Group Lab 2 Team Jury

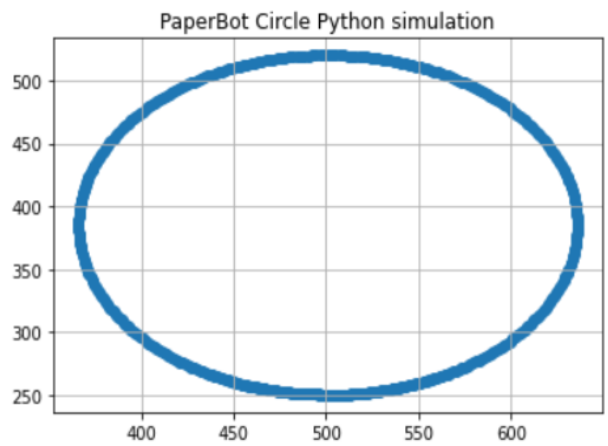
Giacomo Fratus, Chiraag Hebbar, Hieu Nguyen, Jake Smith, Jeremy Wong, Seong Hong, Sidharth Subbarao

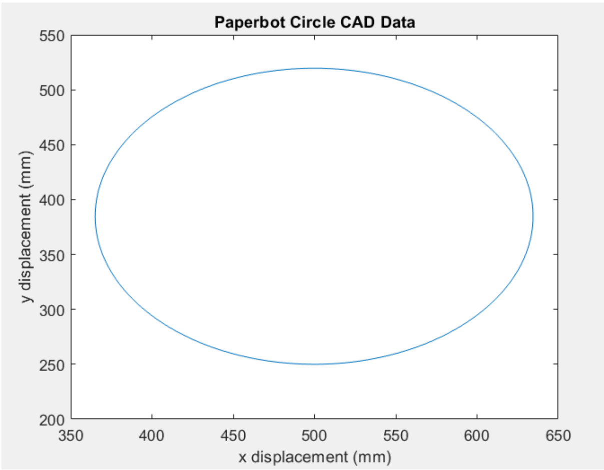
In this lab, we conducted a motion analysis of the Paperbot and Segway vehicles using Solidworks simulation tools and Python. Giving a piecewise angular velocity input to each wheel, we ran 8 paths on each robot, for a total of 16 simulations. Below is a comparison of the Python and CAS simulation results.

Python(EE) Simulation (left) and CAD simulation (right), starting point = (x,y,θ)

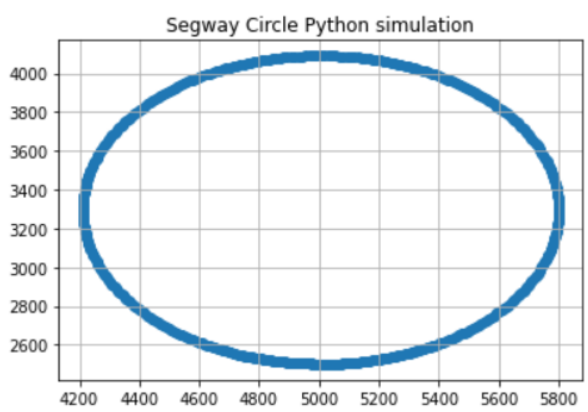
1. Circle

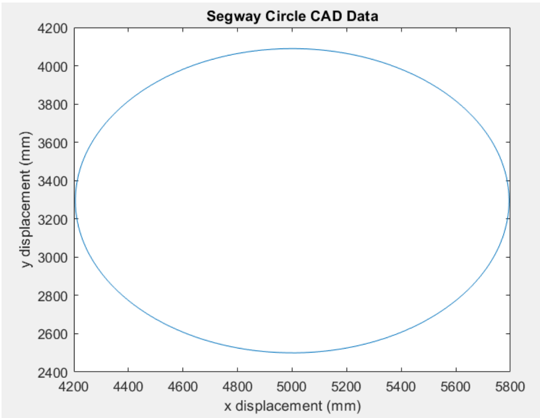
Paperbot, starting point = (500,250,0)





Segway, starting point = (5000,2500,0)



