


# TERRAIN SURVEILLANCE ROBOT

-ADARSH JK (19BEE1023)
-SAMBIT PATRA(19BEE1109)
-SANKAR RAM(19BEE1200)
-ARUL R (50912)

# ABSTRACT

The increasing human population drove humans to find more terrain and places to accommodate human life. The main goal of the rover is to work on different types of terrain and help understand humans more about compatibility of life on the terrain. Rover has computer vision capabilities and sends live feedback to the station about the objects around it. The station has the control to guide the rover in any direction of their choice using IOT protocols. The robot has path planning, object tracking and obstacle avoidance capabilities. It is based on a rocker boggie chassis setup. Further the abilities are enhanced with motion planning and differential drive axle technology. The various sensors that are present in built on the rover perform in-situ tests to check If the terrain is suitable for humans. The most important thing is the rover can perform $CO_2$ detection test that makes it useful to monitor gas levels.

# LIST OF COMPONENTS

| S.No. | Component name | Quantity |
|---|---|---|
| 1. | NODEMCU | 2 |
| 2. | TP405 VOLTAGE REGULATOR | 1 |
| 3. | BME280 | 1 |
| 4. | MQ2 | 1 |
| 5. | HC-SR04 | 1 |
| 6. | L298N | 1 |
| 7. | GPS | 1 |
| 8. | RASPBERRY PI | 1 |
| 9. | PI CAM | 1 |
| 10. | DC MOTOR | 4 |
| 11. | WHEELS | 6 |

The estimated cost for building the prototype is 25,000 INR

# Introduction

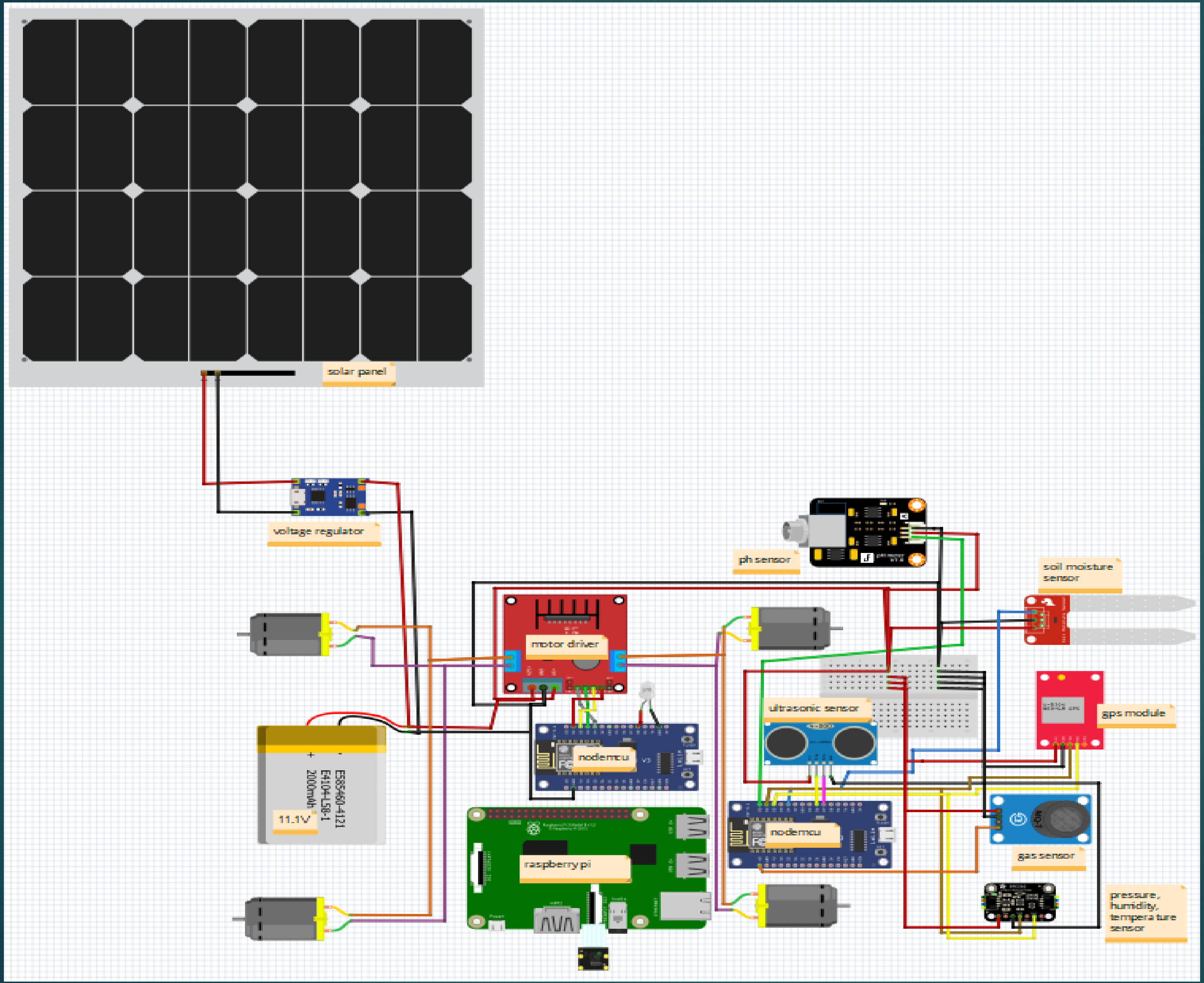
The population of human beings is increasing steadily and there is a necessity to discover new lands for people to live in, but not every terrain is suitable for living. New lands that are discovered need to be examined and tested for compatibility of life. This paper briefs about the rover that can be used for testing the terrain. The rover can either be controlled autonomously or can be controlled manually using IOT protocols.

Machine learning and AI techniques are used to train the rover for autonomous capability. The robot has computer vision technology that assists the rover to perceive the surroundings, the robot has object detection capabilities that can be used to identify its surrounding objects. The robot has pH sensors and soil moisture sensors to test the pH and water content in the soil. The exact location of the robot can be determined with the built in GPS system. The rover has in-situ test modules to test the terrain moisture, terrain pressure, temperature surrounding, humidity level and gas contents and sends it to the base station using IOT. As the rover has computer vision capabilities the rover can detect plant and animal life in the terrain and classify them accordingly.

The rover also has pH sensor and soil moisture sensor. Generally, the standard pH scale ranges from 0-14 with prescribed standards of alkalinity, acidity and neutral nature. The prevailing varieties operate is discussed in the later part. The soil moisture sensor in a view of simplifying the process with integrated which has METER soil sensor which has same research grade accuracy with minimum sensitivity (<10 dS/m bulk EC), soil textures and temperature gradients. This sensor uses high frequency capacitance technology to measure water volume of the soil or the volume basis compared to the total volume of the soil. The technologies used include frequency domain, reflectometry, galvanic cell, soil resistivity, time domain transmission and neutron moisture gauges.

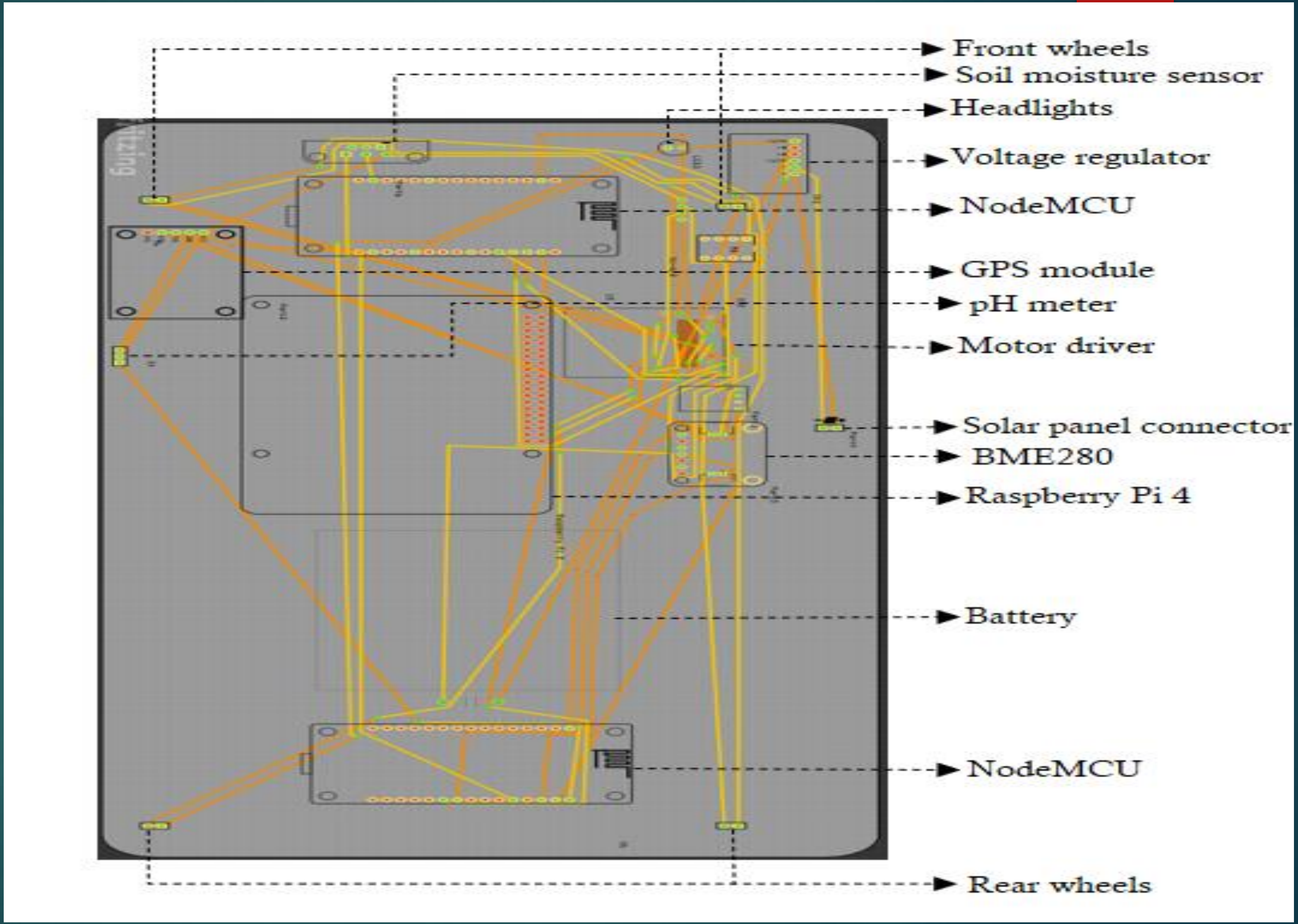
# Description Of The Proposed Method

The proposed model of the rover is based on Raspberry Pi as the main computing board, paired with NodeMCU to receive transfer data from board to the base station on the internet. The Raspberry Pi handles the image processing work and streams the video feed to the base station. The robot has object detection capabilities that helps the robot to interact better with the environment. The robot is programmed with path planning algorithm that helps to create a pursuit for the robot to travel from start pose to goal pose. The robot has object tracking algorithm that helps the robot to track any dynamic object in the environment. The robot has obstacle avoidance technology that helps the robot to avoid collision thus saving time and cost of repair.




