MTE 351 Final Project Report

UNIVERSITY OF
Waterloo

Department of Mechanical and Mechatronics Engineering

July 30, 2025

Andor Siegers - 20990622
Henrique Rodrigues - 21037291
Ian Jinzo Macpherson - 21009699
Josh Morcombe - 20937588
Sean Aitken - 21006546

Task 1: Navigation & DC Motor Models

Initial Assumptions

- The disturbance torque of the motors is assumed to be negligible
- The dampening effect of the brushes within the DC motor is negligible
- The wheels are massless and have negligible moment of inertia
- No slip is present in the wheels

Method

To create the navigation motor drive system, the first step was turning the input voltage to the armature-controlled DC motors into an output torque at the wheels. Using the governing equations for DC motors and considering that back emf will be generated depending on the angular velocity of the motor axle, a Laplace domain equation that relates input voltage to output torque can be created. The output torque of the motor and back emf is scaled by a gear ratio to relate the torque and angular velocity output of the wheel to that of the motor. The Laplace equation can be seen below.

$$\frac{(V_a - k_b N\omega)}{L_a s + R_a} k_t N = \tau_w \tag{1.1}$$

Where:

V_a - Voltage across the motor armature

k_b - Back emf constant

k_t - Motor torque constant

N - Gear ratio

L_a - Armature inductance

R_a - Armature Resistance

ω - Angular velocity of the output wheel

τ_w - Output torque of the drive wheels

To calculate how the output torque of the wheels affects the movement of the rest of the robot, the mass of the robot and its second moment of inertia need to be taken into account. The force of each wheel acting backwards on the ground was calculated simply by dividing the wheel torque by the radius of the wheels, which was assumed to be 5 cm. These forces were then used to calculate the linear and angular acceleration of the robot, assuming that a positive input voltage would move both motors in the forward direction. For angular acceleration, it was decided that the motors would be in line with the center of gravity of the robot to simplify calculations and analysis.

$$a = \frac{T_L + T_R}{mr} \tag{1.2}$$

$$\alpha = \frac{(T_R - T_L)\, L}{I\, r} \tag{1.3}$$

Where:

T_L - Left motor torque

T_R - Right motor torque

r - Radius of drive wheels-

L - Distance from the center of the robot to the wheels

a - Linear acceleration of the robot

α - Angular acceleration of the robot

m - Mass of the robot and load

I - Second moment of inertia of the robot and load

The mass of the robot was assumed to be 20 kg plus the weight of the load, and the second moment of inertia calculation used the mass and the formula for the inertia of a cylinder to calculate a constant value that the simulation can use. It should be noted that the load was treated as evenly distributed along the top surface of the cylinder to simplify calculations. The output linear and angular acceleration can be integrated to find the robot's overall linear and angular velocity. The global angle and jerk can then be calculated through integration and chain differentiation, respectively.

Utilizing the angular and linear velocities, the velocity of each side of the robot where the wheels sat could then be calculated. To find the angular velocities of the left and right wheels, the following equation was used:

$$\omega_w = \frac{V_u \pm \omega_u L}{r} \tag{1.4}$$

Where:

V_u - Velocity of the center of gravity

ω_u - Angular velocity from the center of gravity

ω_w - Output wheel angular velocity

r - Radius of the output wheel

L - Distance from the center of the robot to the wheels

It should be noted that the denominator should feature an additional sign to get right angular velocity, and a minus to get the left angular velocity. The output angular velocity of the wheels are connected through a feedback line on the Simulink model to the output torque equation (equation 1.1).

Finally, the robot's position in relation to the global origin can be calculated by taking the velocity of the centroid of the robot and multiplying it by the sine and cosine of the angle to get the y and x velocities respectively. These can then be integrated and plotted against each other to yield the robot's x and y displacements.

This entire system of equations described was created in simulink as a block diagram, and is viewable in the file "*Motor_and_Nav_Model.slx*".

Task 2: Suspension Subsystems

Initial Assumptions

- The wheel does not slip and is in contact with the ground at all times
- The full weight and force of the robot is divided equally between its two powered wheels
- Caster wheels do not affect the tray's position
- Wheel inertia is neglected

Method

Firstly, the robot's suspension subsystem was simplified to a single spring-damper system. This allows for the tray's height to be treated as an output of a function of the floor's height.

