Homework 8

This homework is due on Saturday, October 21, 2023, at 11:59PM. Self-grades and HW Resubmissions are due on Saturday, October 28, 2023, at 11:59PM.

1. 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

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

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

$$A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad x[i] = \begin{bmatrix} x_1[i] \\ x_2[i] \end{bmatrix}. \tag{2}$$

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, i.e., you can see the states $\vec{x}[t]$ and set 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 for a sequence of ℓ inputs.
Let us define an ℓ-length trajectory to be 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 $\vec{w} = 0$. We assume it is small, so the model is approximately correct and we have

$$\vec{x}[i+1] \approx A\vec{x}[i] + Bu[i]. \tag{3}$$

Say 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 (because of potential, small, disturbances) matrix equation

$$DP \approx S \tag{4}$$

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

$$P := \begin{bmatrix} \vec{p}_1 & \vec{p}_2 \end{bmatrix} = \begin{bmatrix} a_1 & a_3 \\ a_2 & a_4 \\ b_1 & b_2 \end{bmatrix} \tag{5}$$

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 p_1 and p_2?)*

2. 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. In phases 1 and 2, you will build the **legs** of SIXT33N: you will be designing SIXT33N's wheels and developing a linear model for the car system. The wheels will be driven by two 9-Volt DC motors whose driver control circuit is shown in Figure 1 .

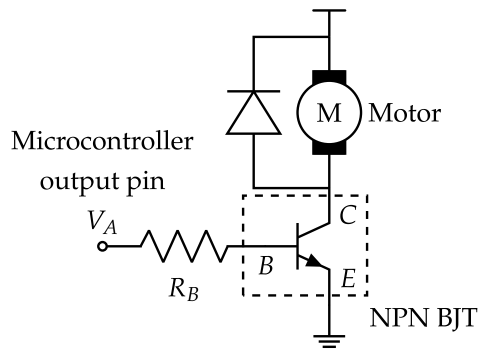


Figure 1: Motor Controller Circuit

There is some minimum voltage required to deliver enough power to the motors to overcome the static friction and start them, but after that point, we treat the motor speed as approximately **linear** with the applied voltage V_A (this will be the basis of the system model you will develop in this problem).

As it is difficult to use a microcontroller (MSP430 in hands-on lab; Arduino in lab sim) to generate a true adjustable DC signal, we will instead make use of its PWM function. A PWM, or pulse-width modulated, signal is a square wave with a variable duty cycle (the proportion of a cycle period for which the power source is turned on, or logic HIGH). PWM is used to digitally change the average voltage delivered to a load by varying the duty cycle. If the frequency is large enough, the on-off switching is imperceptible, but the average voltage delivered to the load changes proportionally with the duty cycle. Hence, changing the duty cycle corresponds to changing the DC voltage supplied to the motor. An example can be seen in figure 2.

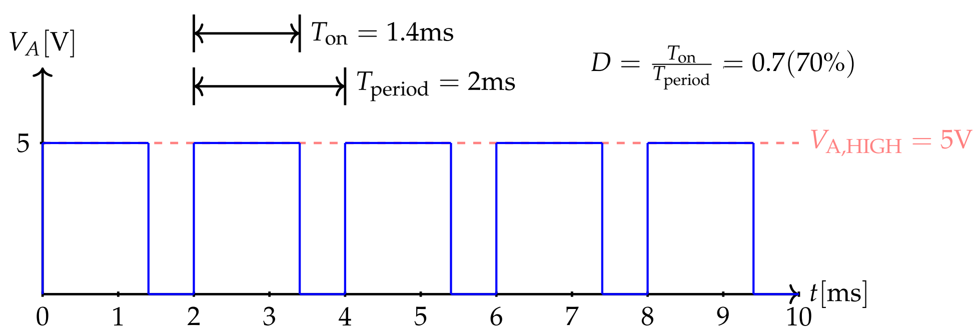
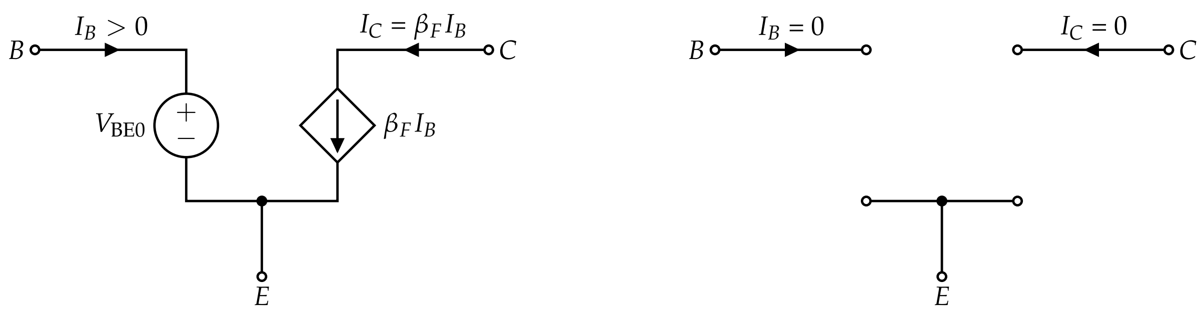