# Circuit Diagram

# PCB DIAGRAM


Front wheels
Soil moisture sensor
Headlights
Voltage regulator
NodeMCU
GPS module
pH meter
Motor driver
Solar panel connector
BME280
Raspberry Pi 4
Battery
NodeMCU
Rear wheels

# Discussion of the neuron model

Neural models are now very common. This forms an address to link the sample to label. This is a numerical optimisation mechanism. So convergence criteria provides the feedback (learning implies feedback). Such techniques are used extensively in maths and statistics (e.g. regression) and they employ mathematical properties such as maximising the gradient of the convergence criteria. Feedback must come from the source of the input. In neural systems the sample and the labels are selected by the prior knowledge of the user (picture of a lake, link to label “lake”).Genetic algorithms (on computers) again they are symbolic and use heuristics (rules) which can eventually find stable states. The heuristics(rules) can become incorporated into members of the population to continue development. These find local optima in a constraint-based system. Again it is not learning as we know it.

- **Hebbian Learning**

The environmental interactions allow a relationship between observations (sensors) and actions to become established if the outcome is OK. This is achieved by reinforcing learning the neural links between the input and its normal response. This process allows the system to adapt to its environment more efficiently.

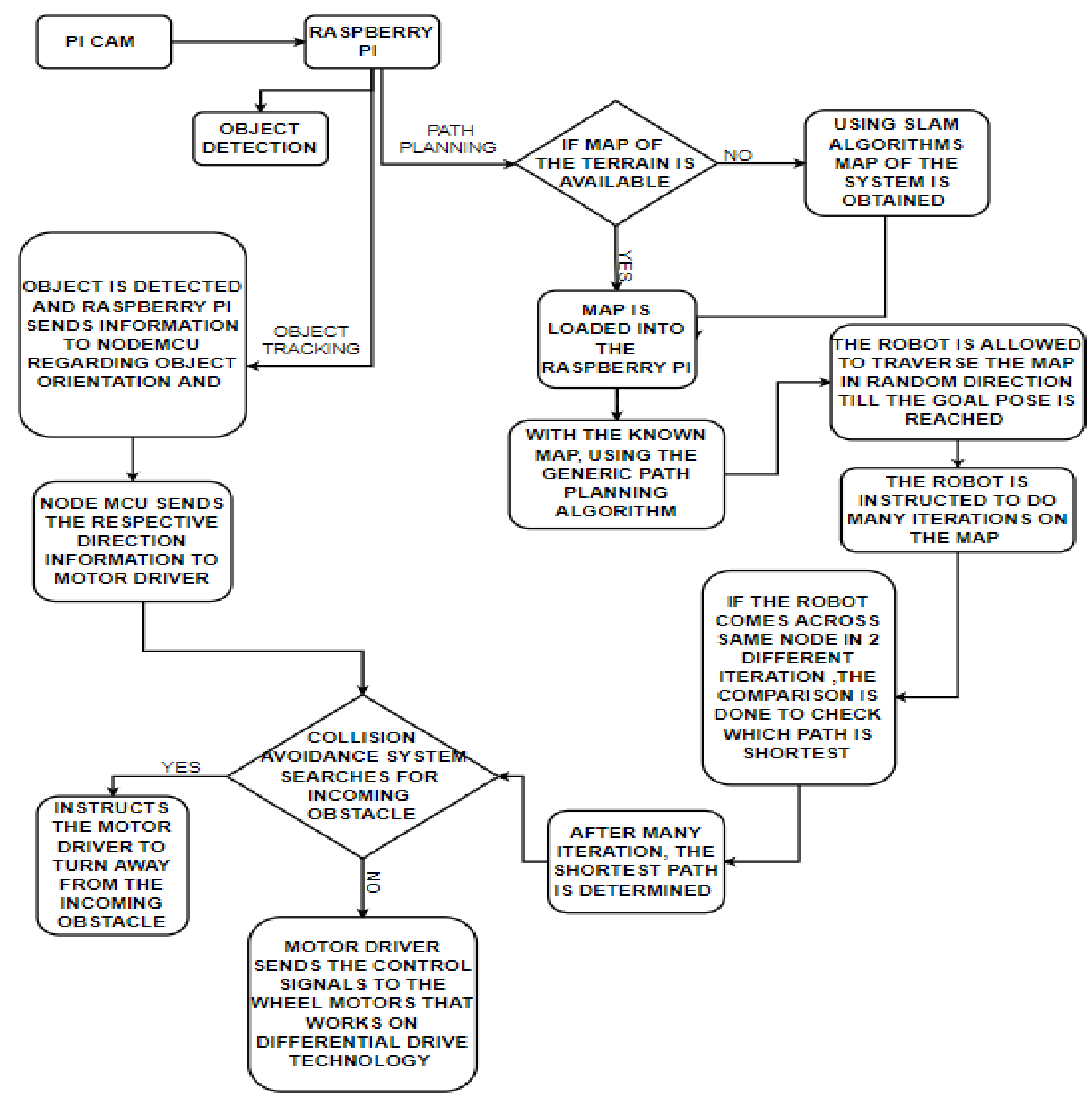
- **Knowledge refinement**

For conscious systems which are aware of the environment, we can build models of a system and obtain feedback through the model. Then we can test the predictions and decide eventually to store the relationship between inputs from sensors to their output interactions (this is knowledge ) so we acquire knowledge and refine and store further knowledge. Basically knowledge is a persistent and appropriate response to an input set. The feedback from interactions with the real world is used to train the system for better performance.

We have extended this to encode knowledge into text and then reading it to acquire potentially new knowledge (without interacting with our environment) This technique encourages socially derived knowledge to be accumulated. All these algorithm are used to move effectively from start pose to end pose.

# SYSTEM FLOWCHART

The object detection is done on Raspberry Pi using Pi cam as camera module. For the object tracking algorithm, ultrasonic sensor sends data to the motor driver about the object location and the motor driver aligns the robot in the direction and moves forward till it reaches the object. For path planning, if the map of the environment is available, it is loaded onto the system if not, using SLAM algorithm, the map of the system is obtained and then loaded onto the system, and the shortest path to the goal pose is planned and moved accordingly. For the obstacle avoidance, the ultrasonic scans for the incoming obstacle and then sends signal to motor driver to turn away from the obstacle.



# Differential Drive

The Path of the wheels cut through a turn. The wheels are not connected, when all the wheel rotate in forward direction, the rover/robot go in the forward motion and when all the four wheels rotate in reverse it moves backward. If the robot want to rotate about the central axis then the  opposite caps should rotate in against each other in opposite direction. Basically the direction of the robot is dependent on the rate and direction of rotation of the two driven wheels. These quantities should be designed and controlled precisely to observe the desired movement. The angular kinetics of the wheels is given in figure.

- ***Forward Kinematics***

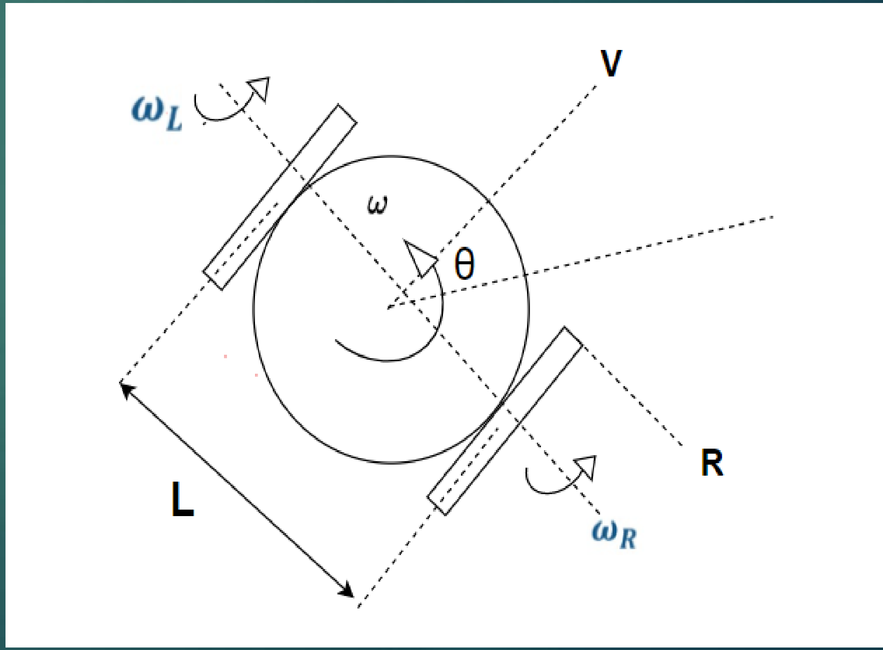
$v=R*(\omega_R+\omega_L)/2$ ................................(1)

$\omega=R*(\omega_R-\omega_L)/L$ ................................(2)

- ***Inverse Kinematics***

$\omega_L=(v/R-\omega*L/2R)$ ..................................(3)

$\omega_R=(v/R+\omega*L/2R)$ ...............................(4)

