1. Reading Lecture Notes

Staying up to date with lectures is an important part of the learning process in this course. Here are links to the notes that you need to read for this week: Note 0A Note 0B

(a) Looking at the different conceptual stages/blocks for dealing with the Cyborg problem, which parts do you think that you are most interested in?

Solution: A sample written by course staff is provided below. Your response should reflect your personal topical interests so self-grade on the basis of having expressed these preferences.
I’m most excited about the notion of discrete representations of data highlighted in (b) and the learning of dynamical models in (e). I’m curious about what is lost when using discrete representations, as well as the notion of what kind of biased dynamical model results from biased data.

(b) How do you think that analogous ideas are important in the development of modern high-performance software systems?

Solution: A sample written by course staff is provided below for further consideration. We recognize your response will unique to you and may not address all conceptual blocks, and that is okay. Self-grade on the basis of judging whether you have made connections between the note material and software systems.

High performance software systems often need to sense what is the state of the world, where the world is the computational platform that they are executing on. The state is often high dimensional in nature and must be dimensionality reduced for real-time decision making. For example, consider long text output logs. This is related to (a) and (c). An engineer can alter performance by understanding the methods used to pre-process and summarize. Inferring what the methods are sensitive to, and what they can or cannot detect allows design decisions to made in a variety of settings. Certain failure modes can be diagnosed and tracked using classification ideas (d). Consider the setting of a warehouse scale computer, utilized by large software companies. Detecting which resources in a cluster of computers are in a condition soon to fail and require replacement can be done so by learning the presentation of such resources. Plans for mapping tasks and requests onto computational resources have to be made (e) to minimize power consumption, guarantee adequate cooling, minimize the time to task completion, and satisfy other resource constraints or performance goals. In the context of load-balancing, while a static rule independent of computational characteristics can be utilized, performance can be increased by dynamically allocating resources depending on the overall state. A plan can directly leverage a learned model of the expected evolution of the mass of computational resources in response to different patterns of tasks and requests. To actually keep things humming along in the face of uncertainty, feedback control is used on fast time scales (f). Perhaps the data collected...
leads to a model that is slightly off, or there is an unexpected slew of requests. Managing these requires sensing the resulting performance and feeding this information back into a request reallocation rule. Networking, and internet-scale networking in particular, is a big user of feedback control ideas. Here is a real life instance from Cloudflare highlighting where such considerations as (e) and (f) will occur. Interpolation ideas in the narrow form of (b) are less common, but combinations of (b) with tomography ideas from 16A definitely figure into making measurements of high-performance software systems without adding too much logging overhead in critical paths.

2. Surveys

Below we have attached two google form links for you to fill out. The first is a demographic survey that will help us understand more about you and your background. Getting to know you will only help us improve your experience in EECS16B. The second is to indicate to if you would like us to match you with a study group. Participating in study groups is completely optional, please choose an option that is best for you.

(a) Demographic Survey - Google Form
(b) Group Formation Survey - Google Form

3. Course Policies

Go to the course website and read the course policies carefully. Leave a followup in the Homework 0, Question 3 thread on Piazza if you have any questions. Are the following situations violations of course policy? Write "Yes" or "No", and a short explanation for each.

(a) Alice and Bob work on a problem in a study group. They write up a solution together and submit it, noting on their submissions that they wrote up their homework answers together.

Solution: Yes, this is a violation as the solutions must always be written up on one’s own.

(b) Carol goes to a homework party and listens to Dan describe his approach to a problem on the board, taking notes in the process. She writes up her homework submission from her notes, crediting Dan.

Solution: No, as Carol has written her homework submission by herself and appropriately credited Dan.

(c) Erin gets frustrated by the fact that a homework problem given seems to have nothing in the lecture, notes, or discussion that is parallel to it. So, she starts searching for the problem online. She finds a solution to the homework problem on a website. She reads it and then, after she has understood it, writes her own solution using the same approach. She submits the homework with a citation to the website.

Solution: Yes, this is a violation as Erin looked at a solution to the homework as found on a website. It doesn’t matter that she wrote the solution after understanding it and didn’t simply copy it. Even if she had written a solution using a different approach, it would still have been a policy violation. Why? Because being able to work to understand the homework and what it is asking is a part of the skills being exercised and developed. Reading a solution shortcuts this in a non-interactive way.

(d) Frank is having trouble with his homework and asks Grace for help. Grace lets Frank look at her written solution. Frank copies it onto his notebook and uses the copy to write and submit his homework, crediting Grace.

Solution: Yes, this is a violation as Grace should not have shared her written solution directly. Frank also should have not used Grace’s solution in writing his own.
(e) Heidi has completed her homework. Her friend Irene has been working on a homework problem for hours, and asks Heidi for help. Heidi sends Irene her photos of her solution through an instant messaging service, and Irene uses it to write her own solution with a citation to Heidi.

**Solution:** Yes, this is a violation as Irene should not have looked at Heidi’s solution before writing hers, and Heidi should not have sent her solution. Both have violated class policies.

4. **Save Baby Yoda!**

Despite our best efforts, we have lost Baby Yoda to former agents of the Galactic Empire. Luckily we were able to conceal a receiver in his locket, so now it’s time to save Baby Yoda using our 16A knowledge!

(a) Baby Yoda has been delivered to an Imperial Star Destroyer. Rebel intel has provided us with access to their internal communication beacons. The ship’s layout is 2-dimensional with 3 beacon locations specified in Table 1.

<table>
<thead>
<tr>
<th>Beacon</th>
<th>Coordinates</th>
<th>Distance to Baby Yoda</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>(5,5)</td>
<td>$\sqrt{20}$</td>
</tr>
<tr>
<td>B</td>
<td>(2,3)</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>(1,1)</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1: Data from Destroyer Beacons and their coordinates.

![Diagram of the Destroyer’s floor-plan with Beacon coordinates marked accordingly.](image)

**Figure 1:** Diagram of the Destroyer’s floor-plan with Beacon coordinates marked accordingly.

Explicitly write out a linear system of equations (in matrix-vector form) using the data above for finding Baby Yoda’s location $\vec{x} = \begin{bmatrix} x \\ y \end{bmatrix}$. Draw a box around your final linear system, then solve for Baby Yoda’s location. Nonlinear terms are not permitted in your final system of equations. You must provide both the system and the location for full credit.

**Solution:**
The three equations are:

[A]:\[(x - 5)^2 + (y - 5)^2 = x^2 - 10x + 25 + y^2 - 10y + 25 = 20,\]
[B]:\[(x - 2)^2 + (y - 3)^2 = x^2 - 4x + 4 + y^2 - 6y + 9 = 1,\]
[C]:\[(x - 1)^2 + (y - 1)^2 = x^2 - 2x + 1 + y^2 - 2y + 1 = 4.\]

**Method A:** We subtract equation [A] from the others to yield:

\[
\begin{align*}
6x - 37 + 4y &= -19 \\
8x - 48 + 8y &= -16
\end{align*}
\rightarrow
\begin{align*}
3x + 2y &= 9 \\
x + y &= 4
\end{align*}
\rightarrow
\begin{bmatrix}
3 & 2 \\
1 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
= 
\begin{bmatrix}
9 \\
4
\end{bmatrix}.
\]

**Method B:** We subtract equation [B] from the others to yield:

\[
\begin{align*}
-6x + 37 - 4y &= 19 \\
2x - 11 + 4y &= 3
\end{align*}
\rightarrow
\begin{align*}
3x + 2y &= 9 \\
x + 2y &= 7
\end{align*}
\rightarrow
\begin{bmatrix}
3 & 2 \\
1 & 2
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
= 
\begin{bmatrix}
9 \\
7
\end{bmatrix}.
\]

**Method C:** We subtract equation [C] from the others to yield:

\[
\begin{align*}
-8x + 48 - 8y &= 16 \\
-2x - 11 - 4y &= -3
\end{align*}
\rightarrow
\begin{align*}
x + y &= 4 \\
x + 2y &= 7
\end{align*}
\rightarrow
\begin{bmatrix}
1 & 1 \\
1 & 2
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
= 
\begin{bmatrix}
4 \\
7
\end{bmatrix}.
\]

All of these systems of equations are acceptable. In the following solution we will work from the Method A system. From this stage there are three ways to solve for Baby Yoda’s location:

1. **Row Reduction:** Use Gaussian elimination on your system.

\[
\begin{bmatrix}
3 & 2 & 9 \\
1 & 1 & 4
\end{bmatrix}
\rightarrow
\begin{bmatrix}
3 & 2 & 9 \\
2 \cdot 1 - 3 & 2 \cdot 1 - 2 & 2 \cdot 4 - 9
\end{bmatrix}
\rightarrow
\begin{bmatrix}
3 & 2 & 9 \\
1 & 0 & 1
\end{bmatrix}
\]

Thus \(x = 1\). From here we can substitute into either equation: \(x + y = 4 \rightarrow y = 3\).

