is inconsistent. An alternative reason is that second and third column are equal, i.e., they are linearly dependent. Since the columns of the matrix are linearly dependent, the inverse does not exist.

2. 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 Kuo and her 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 Kuo 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.

Solution: First, we must apply $R_x(\theta)$ to $\vec{r}$, then $R_z(\phi)$. We are told the angle rotated about the x-axis, θ, is $30°$. Likewise, for the z-axis, $\phi = 60°$. So the matrix is:

$$R_z(60°)R_x(30°) = \begin{bmatrix} \cos 60° & -\sin 60° & 0 \\ \sin 60° & \cos 60° & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 30° & -\sin 30° \\ 0 & \sin 30° & \cos 30° \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{2}\cdot 1+0+0 & 0-\frac{\sqrt{3}}{2}\cdot\frac{\sqrt{3}}{2}+0 & 0+-\frac{\sqrt{3}}{2}\cdot(-\frac{1}{2})+0 \\ \frac{\sqrt{3}}{2}\cdot 1+0+0 & 0+\frac{1}{2}\cdot\frac{\sqrt{3}}{2}+0 & 0+\frac{1}{2}\cdot(-\frac{1}{2})+0 \\ 0+0+0 & 0+0+1\cdot\frac{1}{2} & 0+0+1\cdot\frac{\sqrt{3}}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{3}{4} & \frac{\sqrt{3}}{4} \\ \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{4} & -\frac{1}{4} \\ 0 & \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}$$

(b) Professor Kuo 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.

Solution: Now, we compute the product with $R_z(\phi)$ and $R_x(\theta)$ commuted.

$$R_x(30°)R_z(60°) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 30° & -\sin 30° \\ 0 & \sin 30° & \cos 30° \end{bmatrix} \begin{bmatrix} \cos 60° & -\sin 60° & 0 \\ \sin 60° & \cos 60° & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1\cdot\frac{1}{2}+0+0 & 1\cdot(-\frac{\sqrt{3}}{2})+0+0 & 0+0+0 \\ 0+\frac{\sqrt{3}}{2}\cdot\frac{\sqrt{3}}{2}+0 & 0+\frac{\sqrt{3}}{2}\cdot\frac{1}{2}+0 & 0+0-\frac{1}{2}\cdot 1 \\ 0+\frac{1}{2}\cdot\frac{\sqrt{3}}{2}+0 & 0+\frac{1}{2}\cdot\frac{1}{2}+0 & 0+0+\frac{\sqrt{3}}{2}\cdot 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{3}{4} & \frac{\sqrt{3}}{4} & -\frac{1}{2} \\ \frac{\sqrt{3}}{4} & \frac{1}{4} & \frac{\sqrt{3}}{2} \end{bmatrix}$$

(c) Say the quadcopter was at $\vec{r} = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}$. Where did Professor Kuo intend for the quadcopter to end up? Where did it actually end up? Are they the same?

Solution: Professor Kuo's intended location is given by:

$$R_z(60°)R_x(30°)\vec{r} = \begin{bmatrix} \frac{1}{2} & -\frac{3}{4} & \frac{\sqrt{3}}{4} \\ \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{4} & -\frac{1}{4} \\ 0 & \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} \frac{-1+2\sqrt{3}}{4} \\ \frac{-2+3\sqrt{3}}{4} \\ \frac{1+2\sqrt{3}}{2} \end{bmatrix}$$

Professor Kuo's actual location is given by:

$$R_x(30°)R_z(60°)\vec{r} = \begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{3}{4} & \frac{\sqrt{3}}{4} & -\frac{1}{2} \\ \frac{\sqrt{3}}{4} & \frac{1}{4} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} \frac{1-\sqrt{3}}{2} \\ \frac{-1+\sqrt{3}}{4} \\ \frac{1+5\sqrt{3}}{4} \end{bmatrix}$$

We can see that $\begin{bmatrix} \frac{-1+2\sqrt{3}}{4} \\ \frac{-2+3\sqrt{3}}{4} \\ \frac{1+2\sqrt{3}}{2} \end{bmatrix} \neq \begin{bmatrix} \frac{1-\sqrt{3}}{2} \\ \frac{-1+\sqrt{3}}{4} \\ \frac{1+5\sqrt{3}}{4} \end{bmatrix}$, and that they are not the same.

