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

This homework is due Wednesday, July 01, 2020 at 23:59 PT.
Self-grades are due Sunday, July 05, 2020 at 23:59 PT.

Submission Format
Your homework submission should consist of a single PDF file that contains all of your answers (any hand-written 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 introduces matrix-vector notation for a system of linear equations. Additionally, this homework introduces mathematical proofs.*

1. **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 a kinematic model, described in the following four equations and Figure 1.

$$x[k+1] = x[k] + v[k]\cos(\theta[k])\Delta t \tag{1}$$
$$y[k+1] = y[k] + v[k]\sin(\theta[k])\Delta t \tag{2}$$
$$\theta[k+1] = \theta[k] + \frac{v[k]}{L}\tan(\phi[k])\Delta t \tag{3}$$
$$v[k+1] = v[k] + a[k]\Delta t \tag{4}$$

where

- k, a nonnegative integer, indicates the time step at which we measure each 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)

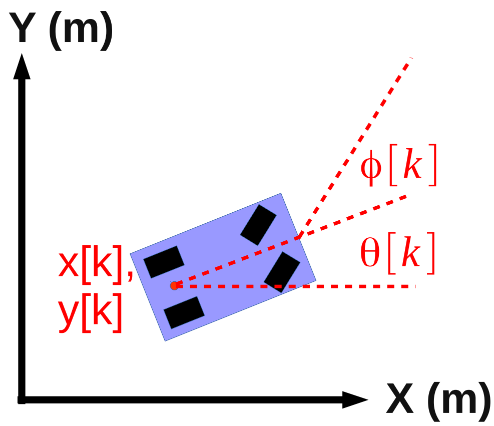

Figure 1: Vehicle Kinematic Model

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 prob1A.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:

$$\sin(\alpha) \approx 0,$$
$$\cos(\alpha) \approx 1,$$
$$\tan(\alpha) \approx 0.$$

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?

Solution:

From Fig. 2, we see that sine is close to zero for angles close to zero. From Fig. 3, we see that cosine is close to one for angles close to zero,

(b) Applying the approximation described in the previous part, write down a system of linear equations 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.

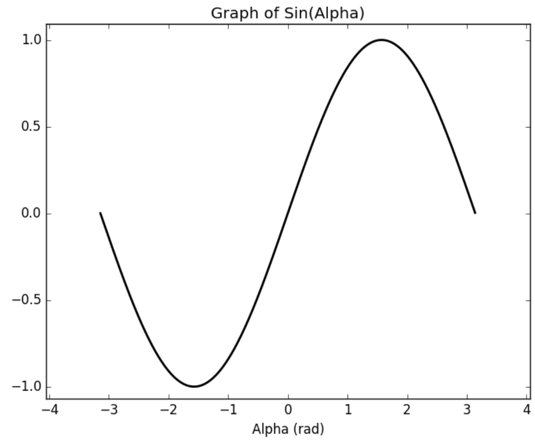

Figure 2: Graph of $sin(\alpha)$

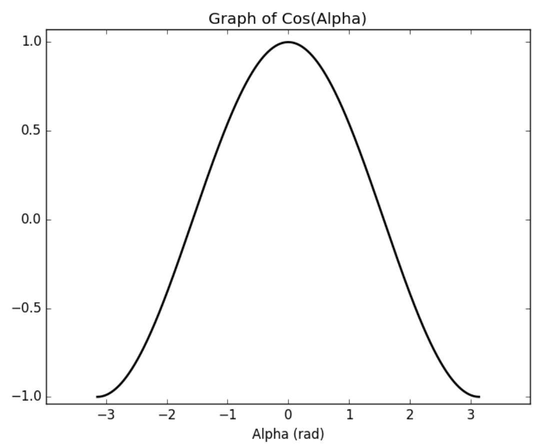

Figure 3: Graph of $cos(\alpha)$

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

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

Solution: In the region where both θ and ϕ are very small, we get that:

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

So the vehicle equations simplify to a linear form:

$$
\begin{aligned}
x[k+1] &= x[k] + v[k]\Delta t \\
y[k+1] &= y[k] + 0 \\
\theta[k+1] &= \theta[k] + 0 \\
v[k+1] &= v[k] + a[k]\Delta t
\end{aligned}
$$

This corresponds to the following linear model:

$$
\begin{bmatrix} x[k+1] \\ y[k+1] \\ \theta[k+1] \\ v[k+1] \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \Delta t \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x[k] \\ y[k] \\ \theta[k] \\ v[k] \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \Delta t & 0 \end{bmatrix} \begin{bmatrix} a[k] \\ \phi[k] \end{bmatrix}
$$

Note to graders: Some students may have $\mathbf{A}$ with element (3,3) set to zero, but otherwise have the right $\mathbf{A}$ and $\mathbf{B}$ matrices. Don't take off credit if this is the case.