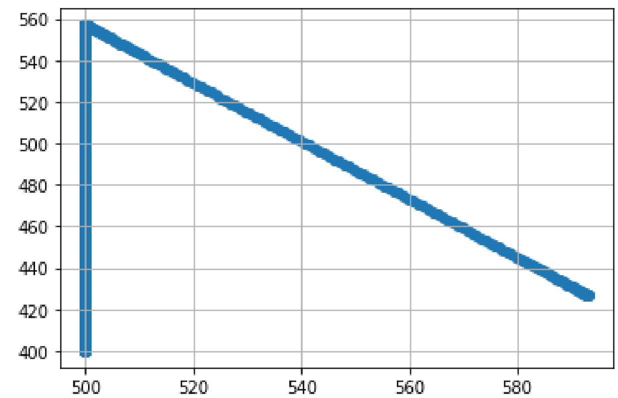


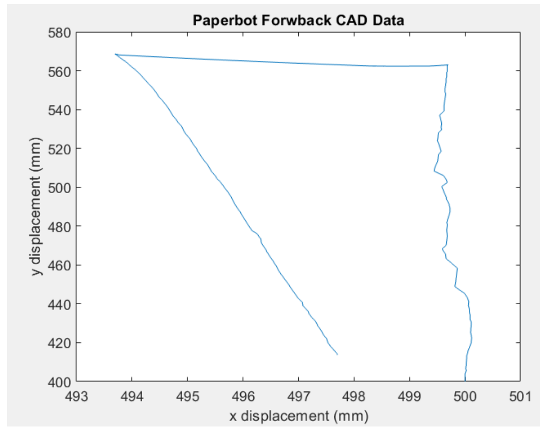
Group Lab 2 Team Jury

Giacomo Fratus, Chiraag Hebbar, Hieu Nguyen, Jake Smith, Jeremy Wong, Seong Hong, Sidharth Subbarao

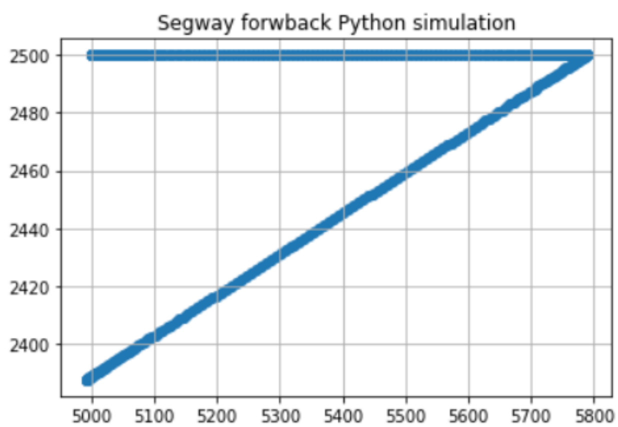
2. Straight line, turn around, come back to initial spot

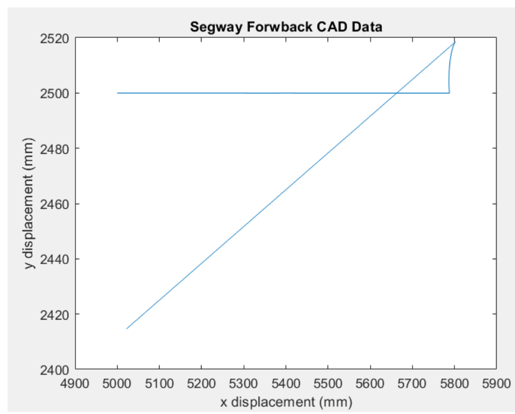
Paperbot, starting point = (500,400,0)





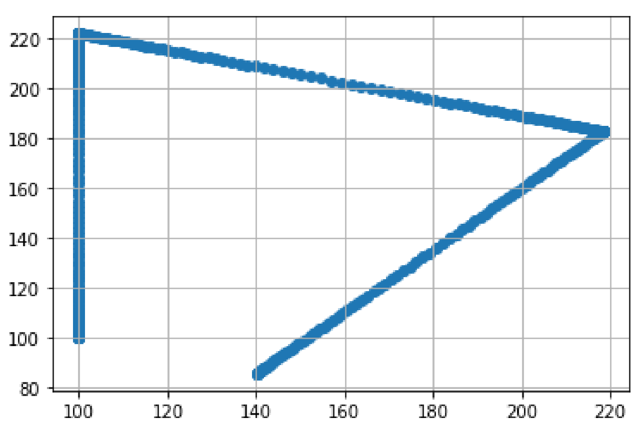
Segway, starting point = (5000,2500,0)