Secondly, free body diagrams of the system's mass, spring, and damper were drawn and the equations of motion were derived. Then the equations were transformed into the Laplace domain to find the system's transfer function. This allows for modelling of the system in Simulink. The maximum mass of the robot was found to be 30kg, and since each wheel has its own suspension system, each wheel only carries half of the total mass, so the mass in each system is set to 15kg.

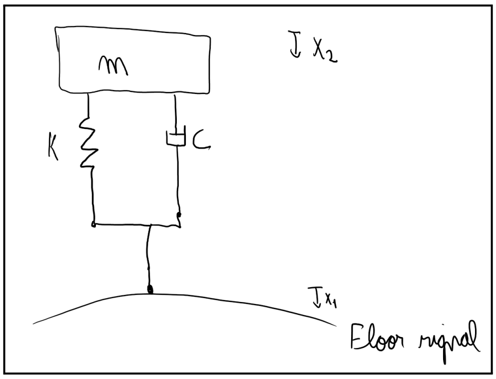

Figure 1: Suspension system diagram

The transfer function of the system, representing the tray's position over the floor's height, was found to be:

$$\frac{X_2}{X_1} = \frac{cs + k}{ms^2+cs+k} \tag{2.1}$$

Task 3: Floor Signal

To build the floor signal, the haversine equation was modeled using MATLAB and inserted into an array of zeros that represents flat ground. The haversine equation was repeated at regular intervals along the entire active distance to simulate the grouting.

The MATLAB script that generates the floor signal is titled "*floorplan.m*", and has been submitted alongside this report. The same script is used to generate the floor signal passed to the suspension system in Task 5.

The plot below shows the floor's depth over time, assuming the robot is moving at 3 m/s for 5 metres, as specified.

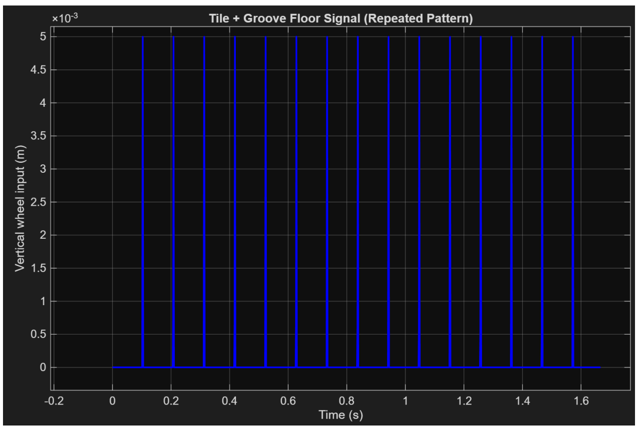

Figure 2: Graph of the floor signal (with height decreasing in the negative y-axis)

Task 4: Door Threshold Signal

The door threshold signal takes distance traveled as the input and contains a single pulse that occurs 2m into the 5m section. This signal was modeled using a haversine profile, as shown below:

$$y(z) = H\,sin^2(\frac{\pi z}{L}) \tag{4.1}$$

Where:

H - Height of the door threshold (0.03m) L - Length of door threshold (0.05m)

z - Distance traversed from the start of the door threshold

The signal was configured using comparator logic to create a single pulse at the correct distance for the specified threshold length. Figure 4 below shows the block diagram and the system response when a ramp function with a constant slope of 1 is used as a test input.