While the intention of the problem was that $\theta[k]$ was very small, but not necessarily zero, this was not clear. The answers to part C and D will be unaffected by the value in $\mathbf{A}_{33}$.

(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?

Solution: Yes, the linear model is a good approximation. This is since we are looking at a state and input that satisfy the two approximations we made with θ and ϕ. For example, the nonlinear model predicts that the state at time 10 is:

$$
\begin{bmatrix} x[10] \\ y[10] \\ \theta[10] \\ v[10] \end{bmatrix} = \begin{bmatrix} 7.449 \\ 10.00 \\ 0.0000245 \\ 3 \end{bmatrix}
$$

The state after 10 steps predicted by the linear model is very close.

$$
\begin{bmatrix} x[10] \\ y[10] \\ \theta[10] \\ v[10] \end{bmatrix} = \begin{bmatrix} 7.45 \\ 10.0 \\ 0.0 \\ 3 \end{bmatrix}
$$

From the plot, we see that both models show the vehicle moving along the x-axis and agree relatively well.

The linear model captures the state evolution well in the case where we drive mostly straight with a tiny heading angle. The models are close because the heading angle is tiny enough that the approximations are valid.

One important thing to note is that if we keep predicting forward in time, the heading angle will increase and the linear model will break down. So we usually use linear models for short predictions near the current state.

(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?

Solution: No, the linear model we found breaks down in this case, because we have a high steering angle (ϕ = 0.5 radians is a steering angle of roughly 30°). The linear model predicts the same next state as in the previous part. The nonlinear model, on the other hand, has a different prediction after ten time steps.

$$\begin{bmatrix} x[10] \\ y[10] \\ \theta[10] \\ v[10] \end{bmatrix} = \begin{bmatrix} 6.88 \\ 11.3 \\ 1.34 \\ 3 \end{bmatrix}$$

The linear model predicts $[7.45, 10, 0, 3]^T$, and is not close to the prediction of the nonlinear model at all. This is because the linear model has no dependence on $\phi[k]$ — notice the column of zeros in the matrix B. Since $\phi[k] = 0.5$ is very different from zero, the approximation breaks down. To approximate this better we would need to approximate sine and tangent better. In the plot, we notice the nonlinear model predicts the car will turn left. The linear model still predicts that the car will go straight.

2. Linear Dependence in a Square Matrix

(a) Suppose Gaussian elimination is applied to a matrix A, and the resulting matrix (in row reduced echelon form) has at least one row of all zeros. Argue that this means that the rows of A are linearly dependent.

Solution: Gaussian elimination consists of repeatedly applying the three elementary row operations:

i. Swapping rows
ii. Multiplying a row by a non-zero scalar
iii. Adding a (non-zero) scalar multiple of one row to another row

After each of these operations, the resulting rows can be written as a linear combination of the original rows, where the coefficients of the linear combination are never all zero. For example, let $\vec{r}_1, \vec{r}_2, \cdots, \vec{r}_n$ be the original rows of A, and let $\vec{r}_1'$ bt the first row after one row operation. We demonstrate that each elementary row operation results in a linear combination of the original rows.

i. $\vec{r}_1' = \vec{r}_2$ (swap row 1 and row 2)
ii. $\vec{r}_1' = \alpha\vec{r}_1$ (scale row 1 by $\alpha \neq 0$)
iii. $\vec{r}_1' = \vec{r}_1 + \alpha\vec{r}_2$ (add a scalar multiple of row 2 to row 1)
iv. $\vec{r}_1' = \vec{r}_1$ (any row operation that does not affect row 1)

Note that the coefficients of the linear combination always have a non-zero element (even when the row operation doesn't affect the row 1).

Gaussian elimination involves repeatedly applying these row operations. Consider what happens when we apply a second row operation to A: The rows after 2 operations are linear combinations of the rows after 1 operation (with coefficients not all zero), and the rows after 1 operation are linear combinations of the original rows (with coefficients not all zero). Therefore, the rows after 2 operations are linear combinations of the original rows (with coefficients not all zero).

