UMIC RECRUITMENT PROJECT COMPLETION TEAM - 4

OUR TEAM

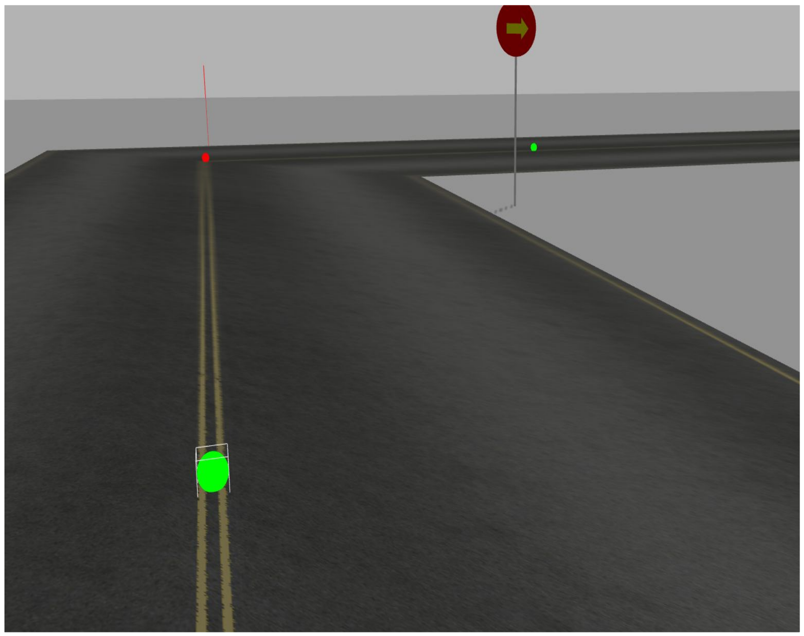
- PRATHMESH BELE (LEADER)
- ARYAN KOLAPKAR
- BISHOP PRAKASH
- SAI GANGADHAR
- PRASANN VISWANATHAN