2. **Inverse:** Compute the inverse matrix, as per our formula for \(2 \times 2\) matrices

\[
\frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.
\]

From this point it becomes a matter of matrix vector multiplication to identify \(\overline{x}\) using the relation \(A\overline{x} = \overline{b} \rightarrow \overline{x} = A^{-1}\overline{b}\).

\[
\begin{bmatrix}
3 & 2 \\
1 & 1
\end{bmatrix}^{-1}
= 
\begin{bmatrix} 1 & -2 \\ -1 & 3 \end{bmatrix}
\rightarrow
\begin{bmatrix} 1 & -2 \\ -1 & 3 \end{bmatrix}
\begin{bmatrix}
9 \\
4
\end{bmatrix}
= 
\begin{bmatrix} 9 - 8 \\ -3 + 12 \end{bmatrix}
= 
\begin{bmatrix} 1 \\
3
\end{bmatrix}
\]

3. **Graphically:** Using Figure 1, draw circles centered at each beacon with the appropriate radius from the beacon data. Their intersection identifies Baby Yoda’s location.

The solution for Baby Yoda’s location is \(\overline{x} = \begin{bmatrix} 1 \\
3 \end{bmatrix}\).
5. Ultrasound Sensing with Op-Amps

The transresistance amplifier is often used to convert a current from a sensor to a voltage. In this problem we will use it to build an ultrasound sensor! When an ultrasonic wave hits our sensor, it generates a current, \( i_{\text{ultra}} \). Whenever no ultrasonic wave hits our sensor zero current is generated, so \( i_{\text{ultra}} = 0 \).

**Note:** An *ideal op-amp* is used in all subparts of this question. You can also assume that \( V_{DD} = -V_{SS} \).

![Figure 2: Transresistance sensor circuit](image)

(a) Calculate the output voltage, \( V_{out} \), of the transresistance sensor circuit shown in Fig. 2, as a function of the reference voltage, \( V_{REF} \), the sensor input current, \( i_{\text{ultra}} \), and the resistor, \( R \), when an ultrasonic wave hits the sensor. Clearly show all your work and justify your answer. Writing only the final expression will not be given full credit.

**Solution:**
Since we have an ideal op-amp in a negative feedback circuit, we can first say that \( u_- = u_+ = V_{REF} \), where \( u_+ \) and \( u_- \) are the voltages at the positive and negative input terminals of the op-amp respectively. Next the current \( i_{\text{ultra}} \) must flow right through \( R \) due to KCL and the golden rules (no current can flow into the op-amp). Thus we establish

\[
V_{out} = u_- - i_{\text{ultra}} R = u_+ - i_{\text{ultra}} R = V_{REF} - i_{\text{ultra}} R. \quad \square
\]

(b) Assume that the amplitude of the ultrasonic wave hitting the sensor is such that the current \( i_{\text{ultra}} \) fluctuates from a minimum value of \( i_{\text{min}} = 1 \cdot 10^{-6} \text{A} \), to a maximum value of \( i_{\text{max}} = 2 \cdot 10^{-6} \text{A} \). Also assume that the reference voltage is set to \( V_{REF} = 1 \text{V} \). In this case, calculate the following:

i. **The maximum value of the resistor, \( R \), so that the output voltage, \( V_{out} \), does not drop below 0V.** Clearly show all your work.

ii. Assuming you picked \( R = 250 \cdot 10^3 \Omega \) (which may or may not be the correct answer to part (i)), calculate the maximum value of the output voltage, \( V_{out} \). Clearly show all your work.

**Solution:**
From part (a) we identified the output voltage of our sensor circuit \( V_{out} = V_{REF} - i_{\text{ultra}} R \).
i. The worst-case scenario (in which the output voltage is most reduced) occurs for $i_{\text{ultra}} = i_{\text{max}} = 2 \cdot 10^{-6}$ A. If we include $V_{\text{REF}} = 1$ V and set our nonnegative condition on $V_{\text{out}}$, we identify the restriction on $R$:

$$V_{\text{out}} = V_{\text{REF}} - i_{\text{max}} R \geq 0 \implies R \leq \frac{V_{\text{REF}}}{i_{\text{max}}} = \frac{1 \text{ V}}{2 \cdot 10^{-6} \text{ A}} = 500,000 \Omega.$$  

ii. Based on our voltage formula, the highest $V_{\text{out}}$ scenario now occurs for the low current condition $i_{\text{ultra}} = i_{\text{min}} = 1 \cdot 10^{-6}$ A. From this point we now substitute into the $V_{\text{out}}$ formula:

$$V_{\text{out}} = V_{\text{REF}} - i_{\text{min}} (250 \cdot 10^3 \Omega) = 1 - (1 \cdot 10^{-6} \text{ A})(250 \cdot 10^3 \Omega) = 1 - 0.25 = 0.75 \text{ V}.$$  

(c) Unfortunately, after a few hours of successful ultrasound sensing, our sensor got damaged. It now constantly generates a huge background current, $I_{\text{damage}}$. So when an ultrasonic wave hits it, the sensor produces $I_{\text{damage}} + i_{\text{ultra}}$, as shown in Fig 3(b). When no ultrasonic wave hits it, the sensor produces just $I_{\text{damage}}$. However, the huge background current causes our output to constantly be $V_{\text{out}} = V_{\text{SS}}$, so we are not able to tell whether an ultrasonic wave is present or not.

We would like to fix this in our circuit by canceling the background current and retaining only the useful signal. For this purpose we are going to use a current source, $I_{\text{fix}}$, shown in Fig. 3(a), whose value we can choose. **How would you connect this current source in your circuit and what value would you pick for it?** Redraw the entire circuit with the new current source, $I_{\text{fix}}$, added and give the value of $I_{\text{fix}}$ in terms of $I_{\text{damage}}, i_{\text{ultra}}, R, V_{\text{REF}}$. **Explain how your design works.**

---

**Solution:**

We want to have only $i_{\text{ultra}}$ flow through $R$. To achieve this we will insert the correcting current source in parallel with the input source $i_{\text{ultra}} + I_{\text{damage}}$ and set it at $I_{\text{fix}} = I_{\text{damage}}$ in opposite polarity, so that KCL gives:

$$I_{\text{damage}} + i_{\text{ultra}} = I_{\text{damage}} + I_R \implies I_R = i_{\text{ultra}}.$$  

---

Homework 0, © UCB EECS 16B, Fall 2021. All Rights Reserved. This may not be publicly shared without explicit permission.
6. Saving Lives with Op-Amps

An electrocardiogram, or ECG, is a medical device used to detect electrical signals in your heart. Typically, the voltage signal from the human heart is only $1 \times 10^{-3}$ V at maximum. However, in order for healthcare professionals to properly interpret ECGs, these signals need to be amplified so that abnormalities are more obvious. In this problem we will do so by using ideal op-amps.

Note: Assume that $V_{DD} = -V_{SS}$ in all subparts.

(a) We need to amplify the voltage signal recorded by the electrodes $V_{in}$ by a factor of 1000. Using the op-amp in Figure 4 below and 2 resistors, draw a circuit that achieves $V_{out} = 1000 \cdot V_{in}$. Write an equation for $V_{out}$ in terms of $V_{in}$ and the resistor(s), label the resistors you use (i.e. $R_1$, $R_2$), and choose their values. You should redraw the entire circuit in your answer sheet, but there is no need to draw the human as long as you label $V_{in}$.

![Figure 4: Unfinished ECG amplification circuit.](image)

Solution:
To achieve this input-output relationship we need to use a non-inverting amplifier like the one shown in Fig. 5, which was analyzed in lecture and gives: $V_{out} = V_{in}(1 + \frac{R_{top}}{R_{bottom}})$.

To get $V_{out} = 1000V_{in}$, we need to size $R_{top}$ and $R_{bottom}$ such that $R_{top} = 999R_{bottom}$.
One such option is selecting $R_{top} = 999 \cdot 10^3 \Omega$ and $R_{bottom} = 1 \cdot 10^3 \Omega$. 
(b) A friend of yours is also working on an ECG amplification circuit, and shows you their design in Figure 6. Their design uses $R_{\text{electrode}} = 1 \times 10^3 \Omega$, $R_1 = 1 \times 10^3 \Omega$, and $R_2 = 1 \times 10^6 \Omega$. They claim their circuit gives, $V_{\text{out}} = -1000 \cdot V_{\text{in}}$. Is their claim true?

- If yes, justify why.
- If no, how would you choose the value of $R_2$ to achieve $V_{\text{out}} = -1000 \cdot V_{\text{in}}$, assuming that both $R_{\text{electrode}}$ and $R_1$ are fixed at $R_{\text{electrode}} = R_1 = 1 \times 10^3 \Omega$? Clearly show your work, and justify your answers.

**Solution:** This is the inverting amplifier topology analyzed in lecture. Using equivalence to lump $R_1$ and $R_{\text{electrode}}$ together we get that:

$$V_{\text{out}} = -\frac{R_2}{R_1 + R_{\text{electrode}}} V_{\text{in}} = -V_{\text{in}} = -500V_{\text{in}} \neq -1000V_{\text{in}}.$$  

So their claim is not true.  