Figure 2: PWM Example with switching frequency 500Hz and 70% duty cycle

The PWM pin (V_A) is connected via a resistor (R_B) to the "Base (B)" of an NPN bipolar junction transistor (BJT). This transistor, in reality, behaves a bit differently from the NMOS with which you are familiar, but for this class, you may assume that it is functionally the same as an NMOS, behaving as a switch. On the BJT, the three terminals are analogous to those of an NMOS: the "Base (B)" is the gate, the "Collector (C)" is the drain, and the "Emitter (E)" is the source.

The BJT in Figure 1 is switching between **ON** and **OFF** modes when V_A is HIGH and LOW respectively. The model for both ON and OFF states are shown in Figure 3. When the BJT turns on, V_{BE} can be modeled as a fixed voltage source with voltage value V_{BE0}. In ON mode, there is a **Current Controlled Current Source** modeled between "Collector (C)" and "Emitter (E)", i.e., current at the "Collector (C)" is an amplified version of current at the "Base (B)" (notice that positive I_B has to flow into the "Base (B)" for the relation $I_C = \beta_F I_B$ to hold). β_F is called the **Common-Emitter Current Gain**.

The diode in parallel with the motor is needed because of the inductive characteristics of the motor. If the motor is on and V_A switches to LOW, the inductive behavior of the motor maintains the current and the diode provides the path to dissipate it as the BJT is turned off. When the BJT turns on, the diode is off so there is no current flow through the diode.



(a) Model of BJT in ON mode (when V_A is logic HIGH)

(b) Model of BJT in OFF mode (when V_A is logic LOW)

Figure 3: Model of NPN BJT in Different Modes

Please use $V_{\text{BE0}} = 0.8\,\text{V}$, $\beta_F = 100$, $V_{A,\text{HIGH}} = 5\,\text{V}$ and $V_{A,\text{LOW}} = 0\,\text{V}$ for all following calculations.

Part 1: Circuit analysis to construct the system model

(a) **Draw the equivalent motor controller circuit when the BJT is ON by substituting in the BJT model from Fig. 3a into Fig. 1. Express I_B and I_C for $V_A = V_{A,\text{HIGH}}$ as a function of R_B.**

(b) **Draw the equivalent motor controller circuit when the BJT is OFF by substituting in the BJT model from Fig. 3b into Fig. 1. Express I_B and I_C for $V_A = V_{A,\text{LOW}}$ as a function of R_B.**

(c) **Derive the average collector current, I_{AVG}, over one period, T_{period}, of the PWM signal, V_A, as a function of R_B and the duty cycle, D, of the PWM signal.** Hint: The time average of some signal $f(t)$ from time t_0 to t_1 is given as $f_{\text{AVG}} = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} f(\tau)\,d\tau$. Figure 2 may be useful.

(d) **In the previous part, explain briefly why is it sufficient to take the average over only one period if we are actually interested in the average collector current over multiple periods?**

(e) **If $R_B = 2\,\text{k}\Omega$, what is the average collector current, I_{AVG}, that drives the motor when the duty cycle of the PWM signal is equal to 25%?**

Part 2: Learning a Car Model from data

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:

$$v_L[i] = \theta_L u_L[i] - \beta_L \tag{6}$$

$$v_R[i] = \theta_R u_R[i] - \beta_R \tag{7}$$

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: **(Note: the β here have nothing to do with the previous part.)**

- i - The current timestep of the model. Since we model the car as a discrete-time system, n will advance by 1 on 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. The duty cycle of the PWM signal (V_A), which is the percentage of the square wave's period for which the square wave is HIGH, is mapped to the range $[0, 255]$. Thus, $u[i]$ 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

$$\text{duty cycle (D)} = \frac{u[i]}{255} \tag{8}$$

- $\theta(\theta_L, \theta_R)$ - Relates change in input to change in velocity. **Its units are ticks/(timestep $\cdot$ 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.**

(f) 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 setup matrix equations of left and right wheel in the form of:

$$D_{\text{data}} P \approx S \tag{9}$$

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.**

(g) **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 for computation.