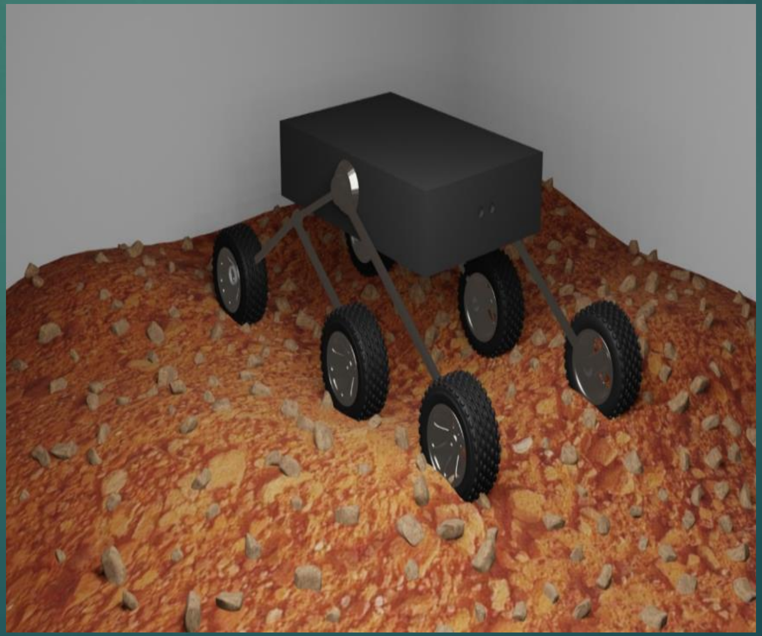


# Rocker Bogie

The rocker bogie has no spring or sub axles for each wheel, which allow the rover to climb over different obstacle that are twice the wheels diameter in size. Here the suspension is not a spring which tilt the stability to the height of the centre of gravity, maximum cases system which uses spring tend to tip more easily compared to the rocker bogie. By using rocker bogie we can reduce the motion by half compared to other suspension system, As the wheels act independently for motion which helps in steering system which in turn allows the vehicle to turn in place as 0 degree turning ratio. It moves slowly as 10cm/sas to minimize the dynamic shock and consequential damage to the rover when moving over sizeable obstacle.

The rocker bogie system can withstand a tilt of at least of 45 degree in any direction . In comparison with other design of chassis, rocker bogie as big ground clearance with help them to travel over the obstacle easily.

- Advantage:
  - Equally distributes the load to the six wheels.
  - It can move over the obstacle which is twice in size.
  - Equally distribute the drive torque.
  - Has no axle and spring.
- Disadvantage
  - It is slow it cannot speed higher than 10cm/s.
- Application
  - It can travel with heavy load easily but slowly.
  - It can be sent venture unmanned areas.



# Autonomous Navigation

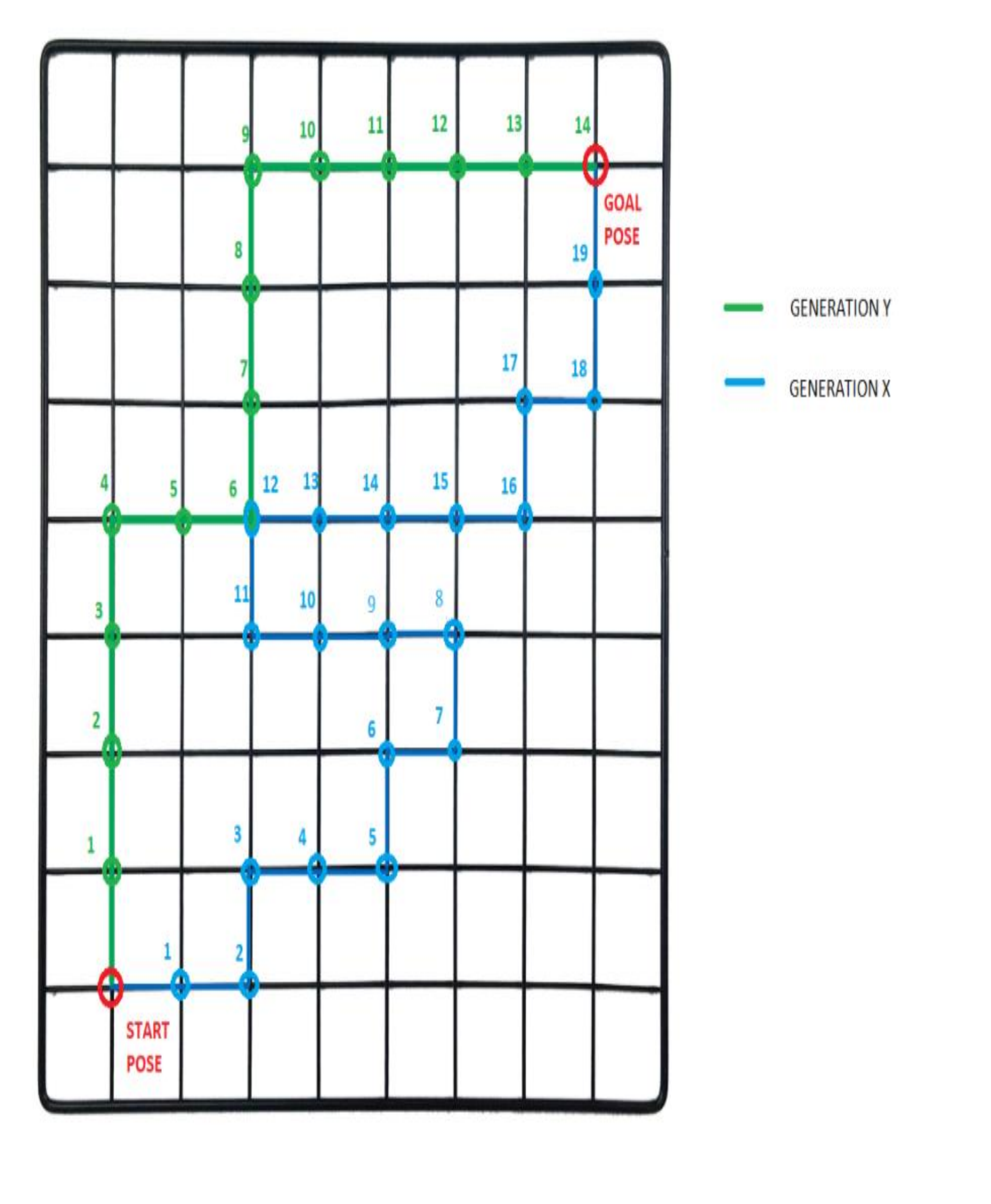
Autonomous navigation is the ability for a vehicle to determine its location and plan a path to some goal without a human. Autonomous navigation spectrum ranges from tele-operations to full autonomy.

- **Heuristic approach**
- **Optimal approach**
- **Challenges**
- **Capabilities of autonomous systems**
- **Estimate Position and Orientation of a Mobile Robot Using a Particle Filter**
- **SLAM Algorithm**

# PATH PLANNING

The goal is to find a path between starting pose to goal pose. For a mobile robot (x,y) location and its orientation decides the starting pose. Path is a sequence of pose state connects the starting and goal pose efficiently. Determining this sequence is called path planning. It's a subset of motion planning.

Here, the starting and the ending pose will be predefined. The map is completely shred into nodes the starting pose is node 0 the system will try to traverse in a random direction and mark it node 1, the distance between node are saved and moves to another point and mark it node 2 and similarly traverse in a random direction until the goal pose is reached. These data will be saved into the memory, then the second iteration will begin, similarly the robot moves in a random direction and mark it node 1 of iteration 2 and works on this algorithm until the goal pose is reached. while traversing, if the some iteration comes across the same node as in the previous iteration, the distance travelled to reach the node is compared and the iteration with less overall distance is considered better performing. The similar example is given in figure 7, clearly generation Y travelled less distance compared to generation X, therefore generation Y is preferred than generation X. Similarly, the robot is instructed to do 200-300 iteration and finds the best path.



