Visvesvaraya National Institute of Technology, Nagpur

Path Planning for Autonomous Navigation of Mobile Robot in Rough Terrain

BT19MEC030: Vaishnavi Desai

BT19MEC056: Tanvi Khurana

BT19MEC069: Richa Mohta

Under the guidance of

Dr. Shital S. Chiddarwar

Acknowledgements

We, final year students of mechanical engineering, at Visvesvaraya National Institute of Technology, Nagpur, would like to express our gratitude towards Dr. Shital S. Chiddarwar for giving us an opportunity to work on this project. We appreciate her constant guidance and support.

We would also like to thank all those who have helped us along the way.

Table of Contents

Abstract

In today's world, the need for automation has spread to all sectors. The ability to increase efficiency, decrease human effort and improve safety is a necessity. Mobile robots, automatic machines capable of locomotion, are making constant progress toward this cause. The project in question aims to work on the path and motion planning of a six-wheeled mobile robot, specifically to further the solution for obstacle avoidance on rough terrain. Rough terrain poses challenges different than those on smooth structured paths and hence requires the inclusion of parameters such as roll to ensure stability and prevent toppling. This project plans to achieve a bettered motion planning system by employing reinforcement learning to control the angular and linear speed variations. Applications will extend to industries, mines, planetory exploration and more.

1 Preface

1.1 Aim

Design and implement a motion planning system for off-road vehicles operating on unpaved surfaces.

1.2 Objectives

1.Assembling and programing a six wheel drive mobile robot used for rough terrain.
2.Modeling of six wheel drive mobile robot kinematics for position and velocity control of robot using simulink.
3.Motion planning for obstacle avoidance using DDPG.
4.Testing of mobile robot on rough terrain and in simulated environment.

1.3 Scope

1.Building a six wheel drive mobile robot
2.Kinematic modeling of the robot
3.Simulations in an in-house built arena

2 Introduction

Robots have the potential to improve the efficiency, safety, and convenience of human endeavors in diverse applications such as military, industry, exploration, and security. Here, the absence of road regulations and lane structure provides the car with unanticipated obstacles, much different than those faces on the road. Path planning is hence a fundamental topic in this ever-evolving field. It determines how safely a mobile robot reaches its goal position taking into consideration several criteria such as the shortest distance. It is essentially the process of constructing a path from a starting point to an endpoint given a full, partial, or dynamic map. Subsequently, another topic is motion planning. This refers to the process of defining the set of actions needed to execute the path planned. Several methodologies have been suggested for path and motion planning, only some considering the off-road rough terrain.

A unique mechanism that combines the computational advantages of discretization and motion constraints is a state lattice search space [Pivtoraiko and Kelly 2005]. It gives a minimal set of primitive paths which in a sequence can generate any other path. While it does not consider dynamic obstacle avoidance, it is far more efficient than the classical grid in its search approach. Considering the multitude of robots, theories, and applications, the planning task solutions are rarely the same. Therefore, a universal solution consisting of a general framework for robot motion planning and control was a step forward [Behere 2010].
Factors such as slip, slope, ground conditions, robot actuator limitations, and dynamics of robot terrain interactions- only specific to unstructured rough terrain environments- are successfully considered in a novel physics-based path planning approach [Sebastian and Ben-Tzvi 2019]. This simulates a closed-loop motion with a low-level controller on a realistic terrain model inside a physics engine. It presents gaps in terms of only addressing static obstacles and assuming known pose information. Fast Lidar-inertial odometry to accurately estimate the real-time pose in unknown complex environments solves one of the gaps [4].

Creating a full map using such a method reduced the performance level. The issue of mapping can be addressed in numerous ways. A 3-dimensional laser finder can be used to map the unknown environment and localize within it [Droeschel, Schwarz, and Behnke 2017]. Determination of drivability is done by analyzing height differences in the allocentric height map developed. It caters to higher distance accuracy as well. The major issue with this proposed method is the necessary initial experiments to drive out statistical error. Another method is employing a combined GPS and LiDAR system to secure the advantages of both [Patoliya *et al.* 2022]. It generates a 2D map of unknown environments minimizing the cumulative error due to the relative measurement of LiDAR data. Although it works well indoor structured spaces, it fails over rough surfaces (e.g.: surfaces with stairs).

The availability of a spatially constrained terrain map allows for dependable long-range navigation [Lacroix *et al.* 2002]. The algorithm employed takes into account concurrent instances of perception, localization, motion generation functionalities, and their integrations. All this essentially leaves is the problem of handling dynamic obstacles. A sampling-based motion planning approach efficiently generates feasible controls that steer the system toward a goal region and handle obstacles in real-time [Yang *et al.* 2019]. This continuous-time nonlinear system in a dynamic environment is managed by the control barrier function. Another proposed way is numerical linearization and inversion of forward models of propulsion, suspension, and motion to achieve generality and efficiency in trajectory generation on rough terrain [Howard and Kelly 2007]. The algorithm can accommodate vehicle dynamics and wheel-terrain interactions. The edge it has is the capability to predict the consequences of its own actions in relatively challenging environments.