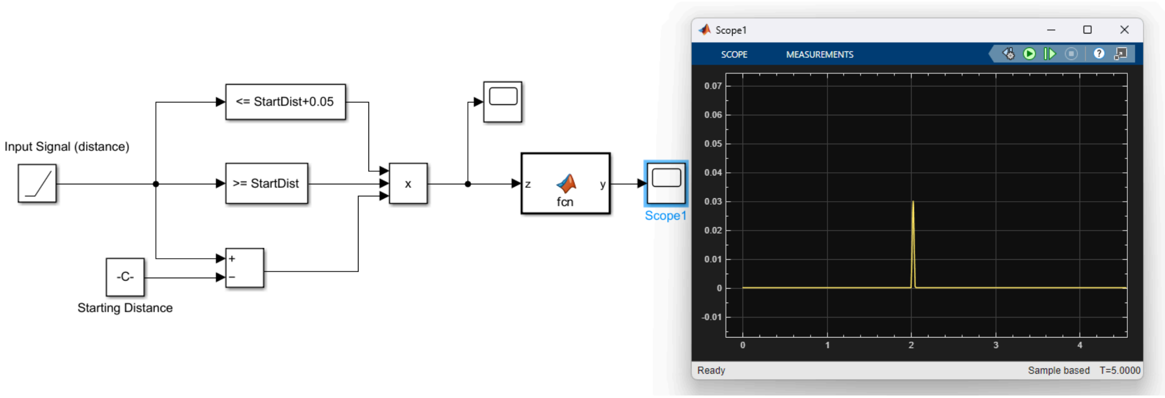

Figure 3: Door Threshold Block Diagram and Response

Task 5: Analysis

To iterate through values for the spring stiffness (*k*) and damping coefficient (*c*), the floor signal created in *Task 3* was passed to Simulink along with the Door Threshold signal found in *Task 4*. Various values for both *k* and *c* were tested by setting them as constants in the system, and observing the system output (tray movement). Figures 4, 5, 6, and 7 show the iterations done while tuning the *k* and *c* values. Note that each iteration assumes the robot is moving linearly at 3 m/s.

The Simulink file used to simulate the suspension system is included alongside this report under the name "*suspension_subsystem.slx*". To run this simulation correctly, the "*floorplan.m*" file must first be run to initialize the variables used by the simulation in the MATLAB workspace.

Initially, the spring constant and damping coefficient were both set to 1000, and mass was set to 15 kg, which represents the maximum weight of the robot with maximum load (30 kg), divided between both wheels. This resulted in the following response:

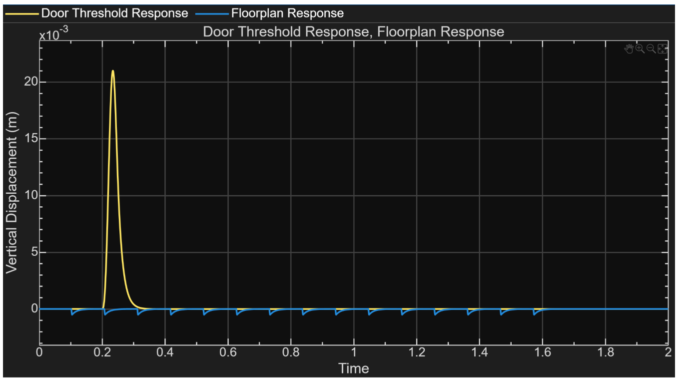

Figure 4: Suspension response when k = 1000 N/m and c = 1000 Ns/m with m = 15 kg

As seen in Figure 4, the maximum displacement was around 2.1 cm, which is above the threshold. The following graph has the same constant values for K and C; however, mass was changed to 10 kg, which is the mass of the robot with no load (20 kg) divided between both wheels. This is done to determine the worst-case displacement.

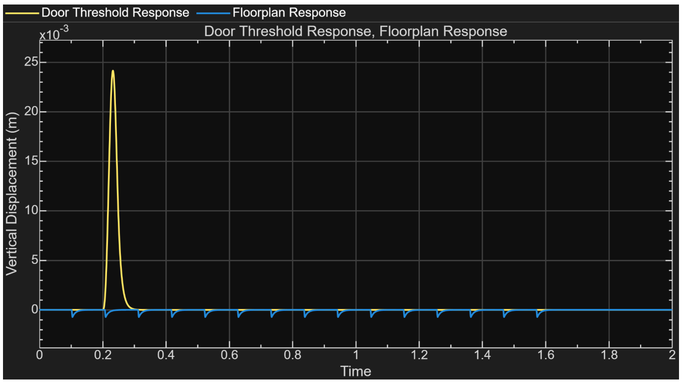

Figure 5: Suspension response when k = 1000 N/m and c = 1000 Ns/m with mass = 10 kg

In Figure 5, the response reached around 2.4 cm, which is higher than the maximum response observed at full load. Therefore, our worst-case scenario for vertical displacement is when the tray is empty, and all the following tests will be done with a load of 10 kg per wheel. The response in Figure 5 also shows that the chosen *c* and *k* values cause the system to be overdamped. The response shows that *c* and *k* are too high, causing a tray displacement well above 0.5 cm, which is not acceptable. *c* and *k* were therefore decreased tenfold for the following test.