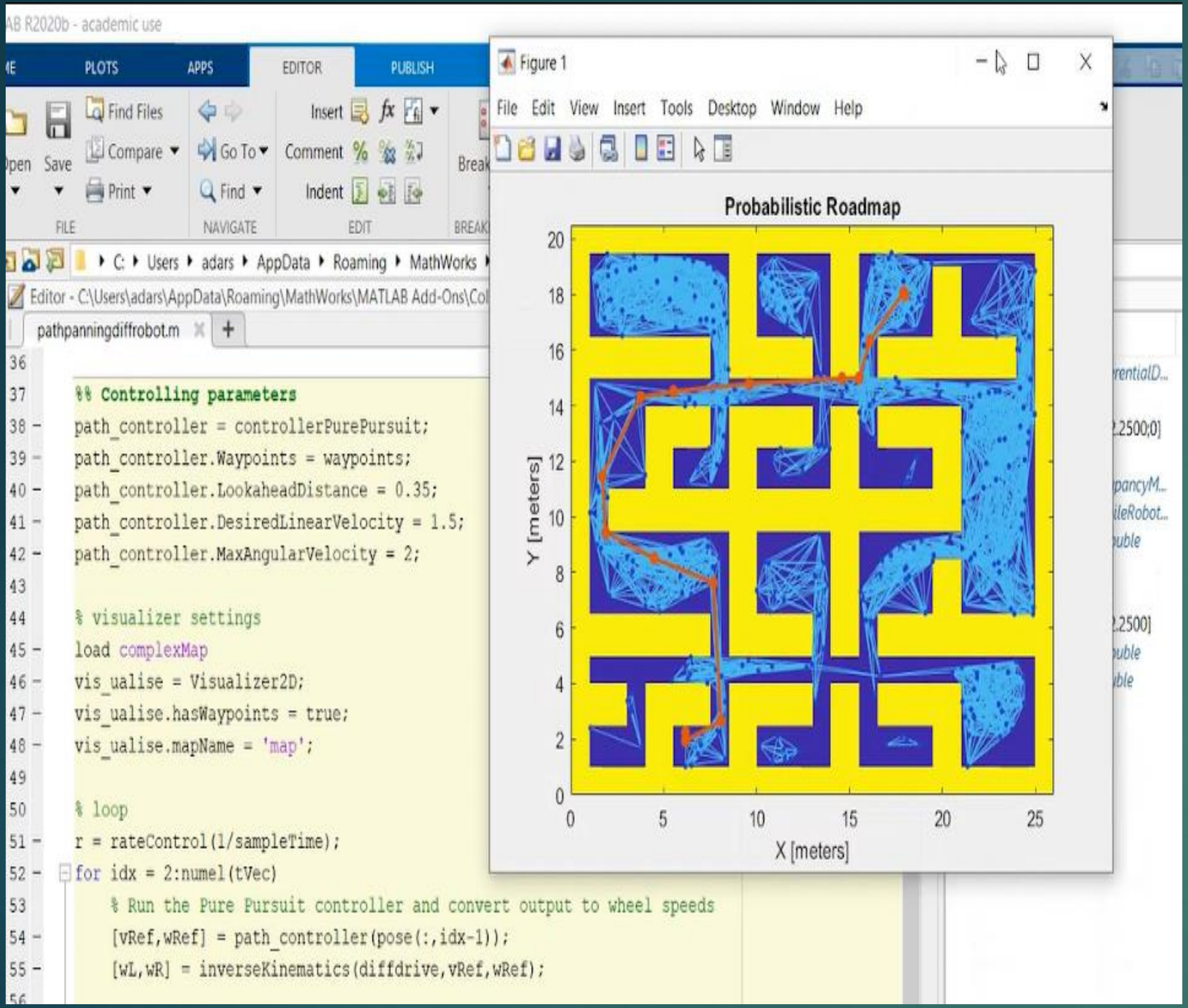


Figure 1
File Edit View Insert Tools Desktop Window Help
Probabilistic Roadmap
Y [meters]
X [meters]
pathpanningdiffrobot.m
%% Controlling parameters
path_controller = controllerPurePursuit;
path_controller.Waypoints = waypoints;
path_controller.LookaheadDistance = 0.35;
path_controller.DesiredLinearVelocity = 1.5;
path_controller.MaxAngularVelocity = 2;
% visualizer settings
load complexMap
vis_ualise = Visualizer2D;
vis_ualise.hasWaypoints = true;
vis_ualise.mapName = 'map';
% loop
r = rateControl(1/sampleTime);
for idx = 2:numel(tVec)
% Run the Pure Pursuit controller and convert output to wheel speeds
[vRef,wRef] = path_controller(pose(:,idx-1));
[wL,wR] = inverseKinematics(diffdrive,vRef,wRef);

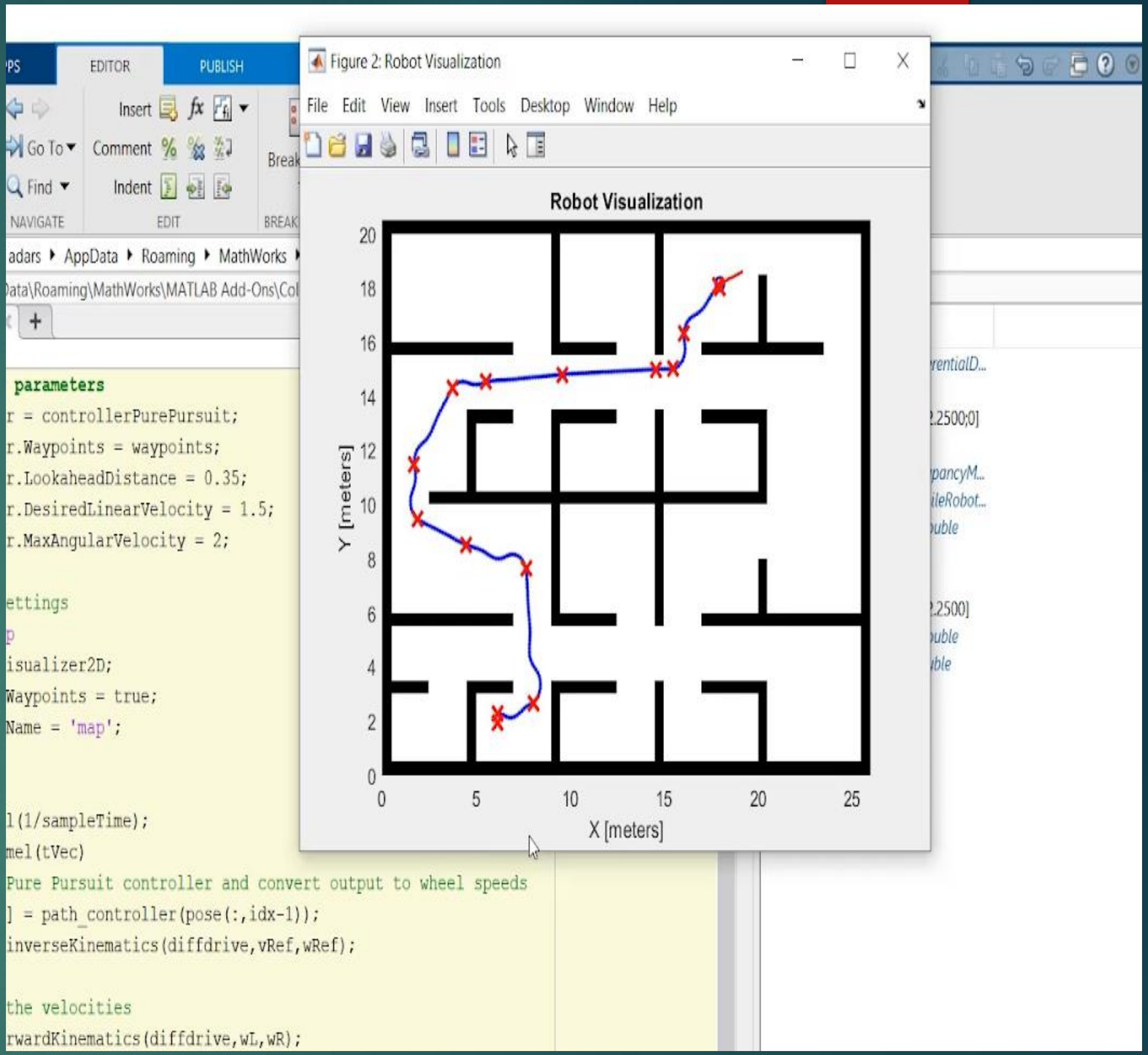
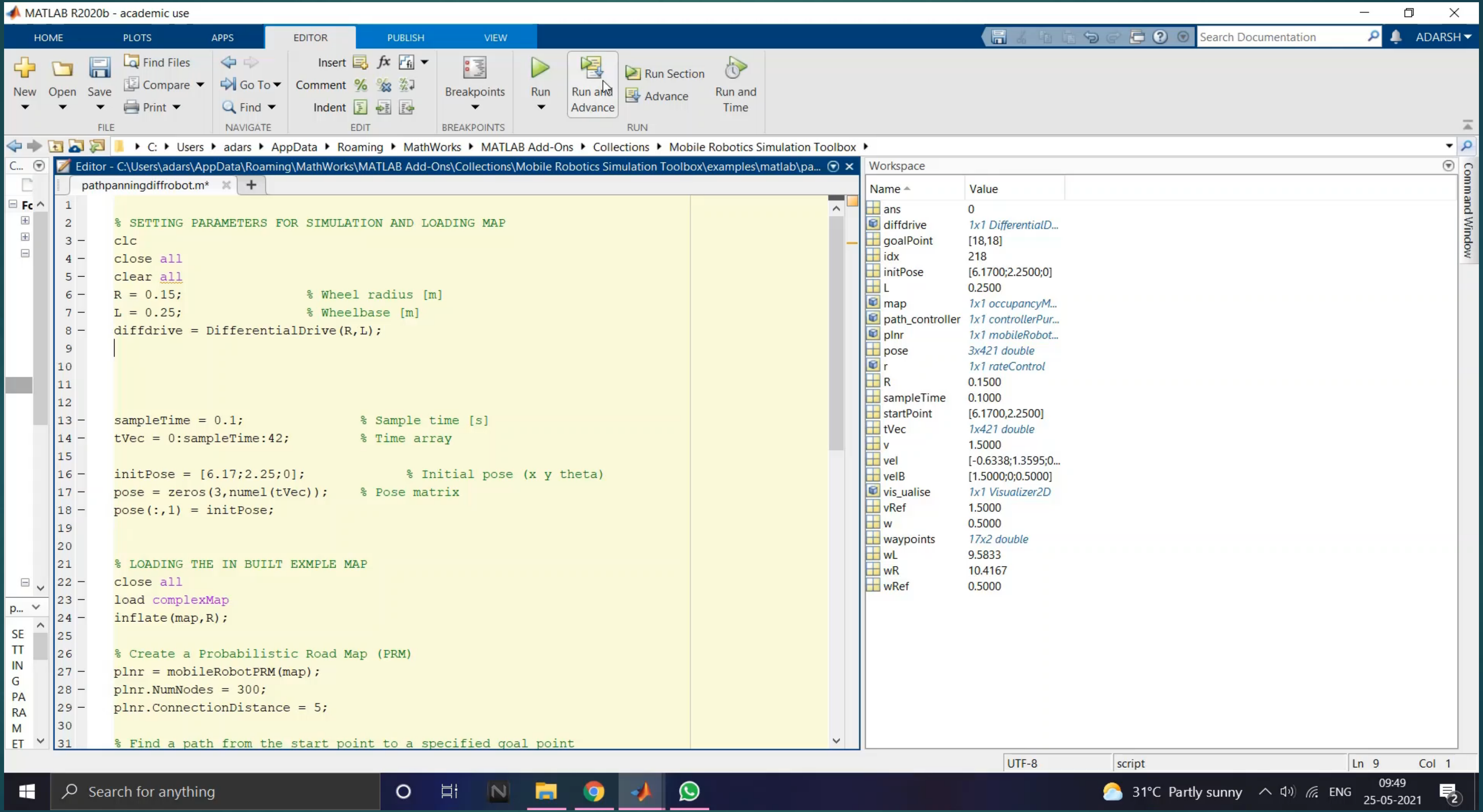