Since we have  

$$V_{\text{out}} = -\frac{R_2}{R_1 + R_{\text{electrode}}} V_{\text{in}}$$

and $R_1, R_{\text{electrode}}$ are fixed at $R_{\text{electrode}} = R_1 = 1 \times 10^3 \Omega$, we need to set $R_2 = 2M\Omega = 2 \times 10^6 \Omega$ in order to get $V_{\text{out}} = -1000V_{\text{in}}$.  

---

**Figure 5:** Complete ECG amplification circuit.

**Figure 6:** An alternative ECG op-amp circuit.
(c) Another configuration often used by healthcare professionals is to attach one electrode to the heart (recording its electrical signal, $V_{in}$) and another electrode to the right leg to serve as a reference voltage, as shown in Figure 7. **What is the output voltage, $V_{out}$, as a function of $V_{in}$, $V_{RL}$, $R_{bottom}$, and $R_{top}$? Clearly show your work.**

![Figure 7: Alternative op-amp ECG topology.](image)

**Solution:**

**Method 1: Superposition**

We can find the output of this circuit by treating $V_{RL}$ as a second input and apply superposition: Zeroing out $V_{in}$ first and looking at $V_{RL}$ we get the following equivalent ckt:

![Figure 8: Alternative op-amp ECG topology with $V_{in}$ zeroed-out](image)

We can see that this is an inverting amplifier, so the output of the circuit is:

$$V_{out,RL} = -\frac{R_{top}}{R_{bottom}} V_{RL}.$$
Next we zero-out and look at the output due to $V_{in}$ only:

![Alternative op-amp ECG topology with $V_{RL}$ zeroed-out](image.png)

Figure 9: Alternative op-amp ECG topology with $V_{RL}$ zeroed-out

Which is a non-inverting amplifier whose output is:

$$V_{out, V_{in}} = (1 + \frac{R_{top}}{R_{bottom}})V_{in}.$$ 

Applying superposition, we get:

$$V_{out} = V_{out, V_{in}} + V_{out, V_{RL}} = V_{in}(1 + \frac{R_{top}}{R_{bottom}}) - V_{RL}(\frac{R_{top}}{R_{bottom}}).$$

**Method 2: NVA**

Alternatively, we can use NVA & golden rules to find the output of this circuit: Since this ckt is in negative feedback, golden rule #2 gives $u_- = V_{in}$, while KCL at $u_-$ gives:

$$I_{R_{bottom}} = I_{R_{top}} \Rightarrow \frac{V_{RL} - u_-}{R_{bottom}} = \frac{u_- - V_{out}}{R_{top}} \Rightarrow \frac{V_{RL} - V_{in}}{R_{bottom}} = \frac{V_{in} - V_{out}}{R_{top}}.$$ 

Solving for $V_{out}$, we get:

$$V_{out} = V_{in}(1 + \frac{R_{top}}{R_{bottom}}) - V_{RL}(\frac{R_{top}}{R_{bottom}}).$$
(d) Even after amplification, certain peaks of your ECG signal are too low to be discerned. You want to sample them and amplify them a bit more. To this end, you use the circuit in Figure 10. The circuit cycles through two phases: in phase 1, switches labeled $\phi_1$ are ON and $\phi_2$ are OFF, while in phase 2, switches labeled $\phi_2$ are ON and $\phi_1$ are OFF. Calculate the output voltage, $V_{\text{out}}$, during phase 2, after steady state has been reached, in terms of $C_1$, $C_2$ and $V_{\text{in}}$. Clearly show your work.

**Solution:**

The equivalent circuit during phase 1 is:

\[
\begin{align*}
V_{\text{in}} & \quad + \\
+ & \quad + \\
- & \quad - \\
\end{align*}
\]

The equivalent circuit during phase 2 is:

\[
\begin{align*}
+ & \quad + \\
& \quad - \\
+ & \quad - \\
\end{align*}
\]

We can see that during phase 2 there are two floating nodes: $u_{\text{mid}}$ and $u_{\text{out}}$. Node $u_{\text{mid}}$ is connected to the “+” plate of $C_1$ and the “−” plate of $C_2$. We will first calculate the
charge on those plates in phase 1:
\[ Q_{\phi_1}^{u_{mid}} = C_1 V_{in} - C_2 V_{in}. \]
And then the charge on that node during phase 2:
\[ Q_{\phi_2}^{u_{mid}} = C_1 u_{mid} - C_2 (u_{out} - u_{mid}). \]
Equating the two we get:
\[ C_1 V_{in} - C_2 V_{in} = C_1 u_{mid} - C_2 (u_{out} - u_{mid}). \]
(1)
Next, we will look at node \( u_{out} \), which is connected to the “+” plate of \( C_2 \) during phase 2. For the charge stored on that \( u_{out} \) during phase 1 we have:
\[ Q_{\phi_1}^{u_{out}} = C_2 V_{in}. \]
The charge on that node during phase 2:
\[ Q_{\phi_2}^{u_{out}} = C_2 (u_{out} - u_{mid}). \]
Equating the two we get:
\[ C_2 V_{in} = C_2 (u_{out} - u_{mid}) \Rightarrow V_{in} = u_{out} - u_{mid}. \]
(2)
Plugging (2) into (1) we get:
\[ C_1 V_{in} - C_2 V_{in} = C_1 u_{mid} - C_2 V_{in} \]
\[ \Rightarrow V_{in} = u_{mid}. \]
(3)
Finally, plugging (3) into (2) we get:
\[ V_{out} = u_{out} - 0 = 2V_{in}. \]

7. Hyperspectral Classification of Tomatoes
You’re a high-tech farmer who just bought a new hyperspectral sensor to monitor your crops.
NOTE: You do not need to understand how a hyperspectral sensor works to solve this problem.

You attach the sensor to a drone and fly it over your crops, taking measurements of the hyperspectral signature for different points along the field. You want to use these measurements to identify which crops are healthy and which crops are getting sick. Your sensor gives you a spectral signature for each plant as a length 5 vector, where each entry of the vector represents a different frequency. Scientists have determined that healthy versus sick tomato plants will have the following spectral signatures as shown in Figure 11.
They can also be represented in vector form:

\[
\vec{s}_h = \begin{bmatrix} 3 \\ 1 \\ 0 \\ 2 \\ 4 \end{bmatrix}, \quad \vec{s}_s = \begin{bmatrix} 1 \\ 0 \\ 2 \\ 4 \\ 3 \end{bmatrix}
\]

(a) Using your spectral sensor, you measure the following spectral signature for one of your tomato plants as shown in Figure 12. This measurement has some noise in it.
The spectral signatures for healthy, sick, and your measured tomato plants can also be represented in vector form as

\[
\vec{s}_h = \begin{bmatrix} 3 \\ 1 \\ 0 \\ 2 \\ 4 \end{bmatrix}, \quad \vec{s}_s = \begin{bmatrix} 1 \\ 0 \\ 2 \\ 4 \\ 3 \end{bmatrix}, \quad \vec{s}_m = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 2 \end{bmatrix}.
\]

Since spectral signatures never exactly match, the standard procedure is to calculate the angle between signature vectors to determine how close they are. Compute the angle between \(\vec{s}_m\) and \(\vec{s}_h\) and the angle between \(\vec{s}_m\) and \(\vec{s}_s\). Is your measured vector closer to the sick plants or the healthy plants? Classify your plant’s health based on the angle between your measured spectral signature (\(\vec{s}_m\)) and the known spectral signatures, (\(\vec{s}_h\), \(\vec{s}_s\)). Show your work and justify your answer.

**NOTE:** Table 2 can be helpful for finding the angles.

<table>
<thead>
<tr>
<th>(\cos(\theta))</th>
<th>(\theta(\degree))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\frac{9}{10})</td>
<td>47.87</td>
</tr>
<tr>
<td>(\frac{\sqrt{180}}{11})</td>
<td>41.81</td>
</tr>
<tr>
<td>(\frac{\sqrt{180}}{12})</td>
<td>34.93</td>
</tr>
<tr>
<td>(\frac{\sqrt{180}}{13})</td>
<td>26.57</td>
</tr>
<tr>
<td>(\frac{\sqrt{180}}{14})</td>
<td>14.31</td>
</tr>
</tbody>
</table>

**Solution:**

The solution to this problem is to use the inner product formula to compute the angles:

\[
\cos(\theta) = \frac{\langle \vec{s}_1, \vec{s}_2 \rangle}{\|\vec{s}_1\| \|\vec{s}_2\|}.
\]
Comparing the measurement with the spectral signature for the healthy plant, we get:

$$\cos(\theta_1) = \frac{\langle \vec{s}_m, \vec{s}_h \rangle}{\|\vec{s}_m\| \|\vec{s}_h\|} = \frac{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 2 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix}}{\sqrt{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 2 \end{bmatrix}} \sqrt{\begin{bmatrix} 3 \\ 1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix}} = \frac{13}{\sqrt{6 \times \sqrt{30}}} = \frac{13}{\sqrt{180}}.$$ 

Comparing the measurement with the spectral signature for the sick plant, we get:

$$\cos(\theta_2) = \frac{\langle \vec{s}_m, \vec{s}_s \rangle}{\|\vec{s}_m\| \|\vec{s}_s\|} = \frac{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}}{\sqrt{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 2 \end{bmatrix}} \sqrt{\begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}} = \frac{11}{\sqrt{6 \times \sqrt{30}}} = \frac{11}{\sqrt{180}}.$$ 

From the cosine table, we get that $\theta_1 = 14.31^\circ$ and $\theta_2 = 34.93^\circ$. The measured tomato plant has a smaller angle with the healthy plants. Therefore, we choose to deem the plant as probably being healthy.

