Homework 8

This homework is due on Monday, March 28, 2022, at 11:59PM. Self-grades and HW Resubmissions are due on Friday, April 1, 2022, at 11:59PM.

1. **Reading Lecture Notes**

   Staying up to date with lectures is an important part of the learning process in this course. Here are links to the notes that you need to read for the homework this week: Note 11, Note 12.

   (a) Is a system of the form

   $$\vec{x}[i+1] = \vec{x}[i] + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u[i] \tag{1}$$

   controllable? Here $\vec{x}[i] \in \mathbb{R}^2$.

   (b) Is the above system stable in the presence of disturbances?

2. **Stability for Information Processing: Solving Least-Squares via Gradient Descent with a Constant Step Size**

In this problem, we will derive a dynamical system approach for solving a least-squares problem which finds the $\vec{x}$ that minimizes $\|A\vec{x} - \vec{y}\|^2$. Here, we assume $A \in \mathbb{R}^{m \times n}$ to be tall and full rank — i.e., it has linearly independent columns.

As covered in EECS16A, this has a closed-form solution:

$$\vec{\hat{x}} = (A^\top A)^{-1} A^\top \vec{y}. \tag{2}$$

In many cases, direct computation of this solution is too slow, because it requires computing $(A^\top A)^{-1}$, which is generally very costly. We will instead solve the problem iteratively using so-called "gradient descent" which turns this particular problem into an analysis of a particular discrete-time state-space dynamical system. *Gradient descent is out of scope for the next few weeks, but the other ideas in this problem are all in scope.*

(a) **Show that $\vec{y} - A\vec{\hat{x}}$ is orthogonal to the columns of A, i.e., show $A^\top(\vec{y} - A\vec{\hat{x}}) = \vec{0}$.**
NOTE: This was shown to you in 16A, but it is important that you see this for yourself again.

(b) In iterative optimization schemes such as gradient descent, we will get a sequence of estimates for $\vec{\hat{x}}$ at each timestep. Let $\vec{x}[i]$ denote our estimate for $\vec{\hat{x}}$ at timestep i.

When implementing the optimization algorithm, we choose the update rule and thus the state space equation. We would like to develop a state space equation for which the state $\vec{x}[i]$ will converge to $\vec{\hat{x}}$, the true least squares solution.

Gradient descent gives us the following update rule:

$$\vec{x}[i+1] = \vec{x}[i] + \alpha A^\top(\vec{y} - A\vec{x}[i]) \tag{3}$$

that gives us an updated estimate using the previous one. Here α is the "step size" or "learning rate" that we choose. For us in 16B, it doesn't matter where we got the update rule, but the important thing to note is that if $\vec{x}[i] = \vec{\hat{x}}$, then by the previous part, $\vec{x}[i+1] = \vec{\hat{x}}$ and the system remains in equilibrium at $\vec{\hat{x}}$ for all time.

To show that $\vec{x}[i] \to \vec{\hat{x}}$, we define a new state variable $\Delta\vec{x}[i] = \vec{x}[i] - \vec{\hat{x}}$. It represents the deviation from where we want to be.
Derive the discrete-time state evolution equation for $\Delta\vec{x}[i]$, and show that it takes the form:

$$\Delta\vec{x}[i+1] = (I - \alpha G)\Delta\vec{x}[i]. \tag{4}$$

What is G?

(c) We would like to change the learning rate α such that $\Delta\vec{x}[i]$ converges to 0. In particular, we want to make sure that we have a stable system. To do this, we need to understand the eigenvalues of $I - \alpha G$. **Show that the eigenvalues of matrix $I - \alpha G$ are $1 - \alpha\lambda_k\{G\}$, where $\lambda_k\{G\}$ are the eigenvalues of G, for $k \in \{1, 2, \dots, n\}$.**

(d) For system eq. (4) to be stable, we need all the eigenvalues of $I - \alpha G$ to have magnitudes that are smaller than 1 (since this is a discrete-time system). **Above what value of α would the system (4) become unstable?** This is what happens if you try to set the α to be too high.