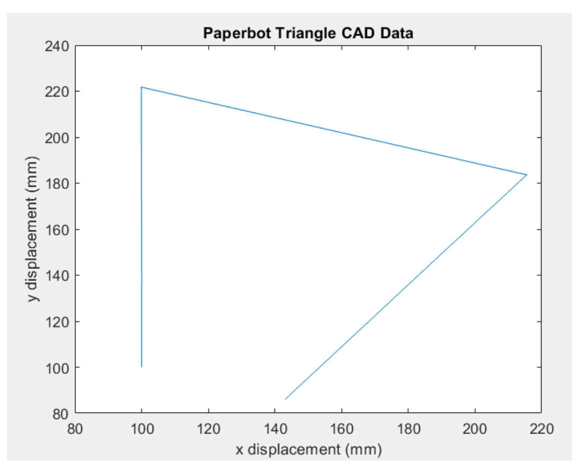




3. Triangle

Paperbot, starting point = (100,100,90)

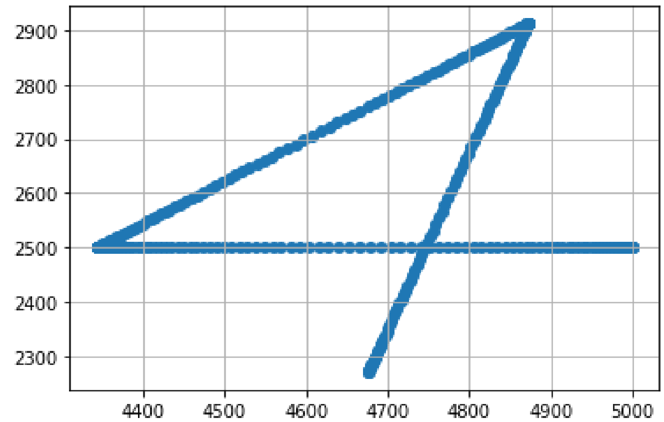


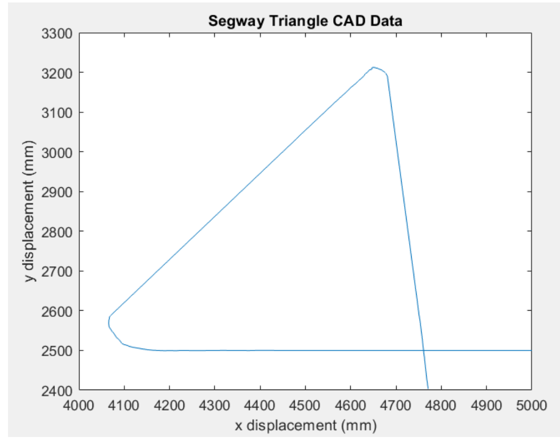


Group Lab 2 Team Jury

Giacomo Fratus, Chiraag Hebbar, Hieu Nguyen, Jake Smith, Jeremy Wong, Seong Hong, Sidharth Subbarao

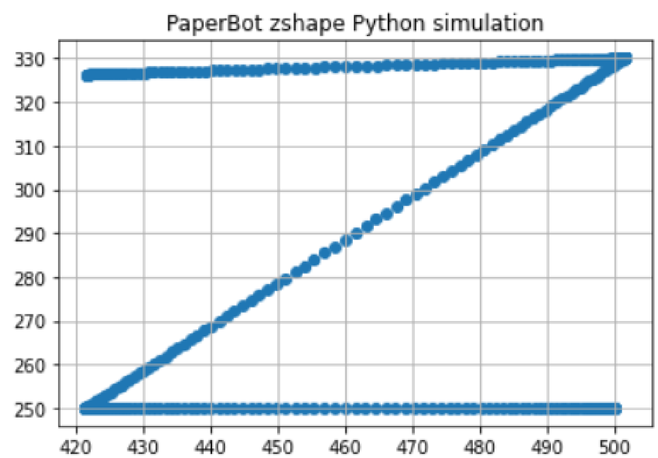
Segway, starting point = (5000,2500,0)