3 Methodology

3.1 Hardware Assembly

The first step involved chassis assembly of the 6 wheeled rough terrain rover. The kit consisted of wheels, suspensions, 12V motors and the frame. Other components like the motor drivers, IMU(Inertial Measurement Unit), 5 MP camera, Raspberry Pi (4B) have been mounted on this chassis. To provide high maneuverability to the rough terrain rover, coil suspension system has been used. The camera and IMU will help keep track of the rovers location and acceleration. After surrounding data has been obtained, it can be used for map generation.

Each pair of wheels is controlled by a single L298N motor driver. The RPi acts as the logic control circuitry and the Motor Driver provides an interface to connect the motor and the RPi. The RPi can be operated with a power source and a WiFi which helps it connect to the source computer. The Raspbian Operating System has been installed on the RPi. It facilitates the integration of the SIMULINK model with the RPi. The wheels can be made to rotate with specified speed and direction depending on our requirement.

Hardware specifications

1. Black 6WD Search Rescue Platform Smart Car Chassis Damping Off-Road Climbing car

   Operating Voltage: 12V
   Maximum load: 6kg
   Rated current: 350mA * 6
   Motor Original speed: 17000 rpm
   Output shaft speed: 500 rpm
   Nominal torque : 5kg/cm

2. Orange 18650 Li-ion 2200mAh 11.1V 3S1P Protected Battery Pack

   Weight: 190g
   Voltage: 11.1-12.6 V
   Cell: 3S1P
   Exhibits good temperature control

3. Raspberry Pi 4 Model

   2 GB RAM
   Standard 40 pin GPIO header
   WiFi and Ethernet supported

4. L298N 2A based motor driver module

   Current sense for each motor
   Heatsink for better performance
   Power-On LED indicator
   Double H-bridge drive chip: L298N
   Operating Voltage(VDC): 5-35
   Peak Current(A): 2
   Continuous current(A): 0-36mA
   No. of channels: 2

Over-current protection: Yes

5.M5Stack 6-Axis IMU Unit (MPU6886)

On-chip temperature sensor
Gyroscope sensitivity error 1
3-axis gravity accelerometer and 3-axis gyroscope

6. Raspberry Pi 5MP Camera Board Module

5 MP resolution capable of 2592 x 1944 pixel static images
Supports 1080p30, 720p60 and 640x480p60/90 video

3.2 Simulation and Modelling

The simulation of the motor and motor driver was carried out on MATLAB Simulink. The DC motor was modelled using its governing equations where voltage (Ua) and reference torque (TL) as input. It also takes into consideration damping torque.

$$Jdw/dt = T_e - T_L - f_r w \tag{1}$$

$$U_a = k_c w + R_a + L_a di_a/dt \tag{2}$$

Motor drivers are used to control the motors. An L298 motor driver which uses a dual H bridge

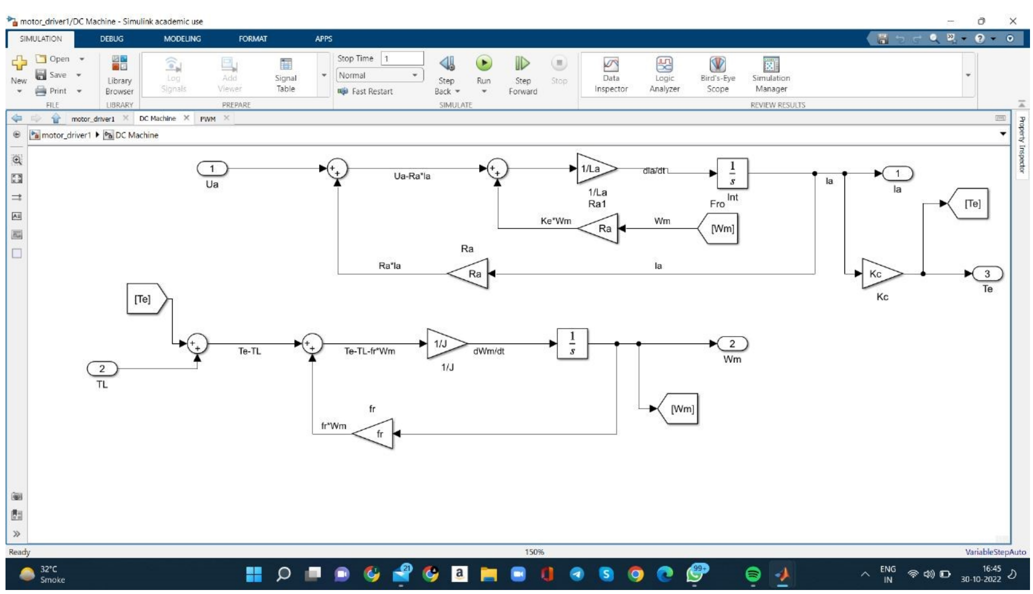


Figure 1. DC motor model

circuit has the ability to control 2 DC motors independently for speed as well as direction. It is modelled on Simulink separately for both.