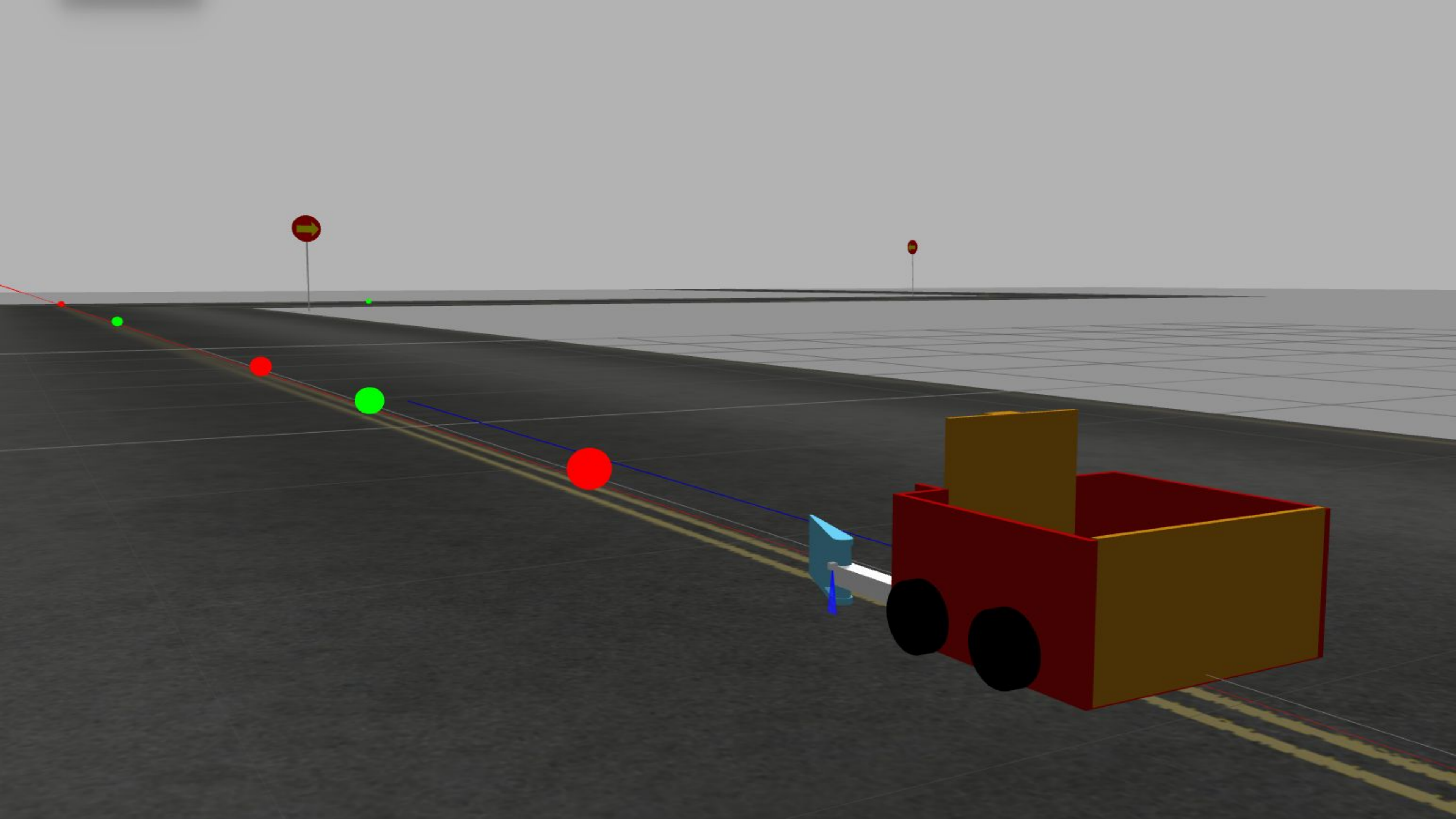
MENTORS

- YASH BHAVSAR
- HARSHIL NAIK

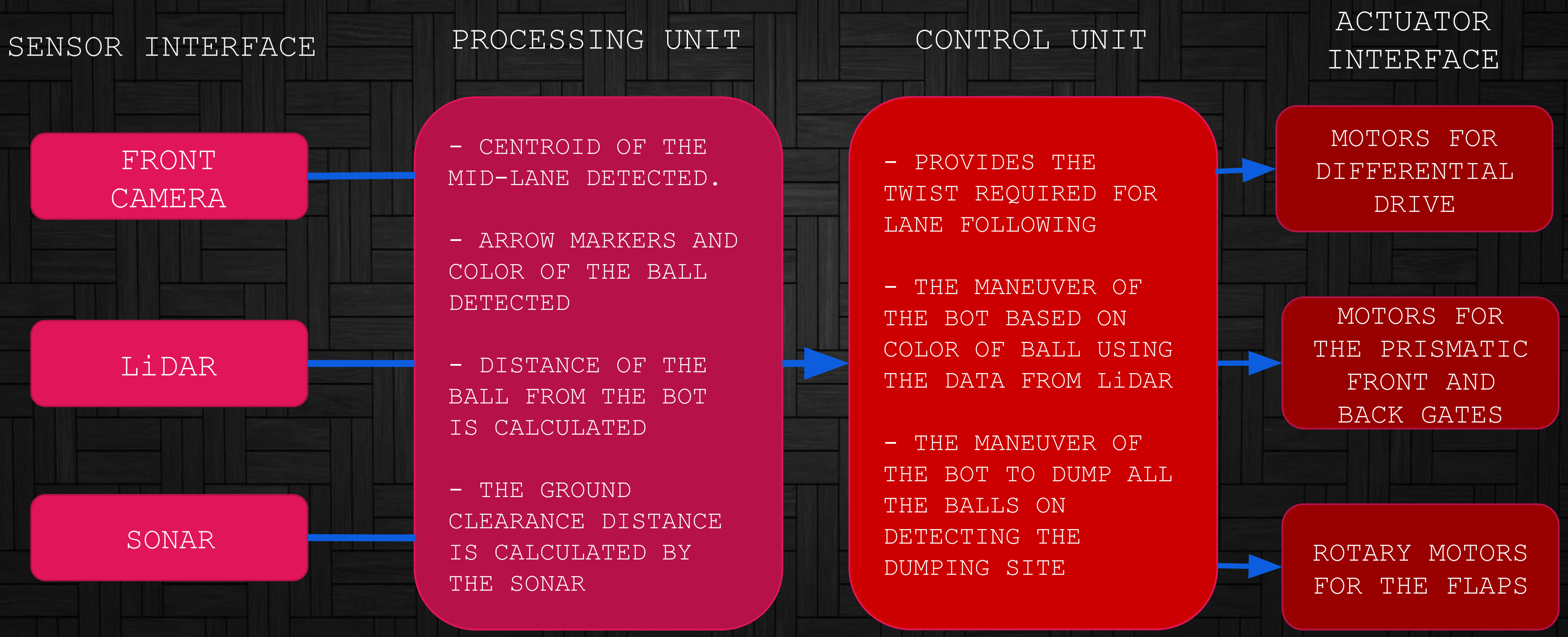
PROBLEM STATEMENT

Design an autonomous bot which can navigate in the environment according to the markers placed. The bot needs to collect the green balls in its path and avoid the red ones, at the end of the simulation it must dump the green balls at a dumping site.