(h) In most advanced systems, we usually use a combination of a physics-based equation and a data-centric approach to build the model. In our case, the velocity of the motor can be written as

$$v[i] = kI_{\text{AVG}}(u[i]) - \beta \tag{10}$$

where $I_{\text{AVG}}(u[i])$ is the average collector current which is the function of the duty cycle that you have already derived in Part 1. k represents the response of your motor speed to the average current. In our simplified motor driver model in Part 1, you have already derived the expression for the I_{AVG} of the motor as a function of the circuit parameters and the duty cycle D. **If we assume that the model from Part 1 holds, determine the resistance ratio $\left(\frac{R_{\text{B,left}}}{R_{\text{B,right}}}\right)$ from the model parameters you identified in Part 2 item (f).** Assume that the left motor and the right motor respond the same, that is, $k_L = k_R$. The only difference is presumed to come from the resistors used.

(i) In order for the car to drive straight, the wheels must be moving at the same velocity. However, the data from Table 1 tell us that two motors cannot run at the same velocity if the duty cycles of driving PWM signals are the same. **Based on the model you extracted in Part 2 item (f), if we want the car to drive straight and $u_L = 100$, what should u_R be?**

3. Stability Criterion

Consider the complex plane below, which is broken into non-overlapping regions A through H. The circle drawn on the figure is the unit circle $|\lambda| = 1$.

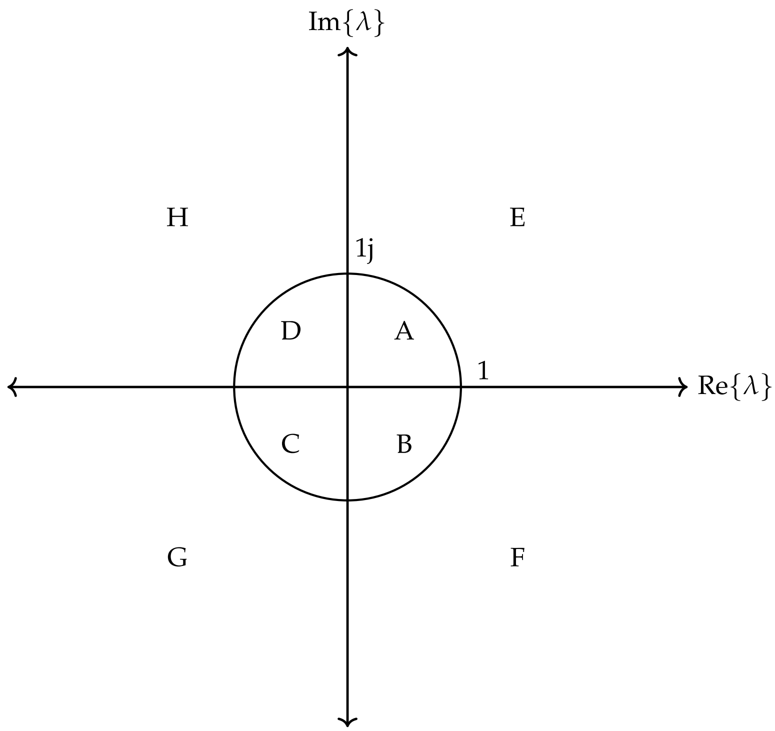


Figure 4: Complex plane divided into regions.

Consider the continuous-time system $\frac{d}{dt}x(t) = \lambda x(t) + v(t)$
and the discrete-time system $y[i+1] = \lambda y[i] + w[i]$. Here $v(t)$ and $w[i]$ are both disturbances to their respective systems.

In which regions can the eigenvalue λ be for the system to be *stable*? Fill out the table below to indicate *stable* regions. Assume that the eigenvalue λ does not fall directly on the boundary between two regions.

	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**
Continuous Time System $x(t)$	◯	◯	◯	◯	◯	◯	◯	◯
Discrete Time System $y[i]$	◯	◯	◯	◯	◯	◯	◯	◯

4. Bounded-Input Bounded-Output (BIBO) Stability

BIBO stability is a system property where bounded inputs lead to bounded outputs. It's important because we want to certify that, provided our system inputs are bounded, the outputs will not "blow up". In this problem, we gain a better understanding of BIBO stability by considering some simple continuous and discrete systems, and showing whether they are BIBO stable or not.