Speed Control

A closed-loop system consisting of numerous components controls the speed of the motors. The main principle is Pulse Width Modulation (PWM). PWM is a method through which variable voltage can be generated by turning on and off the power that's going to the electronic device at a fast rate. The average voltage depends on the duty cycle of the signal, or the amount of time the signal is ON versus the amount of time the signal is OFF in a single period of time. The Simulink model representing PWM consists as input, the voltage across the battery (Udc), and a reference voltage (Ua,ref). This is the voltage that is needed to achieve the desired speed. It returns two modified signals (qa and qb) as the output. Both these signals have varying widths, although their amplitude is 1.

$$d_A = 1/2(1 + U_{aref}/U_{dc}) \tag{3}$$

$$d_A = 1/2(1 - U_{aref}/U_{dc}) \tag{4}$$

$q_A = 1 when d_A > repeated signal and = 0; otherwise$
$q_B = 1 when d_B > repeated signal and = 0; otherwise$

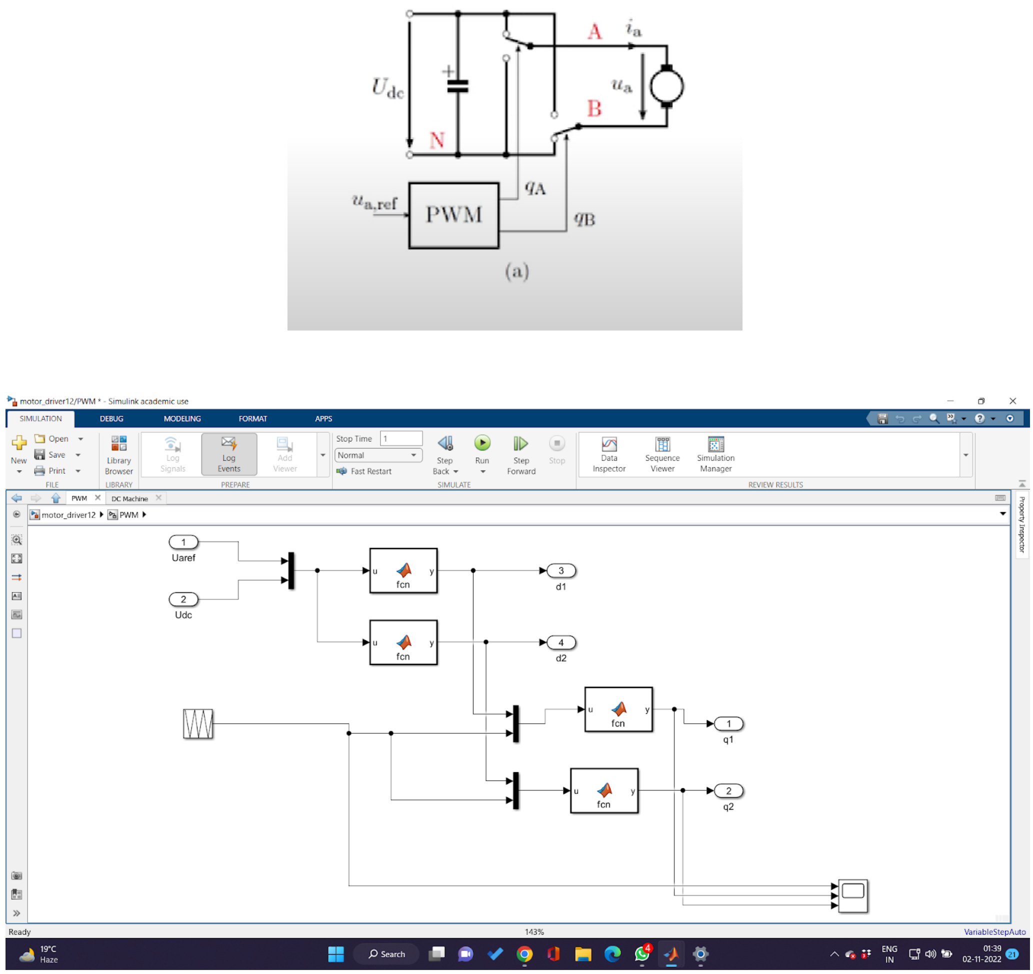


Figure 2. PWM

Next is the DC/DC Converter which amplifies the difference between the two signal outputs by a factor equal to Udc. The output here is the voltage supplied to the Dc motor (Ua). This is done to ensure that the values of Ua and Ua,ref are close.