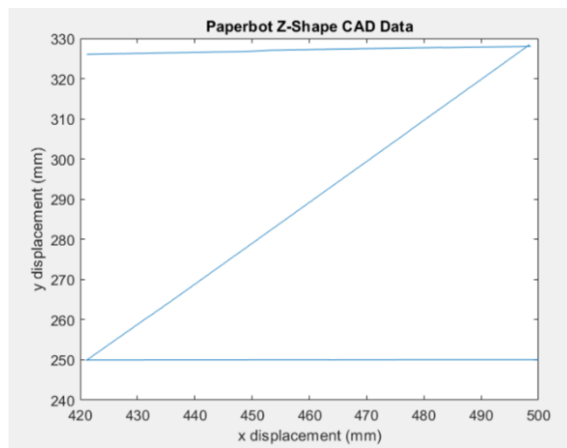




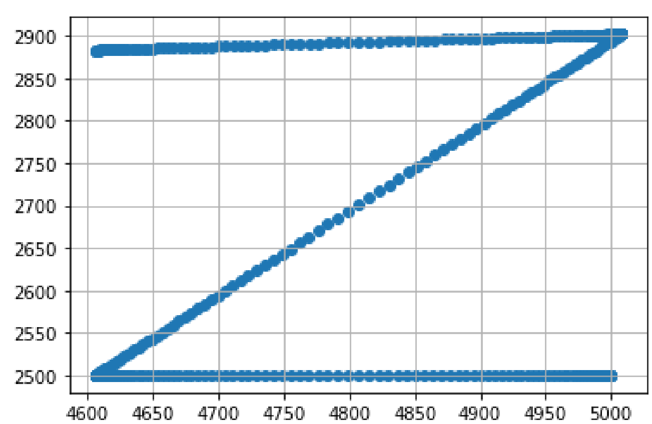
4. Z-Shape

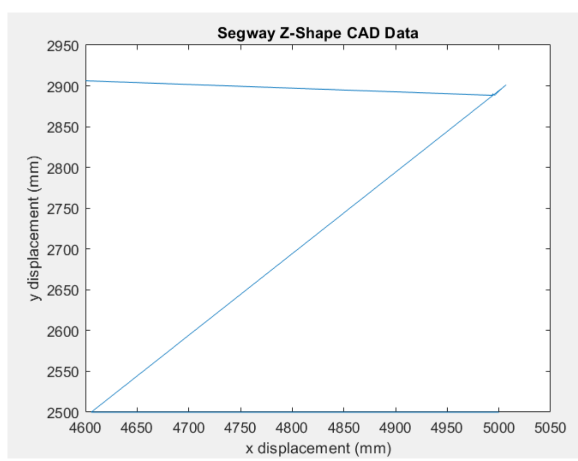
Paperbot, starting point = (500,250,0)





Segway, starting point = (5000,2500,0)



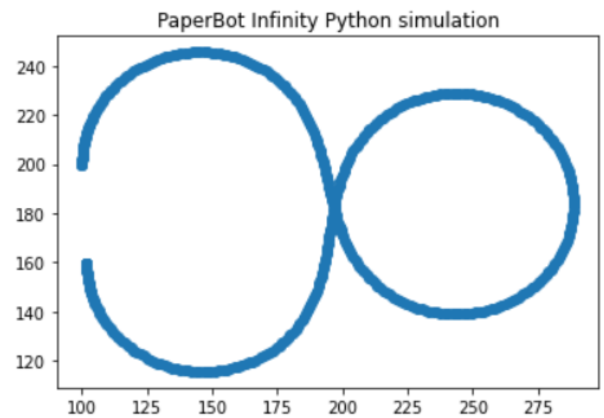


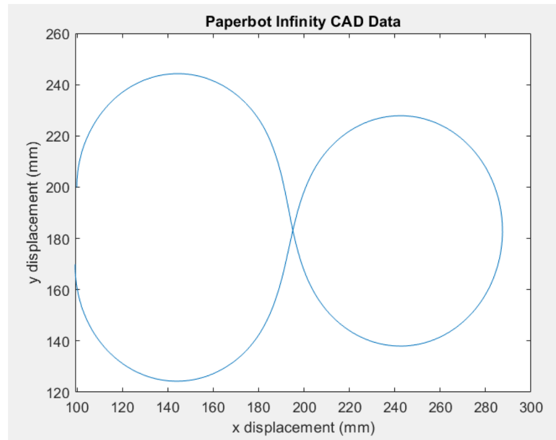
Group Lab 2 Team Jury

Giacomo Fratus, Chiraag Hebbar, Hieu Nguyen, Jake Smith, Jeremy Wong, Seong Hong, Sidharth Subbarao