FLOW CHART OF THE WORKING OF BOT

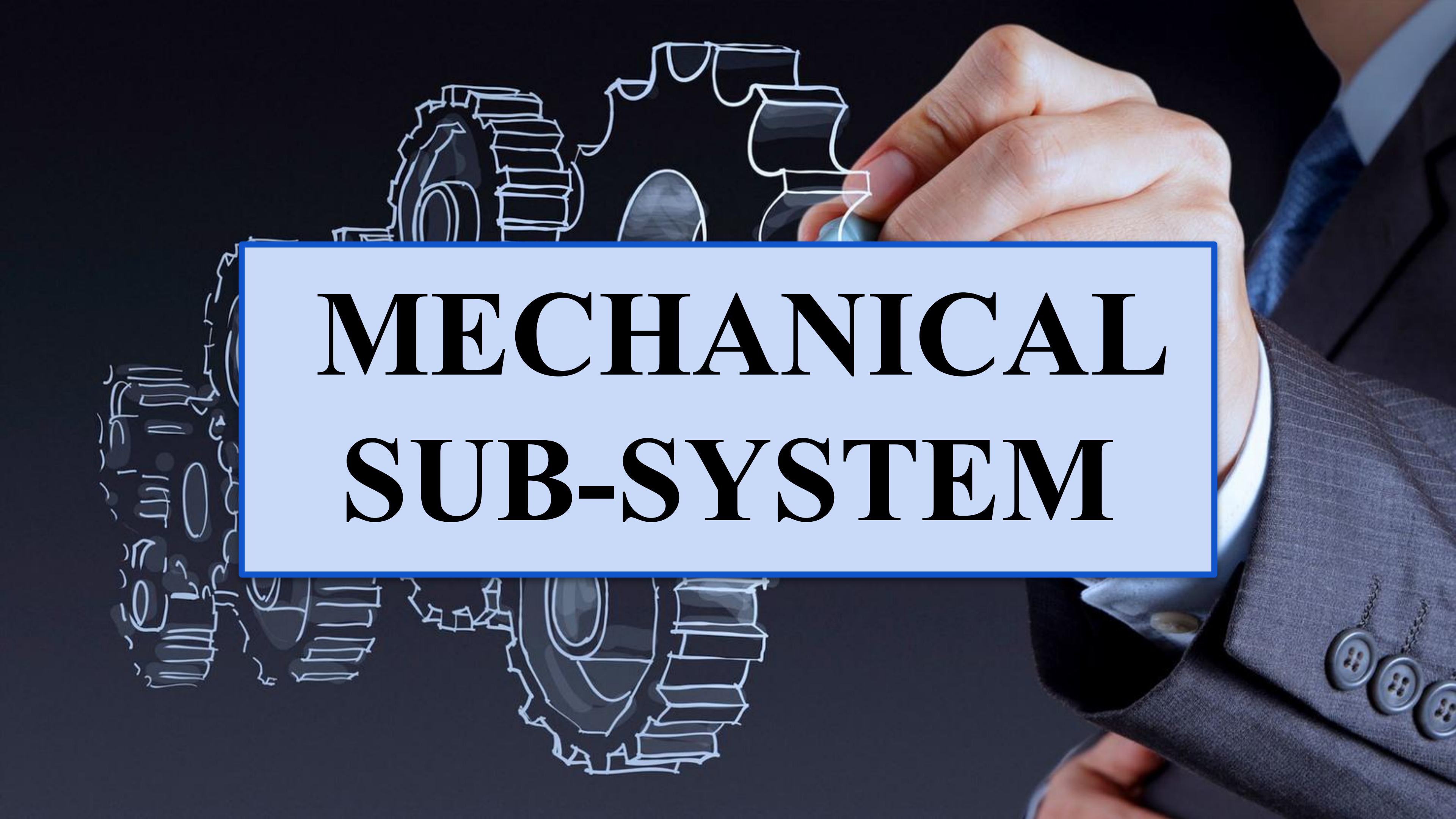


SUB-SYSTEMS AND CONTRIBUTORS

- MECHANICAL AND DESIGN SUBSYSTEM : Bishop
- PERCEPTION SUBSYSTEM : Aryan, Prasann, Prathmesh
- CONTROLS SUBSYSTEM : Gangadhar, Prasann, Aryan
- GAZEBO ENVIRONMENT SUBSYSTEM : Gangadhar, Prathmesh

WORK INVOLVED IN EACH SUB-SYSTEM

MECHANICAL SUB-SYSTEM



BRIEF DESCRIPTION OF OUR APPROACH

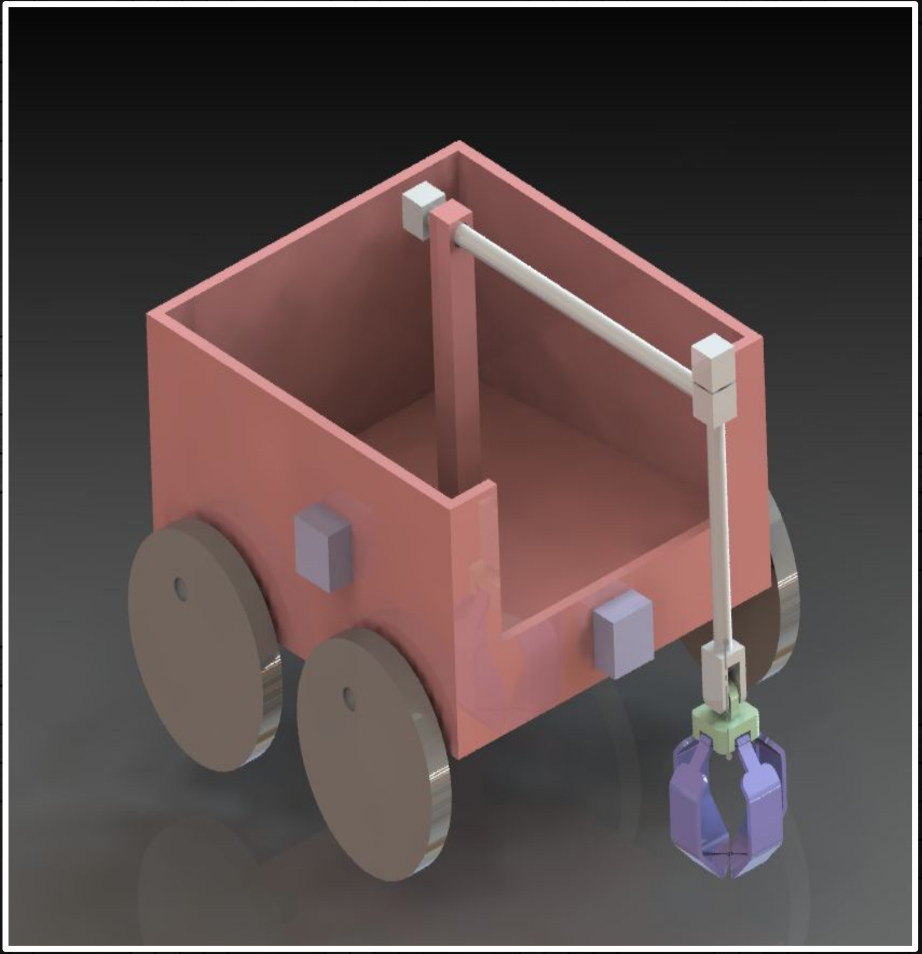
- At first we considered building a bot which would have a ground clearance more than the diameter of the balls and it was supposed to pass over the red balls to avoid them, and pickup the green balls using a claw mechanism.

