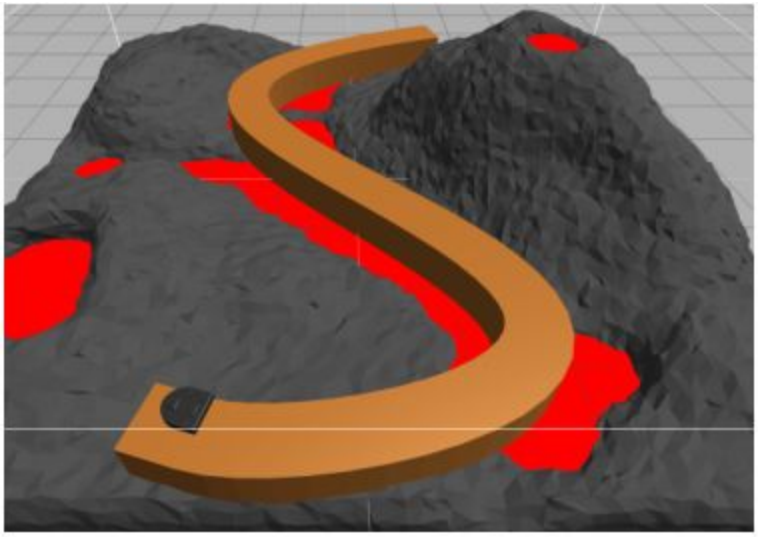
BRIDGE OF DOOM REPORT

Patrick Ogunbufunmi
QEA 2020
4/16/2020

INTRO

The goal of this challenge is to successfully program a Neato to reverse the so called “Bridge of Doom”, defined by a parametric equation, in which one has to send appropriate commands their Neato, equipped with a differential drive, in which it uses appropriate wheel velocities for each wheel to successfully traverse the complex, yet exciting bridge.



The bridge is defined by 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])
, where u=b*t
B is any beta value, which gives our Neato the appropriate velocity and angular velocity to cross the bridge. The equation is also represented in vector notation, courtesy of i hat and j hat symbols, eventually forming an S-like shape.