With c set to 100 Ns/m and k set to 100 N/m, the following response was obtained:

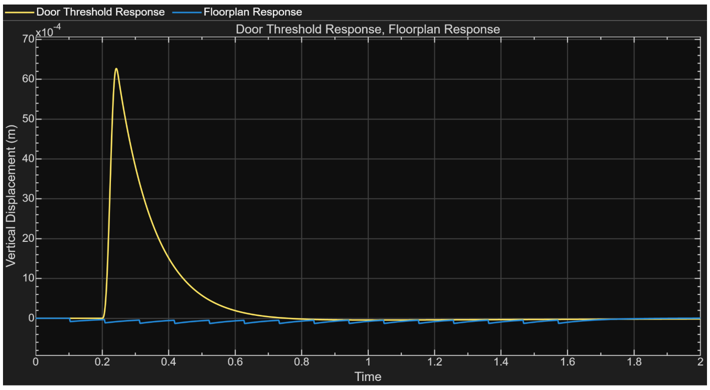

Figure 6: Suspension response when k = 100 N/m and c = 100 Ns/m

For these values, the system is still overdamped and displacing over 0.5 cm vertically, so *k* was changed to 75 N/m, and *c* was changed to 50 Ns/m for the next test. The response due to these new values is shown below:

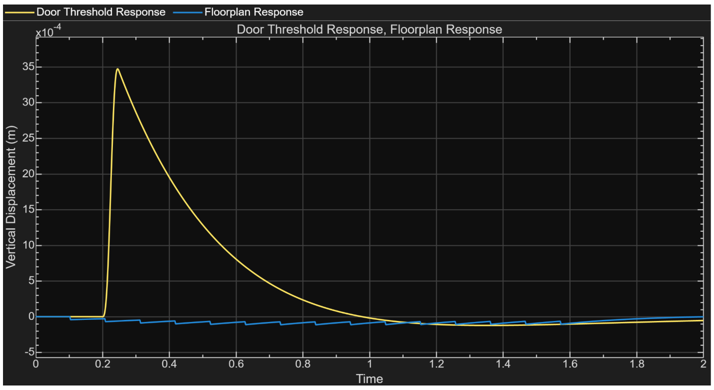

Figure 7: Suspension response when k = 75 N/m and c = 50 Ns/m

This response shows a maximum tray displacement of less than 0.5 cm, fulfilling the requirement while the damper successfully stops the response from oscillating indefinitely.

In conclusion, the final chosen values were **spring stiffness (*k*) = 75 N/m** and **damping coefficient (*c*) = 50 Ns/m.**

The chosen damping coefficient value is interestingly very similar to the critical damping of the system, which was calculated to be 54.8 Ns/m using the following formula:

$$c_{critical} = 2\sqrt{mk} \tag{5.1}$$

The analysis of the suspension system was done assuming a speed of 3 m/s, however, this is decidedly too fast for a restaurant setting. For other tasks the maximum speed of the robot was set to 2 m/s however this decrease in speed should not negatively affect the response of the suspension system, and therefore this analysis is still valid.

The height of the door threshold is also an essential parameter for the suspension response, however, as door thresholds are usually made to follow a standard, it is assumed that every door threshold in the working environment of the robot will have a uniform height.

Task 6: Simulation of Vibration

Vertical acceleration was found by taking the second derivative of the vertical position output by the suspension simulation. The maximum instantaneous acceleration was found to be at most 24 m/s^2, however this acceleration is only reached for a few milliseconds. The plot below shows the instantaneous vertical acceleration felt by the robot over time as it moves along the floor with a speed of 3 m/s for 5 metres.