3. Properties of Pump Systems

Learning Objectives: *This problem illustrates how matrices and vectors can be used to represent linear transformations.*

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.

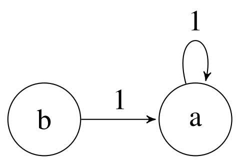


Figure 1: Pump system

(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 $x_i[n]$.

Solution:

$$x_a[n+1] = x_a[n] + x_b[n]$$
$$x_b[n+1] = 0$$

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

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

Solution:

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

(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.

Solution:

$$\vec{x}[1] = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_a[0] \\ x_b[0] \end{bmatrix} = \begin{bmatrix} x_a[1] \\ x_b[1] \end{bmatrix}$$

$$\vec{x}[1] = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \vec{x}[1] = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

(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?

Solution:

No, at timestep 1, configuration 1 and 2 are indistinguishable. If you examine the transition matrix, it's columns are linearly dependent, thus there is not a unique solution to $\mathbf{A}\vec{x}[0] = \vec{x}[1]$ and it is impossible to ascertain how the water was distributed over the states at time $n = 0$ from the states at time $n = 1$.

(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 $\mathbf{A}$?

Solution:

We are told that two different initial states, $\vec{x}[0]$ and $\vec{y}[0]$, lead to the same resulting state $\vec{x}[1]$.

$$\mathbf{A}\vec{x}[0] = \vec{x}[1] \qquad \mathbf{A}\vec{y}[0] = \vec{x}[1]$$

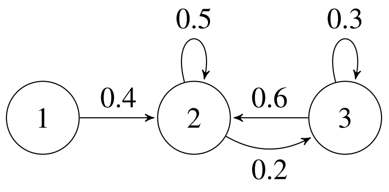
Now, consider the system:

$$\mathbf{A}\vec{x} = \vec{x}[1].$$

We know that this system has two distinct solutions. Hence, from the theorem we proved in class we can say that the columns of $\mathbf{A}$ are linearly dependent. We will see soon that this implies that the matrix $\mathbf{A}$ is not invertible — but you do not need to have mentioned invertibility to give yourself full credit.

(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..



Solution:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0.4 & 0.5 & 0.6 \\ 0 & 0.2 & 0.3 \end{bmatrix}$$

Note that the entries in the columns do *not* sum to one. This is physically interpreted as a "leak" – i.e., the total amount of water is not conserved.

4. Image Stitching

Learning Objective: *This problem is similar to one that students might experience in an upper division computer vision course. Our goal is to give students a flavor of the power of tools from fundamental linear algebra and their wide range of applications.*

Often, when people take pictures of a large object, they are constrained by the field of vision of the camera. This means that they have two options to capture the entire object:

- Stand as far away as they need to include the entire object in the camera's field of view (clearly, we do not want to do this as it reduces the amount of detail in the image)
- (This is more exciting) Take several pictures of different parts of the object and stitch them together like a jigsaw puzzle.

We are going to explore the second option in this problem. Daniel, who is a professional photographer, wants to construct an image by using "image stitching". Unfortunately, Daniel took some of the pictures from different angles as well as from different positions and distances from the object. While processing these pictures, Daniel lost information about the positions and orientations from which the pictures were taken. Luckily, you and your friend Marcela, with your wealth of newly acquired knowledge about vectors and matrices, can help him!

You and Marcela are designing an iPhone app that stitches photographs together into one larger image. Marcela has already written an algorithm that finds common points in overlapping images. **It's your job to figure out how to stitch the images together using Marcela's common points to reconstruct the larger image.**