- This model of bot was extremely unstable in Gazebo and we were unable to add LIDAR in this bot at the level of the ball. We even thought of adding a slanted LIDAR but we could have stuck in calculations of angle and position of LIDAR. An up and down sliding LIDAR would have been an option but it was going to add more links to the base of (already unstable) bot and require more controls.

- Therefore, this idea of bot was scrapped and a new design of bot was taken, which is described in following slides.

THE OLD BOT

- FOUR WHEEL DIFFERENTIAL DRIVE
- 12 cm GROUND CLEARANCE TO PASS OVER THE RED BALLS
- TWO EXTENSION POLES FOR PLACEMENT OF THE CLAW
- 4 FINGER CLAW TO PICK UP THE GREEN BALLS
- PRISMATIC REAR WALL USING FOR DROPPING OUT OF THE BALLS
- 3 CAMERAS FOR LANE DETECTION AND TURNING
- LIDAR FOR CALCULATING BALL DISTANCE
- ULTRASONIC SENSOR FOR DETECTING THE DUMPING SITE
- DC MOTORS FOR OPERATION OF JOINTS WITH APPROPRIATE SPEED CONTROLLERS
- R Pi 4 FOR PROCESSING

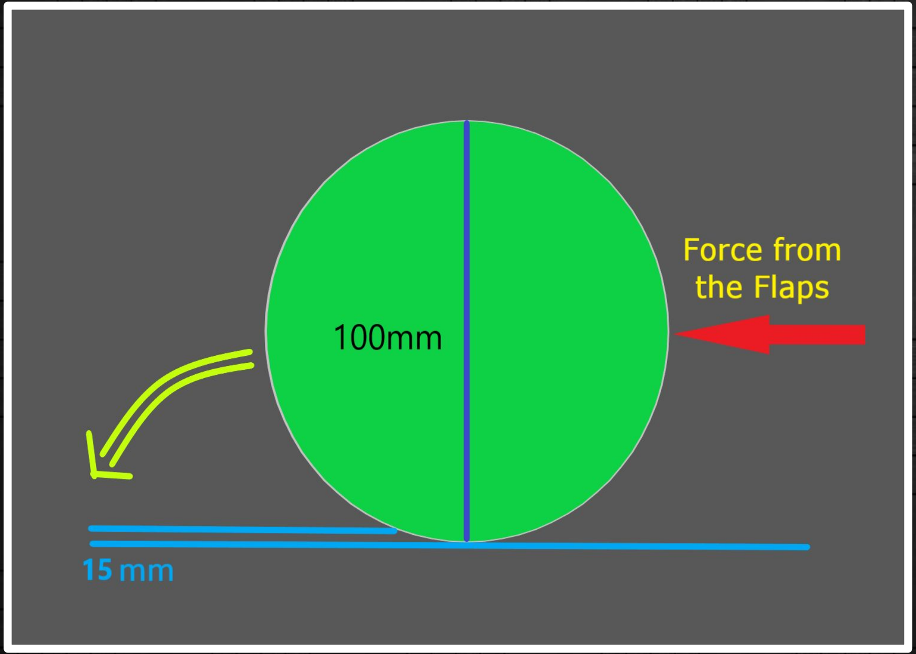


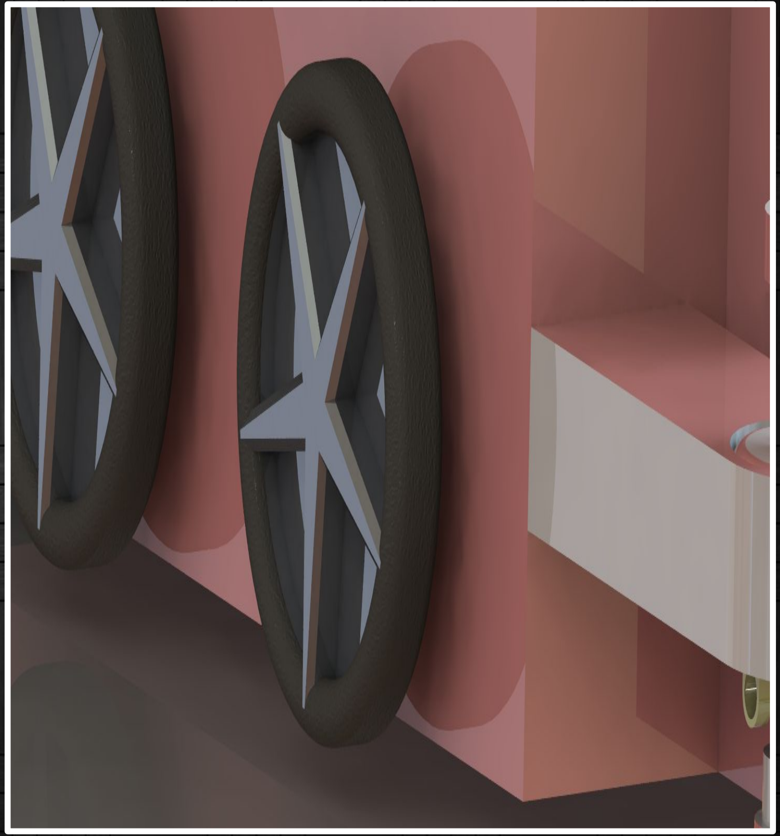
THE BOT

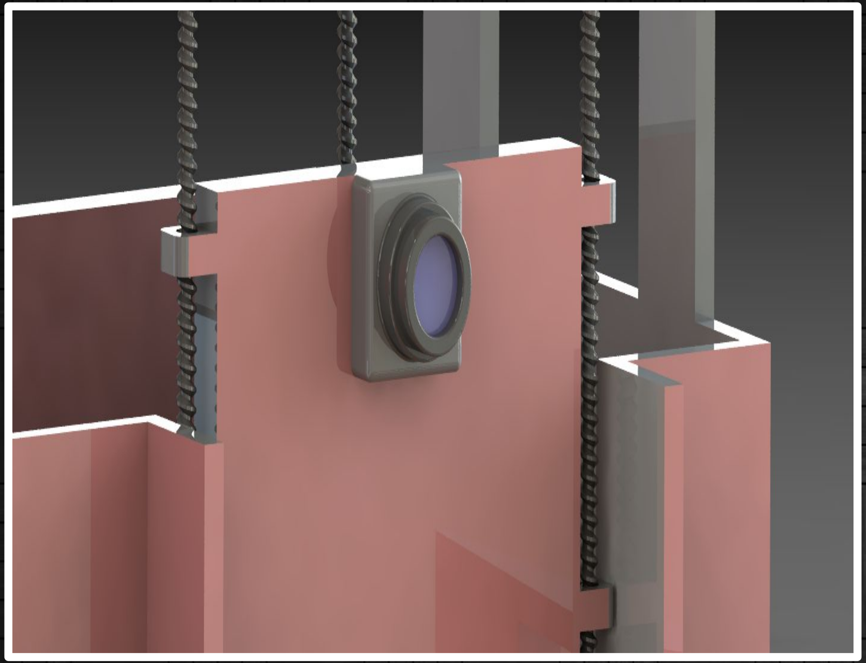
- FOUR WHEEL DIFFERENTIAL DRIVE
- PRISMATIC FRONT AND REAR WALLS USING LEAD SCREW MECHANISM FOR IN AND OUT OF THE BALLS
- TWO FLAP SYSTEM TO PUSH GREEN BALLS IN AND RED BALLS AWAY
- 1 CAMERA FOR LANE DETECTION AND TURNING
- LIDAR FOR CALCULATING BALL DISTANCE
- ULTRASONIC SENSOR FOR DETECTING THE DUMPING SITE
- 10 DC MOTORS FOR OPERATION OF JOINTS WITH APPROPRIATE SPEED CONTROLLERS
- R Pi 4 FOR PROCESSING
- BOT MASS 5 kg Approx