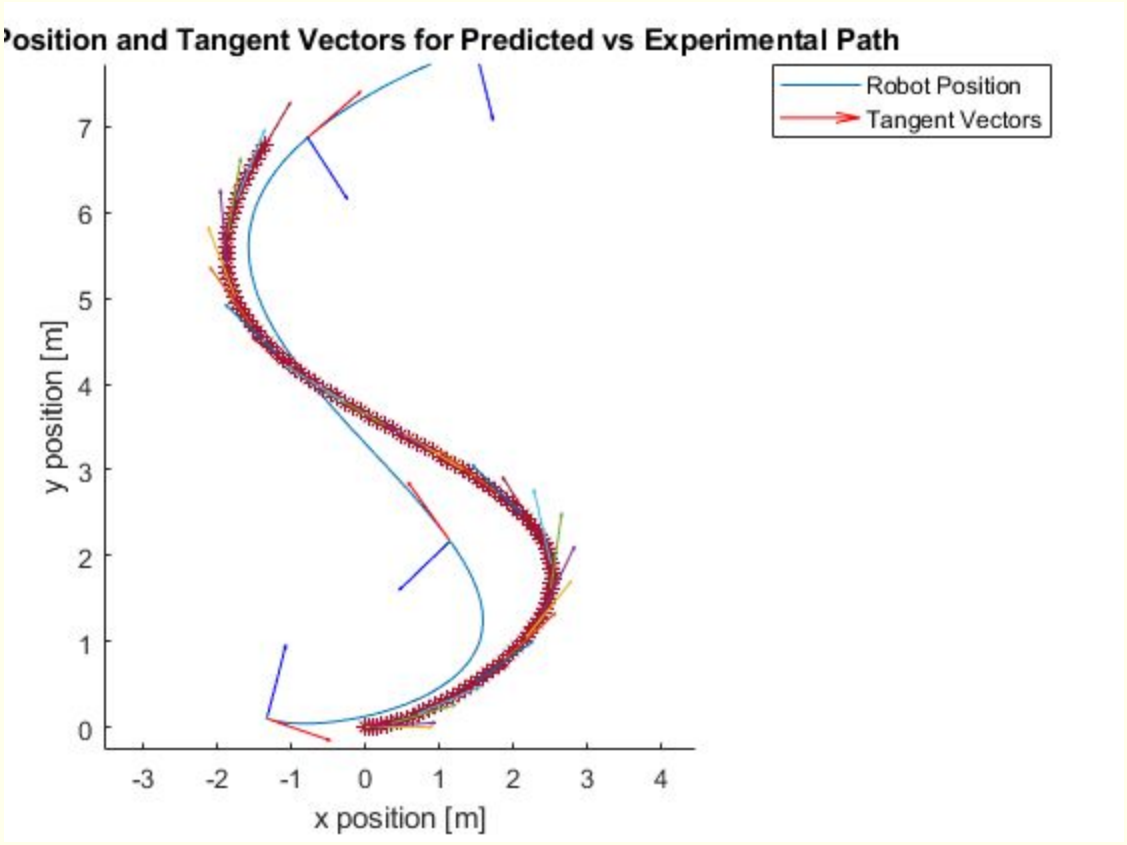
METHODOLOGY

For my methodology, I opted to implement the rostime function in Matlab, to send appropriate Vr and Vl values to my Neato to enable it to move in the desired direction, with Vr and Vl, meaning Right and Left Velocity respectively, result in our Neato moving in a specific direction. I wanted my Neato to drive at constant linear velocity, so I set u=b*t, where b is an experimental value. The equation above gives us the position in relation to time, so we had to find the velocity given at a specific time, which is now represented as r’(t). Normalizing this value gives us the linear speed, which is what we want the speed we want our robot to drive at. Now that we got the linear speed, we needed to find a way to give appropriate velocities to each wheel to produce the desired turning motions at specific times. We worked on the principal where Vr=Linear Speed-Angular Velocity*(d/2) , where d is our wheel diameter, and Vr=Linear Speed+Angular Velocity. Our angular velocity was derived from the cross product of our unit tangent vector and the derivative of that tangent vector. Plugging in each value of t into the equation will give us appropriate Vl and Vr values, enabling our Neato to successfully cross the Bridge of Doom.

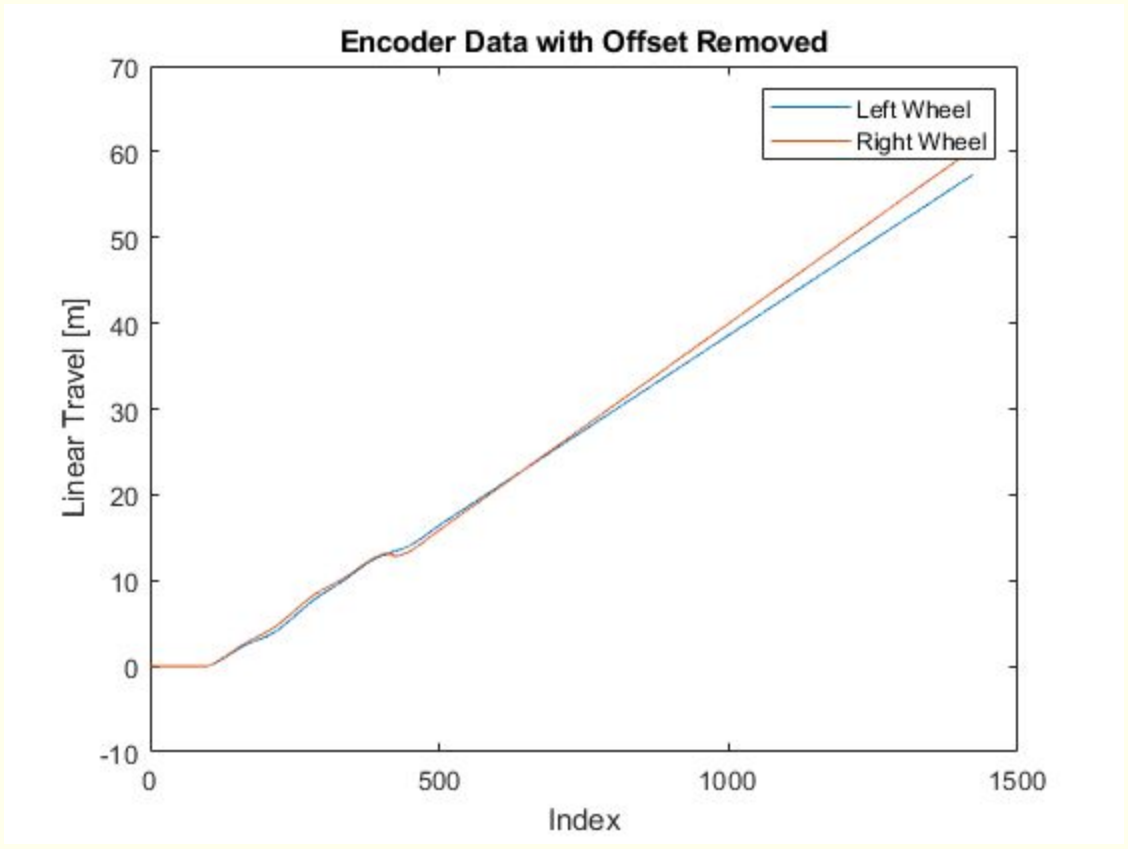
Using r’(t)=v(t)
|r’(t)|=Linear Speed
Angular Velocity=T_hat x dT_hat/dT
d=0.235m
Vl=Linear Speed-Angular Velocity*(d/2)
Vr=Linear Speed+Angular Velocity*(d/2)
Where u=b*t