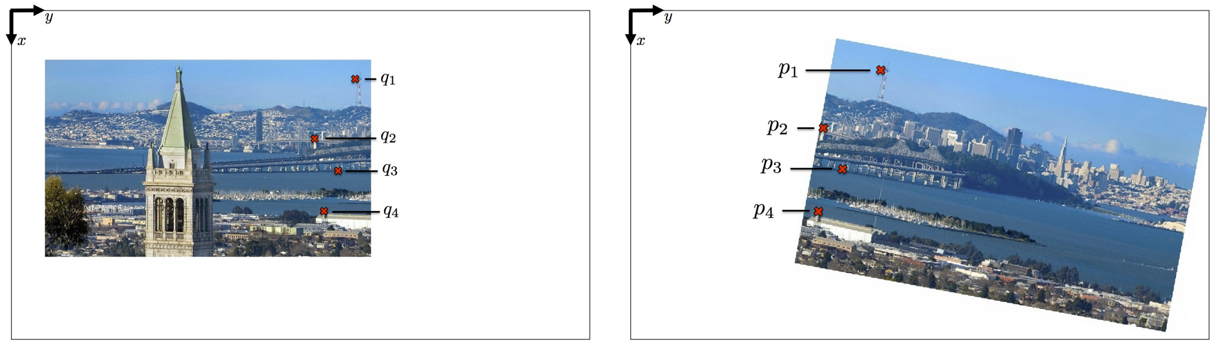


Figure 2: Two images to be stitched together with pairs of matching points labeled.

We will use vectors to represent the common points which are related by a linear transformation. Your idea is to find this linear transformation. For this you will use a single matrix, **R**, and a vector, $\vec{T}$, that transforms every common point in one image to their corresponding point in the other image. Once you find **R** and $\vec{T}$ you will be able to transform one image so that it lines up with the other image.

Suppose $\vec{p} = \begin{bmatrix} p_x \\ p_y \end{bmatrix}$ is a point in one image and $\vec{q} = \begin{bmatrix} q_x \\ q_y \end{bmatrix}$ is the corresponding point in the other image (i.e., they represent the same object in the scene). You write down the following relationship between $\vec{p}$ and $\vec{q}$.

$$\begin{bmatrix} q_x \\ q_y \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{R}_{xx} & \mathbf{R}_{xy} \\ \mathbf{R}_{yx} & \mathbf{R}_{yy} \end{bmatrix}}_{\mathbf{R}} \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \underbrace{\begin{bmatrix} T_x \\ T_y \end{bmatrix}}_{\vec{T}} \tag{1}$$

This problem focuses on finding the unknowns (i.e. the components of **R** and $\vec{T}$), so that you will be able to stitch the image together.

(a) To understand how the matrix $\mathbf{R}$ and vector $\vec{T}$ transform a vector, $\vec{v}_0$, consider this similar equation,

$$\vec{v}_2 = \begin{bmatrix} 2 & 2 \\ -2 & 2 \end{bmatrix} \vec{v}_0 + \vec{v}_1. \tag{2}$$

Using $\vec{v}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $\vec{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, what is $\vec{v}_2$? On a single plot, draw the vectors $\vec{v}_0, \vec{v}_1, \vec{v}_2$ in two dimensions. Describe how $\vec{v}_2$ is transformed from $\vec{v}_0$ (e.g. rotated, scaled, shifted).

Solution:

Plugging in the given vectors and performing the matrix vector multiplication,

$$\vec{v}_2 = \begin{bmatrix} 2 & 2 \\ -2 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{3}$$

We get $\vec{v}_2 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$.

Looking at the plotted vectors, it is observable that $\vec{v}_2$ is a scaled, rotated, and translated version of $\vec{v}_0$.

(b) Multiply Equation (1) out into two scalar linear equations. What are the known values and what are the unknowns in each equation? How many unknowns are there? How many independent equations do you need to solve for all the unknowns? How many pairs of common points $\vec{p}$ and $\vec{q}$ will you need in order to write down a system of equations that you can use to solve for the unknowns?

Solution:

We can rewrite the above matrix equation as the following two scalar linear equations:

$$q_x = p_x R_{xx} + p_y R_{xy} + T_x$$
$$q_y = p_x R_{yx} + p_y R_{yy} + T_y$$

Here, the known values are each pair of points' elements: q_x, q_y, p_x, p_y, and 1. The unknowns are elements of $\mathbf{R}$ and $\vec{T}$: R_{xx}, R_{xy}, R_{yx}, R_{yy}, T_x, and T_y. There are 6 unknowns, so we need a total of 6 equations to solve for them. For every pair of points we add, we get two more equations. Thus, we need 3 pairs of common points to get 6 equations.

