

Assignment for Week 9 (continued lab 8)

- Issued: Tuesday, October 31st
- Tutor Problems due by 1 p.m., Thursday, November 2
- Postlab due Tuesday, November 6th in class.

Dynamic Feedback Control Continued

This lab is an updated version of last week's lab, so you have it for handy reference, with changes for this week. We have indicated the sections that have changed.

Two weeks ago, you used a simple feedback scheme for controlling the robot so that it drove down the center of a narrow corridor at a constant velocity. The scheme we suggested was that you make the robot's rotational velocity equal to the product of a "gain," K , and the measured distance from the center of the corridor. You then developed a mathematical model for the controller in the form of a homogeneous difference equation. An analysis of how the natural frequencies of the homogeneous difference equation changed with K suggested that the simple feedback scheme would not work, regardless of the value of K . It should be noted that for most of the robots, the mathematical analysis produced a qualitatively accurate prediction of how the robot behaved, but as with any physical experiment, the results were not perfectly consistent.

This week you will revisit the robot feedback control problem, but this time you will use the more sophisticated manipulation techniques associated with the Z transform to help you design a more effective controller. This controller should also allow you to change the robot's distance from the walls as it proceeds down the corridor.

As is often the case, the first step will be to develop some useful tools in Python, and then examine some model results before working with the robot. In particular, you will develop methods for manipulating transfer functions and methods for computing difference equations and natural frequencies from the transfer functions.

The summary of tasks for this week are:

- BEFORE LAB: Tutor exercises to practice Z transforms
- IN LAB: Write a Python program to manipulate transfer functions
- IN LAB: Use your program to develop an improved robot controller
- IN LAB: Conduct experiments with the robot to demonstrate your approach
- POST LAB: Describe the design of your controller.
- POST LAB: Final Tutor exercise.

1. Homework in preparation for lab

This section includes problems to complete with the online tutor. Most of these are due before lab on Thursday, but the last one is due on Tuesday after lab. The examples are to help you solidify your understanding of Z transforms.

1.0 Read this whole document!

Please read the entire lab before beginning.

1.1. Exercises with the online tutor

Use the online tutor to complete the first two tutor problems.

2. To do in lab

Today's lab will begin with a mini-quiz that is based on the tutorial problems due for today, and on material covered this week. It should be no great surprise that you will be asked a question about the Z transform. Please keep in mind that the point of the quizzes is as a diagnostic for us (did we succeed in teaching you?) and to encourage you to do the tutorial problems before coming to lab (did you prepare?). After the quiz, you should work as part of a group with a single computer and robot.

2.1 Transfer Function Manipulation Program

Consider starting from a difference equation in a new form we described in class

$$\sum_{k=0}^M a_k y[n+k] = \sum_{l=0}^M b_l x[n+l].$$

Note that, as discussed in class, this new form makes some manipulations easier than when using the very different *normal* form that was used in the difference equation solving and plotting programs you have already written. Review the class notes to understand how to convert from this new form back to the normal form.

We learned to use the Z transform to generate a relation of the form

$$\tilde{Y}(z) = \tilde{H}(z)\tilde{X}(z)$$

where

$$\tilde{H}(z) = \frac{\sum_{l=0}^M b_l z^l}{\sum_{k=0}^M a_k z^k}.$$

We also learned that if the output of one difference equation is used as the input to a second difference equation, then the composite system can be determined by forming the product of the transfer functions. That is, if

$$\tilde{W}(z) = \tilde{H}_1(z)\tilde{X}(z)$$

and

$$\tilde{Y}(z) = \tilde{H}_2(z)\tilde{W}(z)$$

then

$$\tilde{Y}(z) = \tilde{H}_2(z)\tilde{H}_1(z)\tilde{X}(z) = \tilde{H}(z)\tilde{X}(z)$$

where

$$\tilde{H}(z) = \tilde{H}_2(z)\tilde{H}_1(z).$$

We also discussed feedback systems and derived Black's formula. In particular, we defined an error, $e[n]$, by

$$e[n] = x[n] - y[n]$$

where $x[n]$ is an input (what we want) and $y[n]$ is an output (what we got). Transforming the error definition equation yields a simple relation between the Z transform of the error and the Z transform of X and Y ,

$$\tilde{E}(z) = \tilde{X}(z) - \tilde{Y}(z).$$

If the error is used to adjust $y[n]$, then typically the relationship between the error and the output will be given by a difference equation

$$\sum_{k=0}^M a_k y[n+k] = \sum_{l=0}^M b_l e[n+l].$$

Note that the input to the above difference equation is $e[n]$. Computing the Z transform of the difference equation yields a transfer function relating the error and the output

$$\tilde{Y}(z) = \tilde{H}(z)\tilde{E}(z) = \frac{\sum_{l=0}^M b_l z^l}{\sum_{k=0}^M a_k z^k} \tilde{E}(z).$$

Eliminating $\tilde{E}(z)$ yields Black's formula, a relation between the input and output of a feedback system:

$$\tilde{Y}(z) = \frac{\tilde{H}(z)}{1 + \tilde{H}(z)} \tilde{X}(z).$$

For this part of the lab, you will write a Python program to automate the above manipulations, using some helpful routines posted on the 6.081 web site.