$$U_a = (q_A - q_B)U_{dc} \tag{5}$$

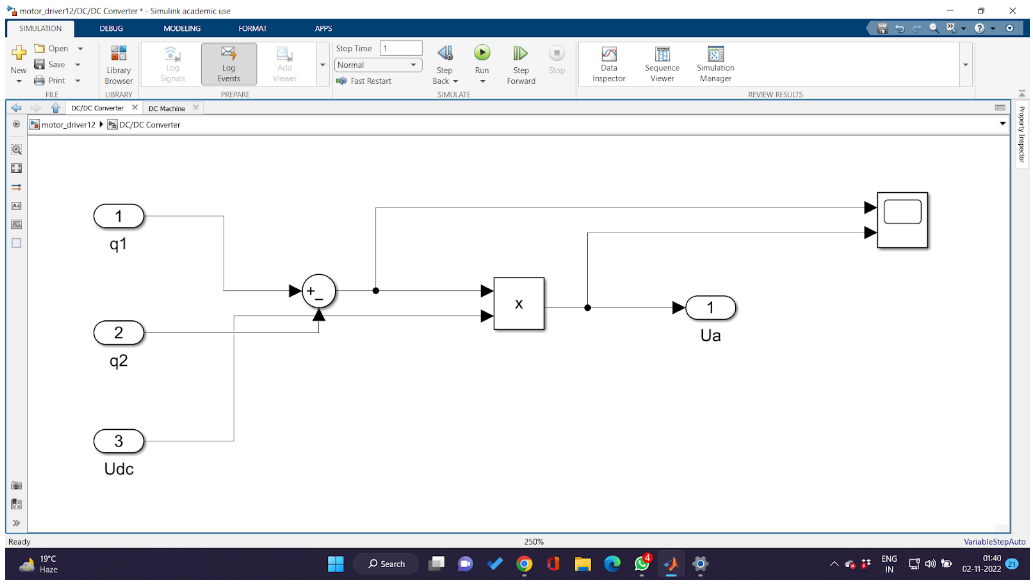


Figure 3. DC/DC Converter

Basically, the PWM and the DC/DC Converter get Ua from Ua,ref since all the other values in the equations are pre-decided based on the hardware employed. This Ua along with TL is sent as

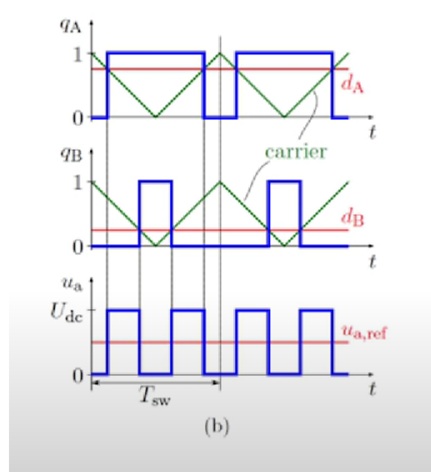


Figure 4. Modulation principle

input to the DC motor to get the present values of rotational speed (w), torque (Te), and current (ia), where rotational speed is the desired parameter to be controlled. To achieve the desired speed, a reference voltage input to the motor (Ua,ref) as mentioned above is needed. It is obtained using two current regulators. Starting with current regulator 1: here, the desired speed is given as an input value (wref), along with the current speed achieved (w). A proportional-integral-differential (P-I-D) controller generates a current that should pass through the motor (ia,ref) which minimizes the difference between w and wref. This current regulator also returns as output the value of TL, which is supplied to the motor. Current regulator 2 uses the value of the desired current (ia,ref) and the value of the current actually passing through the motor (i). It also employs a P-I-D controller to get the value Ua,ref which reduces the difference between the two values.