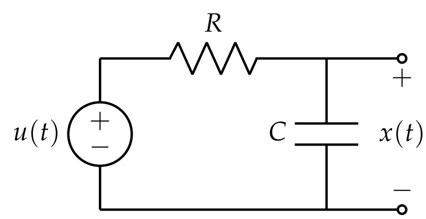
Recall that for the following simple scalar differential equation, we have the corresponding solution:

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = ax(t) + bu(t) \qquad x(t) = \mathrm{e}^{at}x(0) + \int_0^t \mathrm{e}^{a(t-\tau)}bu(\tau)\,\mathrm{d}\tau\,. \tag{11}$$

And for the following discrete system, we have the corresponding solution:

$$x[i+1] = ax[i] + bu[i] \qquad x[i] = a^i x[0] + \sum_{k=0}^{i-1} a^k bu[i-1-k] \tag{12}$$

(a) Consider the circuit below with $R = 1\Omega$, $C = 0.5\mathrm{F}$. Let $x(t)$ be the voltage over the capacitor.



This circuit can be modeled by the differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = -2x(t) + 2u(t) \tag{13}$$

Intuitively, we know that the voltage on the capacitor can never exceed the (bounded) voltage from the voltage source, so this system is BIBO stable. **Show that this system is BIBO stable, meaning that $x(t)$ remains bounded for all time if the input $u(t)$ is bounded. Equivalently, show that if we assume $|u(t)| < \epsilon$, $\forall t \geq 0$ and $|x(0)| < \epsilon$, then $|x(t)| < M$, $\forall t \geq 0$ for some positive constant M.** Thinking about this helps you understand what bounded-input-bounded-output stability means in a physical circuit.

(HINT: eq. (11) may be useful. You may want to write the expression for $x(t)$ in terms of $u(t)$ and $x(0)$ and then take the norms of both sides to show a bound on $|x(t)|$. Remember that norm in 1D is absolute value. Some helpful formulas are $|ab| = |a||b|$, the triangle inequality $|a+b| \leq |a| + |b|$, and the integral version of the triangle inequality $\left|\int_a^b f(\tau)\,\mathrm{d}\tau\right| \leq \int_a^b |f(\tau)|\,\mathrm{d}\tau$, which just extends the standard triangle inequality to an infinite sum of terms.)

(b) Assume $x(0) = 0$. **Show that the system eq. (11) is BIBO unstable when $a = \mathrm{j}2\pi$ by constructing a bounded input that leads to an unbounded $x(t)$.**

It can be shown that the system eq. (11) is unstable for any purely imaginary a by a similar construction of a bounded input.

(c) Consider the discrete-time system and its solution in eq. (12). **Show that if $|a| > 1$, then even if $x[0] = 0$, a bounded input can result in an unbounded output, i.e. the system is BIBO unstable.** *(HINT: The formula for the sum of a geometric sequence may be helpful.)*

(d) Consider the discrete-time system

$$x[i+1] = -3x[i] + u[i]. \tag{14}$$

Is this system stable or unstable? Give an initial condition $x(0)$ and a sequence of non-zero inputs for which the state $x[i]$ will always stay bounded. *(HINT: See if you can find any input pattern that results in an oscillatory behavior.))*

5. Eigenvalue Placement through State Feedback

Consider the following discrete-time linear system:

$$\vec{x}[i+1] = \begin{bmatrix} -2 & 2 \\ -2 & 3 \end{bmatrix} \vec{x}[i] + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u[i]. \tag{15}$$

In standard language, we have $A = \begin{bmatrix} -2 & 2 \\ -2 & 3 \end{bmatrix}, \vec{b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ in the form: $\vec{x}[i+1] = A\vec{x}[i] + \vec{b}u[i]$.

(a) **Is this discrete-time linear system stable in open loop (without feedback control)?**

(b) Suppose we use state feedback of the form $u[i] = \begin{bmatrix} f_1 & f_2 \end{bmatrix} \vec{x}[i] = F\vec{x}[i]$.
Find the appropriate state feedback constants, f_1, f_2 so that the state space representation of the resulting closed-loop system has eigenvalues at $\lambda_1 = -\frac{1}{2}, \lambda_2 = \frac{1}{2}$.

Contributors:

- Nikhil Shinde.
- Ashwin Vangipuram.
- Sally Hui.
- Bozhi Yin.
- Kaitlyn Chan.
- Yi-Hsuan Shih.
- Vladimir Stojanovic.
- Moses Won.
- Sidney Buchbinder.
- Tanmay Gautam.
- Nathan Lambert.
- Anant Sahai.
- Druv Pai.
- Varun Mishra.