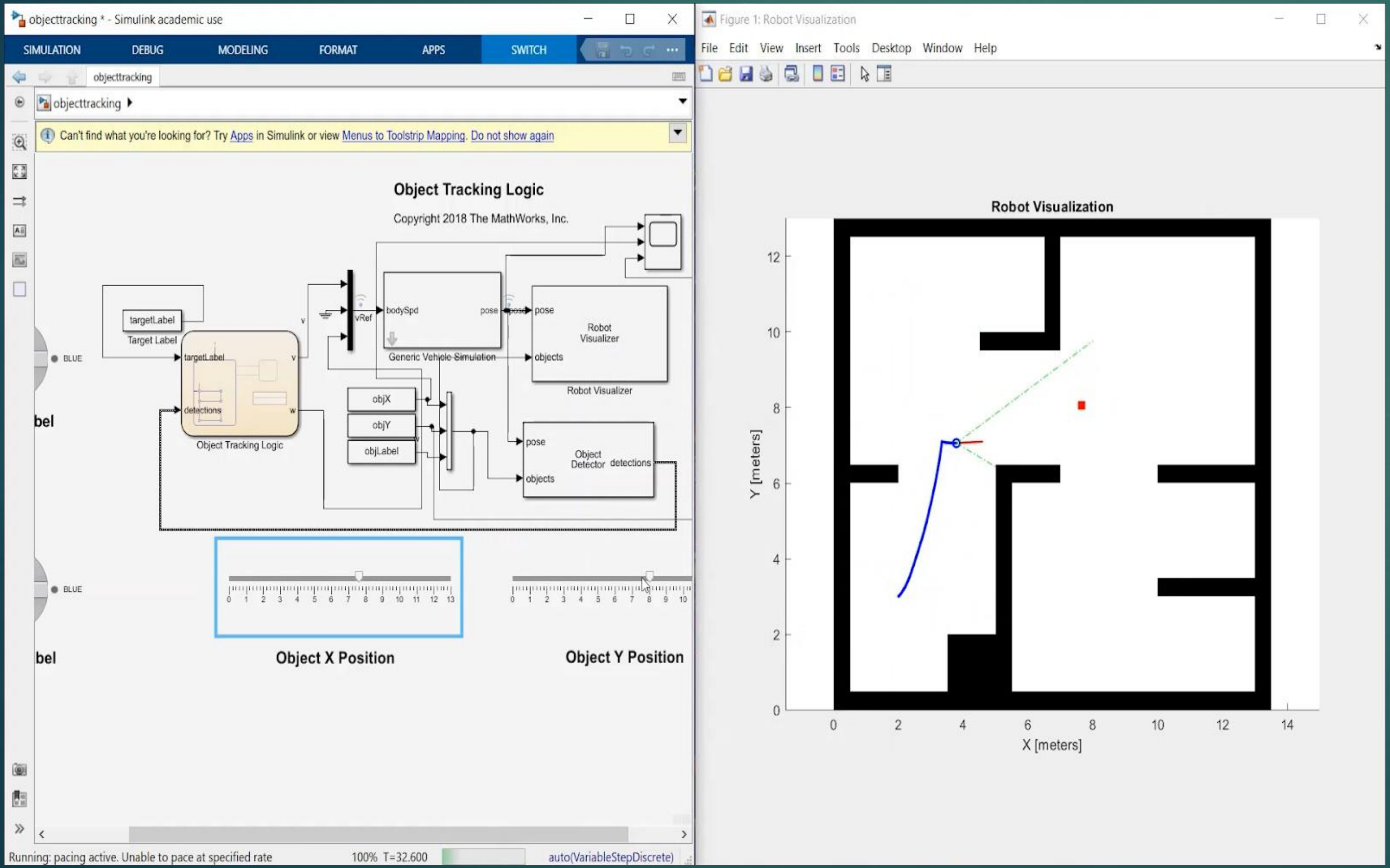
Figure 2: Robot Visualization
File Edit View Insert Tools Desktop Window Help
Robot Visualization
Y [meters]
X [meters]

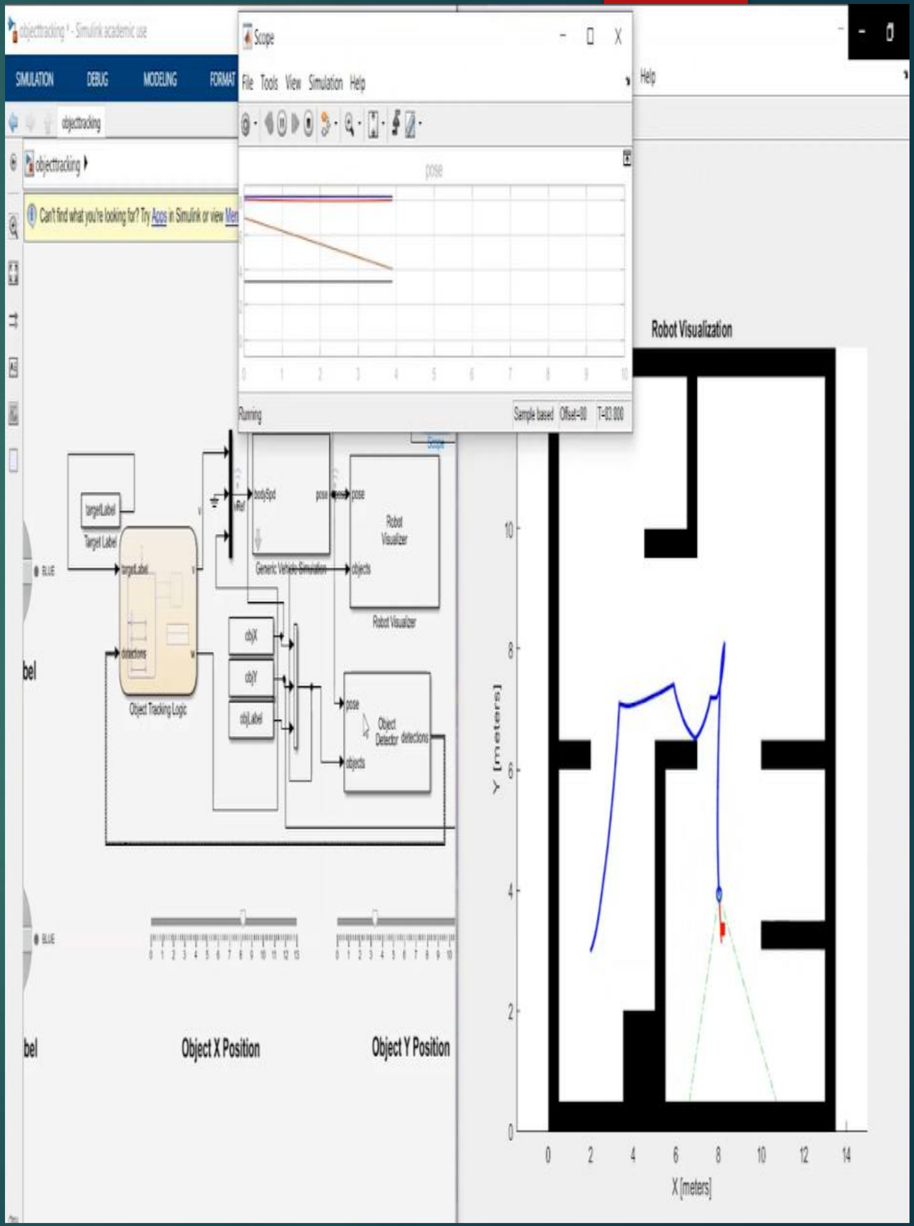
# VIDEO DEMONSTRATION



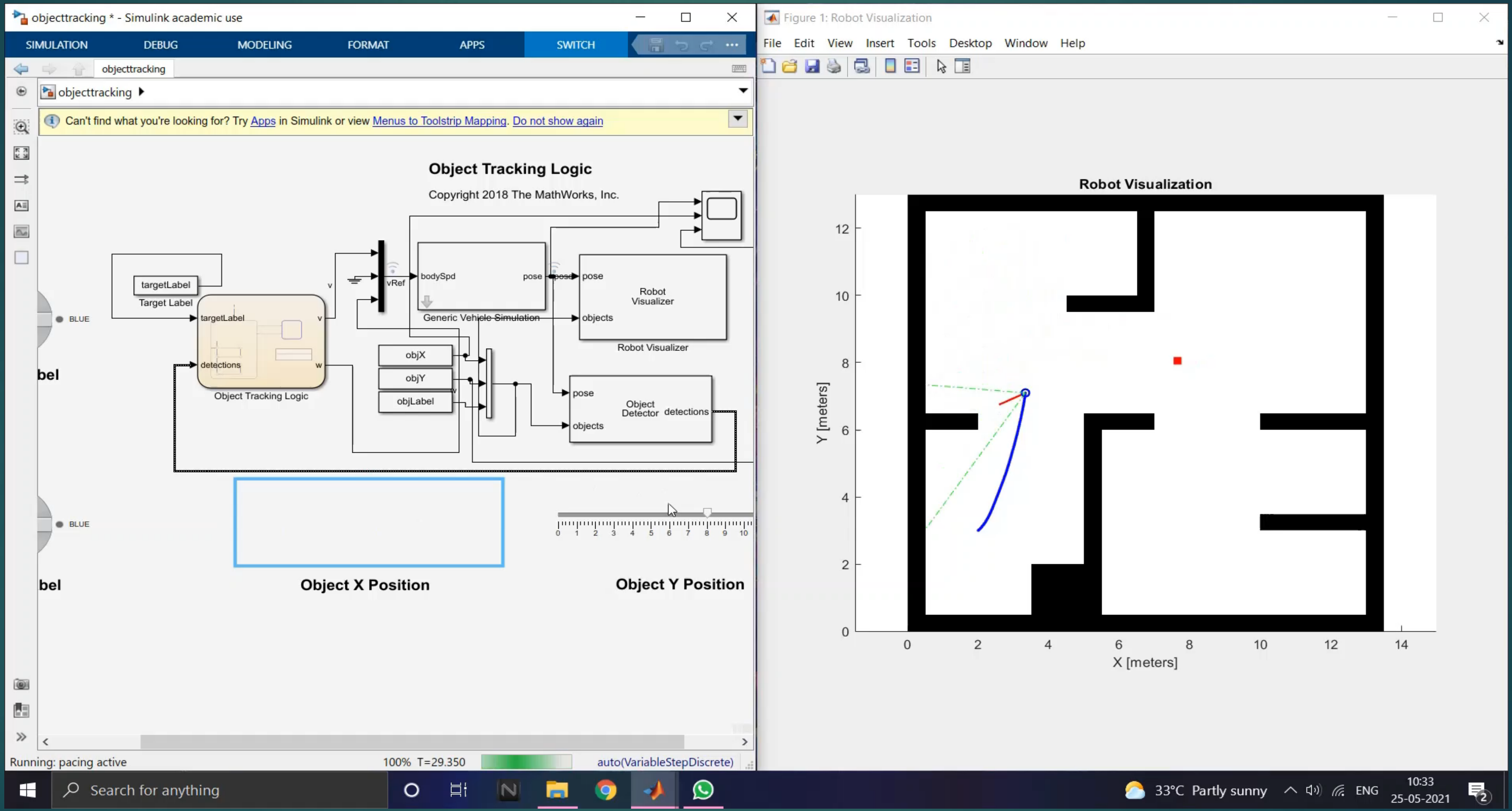
# Object Tracking and Obstacle Avoidance

The ultrasonic sensor scans for the object that is to be tracked and locates it and the control signal is sent to the motor driver. The figure shows the simulation of the object tracking. The simulation results have been obtained from MATLAB 2020b with a mobile robotics simulation toolbox and mobile robot training toolbox.

