Homework 8

This homework is due on Saturday, March 16, 2024, at 11:59PM.

1. Transition Matrix

(a) You are provided a matrix $\mathbf{A} = \begin{bmatrix} 0.2 & 0.8 & 0.2 \\ 0 & 0.4 & 0.2 \\ 0 & 0 & 0.8 \end{bmatrix}$. Matrix $\mathbf{A}$ is transition matrix where $\vec{x}[i+1] = \mathbf{A}\vec{x}[i]$. Additionally, the state vector at timestep $i = 1$ is $\vec{x}[1] = \begin{bmatrix} 8 \\ 1 \\ 4 \end{bmatrix}$. **After infinite timesteps, what is the value of the state vector $\vec{x}[i]$? That is, find $\lim_{i\to\infty} \vec{x}[i]$.**

2. Discrete System Induction

Consider the discrete system given by

$$\vec{x}[i+1] = A\vec{x}[i] + B\vec{u}[i] \tag{1}$$

(a) Write $\vec{x}[1]$ in terms of $\vec{x}[0]$ and $\vec{u}[0]$. Then, write $\vec{x}[2]$ in terms of $\vec{x}[0]$, $\vec{u}[0]$, and $\vec{u}[1]$. Finally, write $\vec{x}[3]$ in terms of $\vec{x}[0]$, $\vec{u}[0]$, $\vec{u}[1]$, and $\vec{u}[2]$.

(b) We can generalize the above process to write $\vec{x}[i]$ in terms of $\vec{x}[0]$ and $\vec{u}[0], \ldots, \vec{u}[i-1]$ as follows:

$$\vec{x}[i] = A^i\vec{x}[0] + \sum_{j=0}^{i-1} A^{i-1-j}B\vec{u}[j] \tag{2}$$

Verify that this equation holds for $i = 1$. *This is equivalent to testing your base case in induction.* (Note that this holds for $i = 0$ because the summation $\sum_{j=0}^{-1}(\cdot)$ is the empty set.)

(c) **Show that if eq. (2) holds for $\vec{x}[i]$, it also holds for $\vec{x}[i+1]$.** That is, write $\vec{x}[i+1]$ in terms of $\vec{x}[i]$ and plug in eq. (2) for $\vec{x}[i]$. Show that this simplifies to eq. (2), where we now replace i with $i+1$. *This is equivalent to verifying our inductive hypothesis.*

3. System Identification

You are given a discrete-time system as a black box. You don't know the specifics of the system but you know that it takes one scalar input and has two states that you can observe. You assume that the system is linear and of the form

$$\vec{x}[i+1] = A\vec{x}[i] + Bu[i] + \vec{w}[i], \tag{3}$$

where $\vec{w}[i]$ is an unknown small external disturbance, $u[i]$ is a scalar input, and

$$A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad x[i] = \begin{bmatrix} x_1[i] \\ x_2[i] \end{bmatrix}. \tag{4}$$

You want to identify the system parameters (a_1, a_2, a_3, a_4, b_1 and b_2) from measured data. However, you can only interact with the system via a black box model. In other words, you can only evaluate the system via observing the states $\vec{x}[i]$ and setting the inputs $u[i]$ that allow the system to move to the next state.

(a) You observe that the system has state $\vec{x}[i] = \begin{bmatrix} x_1[i] & x_2[i] \end{bmatrix}^\top$ at time i. You pass input $u[i]$ into the black box and observe the next state of the system: $\vec{x}[i+1] = \begin{bmatrix} x_1[i+1] & x_2[i+1] \end{bmatrix}^\top$.
Write scalar equations for the new states, $x_1[i+1]$ and $x_2[i+1]$. Write these equations in terms of the a_i, b_i, the states $x_1[i]$, $x_2[i]$ and the input $u[i]$. Here, assume that $\vec{w}[i] = \vec{0}$ (i.e. the model is perfect).