Assume that you start with transfer functions which are ratios of polynomials in z , as in

$$\frac{\sum_{l=0}^M b_l z^l}{\sum_{k=0}^M a_k z^k}.$$

NOTICE CHANGE FROM LAST WEEK – WE HAVE PROVIDED THE PROGRAM

We have written a program that will compose and manipulate transfer functions in the following ways:

- Create a new transfer function from the product of two transfer functions
- Create a new transfer function that is a Black's formula transformation of a transfer function

The result of any of the manipulations is a transfer function in the same form, ratios of polynomials. The transfer functions are represented as numerator and denominator polynomials using a polynomial class.

In addition, our program is able to:

- Compute the natural frequencies associated with a transfer function
- Generate a difference equation in a form suitable for a difference equation solving program which you can call from the SOAR graphing program.

For the second item, notice that we used a difference equation solver that converts from the new difference equation back to the *normal* form of a difference equation.

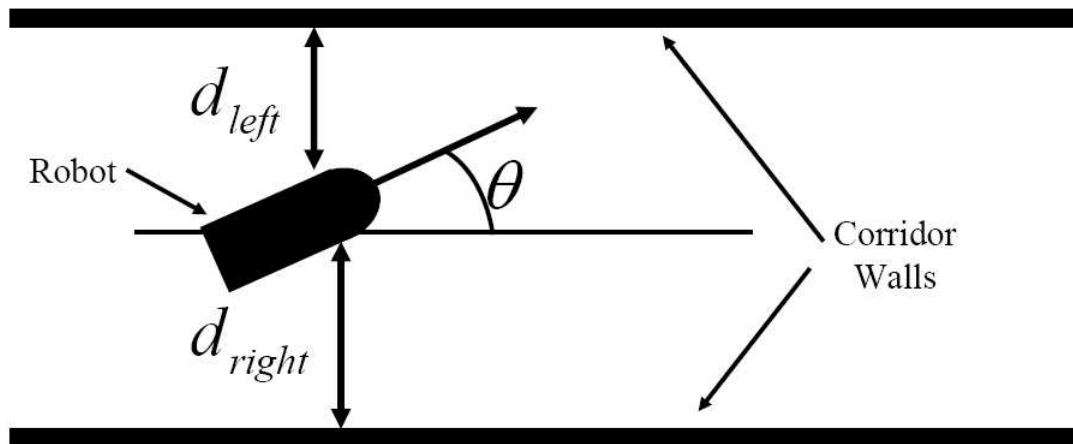


Figure 1: Robot in corridor.

Checkpoint: 2:00 PM

- Download our software from the 6.081 web site (week 9 on the calendar)
- Demonstrate that you understand the *testing* procedure
- Demonstrate that you can use the new software and SOAR's graphing to plot the solution of a difference equation described by its transfer function.

2.2 Robot Feedback System

Just as in the previous lab, you will be controlling the robot to drive down a narrow corridor as shown in Figure 1. Notice that in the figure, we have denoted the forward velocity of the robot, V , the distance to the left wall, d_{left} , the distance to the right wall, d_{right} , and the angle the robot is making with respect to the parallel walls, θ .

Consider the problem of trying to steer the robot straight down the hall so as to maintain a desired distance, $d_{desired}$, while moving forward with a constant velocity. For example, if $d = d_{right} - d_{left}$ is equal to $d_{desired}$, and the robot is pointing in a direction parallel to the walls, one could keep the robot moving forward and maintaining a distance $d_{desired}$ from the corridor center with the command

```
motorOutput(0.1, 0.0),
```

where the first argument is the forward velocity and the second argument is rotational velocity.

If the robot is too close to the right wall, one could issue the command

```
motorOutput(0.1, 0.1),
```

which will keep the robot moving at a constant velocity but will also cause the robot to rotate to the left. If the robot is too close to the left wall, one could issue the command

```
motorOutput(0.1, -0.1)
```

which will keep the robot moving at a constant velocity but will cause the robot to rotate to the right.