THE CONCEPT

- 10 mm GROUND CLEARANCE
- 5 mm BASE FLOOR THICKNESS

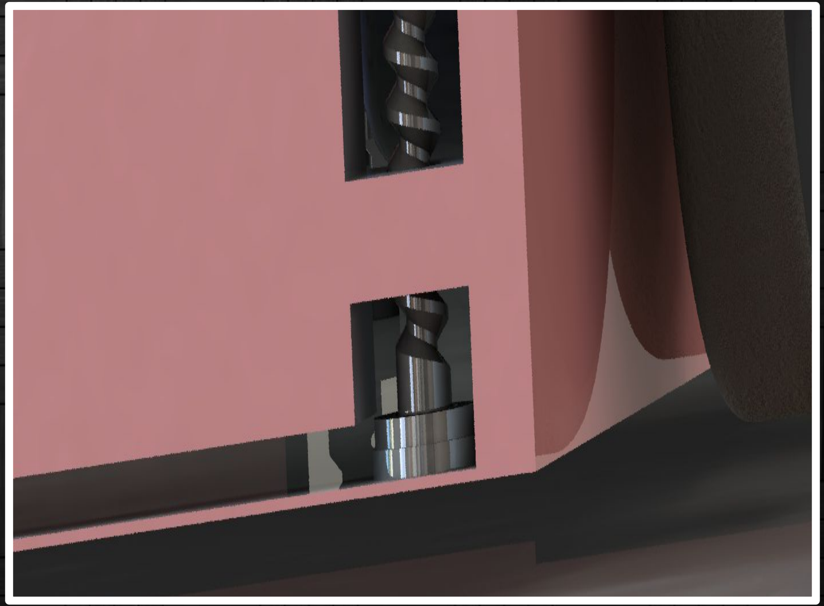




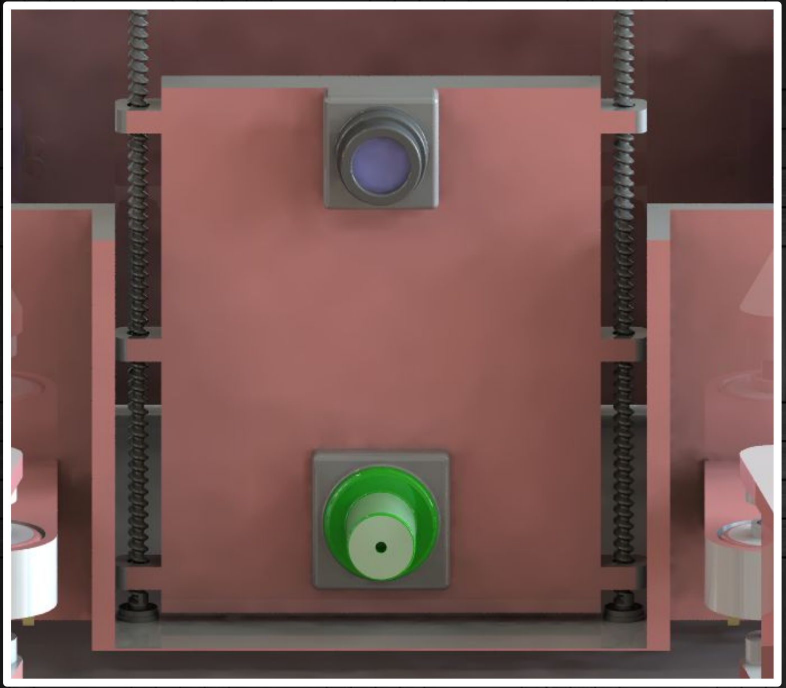


← LEAD SCREW MECHANISM
(Used for both prismatic doors and implemented using long shaft motors)

