EECS 16A Designing Information Devices and Systems I
Fall 2019 Homework 2

This homework is due Friday, September 13, 2019, at 23:59.
Self-grades are due Monday, September 16, 2019, at 23:59.

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.

1. **Gaussian Elimination**

   (a) In this problem we will investigate the relationship between Gaussian elimination and the geometric interpretation of linear equations.

   i Draw the following set of linear equations in the x-y plane. If the lines intersect, write down the point or points of intersection.

$$\begin{aligned} x+2y &= 4 \\ 2x-4y &= 4 \\ 3x-2y &= 8 \end{aligned}$$

   ii Write the above set of linear equations in augmentated matrix form and do the first step of Gaussian elimination to eliminate the x variable from equation 2. Now, one of the three rows has changed. Plot the equations on the same graph as above. What do you notice about the new line you draw?

   iii Complete all of the steps of Gaussian elimination including back substitution. Now plot the new equations created in this last step on the same graph as above. What do you notice?

   (b) Write the following set of linear equations in augmented matrix form and use Gaussian elimination to determine if there are no solutions, infinite solutions, or a unique solution. If any solutions exist, determine what they are. You may do this problem by hand or use a computer. We encourage you to try it by hand to ensure you understand Gaussian elimination.

$$\begin{aligned} x+2y+5z &= 3 \\ x+12y+6z &= 1 \\ 2y+z &= 4 \\ 3x+16y+16z &= 7 \end{aligned}$$

(c) Consider the following system:

$$\begin{aligned} x+2y+5z &= 6 \\ 3x+9y+6z &= 3 \end{aligned}$$

Let the set S contain all vectors $\vec{v}$ of the form $\vec{v} = \begin{bmatrix} 16 \\ -5 \\ 0 \end{bmatrix} + \begin{bmatrix} -11 \\ 3 \\ 1 \end{bmatrix} t$, where $t \in \mathbb{R}$. This can be written mathematically as $S = \left\{ \vec{v} \;\middle|\; \vec{v} = \begin{bmatrix} 16 \\ -5 \\ 0 \end{bmatrix} + \begin{bmatrix} -11 \\ 3 \\ 1 \end{bmatrix} t \quad , \quad t \in \mathbb{R} \right\}$.

Prove that any vector $\vec{v}$ in the set S is a solution to the system given above.

(d) Consider the following system:

$$\begin{aligned} 4x+4y+4z+w+v &= 1 \\ x+y+2z+4w+v &= 2 \\ 5x+5y+5z+w+v &= 0 \end{aligned}$$

If you were to write the above equations in augmented matrix form and use Gaussian elimination to solve the system, you would get the following (for extra practice, you can try and do this yourself):

$$\left[\begin{array}{ccccc|c} 1 & 1 & 0 & 0 & 3 & 16 \\ 0 & 0 & 1 & 0 & -3 & -17 \\ 0 & 0 & 0 & 1 & 1 & 5 \end{array}\right]$$

How many variables are free variables? Determine the solutions to the set of equations.

2. Boser's Optimal Boba

Boser's Optimal Boba has a unique way of serving its customers. To ensure the best customer experience, each customer gets a combination drink personalized to their tastes. Professor Boser knows that a lot of customers don't know what they want, so when customers walk up to the counter, they are asked to taste four standard combination drinks that each contain a different mixture of the available pure teas.

Each combination drink (Classic, Roasted, Mountain, and Okinawa) is made of a mixture of pure teas (Black, Oolong, Green, and Earl Grey), with the total amount of pure tea in each combination drink always the same, and equal to one cup. The table below shows the quantity of each pure tea (Black, Oolong, Green, and Earl Grey) contained in each of the four standard combination drinks (Classic, Roasted, Mountain, and Okinawa).

Tea [cups]	Classic	Roasted	Mountain	Okinawa
Black	$\frac{1}{3}$	$\frac{1}{3}$	0	$\frac{2}{3}$
Oolong	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{2}{5}$	$\frac{1}{3}$
Green	0	$\frac{1}{3}$	$\frac{3}{5}$	0
Earl Grey	$\frac{1}{3}$	0	0	0

Initially, the customer's ratings for each of the pure teas are unknown. The goal of Professor Boser is to determine how much the customer likes each of the pure teas, so that an optimal combination drink can then be made. By letting the customer taste and score each of the four standard combination drinks, Professor Boser can use linear algebra to determine the customer's initially unknown ratings for each of the pure teas. After a customer gives a score (all of the scores are real numbers) for each of the four standard combination drinks, Professor Boser then calculates how much the customer likes each pure tea and mixes up a special combination drink that will maximize the customer's score.