(c) What is the vector of unknown values? Write out a system of linear equations that you can use to solve for the unknown values (you should use multiple pairs of points $\vec{p}$'s and $\vec{q}$'s to have enough equations based on what you found in part b). Transform these linear equations into a matrix equation, so that we can solve for the vector of unknown values.

Solution:

The vector of unknowns:

$$\begin{bmatrix} R_{xx} \\ R_{xy} \\ R_{yx} \\ R_{yy} \\ T_x \\ T_y \end{bmatrix} \tag{4}$$

The system of linear equations:

$$R_{xx}p_{1x} + R_{xy}p_{1y} + T_x = q_{1x} \tag{5}$$
$$R_{yx}p_{1x} + R_{yy}p_{1y} + T_y = q_{1y} \tag{6}$$
$$R_{xx}p_{2x} + R_{xy}p_{2y} + T_x = q_{2x} \tag{7}$$
$$R_{yx}p_{2x} + R_{yy}p_{2y} + T_y = q_{2y} \tag{8}$$
$$R_{xx}p_{3x} + R_{xy}p_{3y} + T_x = q_{3x} \tag{9}$$
$$R_{yx}p_{3x} + R_{yy}p_{3y} + T_y = q_{3y} \tag{10}$$
$$\tag{11}$$

We will label the 3 pairs of point we select as:

$$\vec{q}_1 = \begin{bmatrix} q_{1x} \\ q_{1y} \end{bmatrix}, \quad \vec{p}_1 = \begin{bmatrix} p_{1x} \\ p_{1y} \end{bmatrix} \qquad \vec{q}_2 = \begin{bmatrix} q_{2x} \\ q_{2y} \end{bmatrix}, \quad \vec{p}_2 = \begin{bmatrix} p_{2x} \\ p_{2y} \end{bmatrix} \qquad \vec{q}_3 = \begin{bmatrix} q_{3x} \\ q_{3y} \end{bmatrix}, \quad \vec{p}_3 = \begin{bmatrix} p_{3x} \\ p_{3y} \end{bmatrix}$$

We write the system of linear equations in matrix form.

$$\begin{bmatrix} p_{1x} & p_{1y} & 0 & 0 & 1 & 0 \\ 0 & 0 & p_{1x} & p_{1y} & 0 & 1 \\ p_{2x} & p_{2y} & 0 & 0 & 1 & 0 \\ 0 & 0 & p_{2x} & p_{2y} & 0 & 1 \\ p_{3x} & p_{3y} & 0 & 0 & 1 & 0 \\ 0 & 0 & p_{3x} & p_{3y} & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{xx} \\ R_{xy} \\ R_{yx} \\ R_{yy} \\ T_x \\ T_y \end{bmatrix} = \begin{bmatrix} q_{1x} \\ q_{1y} \\ q_{2x} \\ q_{2y} \\ q_{3x} \\ q_{3y} \end{bmatrix}$$

(d) In the IPython notebook `prob1B.ipynb`, you will have a chance to test out your solution. Plug in the values that you are given for p_x, p_y, q_x, and q_y for each pair of points into your system of equations to solve for the matrix, $\mathbf{R}$, and vector, $\vec{T}$. The notebook will solve the system of equations, apply your transformation to the second image, and show you if your stitching algorithm works. You are not responsible for understanding the image stitching code or Marcela's algorithm.

Solution:

The parameters for the transformation from the coordinates of the first image to those of the second image are $\mathbf{R} = \begin{bmatrix} 1.1954 & .1046 \\ -.1046 & 1.1954 \end{bmatrix}$ and $\vec{T} = \begin{bmatrix} -150 \\ -250 \end{bmatrix}$.

