Bridge of Doom

Sam Coleman, Ruby Eisenbud, Rishita Sarin

April 16, 2020

1 Introduction

We were tasked with driving our Neato robot across the Bridge of Doom, which is shaped as the path of the parametric equation $r(u) = 4 * [0.3960 * cos(2.65 * (u + 1.4))i - 0.99 * sin(u + 1.4)j], (u = [0, 3.2])$. In addition to making the Neato follow this path without falling off the bridge into the lava below, we also had to guarantee that the velocities of each wheel of the Neato did not exceed 2.0 m/s.

2 Methodology

Before any calculations, we redefined $u = beta * t$, where beta is a scaling factor we can use to adjust wheel velocity and time is the new variable. Using the position function $r(t)$ above (previously $r(u)$), we were able to symbolically define velocity ($v = r'(t)$), the unit tangent vector ($\hat{T}$), and the unit normal vector($\hat{N}$). We used those values to symbolically calculate left and right wheel velocity, linear speed, and angular velocity of the Neato. We then substituted in a beta value of 0.3 and an array of time values from 0 to $3.2/beta$ seconds in to our symbolic equations. This division ensures that with the maximum values, $beta * t$ remains in the original u domain from [0, 3.2]. We used the numerical versions of each to plot our theoretical predictions for the Neato (Figures 1-3 below).

After plotting the theoretical predictions, we used the provided starter code for the Neato to place it at the initial position on the bridge. We then modified this code to drive the Neato across the bridge. We used a while loop that used rostime to update the time elapsed and used that time to recalculate the left and right wheel velocities for that given time step. This meant that our wheel velocities properly changed as the Neato traversed the bridge, since we discovered from our theoretical modeling that the one wheel was initially faster than the other but at some point along the bridge it switched to the other wheel being faster. The while loop ran until the time elapsed from rostime exceeded the time required to traverse the bridge. We then sent a message to the Neato to bring both wheel velocities to zero so it did not drive off the end of the bridge.

Once the Neato successfully travelled along the Bridge of Doom, we ran the collectData_sim.m file provided by the teaching team to attain the time and experimental left and right wheel position data. This was achieved by running the collectData_sim function simultaneously with the Neato traversing the Bridge.

Using the experimental wheel position data and time from the encoder, we were able to calculate the experimental right and left wheel velocities which were the derivatives of the position data. We adjusted the experimental data for these calculations to exclude any data collected while the Neato was not moving (at the beginning and end of encoder data). Once we cropped the data we then found the new starting time and subtracted it from all times to set new starting position to $t = 0$. These wheel velocities were used to attain the overall velocity of the Neato from which we determined the experimental linear speed (magnitude of velocity), and experimental angular velocity. We then used those results to determine the heading of the Neato at each time step by integrating angular velocity over time to get the heading theta, $\theta(t + 1) = \theta(t) + \omega\_experimental(t) * (t\_experimental(t + 1) - t\_experimental(t))$. Theta allowed us to calculate the experimental general position of the Neato at each time step along the path using the equation $r(t + 1) = r(t) + v\_experimental(t) * sin(\theta(t)) * (t\_experimental(t + 1) - t\_experimental(t))$, and the $\hat{T}$ at selected points to create the experimental parametric curve the Neato travelled along to cross the Bridge of Doom.

We plotted all of these experimental values on the same plots as our theoretical results and visually compared the accuracy of our simulator and the success of our experiment.