The score that a customer gives for a combination drink is a linear combination of the ratings of the constituent pure teas, based on their proportion. For example, if a customer's rating for black tea is 6 and oolong tea is 3, then the total score for the Okinawa boba drink would be $6 \cdot \frac{2}{3} + 3 \cdot \frac{1}{3} = 5$ because Okinawa has $\frac{2}{3}$ black tea and $\frac{1}{3}$ Oolong tea.

Professor Ranade was thirsty after giving the first lecture, so Professor Ranade decided to take a drink break at Boser's Optimal Boba. Professor Ranade walked in and gave the following ratings:

Combination Drink	**Score**
Classic	7
Roasted	7
Mountain	$7\frac{2}{5}$
Okinawa	$6\frac{1}{3}$

(a) What were Professor Ranade's ratings for each tea? **Work this problem out by hand in terms of the steps. You may use a calculator to do algebra.**

(b) What mystery tea combination could Professor Boser put in Professor Ranade's personalized drink to maximize the customer's score? What score would Professor Ranade give for this drink? There may be more than one correct answer.

3. Finding Charges from Potential Measurements

We have three point charges Q_1, Q_2, and Q_3 whose positions are known, and we want to determine their charges. In order to do that, we take three electric potential measurements U_1, U_2 and U_3 at three different locations. You do not need to understand electric potential to do this problem. The locations of the charges and potentials are shown in Figure 1.

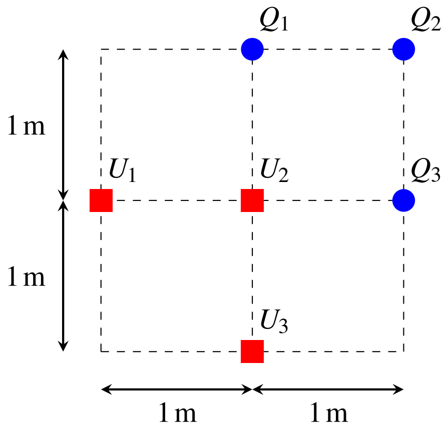

Figure 1: Locations of the charges and potentials.

For the purpose of this problem, the following equation is true:

$$U = k\frac{Q}{r}$$

at a point r meters away (for some fixed physical constant k; this problem does not require the numerical value of k).

Furthermore, the potential contributions from different point charges add up linearly. For example, in the setup of Figure 1, the potential measured at point U_2 is

$$U_2 = k\frac{Q_1}{1} + k\frac{Q_2}{\sqrt{2}} + k\frac{Q_3}{1}.$$

Given that the actual potential measurements in the setup of Figure 1 are

$$\begin{aligned} U_1 &= k\frac{4+3\sqrt{5}+\sqrt{10}}{2\sqrt{5}}, \\ U_2 &= k\frac{2+4\sqrt{2}}{\sqrt{2}}, \\ U_3 &= k\frac{4+\sqrt{5}+3\sqrt{10}}{2\sqrt{5}}, \end{aligned}$$

write the system of linear equations relating the potentials to charges. Solve the system of linear equations to find the charges Q_1, Q_2, Q_3. You may use your IPython notebook to solve the system.

IPython hint: For constants a_i, b_i, c_i, y_i, you can solve the system of linear equations

$$\begin{aligned} a_1x_1 + a_2x_2 + a_3x_3 &= y_1 \\ b_1x_1 + b_2x_2 + b_3x_3 &= y_2 \\ c_1x_1 + c_2x_2 + c_3x_3 &= y_3 \end{aligned}$$

in IPython with the following code:

```
import numpy as np
a = np.array([
    [a1, a2, a3],
    [b1, b2, b3],
    [c1, c2, c3]
])
b = np.array([y1, y2, y3])
x = np.linalg.solve(a, b)
print(x)
```

The square root of a number a can be written as `np.sqrt(a)` in IPython.

4. Kinematic Model for a Simple Car

Learning Goal: Many real world systems are not actually linear and have more complex behaviors. However, linear models can approximate non linear systems under certain conditions.

Building a self-driving car first requires understanding the basic motions of a car. In this problem, we will explore how to model the motion of a car.

There are several models that we can use to model the motion of a car. Assume we use the following kinematic model, given in the following four equations and Figure 2.