5. Segway Tours

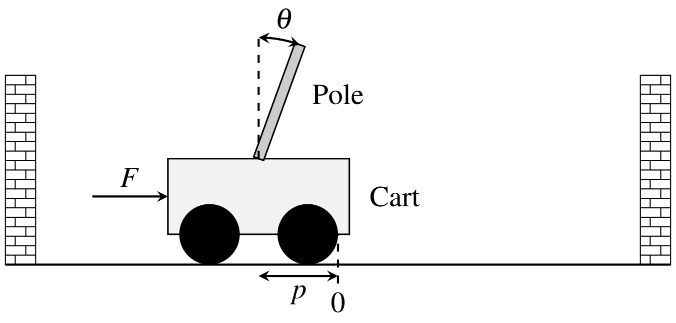
Learning Objective: *The learning objective of this problem is to see how the concept of span can be applied to control problems. If a desired state vector of a linear control problem is in a span of a particular set of vectors, then the system may be steered to reach that particular vector using the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

$$\vec{x}[n+1] = \mathbf{A}\vec{x}[n] + \vec{b}u[n], \tag{12}$$

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.

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.)

Solution:

From Equation (12), we get (by substituting $n = 0$):

$$\vec{x}[1] = \mathbf{A}\vec{x}[0] + \vec{b}u[0] \tag{13}$$

[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.

(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 $u[3]$.

Solution:

From Equation (12), we get (by substituting $n = 1$):

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

By substituting $\vec{x}[1]$ from Equation (13), we get

$$\begin{aligned}\vec{x}[2] &= \mathbf{A}\vec{x}[1] + \vec{b}u[1] \\ &= \mathbf{A}\left(\mathbf{A}\vec{x}[0] + \vec{b}u[0]\right) + \vec{b}u[1] \\ &= \mathbf{A}^2\vec{x}[0] + \mathbf{A}\vec{b}u[0] + \vec{b}u[1] \qquad (14)\end{aligned}$$

From Equation (12), we get (by substituting $n = 2$):

$$\vec{x}[3] = \mathbf{A}\vec{x}[2] + \vec{b}u[2]$$

By substituting $\vec{x}[2]$ from Equation (14), we get

$$\begin{aligned}\vec{x}[3] &= \mathbf{A}\vec{x}[2] + \vec{b}u[2] \\ &= \mathbf{A}\left(\mathbf{A}^2\vec{x}[0] + \mathbf{A}\vec{b}u[0] + \vec{b}u[1]\right) + \vec{b}u[2] \\ &= \mathbf{A}^3\vec{x}[0] + \mathbf{A}^2\vec{b}u[0] + \mathbf{A}\vec{b}u[1] + \vec{b}u[2] \qquad (15)\end{aligned}$$

From Equation (12), we get (by substituting $n = 3$):

$$\vec{x}[4] = \mathbf{A}\vec{x}[3] + \vec{b}u[3]$$

By substituting $\vec{x}[3]$ from Equation (15), we get

$$\begin{aligned}\vec{x}[4] &= \mathbf{A}\vec{x}[3] + \vec{b}u[3] \\ &= \mathbf{A}\left(\mathbf{A}^3\vec{x}[0] + \mathbf{A}^2\vec{b}u[0] + \mathbf{A}\vec{b}u[1] + \vec{b}u[2]\right) + \vec{b}u[3] \\ &= \mathbf{A}^4\vec{x}[0] + \mathbf{A}^3\vec{b}u[0] + \mathbf{A}^2\vec{b}u[1] + \mathbf{A}\vec{b}u[2] + \vec{b}u[3] \qquad (16)\end{aligned}$$

(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 $u[0], \ldots, u[N-1]$.

Solution:

Use the same procedure as above for N steps. You will obtain the following expression:

$$\vec{x}[N] = \mathbf{A}^N\vec{x}[0] + \mathbf{A}^{N-1}\vec{b}u[0] + \ldots + \mathbf{A}\vec{b}u[N-2] + \vec{b}u[N-1] \qquad (17)$$

You might also use the compact expression:

$$\vec{x}[N] = \mathbf{A}^N\vec{x}[0] + \sum_{i=0}^{N-1} \mathbf{A}^i\vec{b}u[N-i-1] \qquad (18)$$

Note that $\mathbf{A}^0$ is the identity matrix.

As a sanity check, plug the values $N = 1, 2, 3$, and 4 to obtain Equations (13), (14), (15), and (16), respectively.