3 Plots

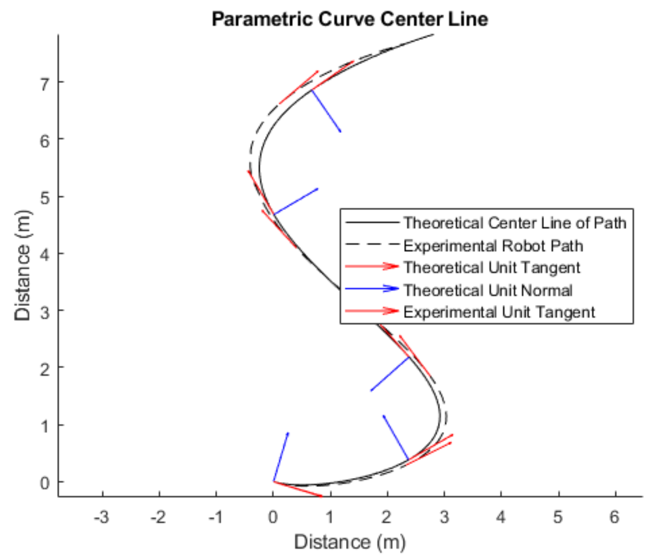

Figure 1: This plot is of the center line of the curve

Figure 1 shows the graph of the theoretical center line and experimental path of the Neato. As the graph shows, the experimental path was overall accurate to the theoretical path we expected. In our code set our initial θ to -.3. This is approximately what $\hat{T}$ is at the origin, and ensures the robot path is rotated in the same way as our theoretical path.

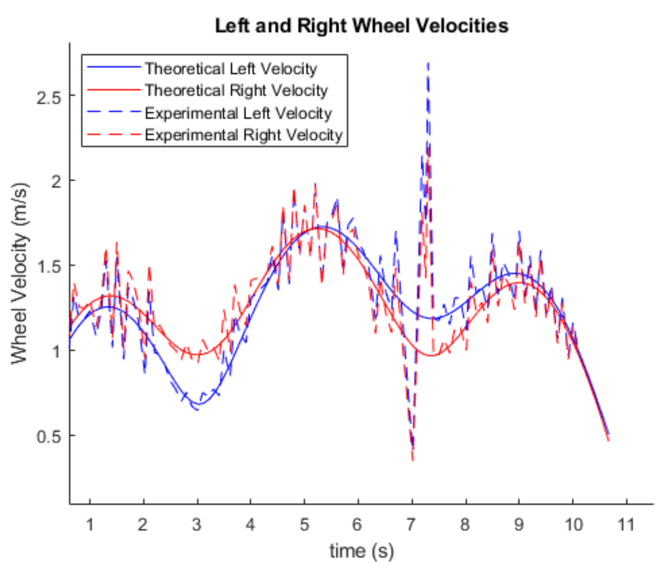

Figure 2: This plot is the left and right wheel velocities over time

Figure 2 shows the graph of the theoretical and experimental wheel velocities over time. The general trend of our experimental velocities align with the theoretical. There are some spikes in our experimental data, but this is due to how the encoder data was collected and the simulation and ROS environment. Ignoring these spikes, however, our experimental data aligns very closely with our theoretical. At about 7 seconds, the experimental right and left wheel velocities exceed 2 m/s, however, this is during a spike in the data and does not follow the general trend.

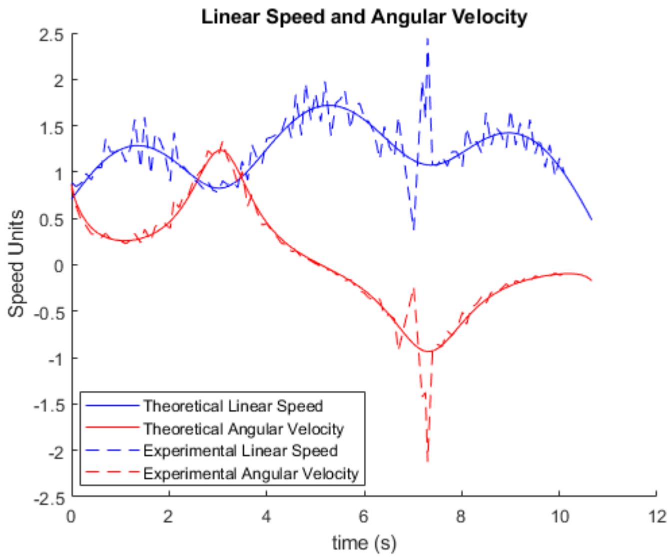

Figure 3: This plot is the linear speed and angular velocity over time

Figure 3 shows the theoretical and experimental values for linear speed and angular velocity over time. The experimental data is very close to what we expected from our theoretical data. Similar to Figure 2, there are some spikes in our data around 7 seconds. Ignoring these spikes, the experimental data is what we would expect.

4 Documentation

- GitHub Repository
- Neato Drive Code
- Calculations and Plots Code
- Cinematic YouTube Video of Neato in Action