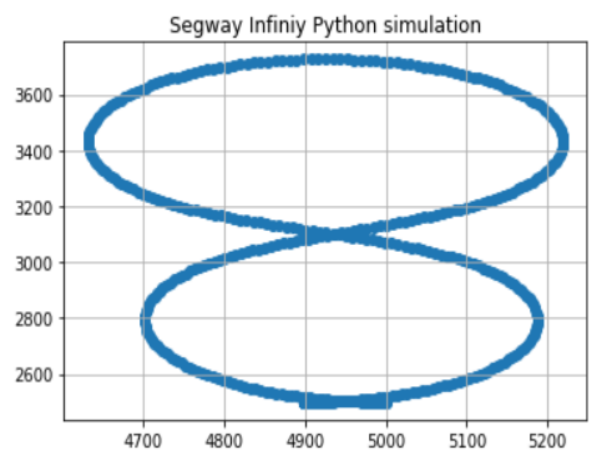
5. Infinity sign

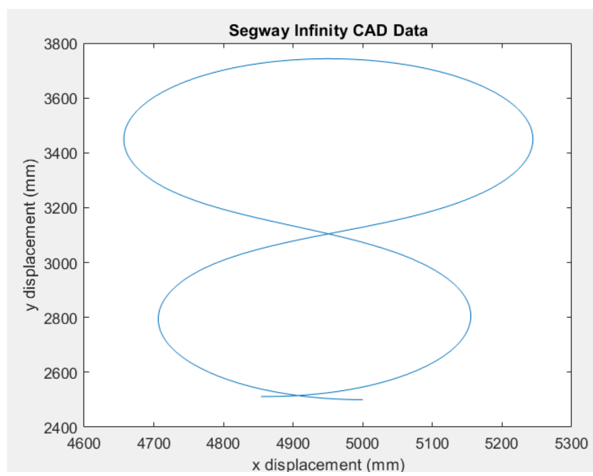
Paperbot, starting point = (100,200,90)





Segway, starting point = (5000,2500,180)



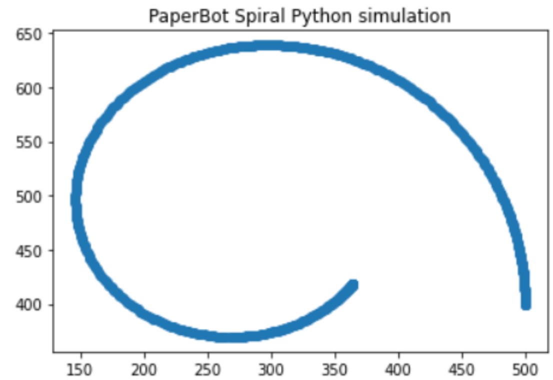


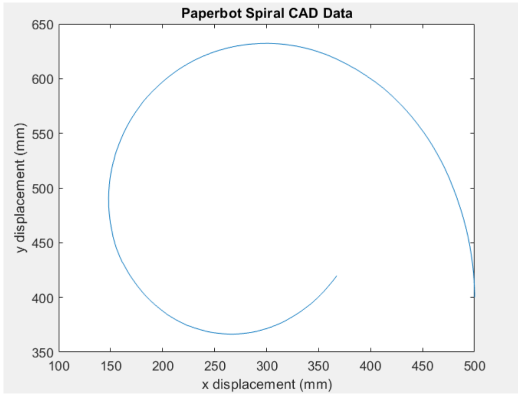
6. Spiral

Paperbot, starting point = (500 ,400,90)