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.*)

Solution:

No.

From Equation (14), we know that $\mathbf{A}^2\vec{x}[0] + \mathbf{A}\vec{b}u[0] + \vec{b}u[1] = \vec{x}[2]$ which is equivalent to $\mathbf{A}\vec{b}u[0] + \vec{b}u[1] = \vec{x}[2] - \mathbf{A}^2\vec{x}[0]$.

This means that in order to reach any state $\vec{x}_f$ in two time steps (that is, $\vec{x}[2] = \vec{x}_f$), we have to solve the following system of linear equations:

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

where $u[0]$ and $u[1]$ are the unknowns.

Since in our case we want to reach $\vec{x}_f = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$, the system of linear equations simplifies to

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

In matrix form, this system of linear equations is

$$\begin{bmatrix} | & | \\ \mathbf{A}\vec{b} & \vec{b} \\ | & | \end{bmatrix} \begin{bmatrix} u[0] \\ u[1] \end{bmatrix} = -\mathbf{A}^2\vec{x}[0],$$

which yields the following augmented matrix:

$$\left[\begin{array}{cc|c} | & | & | \\ \mathbf{A}\vec{b} & \vec{b} & -\mathbf{A}^2\vec{x}[0] \\ | & | & | \end{array}\right].$$

By plugging in the values of $\mathbf{A}$, $\vec{b}$, and $\vec{x}[0]$, we get the following augmented matrix:

$$\left[\begin{array}{cc|c} 0.0208 & 0.01 & 0.02243475295 \\ 0.0557 & 0.21 & -0.30785116611 \\ -0.0572 & -0.03 & 0.0619347608 \\ -0.2385 & -0.44 & 1.38671325508 \end{array}\right].$$

Applying Gaussian elimination, we get the upper triangular form to be

$$\left[\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{array}\right],$$

which means that the system is inconsistent (due to the third row) and that there are no solutions for $u[0]$ and $u[1]$. It is fine if you did not row reduce all the way to the upper triangular form as long as you showed that the system of equations is inconsistent.

(e) Can you reach $\vec{x}_f$ in *three* time steps? (*Hint: Similar to the last part, express* $\vec{x}[3]-\mathbf{A}^2\vec{x}[0]$ *in terms of the inputs* $u[0]$, $u[1]$ *and* $u[2]$. *Then determine if we can obtain valid solutions for* $u[0]$, $u[1]$ *and* $u[2]$.)

Solution:

No.

Similar to the previous part, from Equation (15), we know that $\mathbf{A}^3\vec{x}[0]+\mathbf{A}^2\vec{b}u[0]+\mathbf{A}\vec{b}u[1]+\vec{b}u[2]=\vec{x}[3]$, which is equivalent to $\mathbf{A}^2\vec{b}u[0]+\mathbf{A}\vec{b}u[1]+\vec{b}u[2]=\vec{x}[3]-\mathbf{A}^3\vec{x}[0]$.

This means that in order to reach any state $\vec{x}_f$ in three time steps (that is, $\vec{x}[3]=\vec{x}_f$), we have to solve the following system of linear equations:

$$\mathbf{A}^2\vec{b}u[0]+\mathbf{A}\vec{b}u[1]+\vec{b}u[2]=\vec{x}_f-\mathbf{A}^3\vec{x}[0],$$

where $u[0], u[1]$, and $u[2]$ are the unknowns.

Since in our case we want to reach $\vec{x}_f=\begin{bmatrix}0\\0\\0\\0\end{bmatrix}$, the system of linear equations simplifies to

$$\mathbf{A}^2\vec{b}u[0]+\mathbf{A}\vec{b}u[1]+\vec{b}u[2]=-\mathbf{A}^3\vec{x}[0].$$

In matrix form, this system of linear equations is

$$\begin{bmatrix} | & | & | \\ \mathbf{A}^2\vec{b} & \mathbf{A}\vec{b} & \vec{b} \\ | & | & | \end{bmatrix}\begin{bmatrix}u[0]\\u[1]\\u[2]\end{bmatrix}=-\mathbf{A}^3\vec{x}[0],$$

which yields the following augmented matrix:

$$\left[\begin{array}{ccc|c} | & | & | & | \\ \mathbf{A}^2\vec{b} & \mathbf{A}\vec{b} & \vec{b} & -\mathbf{A}^3\vec{x}[0] \\ | & | & | & | \end{array}\right].$$