(b) It’s a windy day and the drone got pushed as it was taking a measurement, so now the measurement has a linear combination of measurements for several different tomato plants (some of which are healthy
and some of which are sick). So your measurement is

\[ s_m = \alpha s_h + \beta s_s + \vec{e}, \]  

(4)

where \( \vec{e} \) represents an error vector that is unknown.

The values you get for your measurement are:

\[
\begin{bmatrix}
5 \\
1 \\
4 \\
10 \\
10
\end{bmatrix}.
\]

The measurement is also shown in Figure 13.

 Recall that

\[
\begin{bmatrix}
3 \\
1 \\
0 \\
2 \\
4
\end{bmatrix}, \quad \begin{bmatrix}
1 \\
0 \\
2 \\
4 \\
3
\end{bmatrix}.
\]

You want to identify the unknowns \( \alpha \) and \( \beta \). Write a least squares problem in the format \( \mathbf{A}\vec{x} = \vec{b} \) to identify the unknowns \( \alpha \) and \( \beta \). Show your work. You do not have to solve for \( \alpha \) and \( \beta \).

**Solution:** We write out equation (4) into a matrix-vector form and take an approximation to account for the added error:

\[
\begin{aligned}
\vec{s}_m & = \alpha \vec{s}_h + \beta \vec{s}_s + \vec{e} \\
& \approx \alpha \vec{s}_h + \beta \vec{s}_s \\
& = \begin{bmatrix}
\vec{s}_h & \vec{s}_s
\end{bmatrix} \begin{bmatrix}
\alpha \\
\beta
\end{bmatrix}
\end{aligned}
\]
so \( A = \begin{bmatrix}
\vec{s}_h & \vec{s}_s \\
\vec{s}_p & \vec{s}_p & \vec{s}_s & \vec{s}_s \\
\end{bmatrix} = \begin{bmatrix}
3 & 1 \\
1 & 0 \\
0 & 2 \\
2 & 4 \\
4 & 3 \\
\end{bmatrix} , \quad \vec{x} = \begin{bmatrix}
\alpha \\
\beta \\
\end{bmatrix}, \quad \vec{b} = \vec{s}_m = \begin{bmatrix}
5 \\
1 \\
4 \\
10 \\
10 \\
\end{bmatrix}.

(c) Your drone got pushed by the wind again, but this time it was while it was taking a measurement on the border of three adjacent fields - your tomato, pepper, and avocado fields.

Tomato, pepper, and avocado plants have unique spectral signatures with a length of 5. The notations are described as the following:

- \( \vec{s}_h \) and \( \vec{s}_s \) represent the spectral signatures of healthy and sick tomato plants
- \( \vec{s}_{ph} \) and \( \vec{s}_{ps} \) represent the spectral signatures of healthy and sick pepper plants
- \( \vec{s}_{ah} \) and \( \vec{s}_{as} \) represent the spectral signatures of healthy and sick avocado plants

Your measurement is now a linear combination of 6 possible spectral signatures:

\[
\vec{s}_m = \alpha_1 \vec{s}_h + \beta_1 \vec{s}_s + \alpha_2 \vec{s}_{ph} + \beta_2 \vec{s}_{ps} + \alpha_3 \vec{s}_{ah} + \beta_3 \vec{s}_{as}.
\]  

(5)

Here \( \alpha_1, \alpha_2, \alpha_3 \) are the unknown weights of healthy tomato, pepper, and avocado plants respectively. \( \beta_1, \beta_2, \beta_3 \) are the unknown weights of sick tomato, pepper, and avocado plants respectively. **Is it possible to uniquely determine the weights of healthy/sick tomatoes, peppers, and avocados from your measurement in equation 5?** Why or why not? Show your work and justify your answer.

**Solution:** We write out equation (5) into a matrix-vector form:

\[
\vec{s}_m = \begin{bmatrix}
\vec{s}_h \\
\vec{s}_{ph} \\
\vec{s}_{ah} \\
\vec{s}_s \\
\vec{s}_{ps} \\
\vec{s}_{as} \\
\end{bmatrix} = \begin{bmatrix}
\alpha_1 \\
\alpha_2 \\
\alpha_3 \\
\beta_1 \\
\beta_2 \\
\beta_3 \\
\end{bmatrix}.
\]

We will not be able to uniquely determine the unknown weights. This problem is underdetermined and has many possible solutions, since the matrix is of size 5 \( \times \) 6 and we therefore have more unknowns than we have equations.
8. Cross-correlation

We are building our own Acoustic Positioning System.

**NOTE:** The signatures \( \vec{s}_1, \vec{s}_2 \) in each sub-part are different; each prompt is independent from the others.

(a) We have two signatures/gold codes of length-5, given by \( \vec{s}_1 \) and \( \vec{s}_2 \) as in Figure 14. So far we have numerically computed their linear cross-correlation \( \text{Corr}_{\vec{s}_1}(\vec{s}_2) \), yet a few entries have been tragically lost! Fortunately we can compute these omitted terms by hand. **Please compute the missing cross-correlation values at shifts** \( k = -1 \text{ and } k = +2 \). **Show your work and justify your answer.**

\[
\vec{s}_1 = \begin{bmatrix} +1 \\ 0 \\ -1 \\ 0 \\ +1 \end{bmatrix} \quad \vec{s}_2 = \begin{bmatrix} +1 \\ +1 \\ 0 \\ -1 \\ +1 \end{bmatrix}
\]

**Figure 14:** Linear cross-correlation plot of the two signals \( \text{Corr}_{\vec{s}_1}(\vec{s}_2) \). The x-axis represents the shift.

**Solution:**

To find the cross-correlation value at shift \( k = -1 \), we may directly compute it from \( \text{Corr}_{\vec{s}_1}(\vec{s}_2)[-1] = \sum_n \vec{s}_1[n] \vec{s}_2[n + 1] \) over all nonzero terms, so for \( n = 1, 2, 3, 4 \):

\[
\sum_{n=1}^{4} \vec{s}_1[n] \vec{s}_2[n + 1] = (1 \cdot 1) + (0 \cdot 0) + (-1 \cdot -1) + (0 \cdot 1) = 2.
\]

Next, we compute \( \text{Corr}_{\vec{s}_1}(\vec{s}_2)[+2] = \sum_n \vec{s}_1[n] \vec{s}_2[n - 2] \) but now over just \( n = 3, 4, 5 \):

\[
\sum_{n=3}^{5} \vec{s}_1[n] \vec{s}_2[n - 2] = (-1 \cdot 1) + (0 \cdot 1) + (1 \cdot 0) = -1.
\]
(b) We are trying out some new codes $\vec{s}_1$ and $\vec{s}_2$. We only know that the codes are normalized ($\langle \vec{s}_1, \vec{s}_1 \rangle = 1$, $\langle \vec{s}_2, \vec{s}_2 \rangle = 1$) and their inner-product is $\langle \vec{s}_1, \vec{s}_2 \rangle = 0.3$. During our test we have received the signal $\vec{r} = \frac{1}{2} \vec{s}_1 + \frac{1}{3} \vec{s}_2$. Without knowing any more information about our codes, compute $\text{Corr}_{\vec{r}}(\vec{s}_1)$ at the shift $k = 0$. Show your work and justify your answer.

Solution:
We can only compute the requested cross-correlation because at $k = 0$, the correlation reduces to an inner-product: $\text{Corr}_{\vec{r}}(\vec{s}_1)[k = 0] = \langle \vec{r}, \vec{s}_1 \rangle$. Since the inner-product is a linear operation, we can expand out $\vec{r}$ and find the result.

$$\text{Corr}_{\vec{r}}(\vec{s}_1)[k = 0] = \langle \vec{r}, \vec{s}_1 \rangle$$
$$= \langle \frac{1}{2} \vec{s}_1 + \frac{1}{3} \vec{s}_2, \vec{s}_1 \rangle$$
$$= \frac{1}{2} \langle \vec{s}_1, \vec{s}_1 \rangle + \frac{1}{3} \langle \vec{s}_1, \vec{s}_2 \rangle$$
$$= \frac{1}{2} \cdot 1.0 + \frac{1}{3} \cdot 0.3$$
$$= 0.6 \quad \square$$

(c) We again have two new signals $\vec{s}_1$ and $\vec{s}_2$ and are now given the plot of $\text{Corr}_{\vec{s}_1}(\vec{s}_2)$ as shown in Figure 15. Our receiver identified a signal $\vec{r}$ which we know to be related to the code $\vec{s}_2$ by some scaling, shifting, and/or reflection. However, we only know the linear cross-correlation $\text{Corr}_{\vec{s}_1}(\vec{r})$ as shown in Figure 16. Can you express $\vec{r}$ in terms of $\vec{s}_2$? Show your work and justify your answer.

