

Assignment for Week 6

This handout contains

- Software Lab for March 13th
- Prelab exercises due Thursday March 15th before lab
- Post-lab exercises due Tuesday March 20th in Lecture.

Dynamic Feedback Control

Last week, 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:

- Post-lecture software lab on manipulating transfer functions
- Pre-lab tutor exercises to practice with Z transforms
- Robot lab on implementing a better controller
- Post-lab writeup and exercises due Tuesday in lecture.

Tuesday's Software Lab: Transfer Function Manipulation Program

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

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

This new difference equation form makes a variety of manipulations easier as it is more directly associated with the difference equation's transfer function. Recall that the transfer function relates the Z transform of x to the Z transform of y , as in

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

where $\tilde{Y}(z) = \sum_{-\infty}^{\infty} y[n]z^{-n}$, $\tilde{X}(z) = \sum_{-\infty}^{\infty} x[n]z^{-n}$, and

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

is the transfer function.

As we saw in lecture, most ways of combining difference equations correspond to very simple manipulations of the associated transfer functions. For this post-lecture lab, you will write a python transfer function class which automates these manipulations.

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

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

Functions which are ratios of polynomials are referred to as rational functions.

Please write your transfer function class so that it can 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.

Note that for the purposes of this lab, we will assume the Black's formula transformation of a transfer function is given by

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

where $\tilde{H}(z)$ is the original transfer function and $\tilde{H}_B(z)$ is the transformed result. Please be sure that for your class, the result of any of the above manipulations is a rational function.

In addition to manipulating transfer functions, your program should be able to:

- create a transfer function from a difference equation description,
- compute the natural frequencies associated with a transfer function,
- print the associated difference equation in an easily understood format.

You will likely find it easiest to represent the transfer functions as numerator and denominator polynomials, in which case you can use the polynomial class on the 6.081 web site. The polynomial class has functions for adding and scaling polynomials, finding roots of the polynomials, and for polynomial multiplication. Recall that you must execute `import Polynomial.py` to import `polynomial.py`, and that you can see the class interface by typing

`help(Polynomial)`

at the IDLE command line.

As an example to test your program (you should try more than one), consider the two transfer functions

$$H_1(z) = \frac{z+1}{z+2} \quad H_2(z) = \frac{2z+1}{z+3}.$$

The product of the two transfer functions is

$$H_3(z) = H_1(z)H_2(z) = \frac{2z^2 + 3z + 1}{z^2 + 5z + 6}.$$

The Black's formula transformation of $H_3(z)$ is

$$H_4(z) = \frac{H_3(z)}{1 + H_3(z)} = \frac{2z^2 + 3z + 1}{3z^2 + 8z + 7}.$$

Finally, the natural frequencies of $H_4(z)$ are $\frac{4}{3} + j\frac{\sqrt{5}}{3}$ and $\frac{4}{3} - j\frac{\sqrt{5}}{3}$.

To Hand In

You will need your transfer function class in Thursday's lab, please be sure to save a copy where you can access it later. You can hand in a description of your transfer function class on Tuesday March 20th as part of Thursday's lab.

Homework in preparation for lab

This section includes problems to complete with the online tutor. These are due before lab on Thursday. The examples are to help you solidify your understanding of Z transforms.

Read this whole document!

Please read the entire lab before beginning.

Exercises with the online tutor

Use the online tutor to complete the tutor problems.

Thursday's Robot Lab - A better controller

Today's lab will end with a nano-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.

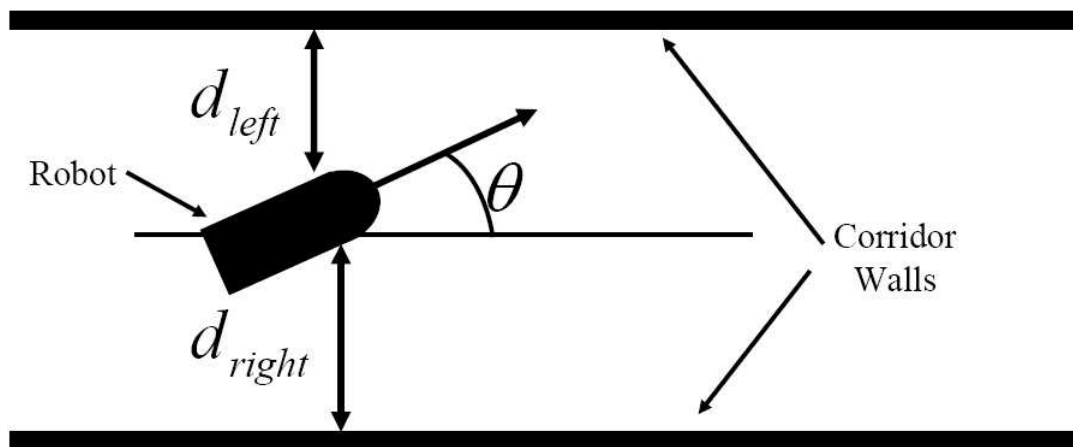


Figure 1: Robot in corridor.