By plugging in the values of $\mathbf{A}$, $\vec{b}$, and $\vec{x}[0]$, we get the following augmented matrix:

$$\left[\begin{array}{ccc|c} 0.024157 & 0.0208 & 0.01 & 0.0064228470365 \\ 0.024363 & 0.0557 & 0.21 & -0.092123298431 \\ -0.083488 & -0.0572 & -0.03 & 0.178491836209001 \\ -0.342448 & -0.2385 & -0.44 & 1.246334243328597 \end{array}\right].$$

Applying Gaussian elimination, we get the upper triangular form to be

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right],$$

which means that the system is inconsistent (due to the fourth row) and that there are no solutions for $u[0], u[1]$, and $u[2]$. It is fine if you did not row reduce all the way to the upper triangular form as long as you showed that the system of equations is inconsistent.

(f) Can you reach $\vec{x}_f$ in *four* time steps? (*Use the hints from the last two parts.*)

Solution:

Yes.

Similar to the previous part, from Equation (16), we know that $\mathbf{A}^4\vec{x}[0] + \mathbf{A}^3\vec{b}u[0] + \mathbf{A}^2\vec{b}u[1] + \mathbf{A}\vec{b}u[2] + \vec{b}u[3] = \vec{x}[4]$ which is equivalent to $\mathbf{A}^3\vec{b}u[0] + \mathbf{A}^2\vec{b}u[1] + \mathbf{A}\vec{b}u[2] + \vec{b}u[3] = \vec{x}[4] - \mathbf{A}^4\vec{x}[0]$.

This means that in order to reach any state $\vec{x}_f$ in four time steps (that is, $\vec{x}[4] = \vec{x}_f$), we have to solve the following system of linear equations:

$$\mathbf{A}^3\vec{b}u[0] + \mathbf{A}^2\vec{b}u[1] + \mathbf{A}\vec{b}u[2] + \vec{b}u[3] = \vec{x}_f - \mathbf{A}^4\vec{x}[0],$$

where $u[0], u[1], u[2]$, and $u[3]$ are the unknowns.

Since in our case we want to reach $\vec{x}_f = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$, the system of linear equations simplifies to

$$\mathbf{A}^3\vec{b}u[0] + \mathbf{A}^2\vec{b}u[1] + \mathbf{A}\vec{b}u[2] + \vec{b}u[3] = -\mathbf{A}^4\vec{x}[0].$$

In matrix form, this system of linear equations is

$$\begin{bmatrix} | & | & | & | \\ \mathbf{A}^3\vec{b} & \mathbf{A}^2\vec{b} & \mathbf{A}\vec{b} & \vec{b} \\ | & | & | & | \end{bmatrix} \begin{bmatrix} u[0] \\ u[1] \\ u[2] \\ u[3] \end{bmatrix} = -\mathbf{A}^4\vec{x}[0].$$

By defining $\mathbf{Q} = \begin{bmatrix} | & | & | & | \\ \mathbf{A}^3\vec{b} & \mathbf{A}^2\vec{b} & \mathbf{A}\vec{b} & \vec{b} \\ | & | & | & | \end{bmatrix}$ and $\vec{u}_4 = \begin{bmatrix} u[0] \\ u[1] \\ u[2] \\ u[3] \end{bmatrix}$, we can now rewrite our system of linear equations as

$$\mathbf{Q}\vec{u}_4 = -\mathbf{A}^4\vec{x}[0].$$

Refer to the code in the solution IPython notebook for a solution of the system above. The solution of the system is

$$\vec{u}_4 = \begin{bmatrix} -13.24875075 \\ 23.73325125 \\ -11.57181872 \\ 1.46515973 \end{bmatrix},$$

that is, the control input sequence is: $u[0] = -13.24875075$, $u[1] = 23.73325125$, $u[2] = -11.57181872$, and $u[3] = 1.46515973$.