Solution:
By inspection of Figure 16, the correlation plot of $\vec{s}_1$ with $\vec{r}$ is the same as the correlation plot of $\vec{s}_1$ with $\vec{s}_2$, only altered by a vertical scaling of $1/2$. This is most evident at the $k = -1$ shift, where
Corr_{\vec{s}_1} (\vec{r})[-1] = 2 \text{ and } Corr_{\vec{s}_1} (\vec{s}_2)[-1] = 4.

So far we’ve concluded that Corr_{\vec{s}_1} (\vec{r}) = \frac{1}{2} Corr_{\vec{s}_1} (\vec{s}_2).

Next we must recognize that correlation is a linear operation, so we can see that

\[ Corr_{\vec{s}_1}(\frac{1}{2} \vec{s}_2)[k] = \sum_n \vec{s}_1[n] \frac{1}{2} \vec{s}_2[n - k] = \frac{1}{2} Corr_{\vec{s}_1} (\vec{s}_2)[k]. \]

Thus, the relationship between \( \vec{r} \) and \( \vec{s}_2 \) is

\[ \vec{r}[n] = \frac{1}{2} \vec{s}_2[n]. \]

(d) With a little effort, we managed to create two good gold codes of length 100, \( \vec{s}_1 \) and \( \vec{s}_2 \). The linear cross-correlation of \( \vec{s}_1 \) and \( \vec{s}_2 \) is small at all shifts while the autocorrelation of each signal is also small, except at shift \( k = 0 \). We receive our first signal \( \vec{r} \), which we know to be a combination of both codes, of the following form:

\[ \vec{r}[n] = \vec{s}_1[n - k_1] + \vec{s}_2[n - k_2]. \] (6)

The linear cross-correlation \( Corr_{\vec{r}} (\vec{s}_1) \) has been computed and plotted in Figure 17, and similarly \( Corr_{\vec{r}} (\vec{s}_2) \) is plotted in Figure 18. Determine the shifts for \( \vec{s}_1 \) and \( \vec{s}_2 \) in the received signal \( \vec{r} \), i.e. solve for \( k_1 \) and \( k_2 \) in equation (6). Explain your answer.

Note: Don’t worry too much about identifying the exact value for \( k_1 \) and \( k_2 \). As long as your answer is reasonable, you will receive full credit.
Solution: Let us start substituting $\vec{r}[n] = \vec{s}_1[n-k_1] + \vec{s}_2[n-k_2]$ into our correlation definitions.

$$\text{Corr}_{\vec{r}}(\vec{s}_1)[k] = \sum_n \vec{r}[n] \vec{s}_1[n-k] = \sum_n \vec{s}_1[n-k_1] \vec{s}_1[n-k] + \sum_n \vec{s}_2[n-k_2] \vec{s}_1[n-k]$$

$$\text{Corr}_{\vec{r}}(\vec{s}_2)[k] = \sum_n \vec{r}[n] \vec{s}_2[n-k] = \sum_n \vec{s}_1[n-k_1] \vec{s}_2[n-k] + \sum_n \vec{s}_2[n-k_2] \vec{s}_2[n-k]$$

We approximate the cancellation of terms since codes $\vec{s}_1$ and $\vec{s}_2$ have a small cross-correlation.

From Figure 17 we note for $\text{Corr}_{\vec{r}}(\vec{s}_1)$ the only significant peak occurs at shift $k = -20$. Since the auto-correlation $\text{Corr}_{\vec{s}_1}(\vec{s}_1)$ is peaked at the zero-shift, it must be (based on the top equation) that $k_1 = k = -20$. With similar reasoning (in regards to Figure 18) we identify $k_2 = +10$.

Thus we arrive at our solution for the received signal:

$$\vec{r}[n] = \vec{s}_1[n+20] + \vec{s}_2[n-10].$$

(e) It appears that making codes orthogonal to each other improves the robustness of our Acoustic Positioning System. Knowing this, we want to use our knowledge of projections to write our first code as $\vec{s}_1 = \vec{a} + \vec{b}$, where $\langle \vec{b}, \vec{s}_2 \rangle = 0$ and $\vec{a} = \alpha \vec{s}_2$ (for some constant $\alpha$) as illustrated in Figure 19.

Compute $\alpha$ and $\vec{b}$ in terms of $\vec{s}_1$ and $\vec{s}_2$. Show your work and justify your answer.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure19}
\caption{2D figure of $\vec{s}_1 = \vec{a} + \vec{b}$.}
\end{figure}

Solution:

From the setup, we may observe that $\vec{a}$ is the projection of $\vec{s}_1$ onto $\vec{s}_2$.

$$\vec{a} = \text{proj}_{\vec{s}_2}(\vec{s}_1) = \left( \frac{\langle \vec{s}_1, \vec{s}_2 \rangle}{\langle \vec{s}_2, \vec{s}_2 \rangle} \right) \vec{s}_2 \rightarrow \alpha = \frac{\langle \vec{s}_1, \vec{s}_2 \rangle}{\langle \vec{s}_2, \vec{s}_2 \rangle}$$

Acquiring the new orthogonal code $\vec{b}$ follows from our result above

$$\vec{b} = \vec{s}_1 - \vec{a} = \vec{s}_1 - \left( \frac{\langle \vec{s}_1, \vec{s}_2 \rangle}{\langle \vec{s}_2, \vec{s}_2 \rangle} \right) \vec{s}_2. \quad \square$$

– Alternate Method –

The constant $\alpha$ can also be determined algebraically using the fact that $\vec{b}$ is orthogonal to $\vec{s}_2$:

$$\langle \vec{s}_1, \vec{s}_2 \rangle = \alpha \langle \vec{s}_2, \vec{s}_2 \rangle + \langle \vec{b}, \vec{s}_2 \rangle.$$
Thus we can write \( \alpha \) in terms of \( \vec{s}_1 \) and \( \vec{s}_2 \): \( \alpha = \frac{\langle \vec{s}_1, \vec{s}_2 \rangle}{\langle \vec{s}_2, \vec{s}_2 \rangle} \).

Finding \( \vec{b} \) from this point follows identically as shown above:

\[
\vec{b} = \vec{s}_1 - \vec{\alpha} \vec{s}_2 = \vec{s}_1 - \left( \frac{\langle \vec{s}_1, \vec{s}_2 \rangle}{\langle \vec{s}_2, \vec{s}_2 \rangle} \right) \vec{s}_2. 
\]

(f) After optimizing two orthogonal codes \( \vec{s}_1 \) and \( \vec{s}_2 \) (i.e. \( \langle \vec{s}_1, \vec{s}_2 \rangle = 0 \)), we would next like to include another code \( \vec{s}_3 \) and make it orthogonal to \( \vec{s}_1 \) and \( \vec{s}_2 \). We can start by writing \( \vec{s}_3 \) as \( \vec{s}_3 = \vec{a} + \vec{b} \), such that \( \vec{a} \) belongs to the span\{\( \vec{s}_1, \vec{s}_2 \)\} and \( \vec{b} \) is orthogonal to \( \vec{s}_1 \) and \( \vec{s}_2 \) (i.e. \( \langle \vec{b}, \vec{s}_1 \rangle = 0 \) and \( \langle \vec{b}, \vec{s}_2 \rangle = 0 \)). Use the idea of projections to write both \( \vec{a} \) and \( \vec{b} \) in terms of \( \vec{s}_1 \), \( \vec{s}_2 \), and \( \vec{s}_3 \), and inner-products thereof. (For full credit your final answer may not contain matrices nor matrix-vector products). Show your work and justify your answer.

**Solution:**

In least-squares method, the minimizing solution \( \hat{x} \) to the system \( A\vec{x} = \vec{y} \) will result in \( A\hat{x} \) producing the projection of vector \( \vec{y} \) onto the column space of \( A \), in which the error vector \( \vec{e} = \vec{y} - A\hat{x} \) is orthogonal to the span of the column vectors in \( A \). In this problem’s context, we need to acquire

\[
\vec{\bar{b}} = \vec{s}_3 - A\hat{x},
\]

in which the original signal \( \vec{s}_3 \) must be projected onto \( A \equiv \begin{bmatrix} \uparrow & \uparrow \\ \vec{s}_1 & \vec{s}_2 \end{bmatrix} \). So we may apply our least-squares formula to acquire \( \vec{\bar{b}} = A\hat{x} = A(A^TA)^{-1}A^T\vec{s}_3 \). But since \( A \) has orthogonal columns, we can substantially simplify this expression:

\[
A\hat{x} = \begin{bmatrix} \uparrow & \uparrow \\ \vec{s}_1 & \vec{s}_2 \end{bmatrix} \begin{bmatrix} \leftarrow & \vec{s}_1^T \rightarrow \\ \vec{s}_1 & \vec{s}_2 \end{bmatrix}^{-1} \begin{bmatrix} \leftarrow & \vec{s}_1^T \rightarrow \\ \vec{s}_1 & \vec{s}_2 \end{bmatrix} \begin{bmatrix} \uparrow & \uparrow \\ \vec{s}_1 & \vec{s}_2 \end{bmatrix} \begin{bmatrix} \rightarrow & \vec{s}_3 \rightarrow \\ \vec{s}_3 \end{bmatrix}
\]