We arrive at our respective Vl and Vr values which help it cross the bridge
Using rostime, we send published VL and Vr values depending on the elapsed time, calculated as current - start, where:
current=rostime(‘now’) inside of our loop
start==rostime(‘now’) outside of our loop

, which sends the appropriate values to our Neatos.



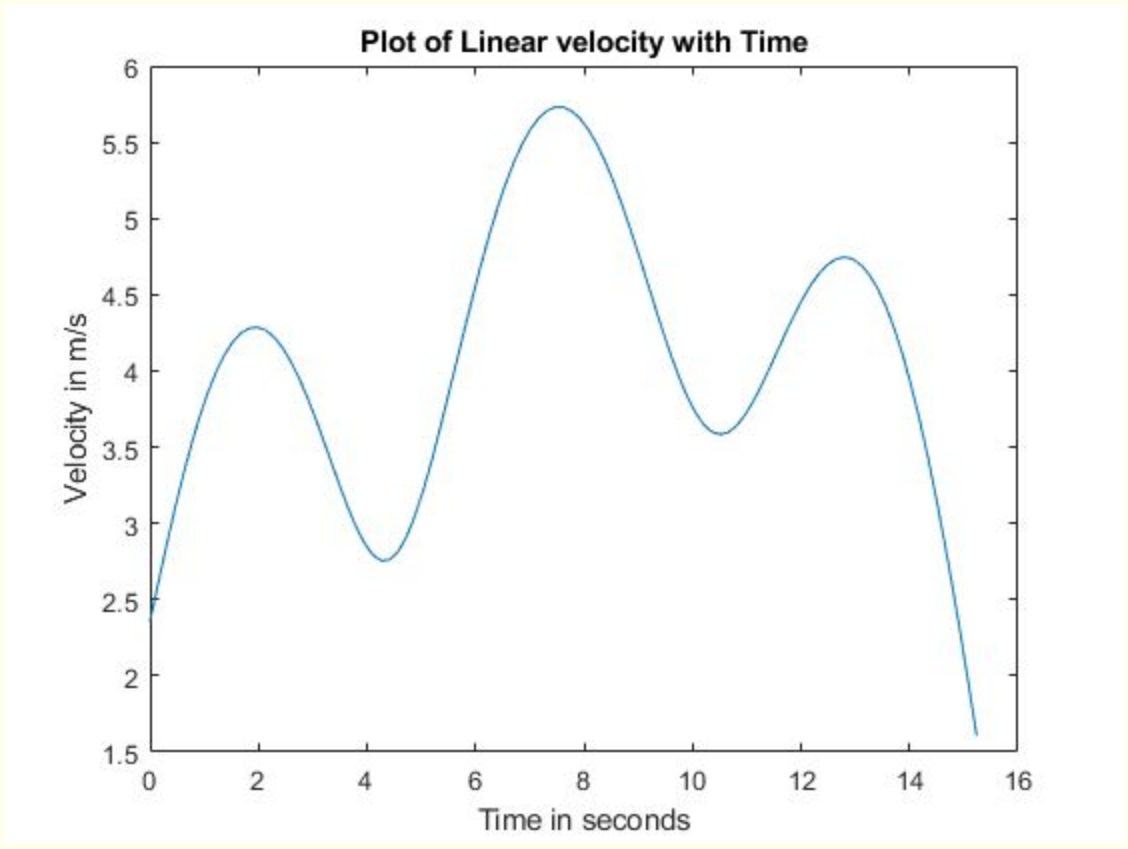
Up there is the graph of our predicted vs actual path for our Neato to drive. Our Neato came pretty close in terms of following the centreline, and using the tangent vectors, it shows it was headed in the right directions.