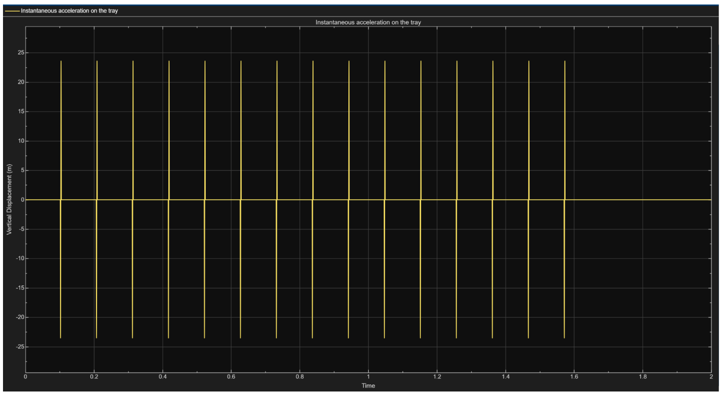

Figure 8: Instantaneous vertical acceleration experienced by the tray

The maximum instantaneous force experienced by the tray was found by multiplying the maximum acceleration by the mass of the robot and dividing by the number of wheels. The robot weighs 20 kg when carrying no load, which is the case where the highest acceleration is found, and this mass is divided between both wheels, so each suspension system supports 10 kg. The maximum force experienced by the tray was therefore calculated to be 240 N. The plot below shows the vertical force by the tray due to the grouting.

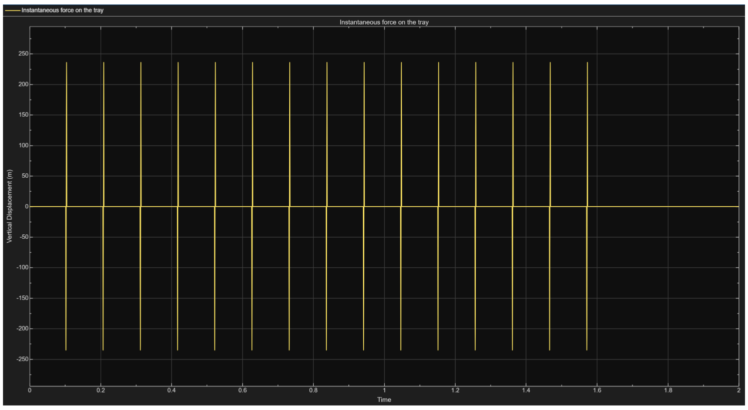

Figure 9: Instantaneous vertical force seen by the tray

Task 8: Input Voltage

Initially, a formula relating output velocity to input voltage was derived. This allows for an input speed signal to be converted to a voltage through a gain. Assuming negligible wheel inertia, brush damping, and disturbance torque, the following formulas for output voltage were derived:

$$V_a = \frac{k}{Nr}v = \frac{kL}{Nr}\omega \tag{8.1}$$

Where:

V_a - voltage across the motor armature
k - motor torque constant
N - gear ratio
r - wheel radius
L - Distance between the wheel and the center of the robot
v - linear velocity of the robot
ω - Angular velocity of the robot

With a gear ratio of one, these formulae were successful in creating a voltage signal that resulted in the desired speed output. However, when the gear ratio was changed, this formula produced undesirable outputs. The reason for this inconsistency was investigated, but no matter what was changed, a consistent output could not be achieved using this method. Therefore, this approach was abandoned, as a gear ratio was required to convert the high torque of the chosen motor to a higher output velocity.

To find the final voltage signal, a trapezoidal input was created. A maximum slope of 20 V/s and 40 V/s was chosen for linear and angular motion profiles, respectively. This was empirically found to result in linear and angular accelerations of **0.9738 m/s^2** and **7.727 rad/s^2**, respectively. A maximum voltage of **40 V** was chosen, which, when sustained, results in linear and angular speeds of **1.9418 m/s** and **7.767 rad/s**, respectively. These values are all within a reasonable range for a food service robot and are well within the operational range of the chosen motors. Once the acceleration ramps were defined, the slew times were calculated based using the following formula:

$$t_{slew} = \frac{D - 2(v_{max}^2/(2a))}{v_{max}} \quad (8.1)$$

Where:

D - distance or angle to traverse
a - linear or angular acceleration
v_{max} - max linear or angular speed

In cases where the distance is too short for a slew phase, we calculate acceleration time as:

$$t_a = \sqrt{\frac{D}{a}}$$

Otherwise, acceleration time is calculated as:

$$t_a = \frac{a}{v_{max}}$$

Combining these formulas, trapezoidal voltage signals can be constructed. The final input voltages for each motor are presented below.