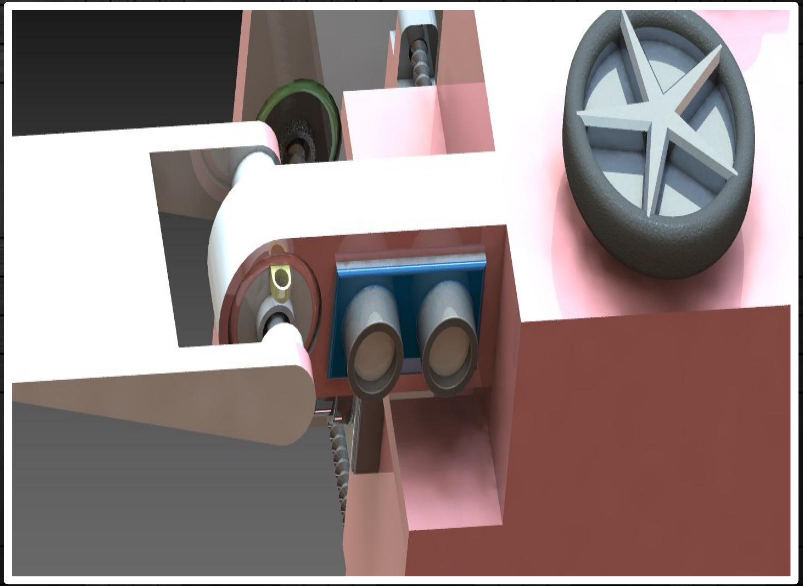
← FRONT PRISMATIC DOOR



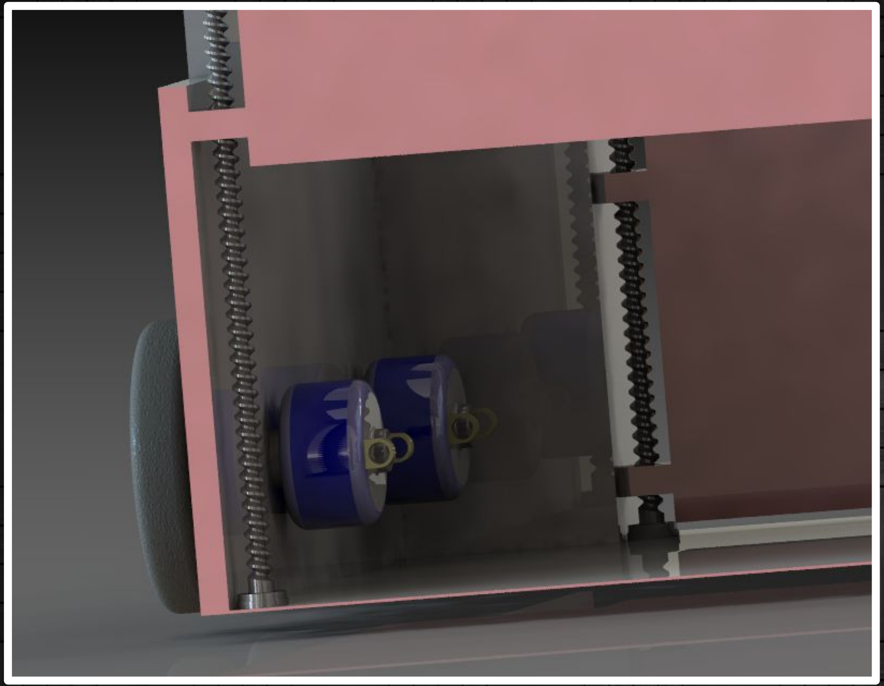
BEARINGS →
(To hold the long shafts in place and provide smooth rotation)



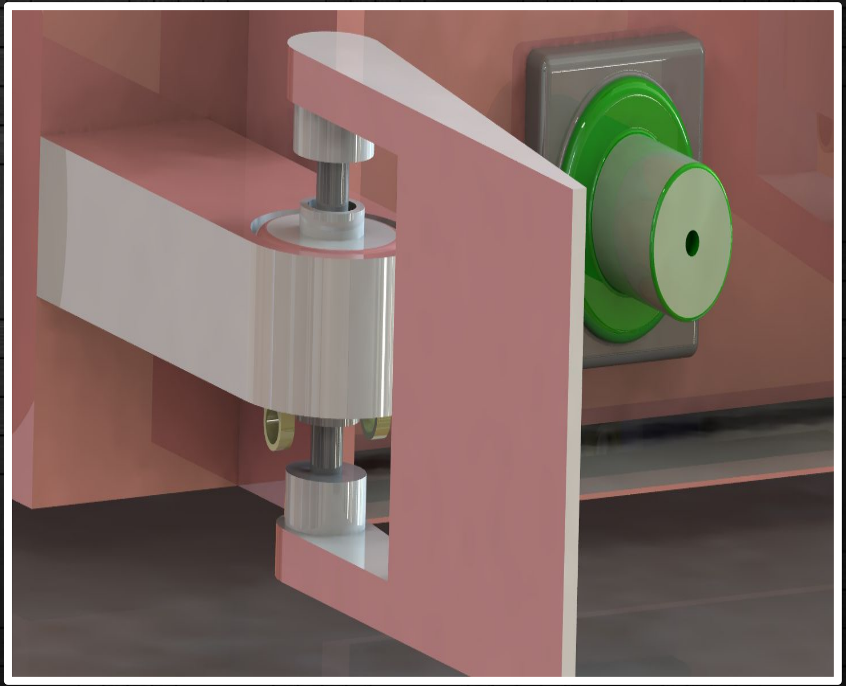
← FRONT CAMERA
(For lane and sign Detection)
← LIDAR
(For Measuring Ball distance)



ULTRASONIC SENSOR →
(For detection of dump-site by depth measurements)



← WHEEL MOTORS
(Would be reduced to two considering differential drive and gears)



FLAP MOTORS →
(Double shaft motors)

By default the bot would move with both the flaps in open position.

If the front camera detects a green ball

- Bot will move towards the ball upto a distance of 13cm.

- Camera feed and LIDAR will be paused

- Front wall would rise (18cm)

