Homework 9

This homework is due on Friday, March 24, 2023, at 11:59PM. Self-grades and HW Resubmissions are due on the following Friday, March 31, 2023, at 11:59PM.

1. **Open-Loop and Closed-Loop Control**

In last week's lab-related System ID problem, we built SIXT33N's motor control circuitry and developed a linear model for the velocity of each wheel. We are one step away from our goal: to have SIXT33N drive in a straight line! We will see how to use the model we developed in the System ID problem to control SIXT33N's trajectory to be a straight line.

More specifically, in this problem, we will explore how to use open-loop and closed-loop control to drive the trajectory of your car in a straight line.

Part 1: Open-Loop Control

An open-loop controller is one in which the input is predetermined using your system model and the goal, and not adjusted at all during operation. To design an open-loop controller for your car, you would set the PWM duty-cycle value of the left and right wheels (inputs $u_L[i]$ and $u_R[i]$) such that the predicted velocity of both wheels is your target wheel velocity (v_t). You can calculate these inputs from the target velocity v_t and the $\theta_L, \theta_R, \beta_L, \beta_R$ values you learned from data. In the System ID problem and lab, we have modeled the velocity of the left and right wheels as

$$v_L[i] = d_L[i+1] - d_L[i] = \theta_L u_L[i] - \beta_L; \tag{1}$$

$$v_R[i] = d_R[i+1] - d_R[i] = \theta_R u_R[i] - \beta_R \tag{2}$$

where $d_{L,R}[i]$ represent the distance traveled by each wheel.

(a) Find the open-loop control that would give us $v_L[i] = v_R[i] = v_t$. That is, **solve the model (Equations (1) and (2)) for the inputs $u_L[i]$ and $u_R[i]$ that make the velocities $v_L[i] = v_R[i] = v_t$.**

In practice, the $\theta_L, \theta_R, \beta_L, \beta_R$ parameters are learned from noisy data, and so can be wrong. This means that we will calculate the velocities for the two wheels incorrectly. When the velocities of the two wheels disagree, the car will go in a circle instead of a straight line. Thus, to make the car go in a straight line, we need the distances traveled by both wheels to be the same at each timestep.

This prompts us to simplify our model. Instead of having two state variables $\vec{v}_L$ and $\vec{v}_R$, we can just have a state variable determining how far we are from the desired behavior of going in a line – a state which we will want to drive to 0.

This prompts us to define our state variable δ to be the *difference* in the distance traveled by the left wheel and the right wheel at a given timestep:

$$\delta[i] := d_L[i] - d_R[i] \tag{3}$$

We want to find a scalar discrete-time model for $\delta[i]$ of the form

$$\delta[i+1] = \lambda_{\text{OL}}\delta[i] + f(u_L[i], u_R[i]). \tag{4}$$

Here λ_{OL} is a scalar and $f(u_L[i], u_R[i])$ is the control input to the system (as a function of $u_L[i]$ and $u_R[i]$).

(b) Suppose we apply the open-loop control inputs $u_L[i], u_R[i]$ to the original system. **Using Equations (1) and (2), write $\delta[i+1]$ in terms of $\delta[i]$, in the form of Equation (4). What is the eigenvalue λ_{OL} of the model in Equation (4)? Would the model in Equation (4) be stable with open-loop control if it also had a disturbance term?**

(HINT: For open-loop control, we set the velocities to $v_L[i] = v_R[i] = v_t$. What happens when we substitute that into Equations (1) and (2) and then apply the definition of $\delta[i]$ and $\delta[i+1]$?)

Part 2: Closed-Loop Control

Now, in order to make the car drive straight, we must implement closed-loop control – that is, control inputs that depend on the current state and are calculated dynamically – and use feedback in real time.

(c) **If we want the car to drive straight starting from some timestep $i_{\text{start}} > 0$, i.e., $v_L[i] = v_R[i]$ for $i \geq i_{\text{start}}$, what condition does this impose on $\delta[i]$ for $i \geq i_{\text{start}}$?**

(d) **How is the condition you found in the previous part different from the condition:**

$$\delta[i] = 0, \qquad i \geq i_{\text{start}}? \tag{5}$$

Assume that $i_{\text{start}} > 0$, and that $d_L[0] = 0, d_R[0] = 0$.

This is a subtlety that is worth noting and often requires one to adjust things in real systems.