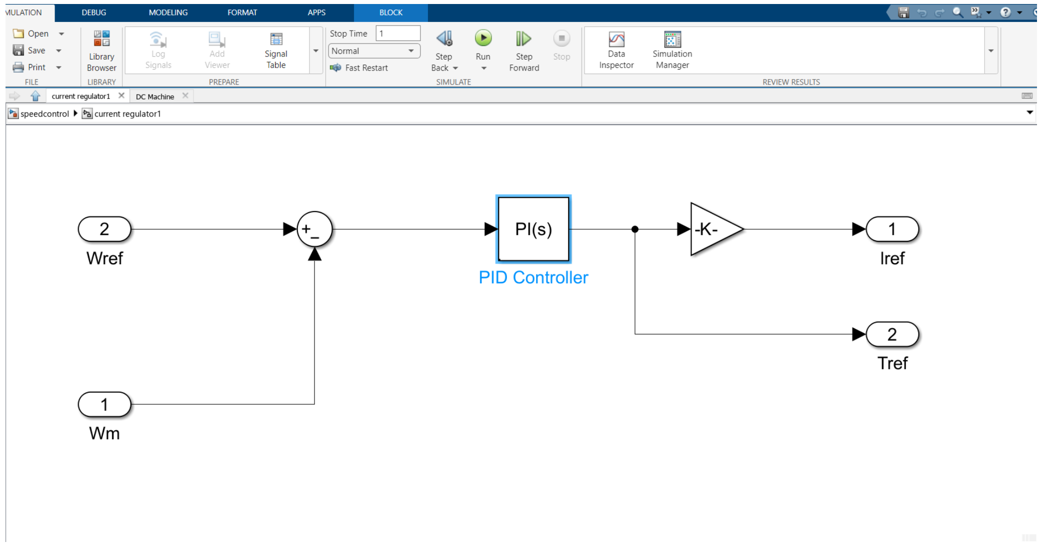


Figure 5. Current Regulator 1

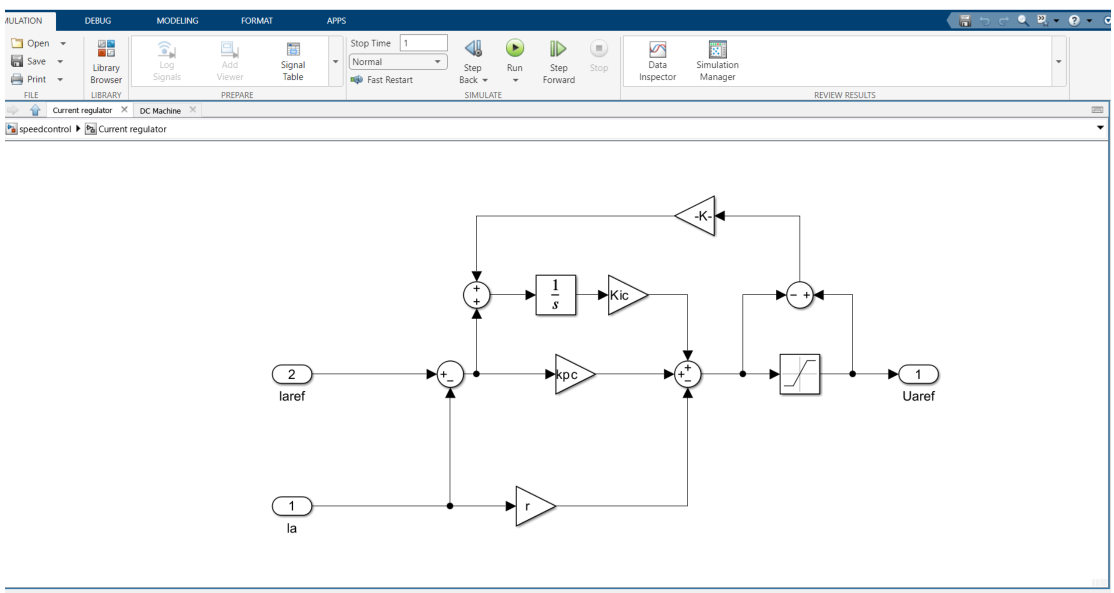


Figure 6. Current Regulator 2

Ua,ref is then supplied to the PWM. This completes the system. Since all values are interconnected and interdependent, the model needs to be initialized with certain values.

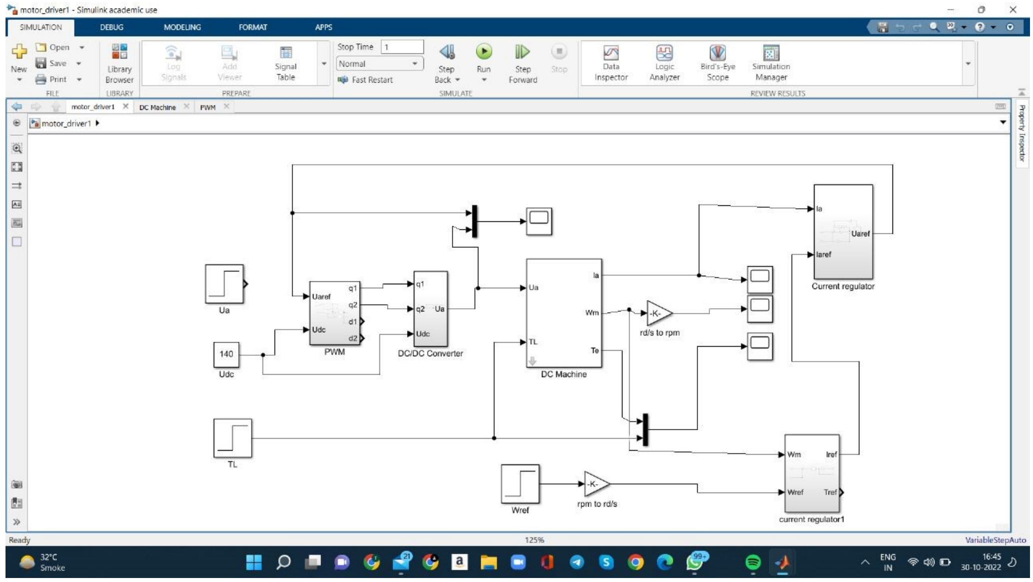
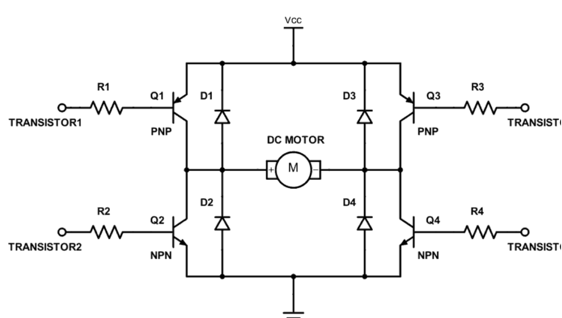


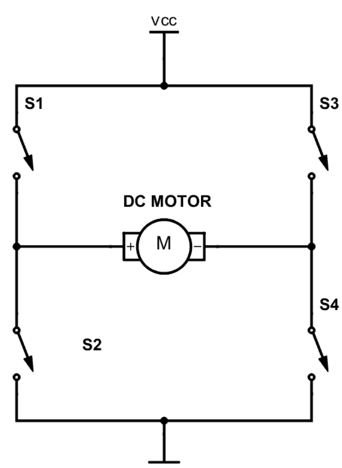
Figure 7. Speed control

Direction Control

The dual H bridge circuit of the L298 motor driver can be essentially reduced to a circuit with four switches as shown below. The direction is altered by selectively turning a series of these switched on and off.