NOTE: Since the matrix G above has a special form, all of the eigenvalues of G are non-negative and real. (You'll see why later in this course.)

(e) **Play with the given Jupyter notebook and comment on what you observe.** How does the size of the learning rate α affect the estimate? At what speed are the estimates converging?

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

In last week's homework (Homework 7, Question 5), 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

$$v_L[i] = d_L[i+1] - d_L[i] = \theta_L u_L[i] - \beta_L; \tag{5}$$

$$v_R[i] = d_R[i+1] - d_R[i] = \theta_R u_R[i] - \beta_R \tag{6}$$

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:

$$\delta[i] := d_L[i] - d_R[i] \tag{7}$$

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:

$$u_L[i] = \frac{v_t + \beta_L}{\theta_L} \tag{8}$$

$$u_R[i] = \frac{v_t + \beta_R}{\theta_R} \tag{9}$$

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:

$$v_L[i] = \theta_L^\star u_L[i] - \beta_L^\star; \tag{10}$$

$$v_R[i] = \theta_R^\star u_R[i] - \beta_R^\star \tag{11}$$

but the inputs $u_{L,R}[i]$ are still using the model's estimates as per equations 8 and 9.

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

$$\frac{\theta_L^\star - \theta_L}{\theta_L} = 0.1. \tag{12}$$

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 8, 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 $\theta_L^\star = 2.2$.

© UCB EECS 16B, Spring 2022. All Rights Reserved. This may not be publicly shared without explicit permission.

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:

$$v_L[i] = v_t - f_L\delta[i] \tag{13}$$
$$v_R[i] = v_t + f_R\delta[i] \tag{14}$$

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:

$$u_L[i] = \frac{v_t - f_L\delta[i] + \beta_L}{\theta_L} \tag{15}$$
$$u_R[i] = \frac{v_t + f_R\delta[i] + \beta_R}{\theta_R} \tag{16}$$

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

$$\delta[i+1] = \delta[i] + \theta_L u_L[i] - \theta_R u_R[i] - \beta_L + \beta_R \tag{17}$$

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 $\theta_L^\star, \theta_R^\star$:

$$\frac{\theta_L^\star - \theta_L}{\theta_L} = +0.1; \tag{18}$$
$$\frac{\theta_R^\star - \theta_R}{\theta_R} = -0.1. \tag{19}$$

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 (18) and (19)? 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.

© UCB EECS 16B, Spring 2022. All Rights Reserved. This may not be publicly shared without explicit permission.

4. **(OPTIONAL) Make Your Own Problem.**

   Write your own problem about content covered in the course thus far, and provide a thorough solution to it.

   NOTE: This can be a totally new problem, a modification on an existing problem, or a Jupyter part for a problem that previously didn't have one. Please cite all sources for anything (including course material) that you used as inspiration.

   NOTE: High-quality problems may be used as inspiration for the problems we choose to put on future homeworks or exams.

5. **Homework Process and Study Group**

   Citing sources and collaborators are an important part of life, including being a student!

   We also want to understand what resources you find helpful and how much time homework is taking, so we can change things in the future if possible.

   (a) **What sources (if any) did you use as you worked through the homework?**

   (b) **If you worked with someone on this homework, who did you work with?**
   List names and student ID's. (In case of homework party, you can also just describe the group.)

   (c) **Roughly how many total hours did you work on this homework? Write it down here where you'll need to remember it for the self-grade form.**

Contributors:

- Kyle Tanghe.
- Anant Sahai.
- Miki Lustig.
- Aditya Arun.
- Nathan Lambert.
- Sidney Buchbinder.
- Bozhi Yin.
- Kaitlyn Chan.
- Yi-Hsuan Shih.
- Vladimir Stojanovic.
- Moses Won.
- Druv Pai.
- Wahid Rahman.

© UCB EECS 16B, Spring 2022. All Rights Reserved. This may not be publicly shared without explicit permission.