\[
= \begin{bmatrix} \uparrow & \uparrow \\ \vec{s}_1 & \vec{s}_2 \end{bmatrix} \begin{bmatrix} ||\vec{s}_1||^2 & 0 \\ 0 & ||\vec{s}_2||^2 \end{bmatrix}^{-1} \begin{bmatrix} \leftarrow & \vec{s}_1^T \rightarrow \\ \vec{s}_1 & \vec{s}_2 \end{bmatrix} \begin{bmatrix} \uparrow & \uparrow \\ \vec{s}_1 & \vec{s}_2 \end{bmatrix} \begin{bmatrix} \rightarrow & \vec{s}_3 \rightarrow \\ \vec{s}_3 \end{bmatrix}
\]

\[
= \begin{bmatrix} \uparrow & \uparrow \\ \vec{s}_1 & \vec{s}_2 \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \langle \vec{s}_1, \vec{s}_1 \rangle \\ \langle \vec{s}_2, \vec{s}_1 \rangle \end{bmatrix} \\ \begin{bmatrix} \langle \vec{s}_1, \vec{s}_2 \rangle \\ \langle \vec{s}_2, \vec{s}_2 \rangle \end{bmatrix} \end{bmatrix}^{-1} \begin{bmatrix} \uparrow \\ \vec{s}_1 \\ \vec{s}_2 \end{bmatrix} \begin{bmatrix} \rightarrow \\ \vec{s}_3 \end{bmatrix}
\]

\[
= \begin{bmatrix} \uparrow & \uparrow \\ \vec{s}_1 & \vec{s}_2 \end{bmatrix} \begin{bmatrix} \langle \vec{s}_1, \vec{s}_1 \rangle \\ \langle \vec{s}_2, \vec{s}_1 \rangle \\ \langle \vec{s}_1, \vec{s}_2 \rangle \\ \langle \vec{s}_2, \vec{s}_2 \rangle \end{bmatrix}^{-1} \begin{bmatrix} \uparrow \\ \vec{s}_1 \\ \vec{s}_2 \end{bmatrix} \begin{bmatrix} \rightarrow \\ \vec{s}_3 \end{bmatrix}
\]

Finally, we can subtract this projection from the original signal \( \vec{s}_3 \) to obtain an orthogonal code to both
9. Warm for the Holidays

Winter is coming, and both you and your roommate are in desperate need of electric heating eye pads to avoid overly dry eyes this holiday. Tragically the circuit for your eye pads broke, yet fortunately you’ve taken EECS16A and have come up with a clever fix by designing a voltage divider and a comparator circuit!

(a) First you build a circuit that converts temperature change to voltage change. Your design is shown in Fig. 20. In your design you use two temperature dependent resistors, whose values are given by $R_0 + \alpha T$, $R_0 - \alpha T$, where $R_0$ is the resistor value at 0 degrees centigrade, $\alpha$ is a thermal coefficient, and $T$ is the temperature of your eye pads.

What is the temperature dependent output voltage, $V_T$, of this circuit, as a function of $V_s$, $R_0$, $\alpha$, and $T$? Is $V_T$ a linear function of $T$? Clearly show all your work.

Solution: This circuit is essentially a voltage divider:

$$V_T = \frac{R_0 + \alpha T}{R_0 + \alpha T + R_0 - \alpha T} V_s = \frac{R_0 + \alpha T}{2R_0} V_s.$$  

This is an affine function of $T$ since there is an offset term in our final expression.

(b) We want to use a comparator to turn the heat ON and OFF, and you set up the circuit in Fig. 21. You process the $V_T$ to make $V_{in} = (1 - \frac{T}{T_0})[\text{Volts}]$, where $T_0 = 30^\circ\text{C}$. The heat will turn on when
$$V_{out} = V_{DD}.$$
For what range of temperatures, $T$, is $V_{out} = V_{DD}$? Give your answer in terms of °C. Clearly show all your work.

$$V_{in} > 0 \rightarrow 1 - \frac{T}{T_0} > 0 \implies T < T_0,$$

which means that $V_{out} = V_{DD}$ (i.e. the heater turns ON) for all temperatures that are below 30°C.

(c) Your TA, Moses, points out that just using the circuit in Figure 21 will cause your heat to turn ON and OFF due to very small fluctuations. Instead, he suggests analyzing the following circuit in Figure 22. Find the voltage $u_+$ at the positive terminal of the comparator, as a function of $V_{out}$, $R_1$, $R_2$, and $V_{ref}$. Clearly show all your work.
A common misconception here is to assume that the circuit is in negative feedback. It is not, since the output connects back to the “+” terminal of the op-amp. We can compute the node voltage using NVA and the fact that no current will flow into the op-amp. Applying KCL at the “+” terminal of the comparator, and because there is no current entering the “+” terminal of the comparator we get:

\[ I_{R_1} = I_{R_2}. \]

Substituting the currents using Ohm’s law we get:

\[
\frac{V_{\text{ref}} - u_+}{R_1} = \frac{u_+ - V_{\text{out}}}{R_2} \implies u_+ = \frac{R_2}{R_1 + R_2} V_{\text{ref}} + \frac{R_1}{R_1 + R_2} V_{\text{out}}.
\]

10. Least Squares for Robotics

Robots rely on sensors for understanding their environment and navigating in the real world. These sensors must be calibrated to ensure accurate measurements, which we explore in this problem.

(a) Your robot is equipped with two forward-facing sensors – a radar and camera. However, the sensors are placed with an offset (i.e. a gap) of \( \ell \) in meters (m), as depicted in Fig. 23, and you want to find its value. The radar returns a range \( \rho \) in meters (m) and heading angle \( \theta \) in radians (rad) with respect to the object. In contrast, the camera only returns an angle, \( \phi \) in radians (rad), with respect to the object.
These relationships are summarized by the following sensor model, where $x_r$ and $y_r$ are the Cartesian coordinates of the object with respect to the radar:

\[ x_r = \rho \cos(\theta), \]
\[ y_r = \rho \sin(\theta), \]
\[ \tan(\phi) = \frac{y_r}{x_r + \ell}. \]

Assuming $\phi \neq 0$, use equations (7), (8), (9) to express $\ell$ in terms of $\rho$, $\theta$, and $\phi$.

**Solution:**
From the sensor model, we have:

\[ (x_r + \ell) \tan(\phi) = y_r \]
\[ \Rightarrow \ell = \frac{yr}{\tan(\phi)} - x_r \]
\[ \Rightarrow \ell = \rho \left( \frac{\sin(\theta)}{\tan(\phi)} - \cos(\theta) \right). \]

**Note:** We stipulate that $\phi \neq 0$ since otherwise division by $\tan(\phi)$ would not be well-defined. When $\phi = 0$, the object would be located right in front of both the radar and camera, and any positive value of $\ell$ would solve the system of equations. This explanation is not required for full credit.

(b) Often it is difficult to precisely identify the value of $\ell$. To learn the value of $\ell$ you decide to take a series of measurements. In particular, you take $N$ measurements and get the equations:

\[ a \ell + e_i = b_i \]

for $1 \leq i \leq N$. Here $a \neq 0$ is a fixed and known constant. Each $b_i$ represents your $i^{th}$ measurement and $e_i$ represents the error in your measurement. While you know all of the $b_i$ values, you do not know the error values $e_i$.

We can write this equation in a vector format as:

\[ A \ell + \vec{e} = \vec{b}, \]

where $A = \begin{bmatrix} a & \vdots \\ \vdots \\ a \end{bmatrix}$, $\vec{e} = \begin{bmatrix} e_1 \\ \vdots \\ e_N \end{bmatrix}$, $\vec{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_N \end{bmatrix}$.

In this simple 1-D case, the least squares solution is a scaled version of the average of $\{b_i\}_{i=1}^N$.

Find the best estimate for $\ell$, denoted as $\hat{\ell}$, using least squares. Simplify your expression and
express \hat{\ell} in terms of \(a, b_i,\) and \(N\). Your answer may not include any vector notation.

*Note: \(A\) is a vector and not a matrix.*

**Solution:**

\(\hat{\ell}\) is given by the least square solution:

\[
\hat{\ell} = (A^T A)^{-1} A^T \tilde{b}
\]

\[
= (Na^2)^{-1} a \sum_{i=1}^{N} b_i
\]

\[
= \frac{\sum_{i=1}^{N} b_i}{aN}.
\]

(c) Now we turn to the task of controlling the robot’s velocity and acceleration, which is a key requirement for navigation.