(a) Actual circuitry



(b) Representative circitry

Figure 8. H bridge for direction control

The direction is set based on the relative values of Ua and Udir, where Ua comes from the motor and Udir is set by the use.

$$U_a > U_{dir} : clockwise \quad rotation \tag{6}$$

$$U_a < U_{dir} : clockwise \quad rotation \tag{7}$$

Ua and Ua,ref are the input values to the deadzone zone. It returns zero if the input is within then specified range and the actual value otherwise. The NOT block reverses the signal supplied. Overall, this logic circuit returns the values of Ua and Ua,ref if they are within the given range and zero otherwise. The range specified here is 6 to 12 Volts; this is because the voltage rating of the motor is 12 Volts and the minimum speed value is achieved at 6 Volts.

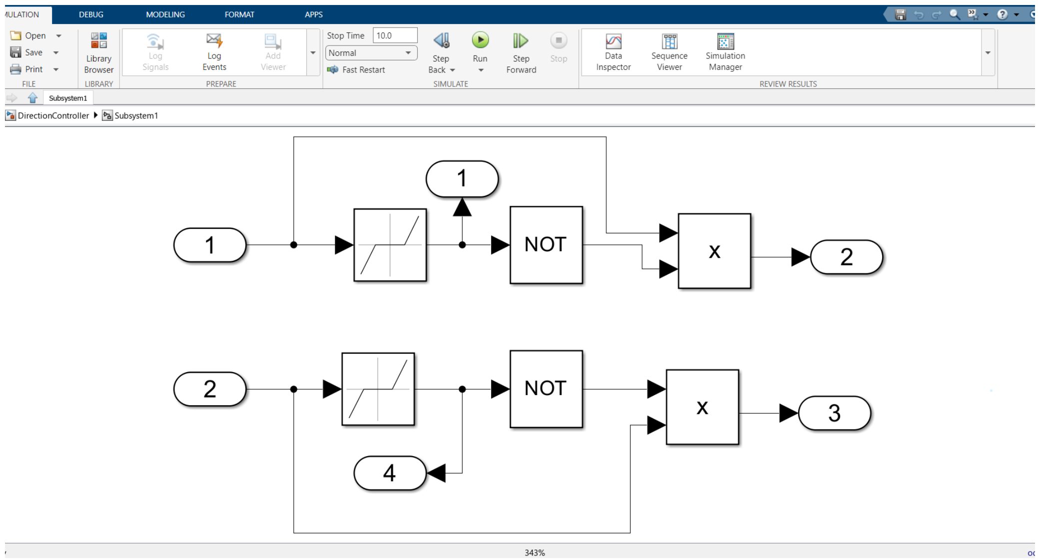


Figure 9. Range control

Udir is subtracted from Ua and only the sign of this operation is carried forward. For the rover to move forward Udir is set less than Ua, the sign is positive and the motors rotate in clockwise direction. Vice-versa for reverse direction. If the motor is to be stopped, Udir is set same as Ua,ref.