(b) Now we want to identify the system parameters. We observe the system at the start state $\vec{x}[0] = \begin{bmatrix} x_1[0] \\ x_2[0] \end{bmatrix}$. We can then input $u[0]$ and observe the next state $\vec{x}[1] = \begin{bmatrix} x_1[1] \\ x_2[1] \end{bmatrix}$. We can continue this process for a sequence of ℓ inputs.
Let us define an ℓ-length trajectory to include an initial condition $\vec{x}[0]$, an input sequence $u[0], \ldots, u[\ell-1]$, and the corresponding states that are produced by the system $x[1], \ldots, x[\ell]$. **Assuming that the model is perfect ($\vec{w}[i] = \vec{0}$), what is the minimum value of ℓ you need to identify the system parameters?**

(c) We now remove our assumption that $\vec{w} = 0$ (that the model is perfect). We assume that $\vec{w}$ is small, so the model is approximately correct and we have

$$\vec{x}[i+1] \approx A\vec{x}[i] + Bu[i]. \tag{5}$$

Say that we feed in a total of 4 inputs $u[0], \ldots, u[3]$, and observe the states $\vec{x}[0], \ldots, \vec{x}[4]$. To identify the system, we need to set up an approximate matrix equation (because of potential small disturbances)

$$DP \approx S \tag{6}$$

using the observed values above and the unknown parameters we want to find. Let our parameter vector be

$$P := \begin{bmatrix} \vec{p}_1 & \vec{p}_2 \end{bmatrix} = \begin{bmatrix} a_1 & a_3 \\ a_2 & a_4 \\ b_1 & b_2 \end{bmatrix} \tag{7}$$

Find the corresponding D and S to do system identification. Write both out explicitly.

(d) Now that we have set up $DP \approx S$, we can estimate a_0, a_1, a_2, a_3, b_0, and b_1. **Give an expression for the estimates of $\vec{p}_1$ and $\vec{p}_2$ (which are denoted $\hat{\vec{p}}_1$ and $\hat{\vec{p}}_2$ respectively) in terms of D and S.** Denote the columns of S as $\vec{s}_1$ and $\vec{s}_2$, so we have $S = [\vec{s}_1 \; \vec{s}_2]$. Assume that the columns of D are linearly independent.

(HINT: Don't forget that D is not a square matrix. It is taller than it is wide.)

(HINT: Can we split $DP = S$ into separate equations for $\vec{p}_1$ and $\vec{p}_2$?)

4. Identifying Systems from their Responses to Known Inputs

In many problems, we have an unknown system, and would like to characterize it. One of the ways of doing so is to observe the system response with different initial conditions (or inputs). This problem is also called system identification. It is a prototypical example of a problem that is today called machine learning — inferring an underlying pattern from data, and doing so well enough to be able to exploit that pattern in some practical setting. Go through the attached Jupyter notebook `demo_system_id.ipynb` and answer the following questions.

(a) In Example 2, we assume that instead of measuring the state $\vec{x}$, we are instead measuring a transformation of the state $\vec{y} = T\vec{x}$ where T is a full rank matrix. Assume that we perform system ID on our observations $\vec{y}[i]$ to recover A_y, B_y such that $\vec{y}[i+1] = A_y\vec{y}[i] + B_y u[i]$. **How do the identified A_y and B_y matrices relate to the original A and B matrices in the dynamics of $\vec{x}$?** Remember that our original state dynamics are $\vec{x}[i+1] = A\vec{x}[i] + B\vec{u}[i]$. (The answer is given in the Jupyter notebook but remember to show your work.)

(b) **Please share your observations on Example 2. Comment on what impact a linear transformation of the state trace has on our ability to perform system identification.**

(c) **Prove that for any full rank transformation matrix T, the eigenvalues of A_y and A from part (a) are the same.**

(d) **Please share your observations on Example 3. Comment on the impact that changing the noise magnitude, number of samples and number of states has on the system identification performance.**

(e) **Please share your observations on Example 4. Comment on how important the model size is for this setting.**

5. (Lab) Motor Driver and System Identification

In the lab project, you will be designing SIXT33N, a mischievous little robot who *might* just do what you want — if you design it correctly. To control the car, we need to build a model of the car first. Instead of designing a complex nonlinear model, we will approximate the system with a linear model to work for small perturbations around an equilibrium point. The following model applies separately to each wheel (and associated motor) of the car:

$$v_L[i] = \theta_L u_L[i] - \beta_L \tag{8}$$
$$v_R[i] = \theta_R u_R[i] - \beta_R \tag{9}$$

Notice that this particular model has no state variables since we are measuring velocity directly here.