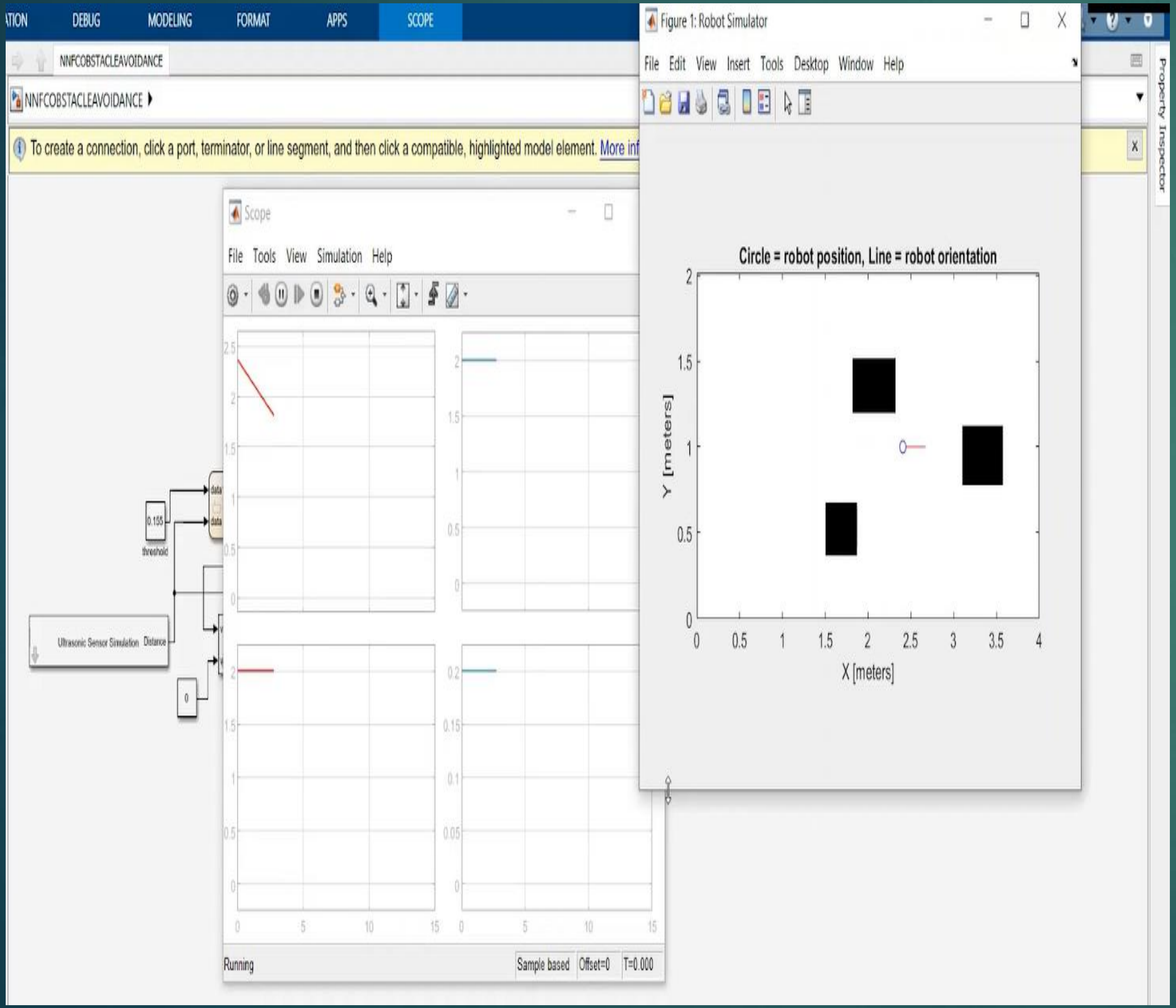


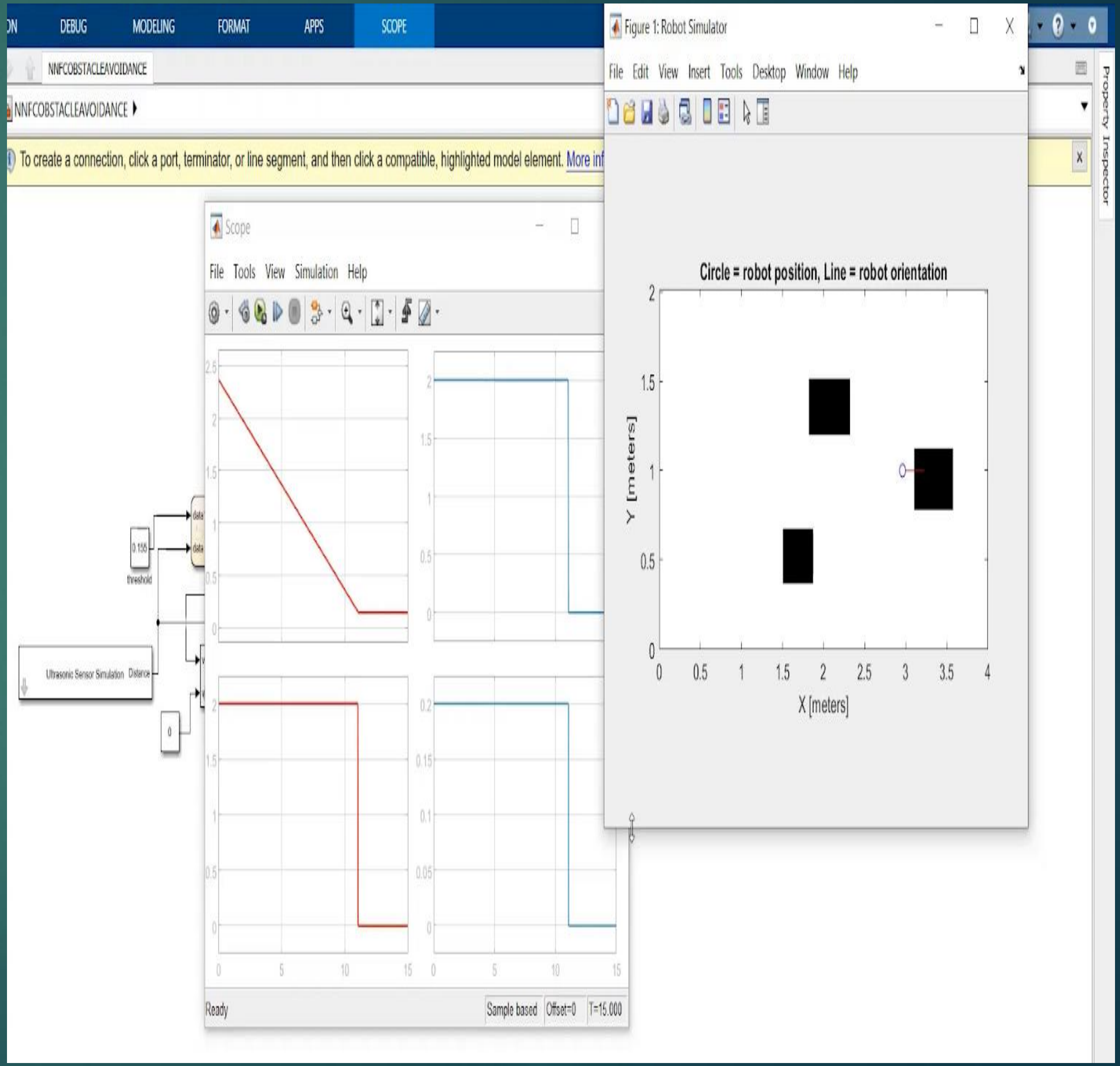



# VIDEO DEMONSTRATION

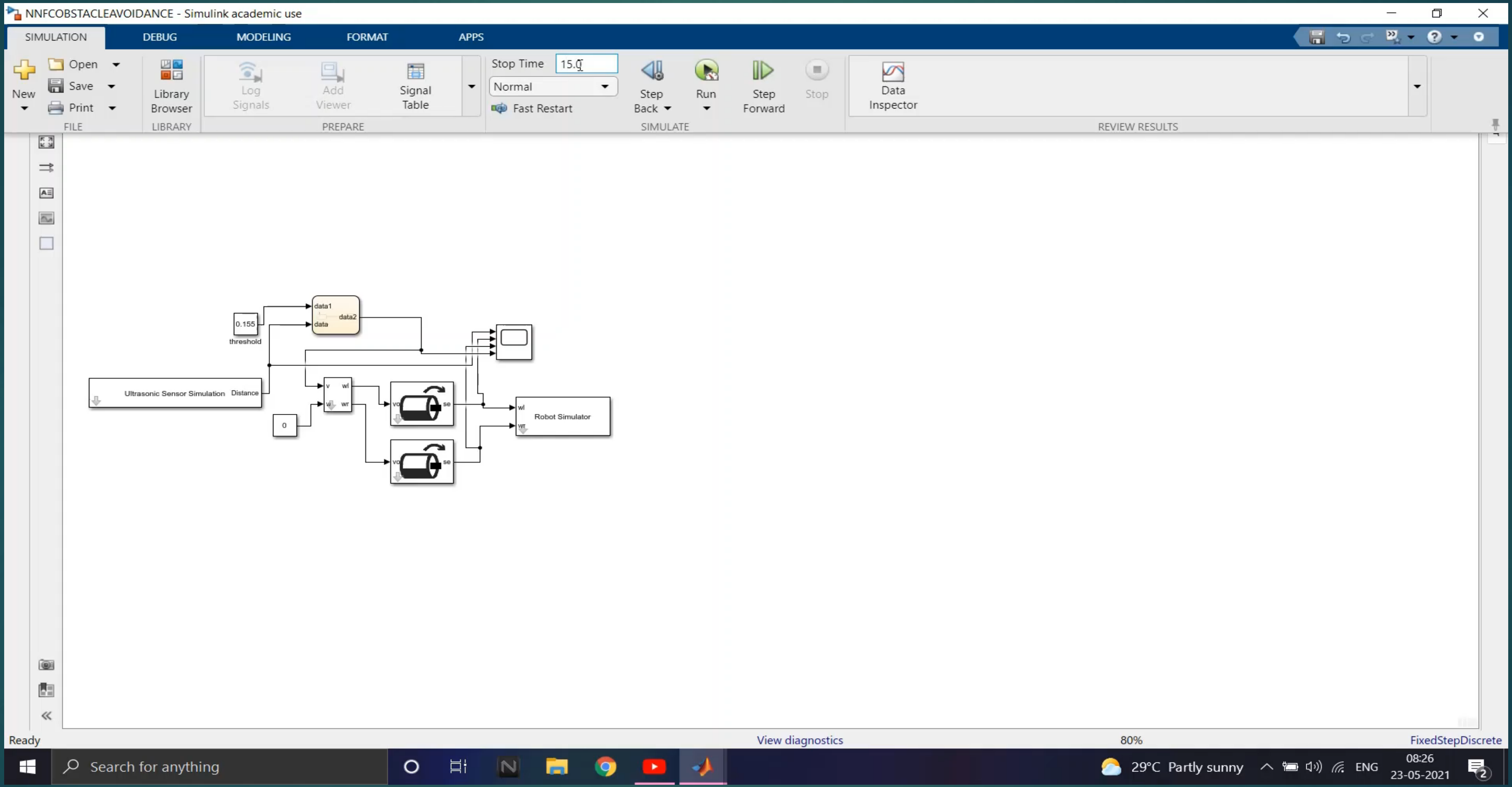


In the event of path planning, the system has to avoid collision therefore the robot has obstacle avoidance system. The ultrasonic sensor scans for the incoming obstacle and sends the control signal to motor driver to turn away from the obstacle to avoid collision. The figure 10a. shows the system traversing through the environment, and figure 10b. shows the system halting to avoid collision.