(e) From here, assume that we have reset the distance travelled counters at the beginning of this maneuver so that $\delta[0] = 0$. We will now implement a feedback controller by selecting two dimensionless positive coefficients, f_L and f_R, such that the closed loop system is stable with eigenvalue λ_{CL}. To implement closed-loop feedback control, we want to adjust $v_L[i]$ and $v_R[i]$ at each timestep by an amount that's proportional to $\delta[i]$. Not only do we want our wheel velocities to be some target velocity v_t, we also wish to drive $\delta[i]$ towards zero. This is in order to have the car drive straight along the initial direction it was pointed in when it started moving. If $\delta[i]$ is positive, the left wheel has traveled more distance than the right wheel, so relatively speaking, we can slow down the left wheel and speed up the right wheel to cancel this difference (i.e., drive it to zero) in the next few timesteps. The action of such a control is captured by the following velocities.

$$v_L[i] = v_t - f_L\delta[i]; \tag{6}$$

$$v_R[i] = v_t + f_R\delta[i]. \tag{7}$$

Give expressions for $u_L[i]$ and $u_R[i]$ as a function of v_t, $\delta[i]$, f_L,f_R, and our system parameters $\theta_L.\theta_R, \beta_L, \beta_R$, to achieve the velocities above.

(f) Using the control inputs $u_L[i]$ and $u_R[i]$ found in part (e), **write the closed-loop system equation for $\delta[i+1]$ as a function of $\delta[i]$. What is the closed-loop eigenvalue $\lambda_{\mathbf{CL}}$ for this system in terms of $\lambda_{\mathbf{OL}}$, f_L, and f_R?**

(g) **What is the condition on f_L and f_R for the closed-loop system in the previous part to be stable in the presence of disturbance?**

Stability in this case means that δ is bounded and will not go arbitrarily high. In fact, if our calculated β and θ are perfectly accurate, then $\delta[i] \to 0$, so the car will (eventually) drive straight!

One question remains – what if our calculated β and θ are *not* perfectly accurate? The answer turns out to be that there is some small steady-state discrepancy that your δ will converge to. You will see how to quantify this in next week's homework.

2. Impact of Model Estimation Error on Open- and Closed-loop Control

In the previous problem, you worked on a System ID problem related to controlling the SIXT33N motor control circuitry to move a car in a straight line. This was done using both open-loop and closed-loop control. Recall that the original system model equations were

$$v_L[i] = d_L[i+1] - d_L[i] = \theta_L u_L[i] - \beta_L; \tag{8}$$

$$v_R[i] = d_R[i+1] - d_R[i] = \theta_R u_R[i] - \beta_R \tag{9}$$

1 where $u_{L,R}[i]$ represent the PWM inputs, $v_{L,R}[i]$ represent velocity outputs, and $\theta_{L,R}$ and $\beta_{L,R}$ represent the model parameters. You are encouraged to re-visit Homework 6, Question 8 for the detailed definitions of these parameters.

Furthermore, in the problem last week, you explored controlling the car using both open-loop and closed-loop systems to keep it driving in a straight line. We simplified the model to use a new state variable $\delta[i]$ as the difference between the left and right wheel distances traveled:

$$\delta[i] := d_L[i] - d_R[i] \tag{10}$$

In the open-loop case, we found that $\delta[i+1] = \delta[i]$, i.e. the open-loop eigenvalue of the discrete-time system is $\lambda_{\text{OL}} = 1$. This did not meet the stability criteria: it forms an unstable system in the presence of disturbances.

One source of disturbance is the error between our model's estimates of parameters θ_L, θ_R, β_L, and β_R vs. their true physical values. Our model uses estimates of these parameters, which themselves are learned from noisy data and have some inherent inaccuracies. This week, we want to understand how much these model inaccuracies impact our car control.

(a) Recall that in the open-loop case, we found simple equations for the inputs:

$$u_L[i] = \frac{v_t + \beta_L}{\theta_L} \tag{11}$$

$$u_R[i] = \frac{v_t + \beta_R}{\theta_R} \tag{12}$$

where v_t is the model's target velocity for both wheels when the car is going straight.

Let $\theta_L^\star, \theta_R^\star, \beta_L^\star, \beta_R^\star$ be the true physical values for the parameters, which our model does not know. Instead, our model uses our best estimates of the parameters, i.e. $\theta_L, \theta_R, \beta_L, \beta_R$ as before. Mathematically, the true physical model of our system is:

$$v_L[i] = \theta_L^\star u_L[i] - \beta_L^\star; \tag{13}$$

$$v_R[i] = \theta_R^\star u_R[i] - \beta_R^\star \tag{14}$$

but the inputs $u_{L,R}[i]$ are still using the model's estimates as per equations 11 and 12.