$$x[k+1] = x[k] + v[k]\cos(\theta[k])\Delta t \tag{1}$$

$$y[k+1] = y[k] + v[k]\sin(\theta[k])\Delta t \tag{2}$$

$$\theta[k+1] = \theta[k] + \frac{v[k]}{L}\tan(\phi[k])\Delta t \tag{3}$$

$$v[k+1] = v[k] + a[k]\Delta t \tag{4}$$

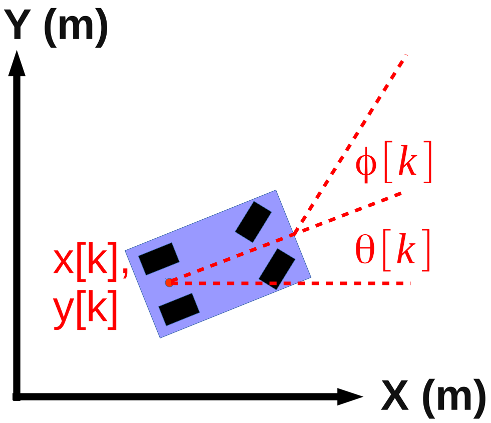

Figure 2: Vehicle Kinematic Model

where

- k, a nonnegative integer, indicates the time step at which we measure the variable (e.g. $v[k]$ is the speed at time step k and $v[k+1]$ is the speed at the following time step)
- $x[k]$ and $y[k]$ denote the coordinates of the vehicle (meters)
- $\theta[k]$ denotes the heading of the vehicle, or the angle with respect to the x-axis (radians)
- $v[k]$ is the speed of the car (meters per second)
- $a[k]$ is the acceleration of the car (meters per second squared)
- $\phi[k]$ is the steering angle input we command (radians)
- Δt is a constant measuring the time difference (in seconds) between time steps $k+1$ and k
- L is a constant and is the length of the car (in meters)

For this problem, let L be 1.0 meter and Δt be 0.1 seconds.

The variables $x[k], y[k], \theta[k], v[k]$ describe the **state** of the car at time step k. The state captures all the information needed to fully determine the current position, speed, and heading of the car. The **inputs** at time step k are $a[k]$ and $\phi[k]$. These are provided by the driver. The current value of these inputs, along with the current state of the vehicle, will determine the state of the vehicle at the next time step.

We note that the problem is nonlinear, due to the sine, cosine and tangent functions, as well as terms including the product of states and inputs.

The purpose of this problem is to show that we can approximate a nonlinear model with a simple linear model and do reasonably well. This is why, despite many systems being nonlinear, linear algebra tools are widely used in practice.

For Parts (b) - (d), fill out the corresponding sections in prob2.ipynb.

(a) We assume that the car has a small heading ($\theta \approx 0$) and that the steering angle is also small ($\phi \approx 0$), where $\approx$ means "approximately equal to." In this case, we could use the following approximations:

$$\begin{aligned} \sin(\alpha) &\approx 0, \\ \cos(\alpha) &\approx 1, \\ \tan(\alpha) &\approx 0. \end{aligned}$$

where α is the small angle of interest. Here, we use a very simple approximation for small angles; in later classes, you may learn better approximations.

Draw, by hand, graphs of of $\sin(\alpha)$ and $\cos(\alpha)$, for α ranging from $-\pi$ to π. Using these graphs can you justify the approximation we are making for small values of α?

(b) Applying the approximation described in the previous part, write down a linear system that approximates the nonlinear vehicle model given above in Equations (1) to (4). In particular, find the 4 x 4 matrix **A** and 4 x 2 matrix **B** that satisfy the equation given below.

$$\begin{bmatrix} x[k+1] \\ y[k+1] \\ \theta[k+1] \\ v[k+1] \end{bmatrix} = \mathbf{A} \begin{bmatrix} x[k] \\ y[k] \\ \theta[k] \\ v[k] \end{bmatrix} + \mathbf{B} \begin{bmatrix} a[k] \\ \phi[k] \end{bmatrix}$$

Hint: *Start with simplifying Equations* (1) *to* (4).

(c) Suppose we drive the car so that the direction of travel is aligned with the x-axis, and we are driving nearly straight, i.e. the steering angle is $\phi[k] = 0.0001$ radians. (Driving exactly straight would have the steering angle $\phi[k] = 0$ radians.) The initial state and input are:

$$\begin{bmatrix} x[0] \\ y[0] \\ \theta[0] \\ v[0] \end{bmatrix} = \begin{bmatrix} 5.0 \\ 10.0 \\ 0.0 \\ 2.0 \end{bmatrix}$$
$$\begin{bmatrix} a[k] \\ \phi[k] \end{bmatrix} = \begin{bmatrix} 1.0 \\ 0.0001 \end{bmatrix}$$

You can use these values in the IPython notebook to compare how the nonlinear system evolves in comparison to the linear approximation that you made. The IPython notebook simulates the car for ten time steps. Are the trajectories similar or very different? Why?

(d) Now suppose we drive the vehicle from the same starting state, but we turn left instead of going straight, i.e. the steering angle is $\phi[k] = 0.5$ radians. The initial state and input are:

$$\begin{bmatrix} x[0] \\ y[0] \\ \theta[0] \\ v[0] \end{bmatrix} = \begin{bmatrix} 5.0 \\ 10.0 \\ 0.0 \\ 2.0 \end{bmatrix}$$

$$\begin{bmatrix} a[k] \\ \phi[k] \end{bmatrix} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$$

You can use these values in the IPython notebook to compare how the nonlinear system evolves in comparison to the linear approximation that you made. The IPython notebook simulates the car for ten time steps. Are the trajectories similar or very different? Why?