Last time you used a very simple control strategy to keep the robot in the center of the corridor, $d_{desired} = 0$, by adjusting the rotational velocity. Now consider the slightly more general problem of trying to control the robot so as to maintain a given distance from the corridor center. One could define an error at step n , $e[n]$, by

$$e[n] = d_{desired}[n] - d[n].$$

Then, the objective would be to try to keep the magnitude of $e[n]$ as small as possible.

We can use almost the same difference equation as last lab to model this slightly more general case. In particular, the center displacement will still be related to the robot angle by

$$d[n + 1] = d[n] + V\delta t\theta[n]$$

where δt is the system's time between samples and V is the robot forward velocity.

If we use the same control strategy as in the last lab, then the robot angle will satisfy a slightly modified difference equation for this more general input case,

$$\theta[n + 1] = \theta[n] + \delta tKe[n]$$

where K is the “gain” of the feedback. You experimented with K in the last lab.

Use the transfer function manipulation program to combine the above difference equations and generate a system that relates $d_{desired}$ and d . Then use your program to compute the natural frequencies of the system and plot $d[n]$ using $d_{desired}[n] = 1^n$ (you can assume the initial values for d are all zero). Please use reasonable estimates of δt and V , but try several values for K .

Please make a careful note of the sign of K , the definition of the error means the sign will not be the same as the lab two weeks ago.

Checkpoint: 2:45 PM

- Plots of your difference equation responses and system natural frequencies for several different K values.
- An explanation for the relation between the natural frequency locations and the response plots.

2.3 Design a Better Controller

In order to design a better controller, one can process the error, $e[n]$, in a more sophisticated way. For example, one could adjust the rotational velocity using some combination of the present

and previous values of the displacement error. The robot angle would then satisfy the difference equation

$$\theta[n + 1] = \theta[n] + \delta t * (K_1 * e[n] + K_2 * e[n - 1])$$

NOTE CHANGE HERE FROM LAST WEEK!

Write a program that will take K_1 and K_2 as inputs, and use the transform manipulation program system to determine natural frequencies and plots of time responses. You can use this program to help you determine effective values for K_1 and K_2 . Be sure to think carefully about what happens to the system natural frequencies as you change K_1 and K_2 .

You may find it difficult to find good values of K_1 and K_2 . There are many approaches. Ask an LA or the TA for a hint.

Checkpoint: 4:00 PM

- Plots of your difference equation responses and system natural frequencies for $x[n] = 1^n$ inputs and using your program which uses K_1 and K_2 values.
- An explanation for what is good about the selected values for K_1 and K_2 .

2.3 Try Your Better Controller on the Robot

Experiment with your improved dynamic feedback control scheme by writing a brain that on every step, sets the robot velocity and updates the robot rotation based on your controller design and a measurement of the distance from the wall (to reduce noisy readings, be sure to using the same minimization over a pair of sensor readings that you used two weeks ago). First try your controller setting $d_{desired} = 0$ and initially offsetting the robot from the center of the corridor. Then, if time permits, try to have the robot move from nearer the left wall to nearer the right wall about halfway down the corridor. Be sure to try your robot in one of our specially designed corridors. Experiment with different constants in your feedback system, not just the ones you designed, and see how much the performance changes.

When doing this section, beware of the following pitfalls:

- $\theta[n]$ corresponds to the controller's desired *angle*. But the second argument to `motorOutputs()` is the robot's angular *velocity*. How will you reconcile this?
- In section 2.2, we asked you to make a "reasonable estimate" of δt . What is the actual time between steps when running on a Pioneer robot? Is this close to your estimate? What will be the consequence if your estimate is off?
- How will you measure $e[n]$?
- In order to produce a difference equation object from a transfer function, we have to supply an *inputSource* function that supplies the inputs — in this case, $e[n]$. This function takes one argument (n), which can be in the past, and returns $e[n]$. How will you implement an *inputSource* function that does this?

Checkpoint: 5 PM

- | |
|---|
| <ul style="list-style-type: none">• Demonstrate your controller on the robot. |
|---|

3. To do before Tuesday following lab

Work on the following problems after Thursday's lab and turn them in at the beginning of Tuesday's class.

3.1. Lab writeup

Please write a brief analysis of your controller, indicating clearly the relation between the constants K_1 and K_2 . In addition, briefly describe your experimental results and comment on the relation between the model behavior and the actual robot behavior.

3.2. Complete the last tutor problems

Be sure to complete the last question of the tutor problems.

Concepts covered in this assignment

Here are the important points covered in this assignment:

- Learn about transfer functions and how to manipulate them
- Learn about how to analyze feedback systems
- Gain a more complete understanding of feedback stability