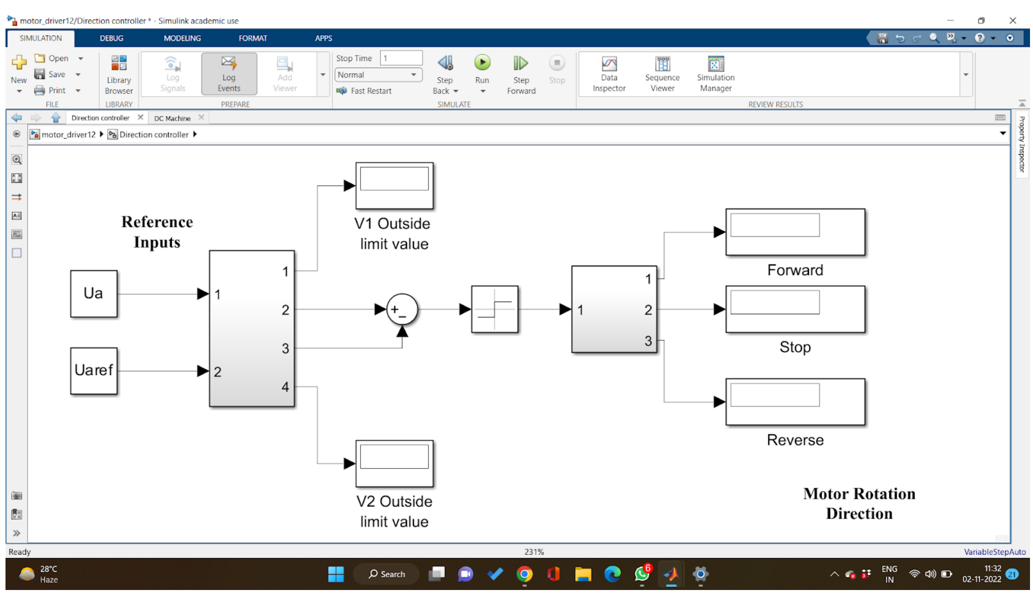
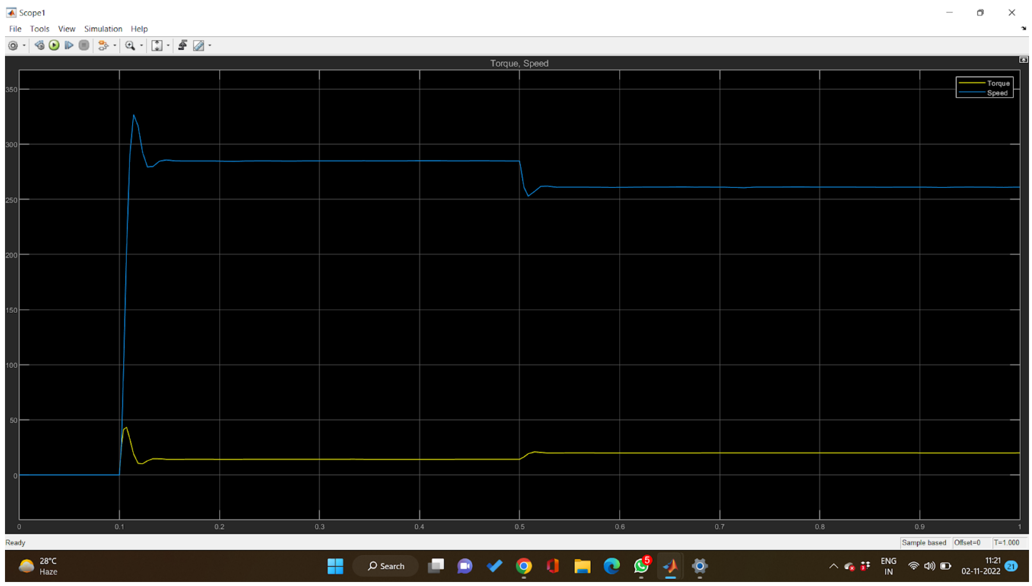


Figure 10. Direction Control

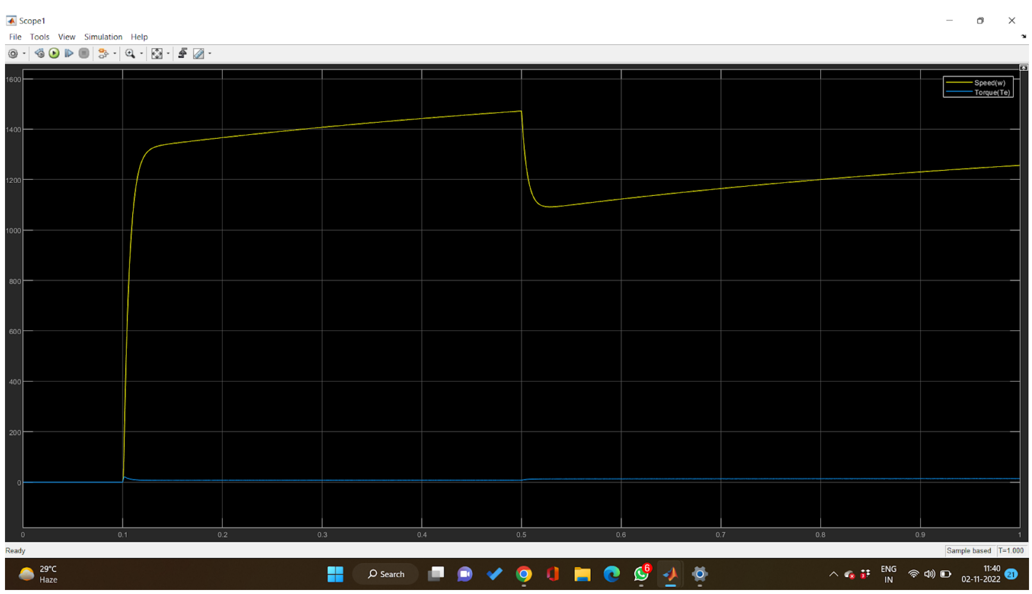
4 Results and Conclusion

4.1 DC Motor Characteristics



4.2 Speed and Torque variation

Given a reference speed (wref) of 2000 rpm, the graph for speed and torque with respect to time is computed. The drop occurs when the first feedback has been received. Since the classical approach to controls has been used, the value of actual speed (w) will reach wref at infinity. In the given time frame of 1 seconds, a speed of 1450 has been achieved.



The value of torque changes with the sudden change of speed.