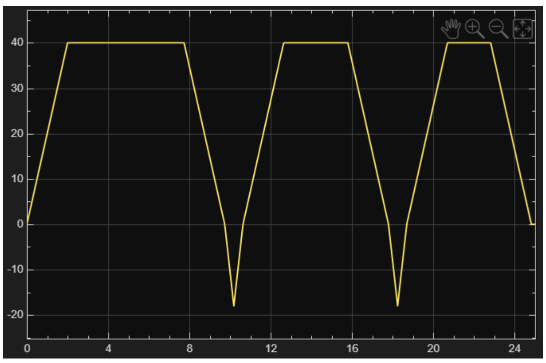

Figure 10: Right Motor Voltage Signal

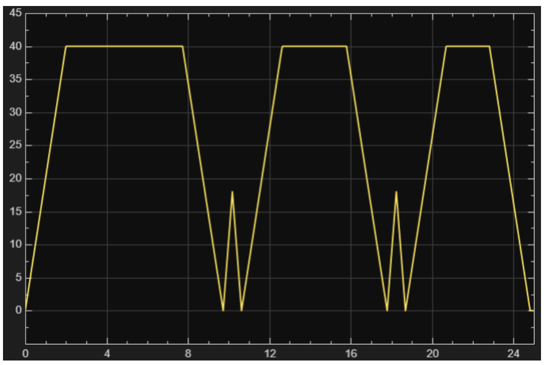

Figure 11: Left Motor Voltage Signal

Task 9: Robot Design and Analysis

The selected motor is the SM0602GSA-KCF-NNV Brushless DC Servo Motor. It has a rated power of 400 W, a rated speed of 3000 rpm, a rated torque of 1.27 Nm, and a resistance of 0.175 Ω. This motor was chosen because it responds well to our chosen input signals, and can easily handle the required loads. The full motor specs can be seen in Appendix A.

To simplify the analysis and calculation of input voltages, the drive wheels will be aligned with the centroid of the robot. Two caster wheels will be placed on the front and back of the robot to prevent tipping.

Additionally, the distance from the center of the robot to the wheels was chosen to be 25 cm, the maximum possible value considering the specified diameter of the robot. This was done to maximize controllability of the robot when performing angular motions, as having the force of the wheels applied further from the center of rotation results in a slower, and more controlled, rotation.

The selected motor is rated at 48 volts. To allow for some margin of error while still using a sizable range of the motor input, the motor voltage was constrained to a maximum of 40 volts. Based on this constraint, a **1:10 gear ratio** was chosen, resulting in an increase in output torque and a decrease in output speed, further improving the controllability of the robot while maintaining a control signal closer to the motor's rated voltage.

After selecting these parameters, the slew phase duration, voltage, and velocity, along with the rms torque can be calculated:

Acceleration and Deceleration Durations	2.014 [s]
Slew Phase Duration (8m, 10m, 15m)	(2.22, 3.18, 5.78) [s]
Slew Phase Voltage	40 [V]
Slew Phase Velocity	1.9418 [m/s]
RMS Torque (for 8m travel)	0.34167 Nm

Table 1: Selected parameters

The calculated rms torque value is well below the rated torque of the motor, therefore the chosen motor will not sustain over-loading related damage when used in this system.

Task 10: Final Simulation

Figures 12 and 13 below show the acceleration and jerk of the robot throughout the path motion.

