QEA Segway Olympics

Coach's Notes for John Dough

Zack Davenport
Meaghen Sausville

November 13, 2017

Figure 1: A portrait of John Dough from his good side.

1 Meet John Dough

We're pleased to present our robot, John Dough, who will be entering his first Segway Olympics this year. John has come from a long history of baking, working at his family's doughnut shop for over two decades. Years of making doughnuts inspired him to spin in circles, a skill he has gotten quite good at. Mr. Dough's true passion, though, is dancing. He is particularly fond of waltzes by Strauss and has chosen the classic Blue Danube Waltz for his debut performance. In the Segway Olympics, John will be participating in the Survivor, the Spin, and the Salsa events.

2 Athlete Demographics

2.1 Block Diagram

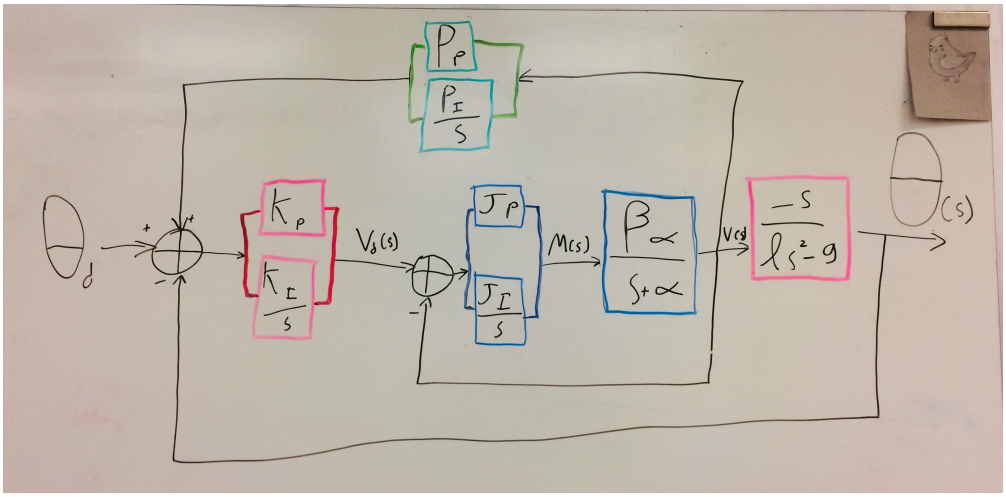


Figure 2: John's block diagram depicts his use of three PI loops to control motors, angles, and position.

John's block diagram, shown in Figure 2, utilizes three instances of proportional-integral (PI) control. The first block, outlined in pink, represents the angle control, and are governed by the parameters Ki and Kp. The process for deriving all parameters are discussed below. The pair of blue blocks in the middle represent the motor control, which are dependent on the parameters Jp and Ji. The pink box on the right side shows the control system for the pendulum. The light green box on the top is used to control the robot's position by integrating the output of the velocity to create a control system that loops back to the initial angle control. This second loop prevents John from drifting.

2.2 Transfer Function

Mr. Dough's transfer function, shown in Equation 1 below, looks like it fell from the ugly tree and hit every branch on the way down, but it is accurate and totally works. There are some things that even the makeup that is Simplify can't fix, but if you want to see it, here it is in all its glory. Kp and Ki control the angle, Jp and Ji control the motors, and Pp and Pi control the position. The calculated values of these parameters, and all others shown in this equation, are further discussed below.

$$-\frac{\alpha B\left(\frac{Ji}{s}+Jp\right)(Ki+Kps)}{s(\alpha+s)\left(\frac{\alpha B(Ji+Jps)(Ki+Kps)}{s(\alpha+s)(Ls^2-g)}+\frac{\alpha B(Ji+Jps)(Ki+Kps)(\Pi+Pps)}{s^3(\alpha+s)}-\frac{\alpha B\left(\frac{Ji}{s}+Jp\right)}{\alpha+s}-1\right)} \tag{1}$$

3 Athlete Performance Information

Most of our values, such as alpha, beta, and the length of the robot from the pivot point to the center of mass were determined in class. The real values we had to play with were the constants that controlled the proportional control and integral control for each of the three main loops. Jp and Ji controlled the velocities of the two wheels (seen in blue in the diagram above). In order to determine these, we first had to find its transfer function. Because that blue part is a relatively simple system that can be approximated to a controller and a plant, we could use Equation 2:

$$Transfer = \frac{Gc * Gp}{(1 + Gc * Gp)} \tag{2}$$

In this case, $Jp + \frac{Ji}{s}$ is Gc and $\frac{\beta\alpha}{s+\alpha}$ is Gp. Inputting these values gives you a denominator of

$$s^2 + (\alpha + Jp * \beta)s + Ji\beta\alpha. \tag{3}$$

We can use the quadratic formula to determine when critical damping is reached– when $b^2 = 4ac$. When applied to our formula, gives us a relationship between Ji and Jp:

$$Ji = \frac{(\alpha + \alpha\beta Jp)^2}{4\alpha\beta} \tag{4}$$

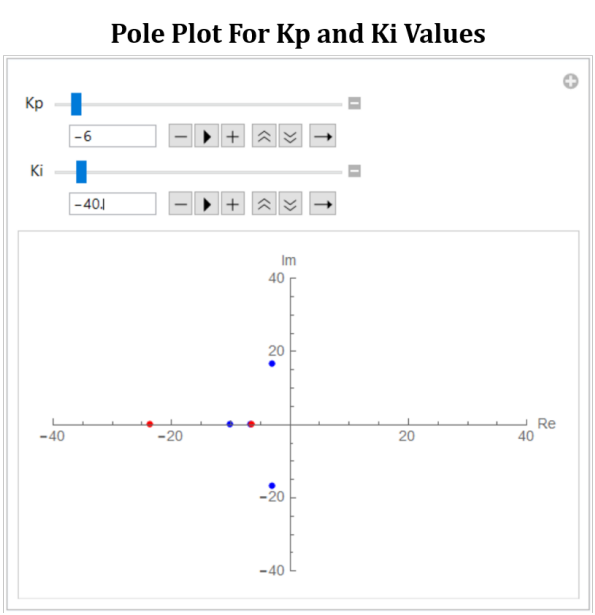
Jp and Ji control the robot's motor speed. These cannot be easily determined with a transfer function, but the values can be derived using knowledge about the motor speeds. Jp multiplied by the motor speed's error should not be more than the maximum motor speed, which is around 300. If we estimate the error to be around $\frac{3}{4}$, then a Jp value of 500 seems reasonable for our values. This can then be substituted back into the above equation to get a Ji of about 5000.

Once we had a Jp and Ji to start with, we could use that to play around with the poles in order to determine the Kp and Ki values (the pink blocks). We made use of Paul Ruvolo's Mathematica notebook that showed how a system's poles changed as Kp and Ki values changed. The poles were determined by finding the roots of the denominator of the transfer function. The ideal poles would be as negative in the real axis and close to zero in the imaginary axis as possible. This makes the system stabilize quickly with little oscillation. When the Kp value is -6 and the Ki value is -40, the plot showed that the poles were stable enough for our purposes. Testing these values on our robot confirmed that our system worked as we anticipated

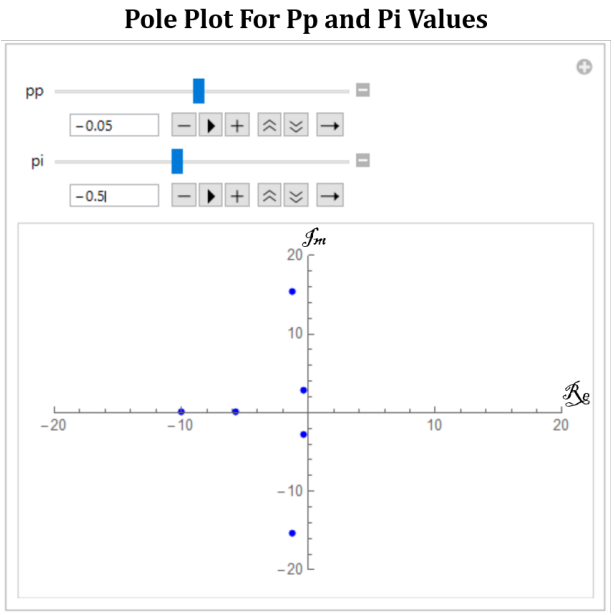
The last values we had to find were the Pp and Pi values (shown in green on the block diagram) for the positional control system. We found these values using the same process as our Kp and Ki values. First, we found the transfer function, used Mathematica to find the roots of the denominator, and plotted those poles. We found Pp and Pi values of -0.05 and -0.5, respectively, to work the best.

3.1 Pole Plots

The plots below in Figures 3a and 3b show the results of plotting our parameter values. We made use of the Manipulate function in to experiment with different values and to determine when all poles were negative and attempt to minimize oscillation. Negative poles indicate a stable system, and the magnitude on the imaginary axis determine the system's oscillation. We chose not to include a pole plot for Jp and Ji because those values were determined based on the maximum motor speed and their pole plots didn't affect our decision about which values to choose.



(a) The above plot for Ki and Kp shows that all poles are negative, indicating stability in our system.



(b) The above plot for Pp and Pi shows that all poles are negative, indicating stability in our system.

4 Training Session Description

We had a runaway issue after Stand-Up Rocky—it would stay straight up but after a while it would accelerate quickly in one direction. We tried fixing this by adding in an extra integration loop that fed into the desired angle. As the robot moved in one direction, we would integrate its speed to find its new position. If its position was not zero, the desired angle would change so the robot would tilt towards its desired position, which would cause the velocity to kick in and drive the robot back to its origin. Unfortunately this caused the robot to oscillate around its origin and never quite settled down. We fixed this by turning the integral control into another PI control– the proportional aspect settled down the oscillations which meant that the robot now oscillates less and less until it settles down to its origin position. This new code allowed John to stand up in roughly the same place for a long period of time.

We also wanted Mr. Dough to spin around in one spot. This took longer than it maybe should have because we hadn't solved the runaway issue yet. Our initial approach was to simply set one of the PWMs to be negative and the other to be positive, but we quickly realized that this didn't do quite what we expected. By changing the signs directly in the line of code that sent the PWMs to the motor, the error values for each wheel were calculated improperly because it could not take into account this sign change. Instead, we chose to modify our speed control variable by adding a "spin factor" to the left wheel speed and subtracting this same speed factor from the right wheel speed. Setting this factor as a variable allowed us to easily control it between tests to determine an appropriate value. We also realized that the PWMs that were being outputted were too high to have it spin in a stable manner, so we multiplied both PWMs by a speed factor that we used to scale down the PWM by a factor of 0.4.

Late in our process of coaching John, he expressed his passion for waltzing to us. He quickly learned how to sing and dance to Strauss's Blue Danube Waltz, a melody he learned using the built-in Pololu Orangutan Buzzer Library. His dance is simple yet elegant: a version of his classic spin but with the positional PI loop removed to allow him to drift across the dance floor.