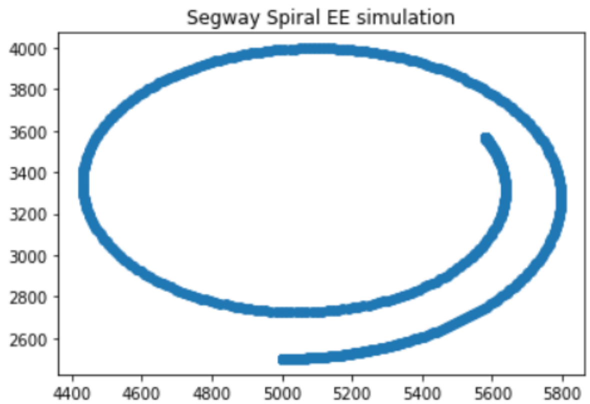
Group Lab 2 Team Jury

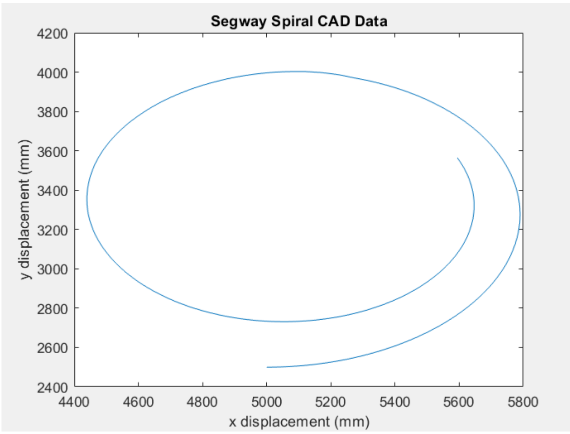
Giacomo Fratus, Chiraag Hebbar, Hieu Nguyen, Jake Smith, Jeremy Wong, Seong Hong, Sidharth Subbarao





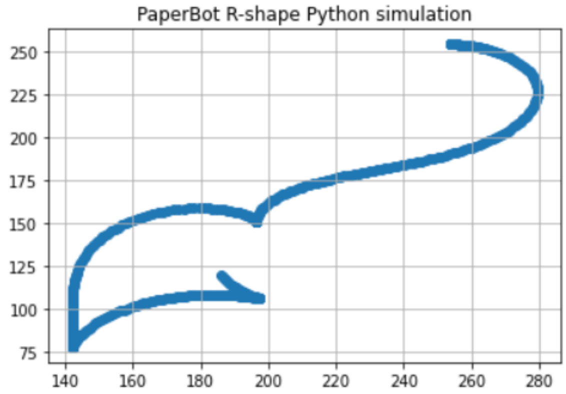
Segway, starting point = (5000,2500,0)

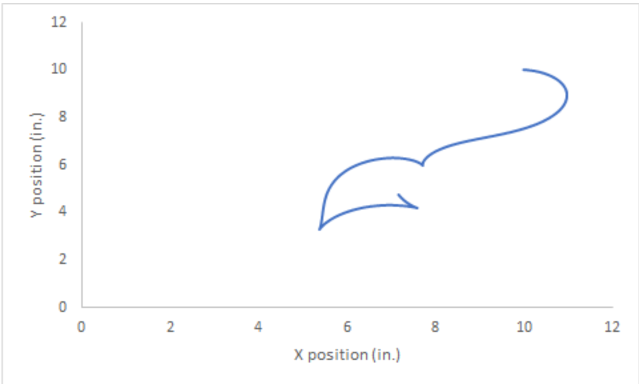




7. Randomized

Paperbot, starting point = (10 in., 10 in.,90)