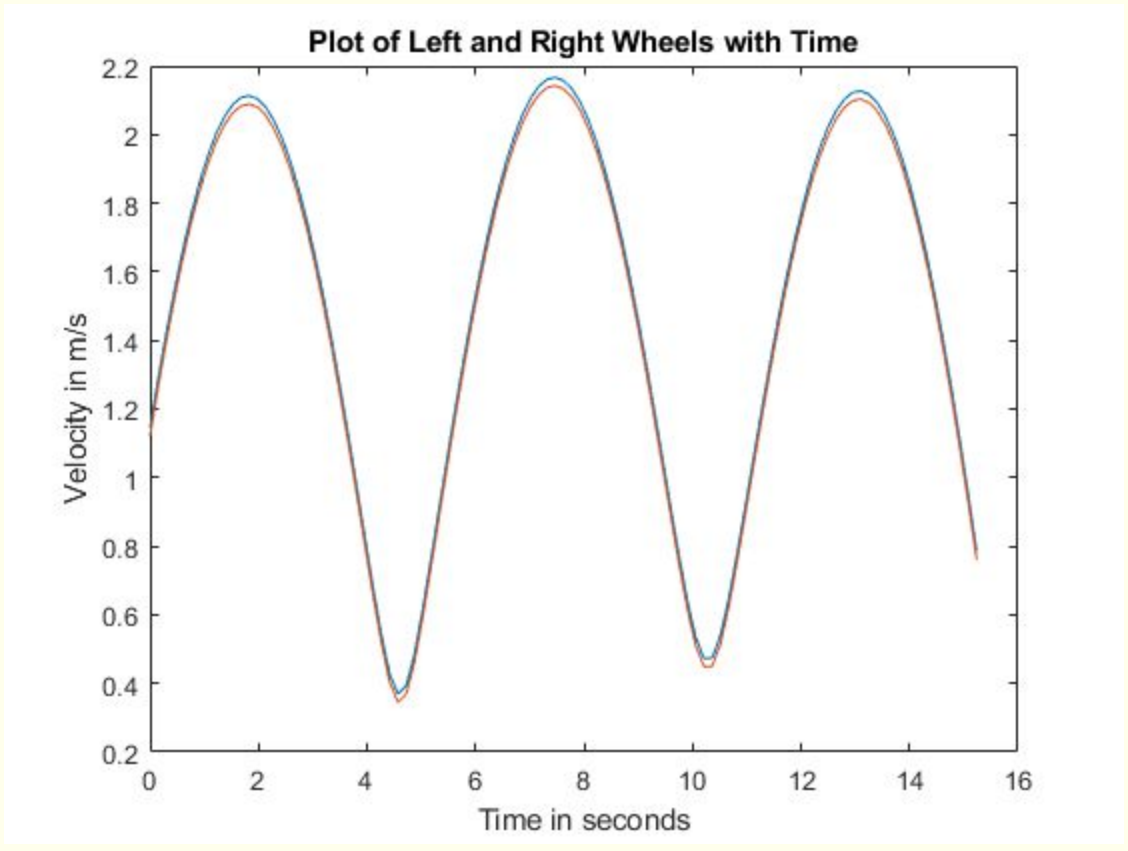
Here is the distance our wheels travelled, and as expected, it seems pretty even as they both covered approximately equal amounts of distance, with our left and right wheel highlighted.

This is a theoretical plot of Linear Velocity with time. It should under ideal conditions be constant in the experimental setting, as shown above, when VL and Vr both have equal rates of distance

with time.



This is also a theoretical plot of Left and Right Wheel velocity with time. It has many oscillations as shown above, as the time step in that case is more than the actual Neato Experiment.



Youtube Link to Neato Journey: https://youtu.be/ZNhi24UFHFU