We use the following model for the robot, which describes how the velocity and acceleration of the robot changes with timestep \(k\):

\[
\begin{bmatrix}
  v[k+1] \\
  a[k+1]
\end{bmatrix} =
\begin{bmatrix}
  1 & 1 \\
  0 & 1
\end{bmatrix}
\begin{bmatrix}
  v[k] \\
  a[k]
\end{bmatrix} +
\begin{bmatrix}
  0 \\
  1
\end{bmatrix} j[k],
\]

where

- \(k\) is the timestep;
- \(v[k]\) is the velocity state at timestep \(k\);
- \(a[k]\) is the acceleration state at timestep \(k\);
- \(j[k]\) is the jerk (derivative of acceleration) control input at timestep \(k\).

We start at a known initial state \([v[0] a[0]]\), and we want to find \(j[0]\) to set \([v[1] a[1]]\) as close to \([0 0]\) as possible. For this, we minimize:

\[
E = \left\| \begin{bmatrix}
  v[1] \\
  a[1]
\end{bmatrix} \right\|^2.
\]

**Find the best estimate for the optimal choice of jerk, \(\hat{j}[0]\), by using least squares method to minimize \(E\). Express your solution in terms of \(v[0]\) and \(a[0]\). Show your work.**

*Hint: Rewrite \(E\) in terms of \(j[0]\) and other relevant terms.*

**Solution:**

Starting from the hint, we try to rewrite the cost \(E\). Applying the dynamics model, we find that:

\[
E = \left\| \begin{bmatrix}
  v[0] + a[0] \\
  a[0] + j[0]
\end{bmatrix} \right\|^2.
\]
\[ = \left\| \begin{bmatrix} 0 \\ 1 \end{bmatrix} j[0] - \begin{bmatrix} -v[0] - a[0] \\ -a[0] \end{bmatrix} \right\|^2 \]
\[ = \left\| A j[0] - \vec{b} \right\|^2. \]

Therefore, \( \hat{j}[0] \) is given by the least square solution:
\[ j[0] = (A^T A)^{-1} A^T \vec{b} \]
\[ = (1)^{-1} \times (-a[0]) \]
\[ = -a[0]. \]

11. Proof

Let \( A, B \in \mathbb{R}^{n \times n} \). The eigenvalues and eigenvectors of \( A \) are given by \((\alpha_1, \vec{v}_1), (\alpha_2, \vec{v}_2), \ldots, (\alpha_n, \vec{v}_n)\), where all the \( \alpha_i \), \( 1 \leq i \leq n \), are distinct. Similarly the eigenvalues and eigenvectors of \( B \) are given by \((\beta_1, \vec{v}_1), (\beta_2, \vec{v}_2), \ldots, (\beta_n, \vec{v}_n)\), where all the \( \beta_i \), \( 1 \leq i \leq n \), are distinct.

\( \text{NOTE: } A, B \text{ have identical eigenvectors. } \)

Prove that:

\[ A B \vec{x} = B A \vec{x}, \]

for any vector \( \vec{x} \in \mathbb{R}^n \).

\( \text{Solution: } \)

To prove that the matrices \( A \) and \( B \) commute for any \( \vec{x} \in \mathbb{R}^n \), we must first be able to write any such \( \vec{x} \) in terms of the shared matrix eigenvectors \( \vec{x} = \sum_{j=1}^{n} c_j \vec{v}_j \). We know that this is true from the theorems proved in lecture and the notes, since \( \{ \vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n \} \) forms a basis for \( \mathbb{R}^n \). You must acknowledge this to receive full credit.

Since eigenvectors \( \vec{v}_j \) form a basis of \( \mathbb{R}^n \), we can write any vector as a linear combination of the eigenvectors, i.e. \( \vec{x} = \sum_{j=1}^{n} c_j \vec{v}_j \).
We know that any $\vec{x} \in \mathbb{R}^n$ can be uniquely expressed in the identical basis of eigenvectors for $A$ and $B$.

$$A B \vec{x} = \sum_{j=1}^{n} c_j A B \vec{v}_j$$

$$= \sum_{j=1}^{n} c_j A \beta_j \vec{v}_j$$

$$= \sum_{j=1}^{n} c_j \beta_j A \vec{v}_j$$

$$= \sum_{j=1}^{n} c_j \beta_j \alpha_j \vec{v}_j.$$  

Similarly,

$$B A \vec{x} = \sum_{j=1}^{n} c_j B A \vec{v}_j$$

$$= \sum_{j=1}^{n} c_j B \alpha_j \vec{v}_j$$

$$= \sum_{j=1}^{n} c_j \alpha_j B \vec{v}_j$$

$$= \sum_{j=1}^{n} c_j \alpha_j \beta_j \vec{v}_j$$

$$= \sum_{j=1}^{n} c_j \beta_j \alpha_j \vec{v}_j.$$  

Therefore both the expressions are equal. The key property we are exploiting is that each $\vec{v}_j$ is simultaneously an eigenvector of $A$ and $B$.

The only other property we needed was matrix linearity $A(a \vec{x} + b \vec{y}) = a A \vec{x} + b A \vec{y}$. This concludes the proof.
In case you want a reminder on how to show that \( \{ \vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n \} \) forms a basis for \( \mathbb{R}^n \):

It was shown in lecture that the eigenvectors of a matrix with entirely distinct eigenvalues are all mutually linearly independent and thus form a basis for \( \mathbb{R}^n \).

i. Fundamental Idea: Since each \( \vec{v}_j \) lives in \( \mathbb{R}^n \) and there are \( n \) such vectors, the set of eigenvectors will form a basis of \( \mathbb{R}^n \) if and only if they are linearly independent.

ii. Prepare contradiction: Suppose the opposite were true; the eigenvectors are linearly dependent so there is a set of constants \( d_j \) (not all of which are zero) we can choose such that \( \sum_{j=1}^n d_j \vec{v}_j = \vec{0} \). Now we know multiplying \( \vec{0} \) by any matrix will still result in zero, and furthermore any inner product \( \langle \vec{a}, \vec{0} \rangle \) will also be zero (regardless of \( \vec{a} \)).

iii. Arriving at absurdity: Without loss of generality, say \( \alpha_n \) is the eigenvalue of \( A \) with the greatest absolute value \( |\alpha_n| > |\alpha_j| \) for any \( j \in 1, 2, \ldots, n-1 \). 

Next compute the inner product of the expression \( \langle \vec{v}_n, \frac{1}{\alpha_n} A^N \vec{0} \rangle \) (which must always be zero) for any positive integer \( N \).

\[
0 \equiv \langle \vec{v}_n, \frac{1}{\alpha_n} A^N \vec{0} \rangle = \sum_{j=1}^n \frac{d_j}{\alpha_n} \langle \vec{v}_n, A^N \vec{v}_j \rangle = \sum_{j=1}^n d_j \left( \frac{\alpha_j}{\alpha_n} \right)^N \langle \vec{v}_n, \vec{v}_j \rangle.
\]

In the large \( N \) limit, the term in parentheses will vanish for all terms except for \( j = n \). The only remaining term is \( \left( \frac{\alpha_n}{\alpha_n} \right)^N = 1 \). Thus we arrive at the final contradiction

\[
0 \equiv d_n |\vec{v}_n|^2 > 0. \quad \square
\]

This means there is no choice of \( d_j \) such that \( \sum_{j=1}^n d_j \vec{v}_j = \vec{0} \).

Q.E.D.

Small note:

In the event that our linear combination happened to have \( d_n = 0 \), then we can return to step 'ii.' with \( n - 1 \) in place of \( n \) (since \( d_n = 0 \)).

While it could be that \( d_{n-1} = 0 \) as well, the procedure here can be continuously applied until you reach a nonzero \( d_j \). Further, there must be a nonzero \( d_j \) as required by the very definition of linear dependence!!

12. 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 friends have decided to start a new SF tour business, and you suggest they use segways. They become intrigued by your idea and asks you how a segway works. A segway is essentially a stand on two wheels.

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 talk to a 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 $\theta$, 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 $n$, which is the force $F$ applied to the cart (or base of the segway).

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

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

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

The control $u[n]$ allows us to move the state ($\vec{x}$) in the direction of $\vec{b}$. 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 state vector, $\vec{x}$, 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 $u[0]$.

**Solution:**

From Equation (10), we get (by substituting $n = 0$):

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

---

1You 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)  

i. Express $\vec{x}[2]$ in terms of *only* $\vec{x}[0]$ and the inputs, $u[0]$ and $u[1]$.

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

iii. Finally express $\vec{x}[4]$ in terms of *only* $\vec{x}[0]$ and the inputs, $u[0], u[1], u[2]$, and $u[3]$.

Your expressions can have other relevant variables (e.g. $A, \vec{b}$ etc) and mathematical operators.