Suppose that there is a 10% relative error between θ_L in the model and $\theta_L^\star$ in the physical system. That is,

$$\frac{\theta_L^\star - \theta_L}{\theta_L} = 0.1. \tag{15}$$

Also assume there is no relative error between β_L in the model and $\beta_L^\star$ in the physical system. That is, $\beta_L = \beta_L^\star$.

If we used the open-loop control inputs from 11, what would be the resulting velocity relative error, $\frac{v_L[i]-v_t}{v_t}$?

NOTE: For concreteness, use the values $\theta_L = 2$, $\beta_L = \beta_L^\star = -2.5$, and $v_t = 200$, but $\theta_L^\star = 2.2$.

As you saw above, there is some discrepancy between what the model thinks the velocity of the left wheel is vs. what is is in reality for the open-loop controller. The same can be said for the right wheel - and it could be a different discrepancy from the left. This would result in our car not going straight.

Last week, we introduced closed-loop control to stabilize the system. This helps the controller achieve the desired result in the presence of disturbances, such as the parameter estimation errors we introduced here. Let's see how well our closed-loop controller does in the presence of these errors.

Recall our closed-loop controller velocity equations:

$$v_L[i] = v_t - f_L\delta[i] \tag{16}$$
$$v_R[i] = v_t + f_R\delta[i] \tag{17}$$

where v_t is the target velocity (for both wheels) and $f_{L,R}$ are the feedback coefficients for each wheel. This yielded our left and right closed-loop control inputs:

$$u_L[i] = \frac{v_t - f_L\delta[i] + \beta_L}{\theta_L} \tag{18}$$
$$u_R[i] = \frac{v_t + f_R\delta[i] + \beta_R}{\theta_R} \tag{19}$$

Moreover, you showed that the closed-loop discrete-time system is:

$$\delta[i+1] = \delta[i] + \theta_L u_L[i] - \theta_R u_R[i] - \beta_L + \beta_R \tag{20}$$

which you then showed could be stabilized for a range of $f_L + f_R$.

(b) Suppose we have 10% relative error between estimated model parameters θ_L, θ_R and the real model parameters $\theta_L^\star, \theta_R^\star$:

$$\frac{\theta_L^\star - \theta_L}{\theta_L} = +0.1; \tag{21}$$
$$\frac{\theta_R^\star - \theta_R}{\theta_R} = -0.1. \tag{22}$$

Also suppose there is no relative error between estimated model parameters β_L, β_R and real model parameters $\beta_L^\star, \beta_R^\star$ in the physical system. That is, $\beta_L = \beta_L^\star$ and $\beta_R = \beta_R^\star$. **Given these estimation errors, what is the true system equation? What is the closed-loop eigenvalue λ_{CL} of the actual system?**

(c) If there were no estimation errors in the model parameters and there were no other source of disturbances, the state variable $\delta[i]$ would eventually converge to 0 assuming the system is stable, i.e., $|\lambda_{\text{CL}}| < 1$. However, with the given estimation error, $\delta[i]$ may not converge to 0 but to some other constant, which is called the steady state error $\delta_{SS} = \lim_{i\to\infty} \delta[i]$.

Remember, BIBO stability just promises that a bounded disturbance gives rise to a bounded output — it doesn't say that the result will be zero.

What is the steady state error δ_{SS} given 10% estimation error in θ_L and θ_R as in Equations (21) and (22)? Assume that even with the estimation error, you have chosen f_L and f_R such that $|\lambda_{\text{CL}}| < 1$.

You should see that this is not zero, but instead depends on the target velocity v_t as well as the β_R and β_L constants. Physically, this reflects the fact that the car will go straight, but it might turn a little before starting to go straight.

3. Cayley-Hamilton and Controllability Matrix

(a) We can define the *characteristic polynomial* of a matrix $A \in \mathbb{R}^{n\times n}$ as

$$p_A(\lambda) = \lambda^n + c_{n-1}\lambda^{n-1} + \cdots + c_1\lambda + c_0\lambda^0 \tag{23}$$

where each $c_i \in \mathbb{R}$ is a constant. The characteristic polynomial has roots that are the eigenvalues of A. That is, we can equivalently define

$$p_A(\lambda) = \det\{\lambda I - A\} \tag{24}$$

We say that any of the eigenvalues of A "satisfy" the characteristic polynomial in that

$$p_A(\lambda_i) = 0 \tag{25}$$

where λ_i is the ith eigenvalue of A. Now, let A be a diagonalizable matrix, where we may write $A = V\Lambda V^{-1}$. **Prove that A satisfies its own characteristic polynomial.** In other words, prove that $p_A(A) = 0_{n\times n}$, where $0_{n\times n}$ is a $n \times n$ matrix of zeros.