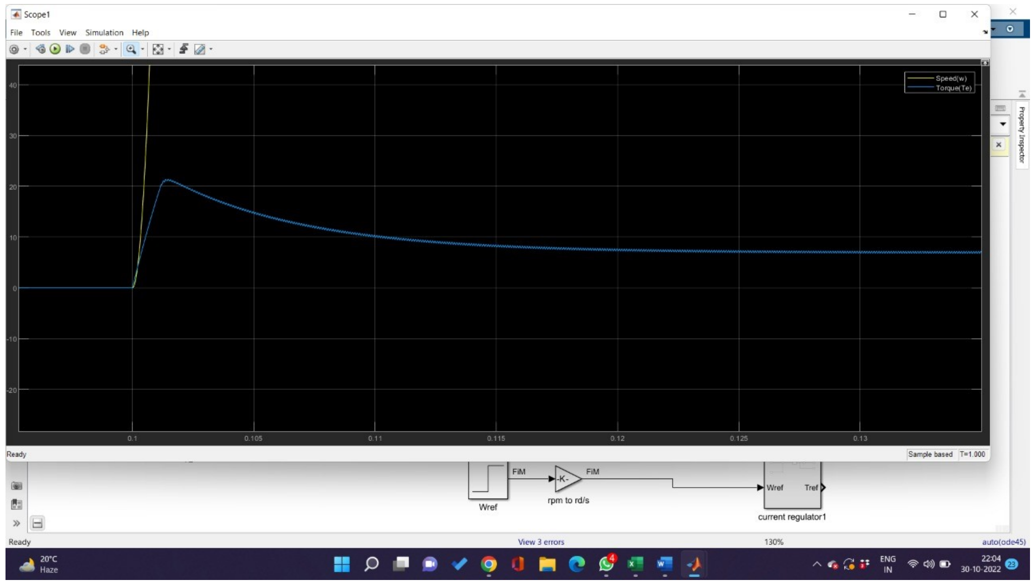


Figure 11. Scaled representation

5 Plan for Phase-II

The process of saving contours from the surrounding vicinity is called mapping. Gazebo is a platform which converts the complex 3D world into a much simpler map which is easy to comprehend for the rover.The real-time data obtained from camera and IMU can be used to generate such a map. The generated map will help the robot localize itself with its field of interest.

5.1 Obstacle Avoidance

It will involve use of Deep Deterministic Policy Gradient agents i.e. Reinforcement Learning Algorithm. This is followed by simulation of robot in the map and avoiding all static obstacles.

The deep deterministic policy gradient (DDPG) algorithm is a model-free, online, off-policy reinforcement learning method. A DDPG agent is an actor-critic reinforcement learning agent that searches for an optimal policy that maximizes the expected cumulative long-term reward. DDPG based reinforcement learning will be used to develop a strategy for a mobile robot to avoid obstacles. Reinforcement Learning(RL) is a type of machine learning technique that enables an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences.

The mobile robot will be trained to avoid obstacles given range sensor readings that detect obstacles in the map. The objective of the reinforcement learning algorithm is to learn what controls (linear and angular velocity) the robot should use to avoid colliding into obstacles.

An occupancy map (modelled on Gazebo) of a known environment to capture range sensor readings, detect obstacles, and check collisions the robot may make will be used. The range sensor readings will act as the observations for the DDPG agent, and the linear and angular velocity controls will be the action.

5.2 Testing

The final step of the project would be hardware testing of our six wheel drive mobile robot on rough terrain.

References

Behere, S. 2010. *A Generic Framework for Robot Motion Planning and Control.*

Droeschel, D., M. Schwarz, and S. Behnke. 2017. "Continuous mapping and localization for autonomous navigation in rough terrain using a 3D laser scanner." *Robotics and Autonomous Systems* 88:104–115.

Howard, T. M., and A. Kelly. 2007. "Optimal rough terrain trajectory generation for wheeled mobile robots." *The International Journal of Robotics Research* 26 (2): 141–166.

Lacroix, S., A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury, M. Herrb, and R. Chatila. 2002. "Autonomous rover navigation on unknown terrains: Functions and integration." *The International Journal of Robotics Research* 21 (10-11): 917–942.

Patoliya, J., H. Mewada, M. Hassaballah, M. A. Khan, and S. Kadry. 2022. "A robust autonomous navigation and mapping system based on GPS and LiDAR data for unconstraint environment." *Earth Science Informatics,* 1–13.

Pivtoraiko, M., and A. Kelly. 2005. "Efficient constrained path planning via search in state lattices." In *International Symposium on Artificial Intelligence, Robotics, and Automation in Space,* 1–7. Munich Germany.

Sebastian, B., and P. Ben-Tzvi. 2019. "Physics based path planning for autonomous tracked vehicle in challenging terrain." *Journal of Intelligent & Robotic Systems* 95 (2): 511–526.

Yang, G., B. Vang, Z. Serlin, C. Belta, and R. Tron. 2019. "Sampling-based motion planning via control barrier functions." In *Proceedings of the 2019 3rd International Conference on Automation, Control and Robots,* 22–29.