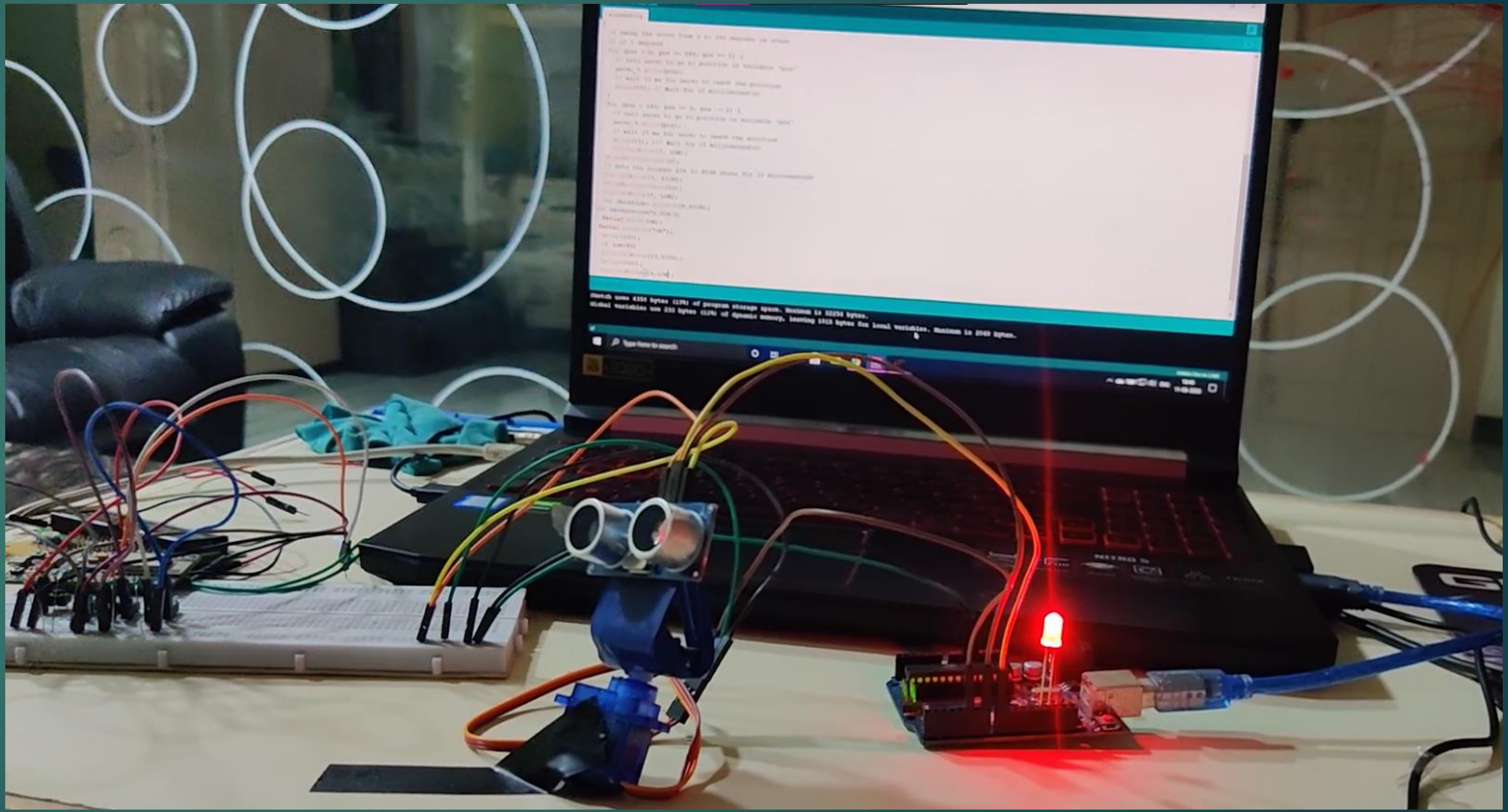


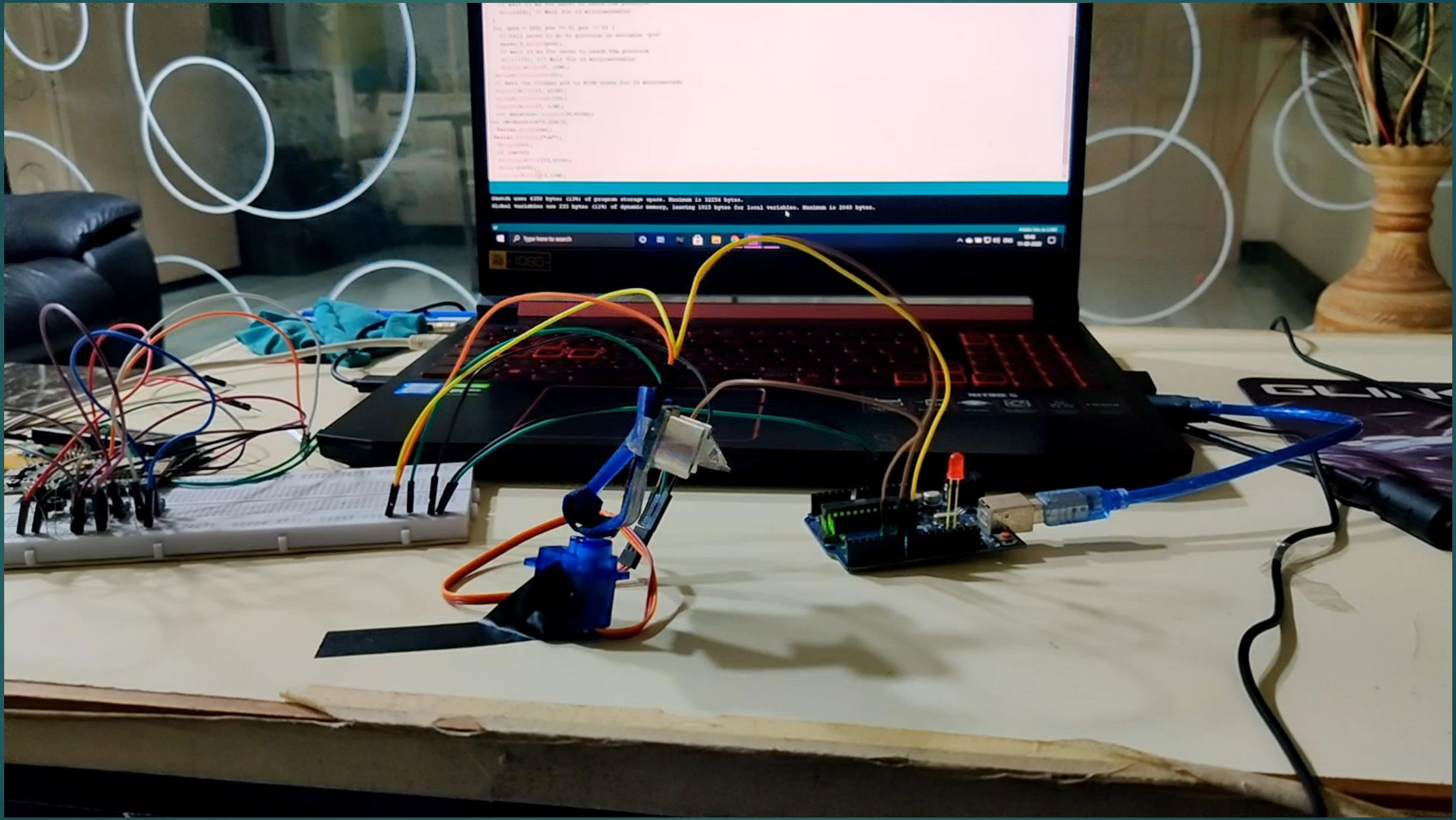
# VIDEO DEMONSTRATION



The figure 11 shows the obstacle avoidance system in action, when the ultrasonic sensor senses an obstacle in front, the red LED glows indicating the brakes have been applied. This avoids the system to get damaged from collision of an obstacle or a dynamic object thus saving time and cost to repair.