(HINT: It is not correct to simply plug in $\lambda = A$ into $\det\{\lambda I - A\}$.)

(b) Now, consider some vector $\vec{b} \in \mathbb{R}^n$. **Using the result from the previous part, show that $A^n\vec{b}$ is linearly dependent on $A^{n-1}\vec{b}, A^{n-2}\vec{b}, \ldots, A\vec{b}, \vec{b}$.**

(c) Instead of setting $\vec{b}$ to be a vector, let it be a matrix $B \in \mathbb{R}^{n\times m}$. **Now, show that the columns of A^nB are linearly dependent on the columns of $A^{n-1}B, A^{n-2}B, \ldots, AB, B$.**

(HINT: If we were to write $B = \begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \cdots & \vec{b}_m \end{bmatrix}$ where each column is n-dimensional, we can write $A^iB = \begin{bmatrix} A^i\vec{b}_1 & A^i\vec{b}_2 & \cdots & A^i\vec{b}_m \end{bmatrix}$. Make sure you convince yourself of this.)

(d) Consider a discrete time system of the form

$$\vec{x}[i+1] = A\vec{x}[i] + B\vec{u}[i] \tag{26}$$

where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$. The controllability matrix for this discrete time system is given by

$$\mathcal{C} = \begin{bmatrix} A^{n-1}B & A^{n-2}B & \cdots & AB & B \end{bmatrix} \tag{27}$$

Conclude that the rank of your controllability matrix will not change if, instead, you made your controllability matrix $\begin{bmatrix} A^nB & A^{n-1}B & \cdots & AB & B \end{bmatrix}$ (i.e., you prepended A^nB to your original controllability matrix).

4. CCF Transformation and Controllability

(a) Consider the following discrete time system

$$\vec{x}[i+1] = A\vec{x}[i] + B\vec{u}[i] \tag{28}$$

Suppose we define a change of basis operation given by $M\vec{z}[i] = \vec{x}[i] \iff \vec{z}[i] = M^{-1}\vec{x}[i]$. This yields a new discrete time system of the form

$$\vec{z}[i+1] = \widetilde{A}\vec{z}[i] + \widetilde{B}\vec{u}[i] \tag{29}$$

for some $\widetilde{A}$ and $\widetilde{B}$ defined in terms of M, A, and B. **What is the controllability matrix for the system in eq. (29), in terms of M, A, and B?**

(b) Consider the change of basis given by $\vec{z}[i] = T^{-1}\vec{x}[i]$ where, under this change of basis transformation, we have the following discrete time system

$$\vec{z}[i+1] = A_{\text{CCF}}\vec{z}[i] + B_{\text{CCF}}\vec{u}[i] \tag{30}$$

Using the result from the previous part, determine an expression for T in terms of $\mathcal{C}$, the controllability matrix of the original system in eq. (28), and $\mathcal{C}_{\text{CCF}}$, the controllability matrix of the system in eq. (30).

(c) We know that the controllability matrix for a system in CCF will always be full rank. **Using this, prove that you can find a transformation matrix T as in the previous part if and only if your original system is controllable.** *(HINT: To prove this, you can first show that, if such a T exists, then your original system is controllable. Then, you can show that, if your original system is controllable, there will exist such a transformation matrix T.) (HINT: Recall that T must be invertible (equivalently, full rank) in order for it to be a valid transformation matrix. You may use without proof the fact that* $\text{rank}(AB) = \min(\text{rank}(A), \text{rank}(B))$.*)*

(d) Consider the following discrete-time dynamics model:

$$\vec{x}[i+1] = \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_{A} \vec{x}[i] + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\vec{b}} \vec{u}[i] \tag{31}$$

Find the transformation matrix T such that the dynamics model for $\vec{z}[i] = T^{-1}\vec{x}[i]$ is in CCF. You may use a calculator/computer to perform any computations, if you wish.

(HINT: First, find the characteristic polynomial of A. Use this to determine what A_{CCF} and $\vec{b}_{\text{CCF}}$ should be, and then use this to determined $\mathcal{C}_{\text{CCF}}$.)

Contributors:

- Bozhi Yin.
- Kaitlyn Chan.
- Yi-Hsuan Shih.
- Vladimir Stojanovic.
- Moses Won.
- Druv Pai.
- Wahid Rahman.
- Anish Muthali.