(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!*

Solution:

See the solution to the previous part.

(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?

Solution: You can move in directions $\vec{b}$ and $\mathbf{A}\vec{b}$. Hence you can reach all positions that are in $\text{span}\{\vec{b}, \mathbf{A}\vec{b}\}$.

(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.

Solution: In N timesteps you can reach

$$\text{span}\left\{\vec{b}, \mathbf{A}\vec{b}, \mathbf{A}^2\vec{b}, \ldots, \mathbf{A}^{N-1}\vec{b}\right\}.$$

You will also have an offset of $A^N\vec{x}[0]$. Taking this into consideration you can reach

$$\text{span}\left\{\vec{b}, \mathbf{A}\vec{b}, \mathbf{A}^2\vec{b}, \ldots, \mathbf{A}^{N-1}\vec{b}\right\} + A^N\vec{x}[0].$$

You can give yourself full credit even if you do not have the offset term for $A^n\vec{x}[0]$.

(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?

Solution: This builds on the previous parts. Since now you want to reach anywhere in $\mathbb{R}^4$, and you know the possible positions you can reach are given by

$$\text{span}\left\{\vec{b}, \mathbf{A}\vec{b}, \mathbf{A}^2\vec{b}, \ldots, \mathbf{A}^{N-1}\vec{b}\right\},$$

you need this span to be $\mathbb{R}^4$.

P.S.: Congratulations! You have just derived the condition for "controllability" for systems with linear dynamics. When dealing with a system that evolves over time, we can sometimes influence the behavior of the system through various control inputs (for example, the steering wheel and gas pedal of a car or the rudder of an airplane). It is of great importance to know what states (think positions and velocities of a car or configurations of an aircraft) that our system can be controlled to. Controllability is the ability to control the system to any possible state or configuration.

A more detailed argument follows, but the above argument is sufficient to conclude the result.

More detailed argument:

The key step here is to rewrite the equation you derived in part ((c)) (Equation (18)) as

$$\sum_{i=0}^{N-1} \mathbf{A}^i \vec{b} u[N-i-1] = \vec{x}[N] - \mathbf{A}^N x[0].$$

$\vec{x}[N] = \vec{x}_f$ can be anything in $\mathbb{R}^4$. Therefore, the system of linear equations can be written as

$$\sum_{i=0}^{N-1} \mathbf{A}^i \vec{b} u[N-i-1] = \vec{x}_f - \mathbf{A}^N x[0].$$

If we extend this sum, we get

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

This system of linear equations can be further rewritten as

$$\begin{bmatrix} | & | & \cdots & | & | \\ \mathbf{A}^{N-1}\vec{b} & \mathbf{A}^{N-2}\vec{b} & \cdots & \mathbf{A}\vec{b} & \vec{b} \\ | & | & \cdots & | & | \end{bmatrix} \begin{bmatrix} u[0] \\ u[1] \\ \vdots \\ u[N-1] \end{bmatrix} = \vec{x}_f - \mathbf{A}^N x[0].$$

For this system to be solvable, we need $\vec{x}_f - \mathbf{A}^N x[0] \in \text{span}\left\{\vec{b}, \mathbf{A}\vec{b}, \mathbf{A}^2\vec{b}, \ldots, \mathbf{A}^{N-1}\vec{b}\right\}$. Since $\vec{x}_f$ can be any vector in $\mathbb{R}^4$, it also means that $\vec{x}_f - \mathbf{A}^N x[0]$ can be any vector in $\mathbb{R}^4$. This means that in order to be able to reach any state $\vec{x}_f \in \mathbb{R}^4$, the range (column space) of the matrix $\begin{bmatrix} | & | & \cdots & | & | \\ \mathbf{A}^{N-1}\vec{b} & \mathbf{A}^{N-2}\vec{b} & \cdots & \mathbf{A}\vec{b} & \vec{b} \\ | & | & \cdots & | & | \end{bmatrix}$ has to be all of $\mathbb{R}^4$.

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?

Solution:

I worked on this homework with...

I first worked by myself for 2 hours, but got stuck on problem 5, so I went to office hours on...

Then I went to homework party for a few hours, where I finished the homework.