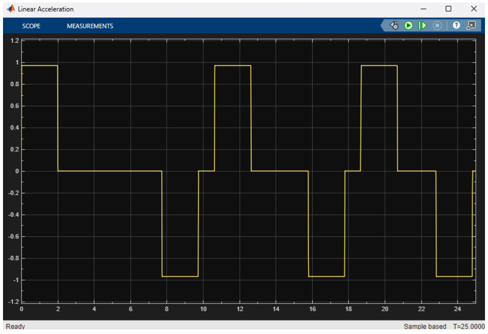

Figure 12: Acceleration Plot

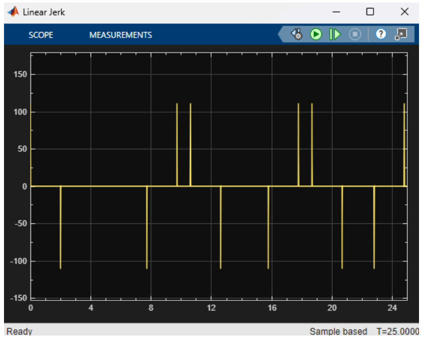

Figure 13: Jerk Plot

Assuming that the origin is located at the starting position of the centre of mass of the robot, the x-axis is facing towards the front of the robot, and the y-axis is facing towards the left wheel (following the convention followed in [1]), the following path plot was generated:

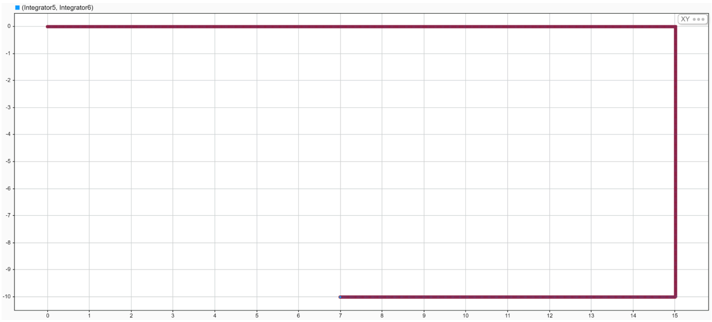

Figure 14: Final path performed by the robot

Conclusion

In summary, a model was built to model the response of a two-wheeled differential drive robot to voltage inputs. This model was used to find a time-varying input voltage signal that resulted in the robot successfully following a predefined path. Additionally, a suspension system was modelled to predict the robot's performance when traversing uneven terrain, including a door threshold and tile gaps. This allowed for the ability of the robot to safely carry food to be evaluated. These models were respectively refined and improved through iterative testing and analysis, resulting in two final models which successfully model the behaviour of a two-wheeled differential drive robot, albeit with some simplifications. Throughout this process, many of the nuances of applying physical equations in practice were better understood, and a lot of useful practical experience using MATLAB and Simulink was gained.

An aspect of the results that can be safely ignored when being applied to a real-world scenario is the high accelerations resulting from the wheels being modelled as infinitesimally small points that perfectly trace the floor. It should also be noted that the input voltages were calculated with the robot carrying a static load of 10 kg, and a more dynamic, more complex, model would allow for variation of the input voltages based on the load.

Additionally, some potential improvements to the design of the robot were identified. Sensors could be added to provide the robot with acceleration and velocity data, allowing it to better estimate its current state and ensure that it does not exceed the maximum acceleration or jerk parameters. Adding additional damping systems between the main body of the robot could also improve the design by further decreasing the risk of spilling.

Works Cited

[1] R. Dhaouadi and A. Abu Hatab, "Dynamic Modelling of Differential-Drive Mobile Robots using Lagrange and Newton-Euler Methodologies: A Unified Framework," Advances in Robotics & Automation, 2013.

[2] Moon's, "Moons Industries," [Online]. Available: https://www.moonsindustries.com/p/sm60-series-brushless-dc-servo-motors/sm0602gsa-kcf-nnv-000004611170006890?_gl=1*csavx6*_up*MQ..*_ga*NDYzNTc1MTIxLjE3NTM1NjYyNzY.*_ga_7GZR52883K*czE3NTM1NjYyNzUkbzEkZzAkdDE3NTM1NjYyNzUkajYwJGwwJGgxNDI0Nzc2Nzgw. [Accessed 26 July 2025].

Appendix A

Parameter	Unit	Value
Frame Size		60
Supply Voltage		48
Rated Voltage		48
Rated Power	W	400
Rated Speed	rpm	3000
Max. Speed	rpm	4000
Rated Torque	Nm	1.27
Max. Torque	Nm	3.8
Rated Current	A	12.6
Voltage Constant	V rms/ K rpm	6.13
Torque Constant	mNm/A	103
Resistance	Ohms ±10%@20℃	0.175
Terminal Inductance	mH	0.35