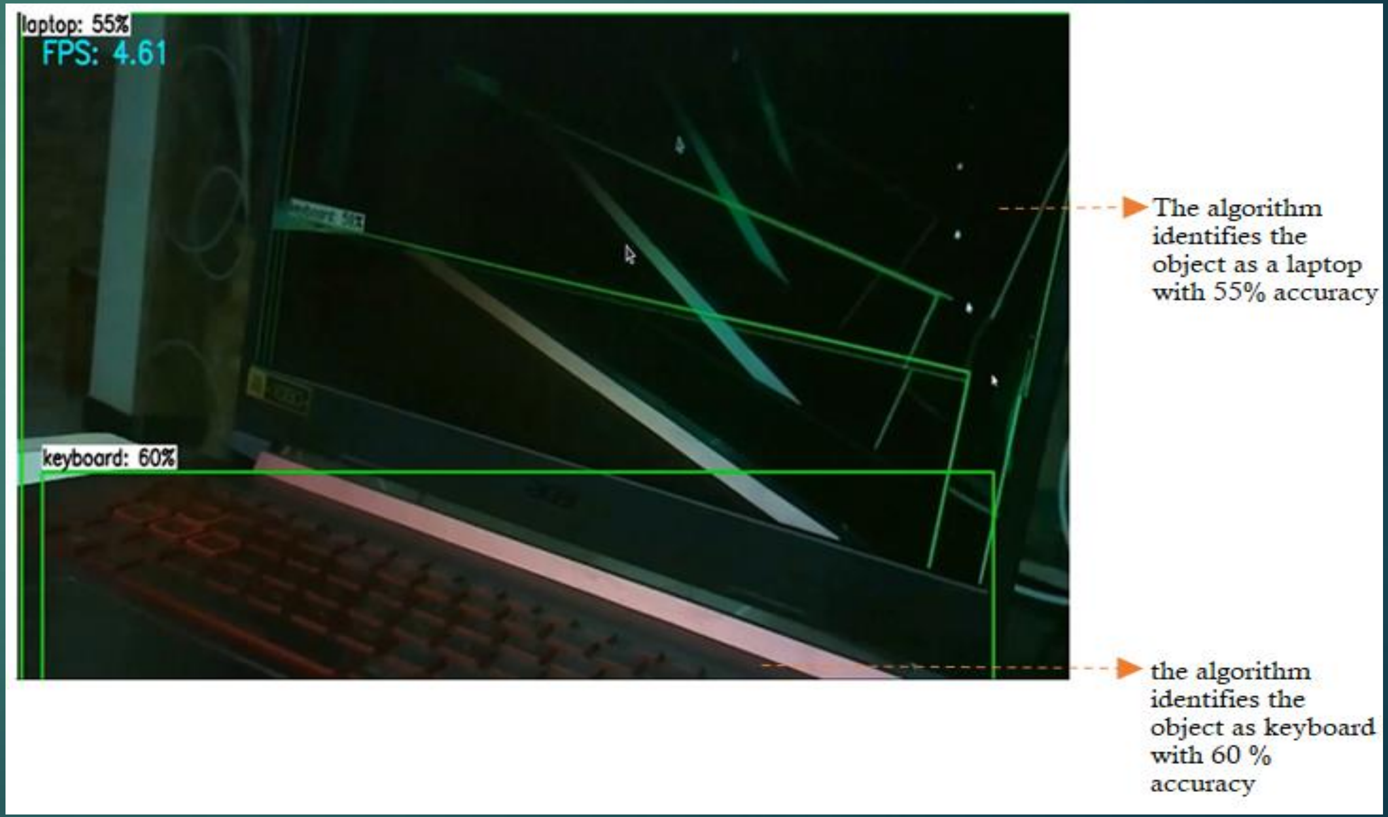
# VIDEO DEMONSTRATION



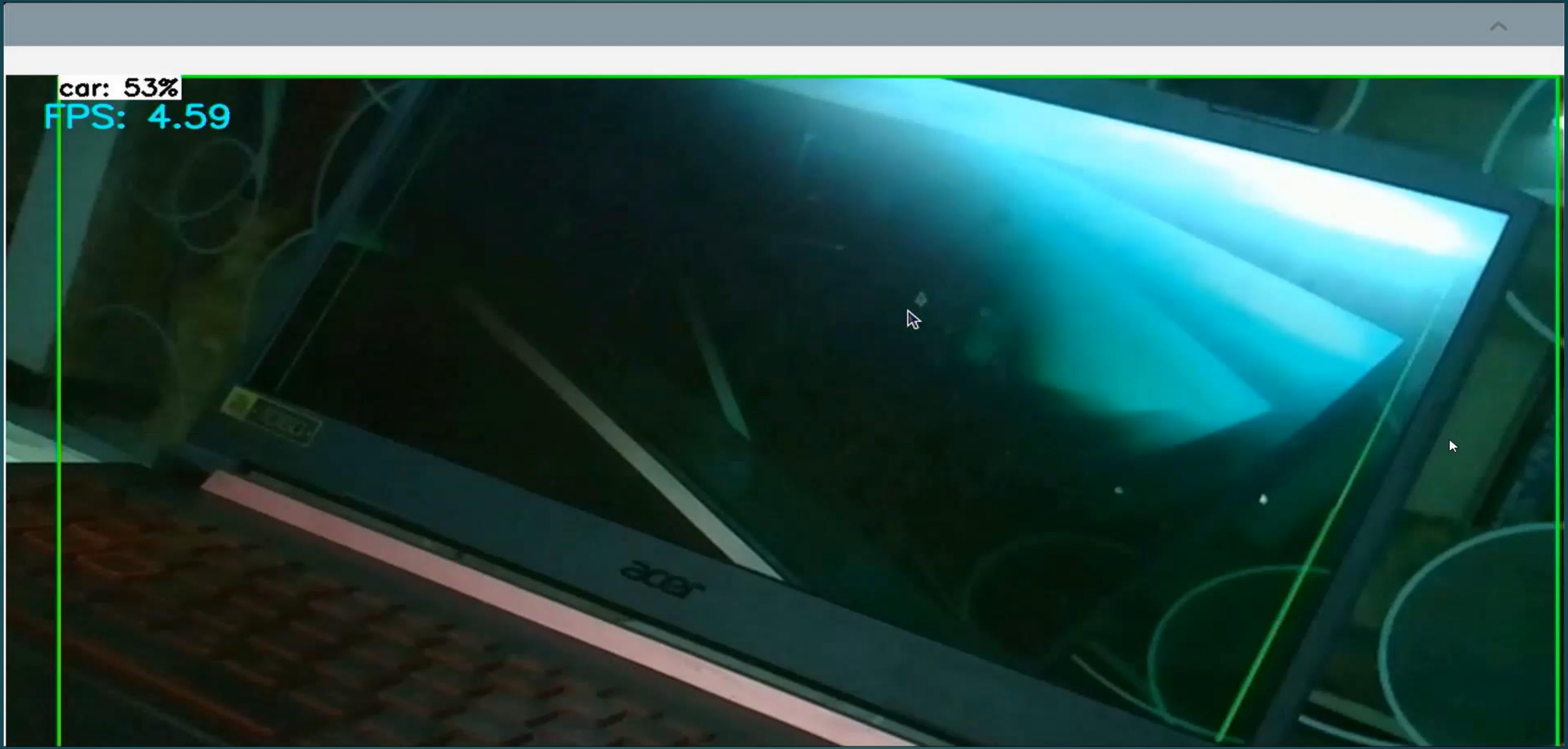
# Object Detection

The object detection capability of the rover enhances the way it interacts with the environment. It generates the small segments in the input. Feature extraction is carried out for each segmented rectangular area to predict if the rectangle contains a valid object. Boxes that overlap are combined into one bounding rectangle. The pre-trained models in their framework which are mentioned as Model Zoo, includes a collection of pre-trained models trained on various datasets such as the

- COCO (Common Objects in Context) dataset,
- the KITTI dataset,
- and the Open Images Dataset.



# VIDEO DEMONSTRATION



# Future Enhancements

The following enhancements will be made to the robot for more efficiency and to handle increased workload:

- Satellite communication instead of internet
- Change the base of the robot from Raspberry Pi to JETSON NANO to handle more tasks effectively.
- Addition of an end effector like robotic arm to make the robot interact with the environment more effectively.
- To increase the scientific test from life compatibility estimation to plant trees and observe their growth using machine learning.

# Conclusion

A land of varied geographic and topographic terrain needs equal interaction with the advancement of technology for the sustainable state of the nature. This kind of rover design or the modified prototype helps to gain a updated and a dynamic knowledge of the undiscovered terrains and acts as a boom to the many sectors in relation with adventure sporting and tourism. It also has a great highlight of data and the report when stored and observed in a database domain. Considering future prospects, the project  has been integrated with a  new visionary approach and the outcomes are achieved.

# References


[1] Kopetz, Hermann. "Internet of things." Real-time systems. Springer US, 2011. 205-249.

[2] S. Yerazunis, J. Krajewski, "Control of an autonomous rover", IEEE Automatic Computation and Control, Milwaukee, pp. 9-15

[3] Matthies, , J. Papon, R. Detry “Task-oriented grasping and geometric scene understanding” International Conference on Intelligent Robots and Systems (IEEE, 2017), pp. 3329-3345

[4] Y. Yu and G.Li , “contrast learning for salient object detection,” CVPR, 2016.

[5] Yukumoto and Osamu, "Agricultural machinery current status and problems", AMA-Agricultural Mechanization in Asia Africa and geographical area, vol. 1, 2014.

[6] Yoshikawa T.Manipulability of robotic mechanisms Int. J. Robot. Res., 4 (2) (1985), pp. 5-7.

[7] M. Lavalle, “Rapidly-exploring random trees”, engineering science Department, Iowa State University, Ames, USA, 1998.

[8] Y. L. Maske, S. V. Patil, S. Deshmukh, “Simulating stair climbing robot with rocker bogie Mechanism ” ,International Journal of Innovative Research in Technology


THANK YOU