By repeating this thought process, we conclude that the rows of the final matrix after Gaussian elimination are linear combinations of the original rows, where the coefficients are not all zero. This means that if there is a row of all zeros in the row reduced echelon form of the matrix, then there exists a linear combination of the rows of A that equals all zeros, and the coefficents of this linear combination are not all zero. This is the definition of linear dependence.

(b) Let A be a square $n \times n$ matrix, (i.e. both the columns and rows are vectors in $\mathbb{R}^n$). Suppose we are told that the columns of A are linearly dependent. Prove, then, that the rows of A must also be linearly dependent. You can use the conclusion from part (a) in your proof.

(**Hint**: Can you use the linear dependence of the columns to say something about the number of solutions to $A\vec{x} = \vec{0}$? How does the number of solutions relate to the result of Gaussian elimination?)

Solution:

Let $\vec{a}_1, \vec{a}_2, \ldots \vec{a}_n$ be the columns of A. By the definition of linear dependence, there exist scalars, $c_1, c_2, \ldots c_n$, not all zero, such that

$$c_1\vec{a}_1 + c_2\vec{a}_2 + \cdots + c_n\vec{a}_n = \vec{0} \tag{5}$$

We define $\vec{c}$ to be a vector containing the c_i's as follows: $\vec{c} = \begin{bmatrix} c_1 & c_2 & \ldots & c_n \end{bmatrix}^T$, where $\vec{c} \neq \vec{0}$ by the definition of linear dependence. We can write Eq. 5 in matrix vector form:

$$A\vec{c} = \vec{0} \tag{6}$$

Let's use the first hint: How many solutions are there to the equation $A\vec{x} = \vec{0}$? We know from Eq. 6 that $\vec{c}$ is a solution, but we can also show that $\alpha\vec{c}$ is a solution for any α:

$$A(\alpha\vec{c}) = \alpha\vec{0} = \vec{0} \tag{7}$$

Since $\vec{c}$ is not zero, every multiple of $\vec{c}$ is a different solution. Therefore there are infinite solutions to the equation $A\vec{x} = \vec{0}$.

What can we say about the result of Gaussian elimination if there are infinite solutions? We know that if there are infinite solutions, there must be a free variable after Gaussian elimination. In other words, there must be a column in the row reduced matrix with no leading entry. Therefore, there must be fewer leading entries than the number of columns. Since the matrix A is square, it has the same number of rows as columns, so there must be fewer leading entries than the number of rows. That means there is at least one row with no leading entry, which is equivalent to saying there must be one row that's all zeros in the row reduced matrix.

Finally, in part (a), we showed that if there is a row of all zeros in the row reduced matrix, then the rows of A must be linearly dependent.

(c) Given some set of vectors $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$, show the following:

$$\operatorname{span}\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\} = \operatorname{span}\{\vec{v}_1 + \vec{v}_2, \vec{v}_2, \ldots, \vec{v}_n\}$$

In other words, we can replace one vector with the sum of itself and another vector and not change their span.

Solution:

Suppose $\vec{q} \in \operatorname{span}\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$. For some scalars a_i:

$$\vec{q} = a_1\vec{v}_1 + a_2\vec{v}_2 + \cdots + a_n\vec{v}_n = a_1(\vec{v}_1 + \vec{v}_2) + (-a_1 + a_2)\vec{v}_2 + \cdots + a_n\vec{v}_n$$

We can change the scalar values to adjust for the combined vectors. Thus, we have shown that $\vec{q} \in \operatorname{span}\{\vec{v}_1 + \vec{v}_2, \vec{v}_2, \ldots, \vec{v}_n\}$. Therefore, we have $\operatorname{span}\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\} \subseteq \operatorname{span}\{\vec{v}_1 + \vec{v}_2, \vec{v}_2, \ldots, \vec{v}_n\}$. Now, we must show the other direction. Suppose we have some arbitrary $\vec{r} \in \operatorname{span}\{\vec{v}_1 + \vec{v}_2, \vec{v}_2, \ldots, \vec{v}_n\}$. For some scalars b_i:

$$\vec{r} = b_1(\vec{v}_1 + \vec{v}_2) + b_2\vec{v}_2 + \cdots + b_n\vec{v}_n = b_1\vec{v}_1 + (b_1 + b_2)\vec{v}_2 + \cdots + b_n\vec{v}_n.$$

Thus, we have shown that $\vec{r} \in \operatorname{span}\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$. Therefore, we have $\operatorname{span}\{\vec{v}_1 + \vec{v}_2, \vec{v}_2, \ldots, \vec{v}_n\} \subseteq \operatorname{span}\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$. Combining this with the earlier result, the spans are thus the same.

3. 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.