**Solution:**

From Equation (10), we get (by substituting $n = 1$):

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

By substituting $\vec{x}[1]$ from Equation (11), we get

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

(12)

From Equation (10), we get (by substituting $n = 2$):

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

By substituting $\vec{x}[2]$ from Equation (12), we get

$$\vec{x}[3] = \vec{A}\vec{x}[2] + \vec{b}u[2]
= \vec{A} (\vec{A}^{2}\vec{x}[0] + \vec{A}\vec{b}u[0] + \vec{b}u[1]) + \vec{b}u[2]
= \vec{A}^{3}\vec{x}[0] + \vec{A}^{2}\vec{b}u[0] + \vec{A}\vec{b}u[1] + \vec{b}u[2]$$

(13)

From Equation (10), we get (by substituting $n = 3$):

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

By substituting $\vec{x}[3]$ from Equation (13), we get

$$\vec{x}[4] = \vec{A}\vec{x}[3] + \vec{b}u[3]
= \vec{A} (\vec{A}^{3}\vec{x}[0] + \vec{A}^{2}\vec{b}u[0] + \vec{A}\vec{b}u[1] + \vec{b}u[2]) + \vec{b}u[3]
= \vec{A}^{4}\vec{x}[0] + \vec{A}^{3}\vec{b}u[0] + \vec{A}^{2}\vec{b}u[1] + \vec{A}\vec{b}u[2] + \vec{b}u[3]$$

(14)

(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 $u[0], \ldots, u[N−1]$. Your expression can have other relevant variables (e.g. $A, \vec{b}$ etc) and mathematical operators.

**Solution:**

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

$$\vec{x}[N] = \vec{A}^{N}\vec{x}[0] + \vec{A}^{N−1}\vec{b}u[0] + \cdots + \vec{A}\vec{b}u[N−2] + \vec{b}u[N−1]$$

(15)
You might also use the compact expression:

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

(16)

Note that \(A^0\) is the identity matrix.

As a sanity check, plug the values \(N = 1, 2, 3,\) and \(4\) to obtain Equations (11), (12), (13), and (14), respectively.

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

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

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

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 \(u[0], u[1], \ldots\) etc. 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 \((u[0], u[1], \ldots\) etc\), such that we get \(\vec{x}[N]\) (i.e. state vector at \(N\)th time step) equal to \(\vec{x}_f = \vec{0}\).

Note: Please use the Jupyter notebook to solve parts (d) - (g) of the problem. You may use the function we provided \texttt{gauss\_elim(matrix)} to help you find the upper triangular form of matrices. An example of Gaussian Elimination using \texttt{(gauss\_elim(matrix))}\ is provided in the Jupyter notebook under section Example Usage of \texttt{gauss\_elim}. You may also use the function \texttt{(np.linalg.solve)} to solve the equations.

(d) Can you reach \(\vec{x}_f\) in two time steps? Show work to justify your answer. You should manipulate the equations on paper, but then use the Jupyter notebook for numerical computations.

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

\textbf{Solution:}

\textit{No.}

From Equation (12), we know that \(A^2 \vec{x}[0] + A \vec{b}u[0] + \vec{b}u[1] = \vec{x}[2]\) which is equivalent to \(A \vec{b}u[0] + \vec{b}u[1] = \vec{x}[2] - A^2 \vec{x}[0].\)

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 \(\vec{x}_f = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}\), 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

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

which yields the following augmented matrix:

\[
\begin{bmatrix}
\mathbf{A} \vec{b} & \vec{b} & -\mathbf{A}^2 \vec{x}[0] \\
\end{bmatrix}.
\]

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

\[
\begin{bmatrix}
0.0208 & 0.01 & 0.02243475295 \\
0.0557 & 0.21 & -0.30785116611 \\
-0.0572 & -0.03 & 0.0619347608 \\
-0.2385 & -0.44 & 1.38671325508 \\
\end{bmatrix}.
\]

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

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
\end{bmatrix},
\]

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? Show work to justify your answer. You should manipulate the equations on paper, but then use the Jupyter notebook for numerical computations.

(Hint: Similar to the last part, express \(\vec{x}[3] - \mathbf{A}^3 \vec{x}[0]\) 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 (13), we know that $A^3 \vec{x}[0] + A^2 \vec{b}u[0] + A\vec{b}u[1] + \vec{b}u[2] = \vec{x}[3]$, which is equivalent to $A^2 \vec{b}u[0] + A\vec{b}u[1] + \vec{b}u[2] = \vec{x}[3] - A^3 \vec{x}[0]$.

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:

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

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

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

In matrix form, this system of linear equations is

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

which yields the following augmented matrix:

$$\begin{bmatrix} A^2 \vec{b} & A\vec{b} & \vec{b} & -A^3 \vec{x}[0] \end{bmatrix}.$$ 

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

$$\begin{bmatrix} 0.024157 & 0.0208 & 0.01 & 0.0064228470365 \\ 0.024363 & 0.0557 & 0.21 & -0.092123298431 \\ -0.083488 & -0.0572 & -0.03 & 0.178491836209001 \\ -0.342448 & -0.2385 & -0.44 & 1.246334243328597 \end{bmatrix}.$$

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

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

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? Show work to justify your answer. You should manipulate the equations on paper, but then use the Jupyter notebook for numerical computations. (Use the hints from the last two parts.)
Solution:

Yes.

Similar to the previous part, from Equation (14), we know that $\mathbf{A}^4\mathbf{x}[0] + \mathbf{A}^3\mathbf{b}u[0] + \mathbf{A}^2\mathbf{b}u[1] + \mathbf{b}u[2] + \mathbf{b}u[3] = \mathbf{x}[4]$ which is equivalent to $\mathbf{A}^3\mathbf{b}u[0] + \mathbf{A}^2\mathbf{b}u[1] + \mathbf{b}u[2] + \mathbf{b}u[3] = \mathbf{x}[4] - \mathbf{A}^4\mathbf{x}[0]$.

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

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

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

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

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

In matrix form, this system of linear equations is

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

Refer to the code in the solution IPython notebook for the values of the augmented matrix. After performing Gaussian elimination, we get the upper triangular form to be

$$\begin{bmatrix} 1 & 0 & 0 & -13.24875075 \\ 0 & 1 & 0 & 23.73325125 \\ 0 & 0 & 1 & -11.57181872 \\ 0 & 0 & 0 & 1.46515973 \end{bmatrix}.$$

This indicates that there exists a unique solution to the system of equations, which is

$$\begin{bmatrix} u[0] \\ u[1] \\ u[2] \\ u[3] \end{bmatrix} = \begin{bmatrix} -13.24875075 \\ 23.73325125 \\ -11.57181872 \\ 1.46515973 \end{bmatrix}.$$

(g) If you have found that you can get to the final state in 4 time steps, find the required correct control inputs, i.e. $u[0], u[1], u[2]$ and $u[3]$, using Jupyter and verify the answer by entering these control inputs into the Plug in your controller section of the code in the Jupyter notebook. You need to just show that you reached the desired final state $\mathbf{x}_f$ by plugging in the control inputs. The code has been already 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:

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

The control allows us to move the state at time step \( n + 1 \) by \( u[n] \) in direction \( \vec{b} \), remember \( u[n] \in \mathbb{R} \) is just a scalar. We know from part (c) that:

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

Again, here \( u[0], u[1] \in \mathbb{R} \) can be thought of as arbitrary scalars, and \( A\vec{b}u[0] + \vec{b}u[1] \) can be thought of as the set of all linear combinations of the vectors \( \vec{b} \) and \( A\vec{b} \). Using this observation, can you express the possible states you can arrive at in two time steps using the span of exactly *two* vectors plus a vector offset?

**Solution:** You can move in directions \( \vec{b} \) and \( A\vec{b} \). We also need to account for the zero input behavior of the system reflected by the term \( A^2\vec{x}[0] \). Hence you can reach all positions that are in

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

(i) Let’s try to generalize the idea in the previous part. Express the states you can reach in \( N \) timesteps as a span of some vectors plus a vector offset. (Hint: Consider the direction that each control input \( u[0], \ldots, u[N - 1] \) can move \( \vec{x}[N] \) by.)

**Solution:** In \( N \) timesteps, the control inputs \( u[0], \ldots, u[N - 1] \) allow you to move the state \( \vec{x}[N] \) by any vector in

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

You will also have an offset of \( A^N\vec{x}[0] \), which captures the zero input system behavior. Taking this into consideration you can reach all states that are in

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

(j) (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 \( \vec{x}_0 \) in \( N \) time steps? 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 states you can reach are given by

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

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 (16)) as

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

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

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

If we extend this sum, we get

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

This system of linear equations can be further rewritten as

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

For this system to be solvable, we need \( \vec{x}_f - A^N x[0] \in \text{span} \{ \vec{b}, A\vec{b}, A^2 \vec{b}, \cdots, A^{N-1} \vec{b} \} \). Since \( \vec{x}_f \) can be any vector in \( \mathbb{R}^4 \), it also means that \( \vec{x}_f - A^N x[0] \) 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

\[
\begin{bmatrix}
\vec{b} & A\vec{b} & A^2 \vec{b} & \cdots & A^{N-1} \vec{b}
\end{bmatrix}
\]

has to be all of \( \mathbb{R}^4 \).

13. **Homework Process and Study Group**

Citing sources and collaborators are an important part of life, including being a student! We also want to understand what resources you find helpful and how much time homework is taking, so we can change things in the future if possible.

(a) What sources (if any) did you use as you worked through the homework?

(b) If you worked with someone on this homework, who did you work with?

List names and student ID’s. (In case of homework party, you can also just describe the group.)

(c) Roughly how many total hours did you work on this homework? Write it down here where you’ll need to remember it for the self-grade form.