To do system ID, we decide to use the exact same input $u_L[i] = u_R[i] = u[i]$ for both motors. We measure both velocities, however.

Meet the variables at play in this model:

- i - The current timestep of the model. Since we model the car as a discrete-time system, i will advance by 1 for every new sample in the system.
- $v_L[i]$ - The discrete-time velocity (in units of ticks/timestep) of the left wheel, reading from the motor.
- $v_R[i]$ - The discrete-time velocity (in units of ticks/timestep) of the right wheel, reading from the motor.
- $u[i]$ - The input to each wheel, which takes a value in $[0, 255]$ representing the duty cycle. For example, when $u[i] = 255$, the duty cycle is 100 %, and the motor controller just delivers a constant signal at the system's HIGH voltage, delivering the maximum possible power to the motor. When $u[i] = 0$, the duty cycle is 0 %, and the motor controller delivers 0 V. The duty cycle (D) can be written as
$$\text{duty cycle (D)} = \frac{u[i]}{255} \tag{10}$$
- $\theta(\theta_L, \theta_R)$ - Relates change in input to change in velocity. **Its units are ticks/(timestep · duty cycle).** Since our model is linear, we assume that θ is the same for every unit increase in $u[i]$. This is empirically measured using the car. **You will have a separate θ for your left and your right wheel(θ_L, θ_R).**
- $\beta(\beta_L, \beta_R)$ - Similarly to θ, β is dependent upon many physical phenomena, so we will empirically determine it using the car. β represents a constant offset in the velocity of the wheel, and hence **its units are ticks/timestep**. Note that you will also typically have a different β for your left and your right wheel (i.e. $\beta_L \neq \beta_R$). **These β_L and β_R are different from the β_F of the transistor.**

(a) By measuring the car with a PWM signal at different duty cycles, we can collect the velocity data of the left and right wheel, as shown in the following table:

Table 1: The velocity of the left and the right wheel at different duty cycles of PWM signal

Duty Cycle $\times 255$ ($u[i]$)	Velocity of the left wheel ($v_L[i]$)	Velocity of the right wheel ($v_R[i]$)
80	147	127
120	218	187
160	294	253
200	370	317

Since the same input is applied to both the wheels, we can take advantage of the same "horizontal stacking" trick you've seen before to be able to reuse computation. To identify the system, we need to set up matrix equations for the left and right wheels in the form of:

$$D_{\text{data}} P \approx S \tag{11}$$

where $P = \begin{bmatrix} \theta_L & \theta_R \\ \beta_L & \beta_R \end{bmatrix}$. **Find the matrix D_{data} and matrix S needed to perform system identification to get the matrix of parameters of the left and right wheel, P.**

(b) **Solve the matrix equation $D_{\text{data}} P \approx S$ with least squares to find θ_L, θ_R, β_L, and β_R.** You may use a jupyter notebook or other tool for computation.

(c) In order for the car to drive straight, the wheels must be moving at the same velocity. However, the data from Table 1 tells us that the two motors do not run at the same velocity if the duty cycles of driving PWM signals are the same. **Based on the model you extracted, if we want the car to drive straight and $u_L = 100$, what should u_R be?**

Contributors:
- Nikhil Shinde.
- Ashwin Vangipuram.
- Sally Hui.
- Alex Devonport.
- Tanmay Gautam.
- Bozhi Yin.
- Kaitlyn Chan.
- Yi-Hsuan Shih.
- Vladimir Stojanovic.
- Moses Won.