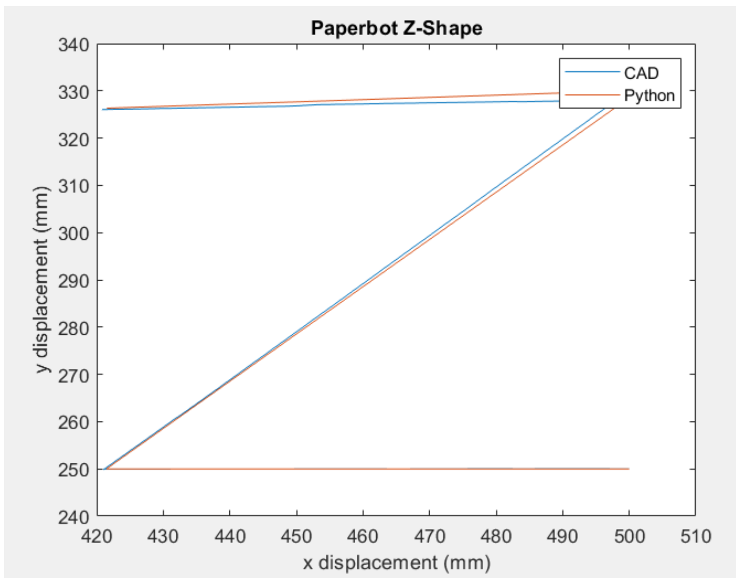
4.1 Statistical Analysis between Analytical and Numerical Simulation

Group Lab 2 Team Jury

Giacomo Fratus, Chiraag Hebbar, Hieu Nguyen, Jake Smith, Jeremy Wong, Seong Hong, Sidharth Subbarao

For the Paperbot, the two simulations yield very similar trajectories, with an error of the 5mm on a square 1m floor. For the Segway, error is within 0.05m on a square 10m floor, with higher errors during certain trajectories. Trajectories on the Segway like an in-place turn tended to have higher error compared to the analytical simulation.

The results match well with the mathematical formulation on Paperbot because the model of the vehicle is simple to relate geometrically and model in CAD. However, the Segway model has more complexity in modeling. This is due to the higher center of mass, a more complex omni wheel-floor contact, and a specified rubber-on-rubber contacts. This made conditions like tipping and slipping more likely in the Segway CAD simulations.



We could improve errors between analytical and numerical results by implementing input variation and slip conditions from friction coefficients into the mathematical model. We could also feed inputs to the simulation from the input angular velocities of the wheels. Currently our simulation takes the user defined data points and linearly interpolates between them. Increased error may be introduced if SolidWorks and our simulation interpolate differently. Another way we could improve would be by not using uniform time stepping in the simulation to match the behavior of SolidWorks Motion Analysis more closely.

4.2 Statistical Analysis between Segway and Paperbot Simulation

The Segway analysis had more error than Paperbot compared to the numerical simulation. In general, the Paperbot analytical solution converged with the Solidworks simulation. The Segway, however, tended to diverge when the robot had to turn in place, as seen in the Segway's Forward-Backward and Triangle trajectories. These corners are rounded out rather than centered around the wheelbase center of the robot, which is likely due to inertia differences between the Paperbot and Segway.

Group Lab 2 Team Jury

Giacomo Fratus, Chiraag Hebbar, Hieu Nguyen, Jake Smith, Jeremy Wong, Seong Hong, Sidharth Subbarao

Our mathematical formulation from lab 1 is based on relating the wheel diameter contact with the ground with distance traveled in the floor frame of reference. Since our inputs were such that our wheels maintained contact with the ground, all movements were accounted for in both models and the resulting trajectories matched well. Higher speeds and sharper turns would cause the two results to diverge, more so for the Segway simulation.

4.3 System Errors and Potential Simulation Scenario (For this section, you don't necessary to show additional simulations. You may do that in the future labs.)

We did not include any noise in the analytical simulation, as we focused on inputs that avoided slip or tip conditions to follow the mathematical formulation more closely. High speeds and sharp turns would show more drastic differences in trajectory due to possible slip conditions, as coordinate system movement would not correlate with wheel inputs and would be lost in the mathematical model.

Some noises and disturbances that cannot be described well are dirt or debris on the track, poorly oiled gears and wheels, faulty wire connections, or situations where natural disasters or human interference occurs. However, these situations can be approximated with a Gaussian distribution based on the magnitude of the interaction. We could include noise in the wheel inputs by adding variation between the requested input and the signal on the wheel motor. We could also model wind as an external force that has a probability to its magnitude and chance of occurring.

4.4 Design Process

Through this Joint Lab 2, we are very confident to start prototyping, developing algorithms, and simulating a system. These simulations are an intuitive way to see how the output of a system differs from the expected output; we can iteratively improve our models effectively in the design process rather than attempting to work primarily in theory and mathematical formulation.

Code:

https://github.com/giacomofratus/Jury