Robot Feedback System with Input

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 speed 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 . Unlike last week, when you tried to steer the robot to drive straight down the center of the hallway, this week you will be trying to steer the robot to stay a desired distance, $d_{desired}$, from the center of the hallway (if $d_{desired} = 0$ the problem is equivalent to last week's problem). So, if $d = 0.5 * (d_{right} - d_{left})$ (we forgot the $1/2$ in last week's lab) 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),
```

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 forward speed 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 speed 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 speed based on the current value of $d[n]$. You discovered that a model of such a controller predicted robot behavior that would oscillate and eventually hit one of the walls. This week, you will design a better controller.

Using Z transforms to analyze the simple controller

Consider trying to steer the robot down a hallway so as to maintain a desired distance, $d_{desired}$, from the center of the hallway while moving forward with a constant speed. Let the error at step n , $e[n]$, be defined as

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

where $d[n]$ is the measured distance at time n from the center of the hallway. Then, the objective would be 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 more general case of an adjustable desired displacement. 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 speed.

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,

$$\theta[n + 1] = \theta[n] + \delta t K e[n]$$

where K is the “gain” of the feedback. You experimented with K with different values of K in the last lab. BE CAREFUL ABOUT THE SIGN, we flipped it! Notice that if $d_{desired} = 0$, then $K e[n] = -K d[n]$.

Demonstrate that you can use your transfer function manipulation program to combine the above difference equations, and generate a system that relates $d_{desired}[n]$ to $d[n]$. Also demonstrate that you can compute the system's natural frequencies. Note that you will have to determine δt and pick numerical values for the gain K .

Checkpoint: 2:45 PM

- Show your LA that you can use your transfer function class to determine the relation between gain and natural frequencies for the simple controller.

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 speed using some combination of the present and previous values of the displacement error. The robot angle would then satisfy the difference equation

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

Use your transform manipulation program and the system natural frequencies to help you determine effective values for K_1 and K_2 . Brute-force search methods can solve this problem (particularly if you write a program to search for you), but an alternative is to work the problem in reverse. Start with natural frequencies that you like, and then derive an associated difference equation.

Checkpoint: 3:45 PM

- Values of K_1 and K_2 ,
- An explanation for what is good about the selected values for K_1 and K_2 .

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 speed and updates the robot rotation based on your controller design and a measurement of the distance from the wall.

Try your controller on the robot in one of our corridors, with $d_{desired}[n] = 0$ and with the robot initially offset from center. In your experiment, you should compare the performance of your new controller with the performance of the simple controller from last time, so you should design your program so that you can easily switch between methods. Also, experiment with different constants for K_1 and K_2 in your feedback system, not just the ones you designed, and see how much the performance changes.

Checkpoint: 4:45 PM

- Demonstrate your controller on the robot.

If Time Permits

If you have demonstrated a working controller, then you can try the following problems.

Changing the input

Try to have the robot move from nearer the left wall to nearer the right wall about halfway down the corridor.

Plotting using your transfer function

Since a difference equation can be generated from a transfer function, it is possible to use the difference equation to generate plots of the output behavior given the input and the initial conditions. To do this, first try adding the ability to solve difference equations with inputs to the recursive difference equation solver from two weeks ago, and then plot the output using SoaR's graphingWindow utilities.

Once you have succeeded in plotting solutions to difference equations with inputs, try to add a function to your transfer function class that will plot the associated difference equation solution given an input and initial conditions. However, BE CAREFUL. The difference equation form for the transfer functions is not the same as the difference equation form used in the difference equation solver.

To Hand in on Tuesday March 20th

Hand in your transfer function program as well as a demonstration that your program produces the correct output for the example given in the Tuesday lab description as well as for the difference equations generated by the simple and improved controller.

Also, write a brief analysis of your controller, indicating clearly how you derived good values for 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.

(Optional) There is an intuitive reason for why the simple controller failed. The simple controller tries to drive the robot to the center of the corridor, but ignores the fact that the robot might be pointing to the right or to the left when arriving at the corridor center. How does your new controller try to ensure that the robot moves back to the desired corridor position while also trying to keep the robot pointing straight down the corridor?

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