5. Fountain Codes

Learning Goal: Linear algebra shows up in many important engineering applications. Wireless communication and information theory heavily rely on principles of linear algebra. This problem illustrates some of the techniques used in wireless communication.

Alice wants to send a message to her friend Bob. Alice sends her message $\vec{m}$ across a wireless channel in the form of a transmission vector $\vec{w}$. Bob receives a vector of symbols denoted as $\vec{r}$. Alice knows some of the symbols in the transmission vector that she sends may be corrupted, so she needs a way to protect her message. A schematic of how this can potentially be done is shown in Figure 3. Transmission corruptions occur commonly in real wireless communication systems, for instance, when your laptop connects to a WiFi router, or your cellphone connects to the nearest cell tower.) Ideally, Alice can come up with a transmission vector such that if some of the symbols get corrupted, Bob can still figure out what Alice is trying to say!

One way to accomplish this goal is to use fountain codes, which are part of a broader family of codes called error correcting codes. Fountain codes are based on principles of linear algebra, and were actually developed right here at Berkeley! The company that commercialized them, Digital Fountain, (started by a **Berkeley grad, Mike Luby**), was later acquired by Qualcomm. In this problem, we will explore some of the underlying principles that make fountain codes work in a very simplified setting.

The message that Alice wants to send to Bob are the three numbers a, b, and c. The message vector representing these numbers is $\vec{m} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$.

(a) Since Alice has three numbers she wishes to transmit, she could transmit six symbols in her transmission vector for redundancy. One of the most naive codes is called the repetition code. If Alice uses the repetition code, her transmission vector is $\vec{w} = \begin{bmatrix} a \\ b \\ c \\ a \\ b \\ c \end{bmatrix}$.

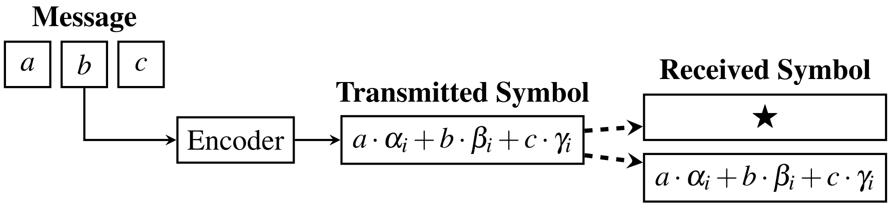
Message
a
b
c
Encoder
Transmitted Symbol
$a \cdot \alpha_i + b \cdot \beta_i + c \cdot \gamma_i$
Received Symbol
★
$a \cdot \alpha_i + b \cdot \beta_i + c \cdot \gamma_i$

Figure 3: Each symbol in a transmission vector $\vec{w}$ is a linear combination of a, b, and c. Each transmitted symbol is either received exactly as it was sent, or it is corrupted. A corrupted symbol is denoted by ★. The ith row of a symbol generating matrix G determines the values of α_i, β_i, and γ_i.

If only the first symbol is corrupted, Bob would receive the vector $\vec{r} = \begin{bmatrix} \bigstar \\ b \\ c \\ a \\ b \\ c \end{bmatrix}$ (the $\bigstar$ symbol represents a corrupted symbol). Using this repetition scheme, give an example where a is unrecoverable but b and c are still recoverable.

(b) Alice can generate $\vec{w}$ by multiplying her message $\vec{m}$ by a matrix. Write a matrix-vector multiplication that Alice can use to generate $\vec{w}$. Specifically, find a symbol generating matrix G_R such that $G_R\vec{m} = \vec{w}$

(c) Instead of a repetition code, it is also possible to use other codes (e.g. fountain codes). Alice and Bob can choose any symbol generating matrix, as long as they agree upon it in advance. Each different matrix represents a different "code." Alice's TA recommends using the symbol generating matrix

$$G_F = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Suppose Bob receives the vector

$$\vec{r} = \begin{bmatrix} 7 \\ \bigstar \\ \bigstar \\ 3 \\ 4 \\ \bigstar \\ \bigstar \end{bmatrix}$$

Write a system of linear equations that Bob can use to recover the message vector $\vec{m}$. Solve it to recover the three numbers that Alice sent.

(d) Can Bob recover the message vector $\vec{m}$ from any three symbols he receives? What about any four received symbols? Justify your answer by giving examples and/or counter-examples. No need to provide a rigorous proof.

(e) Which symbol generating matrix do you prefer? G_R or G_F? Justify why.

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?