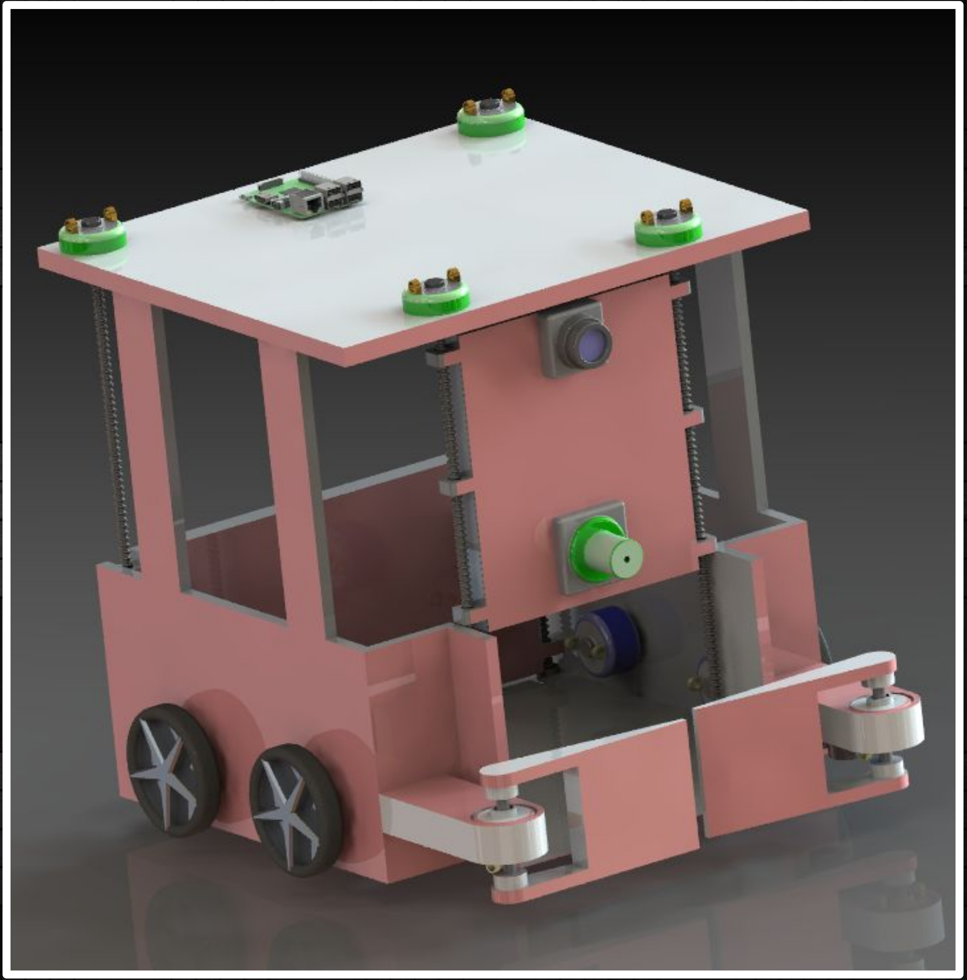
- The flaps will push the green ball inside

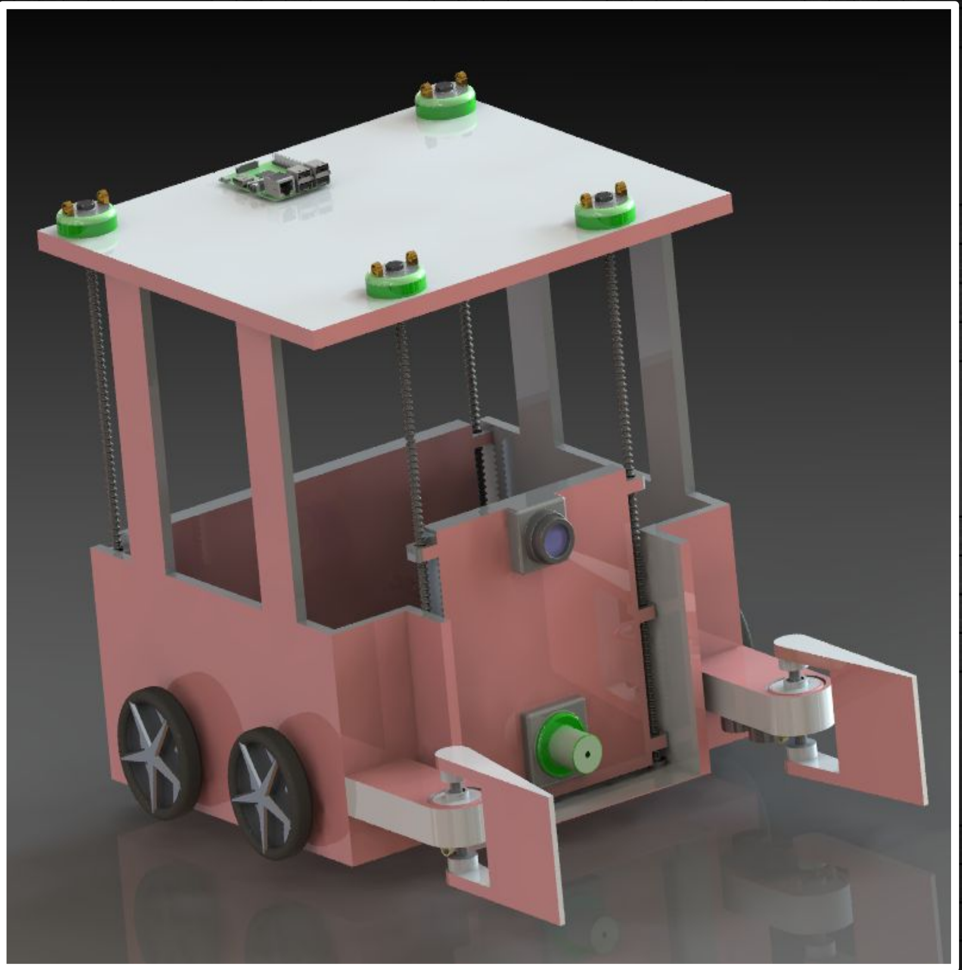
- Front wall would come down,

- The flaps would open up

- Camera feed and the LIDAR would resume

- Bot will start moving.





If the camera detects a **red ball**

- Bot would move towards it upto 25cm

- Camera feed and the LIDAR will be paused

- The flaps will get closed

- Bot would move 10 cms forward

- The flaps would open up, pushing the red balls away

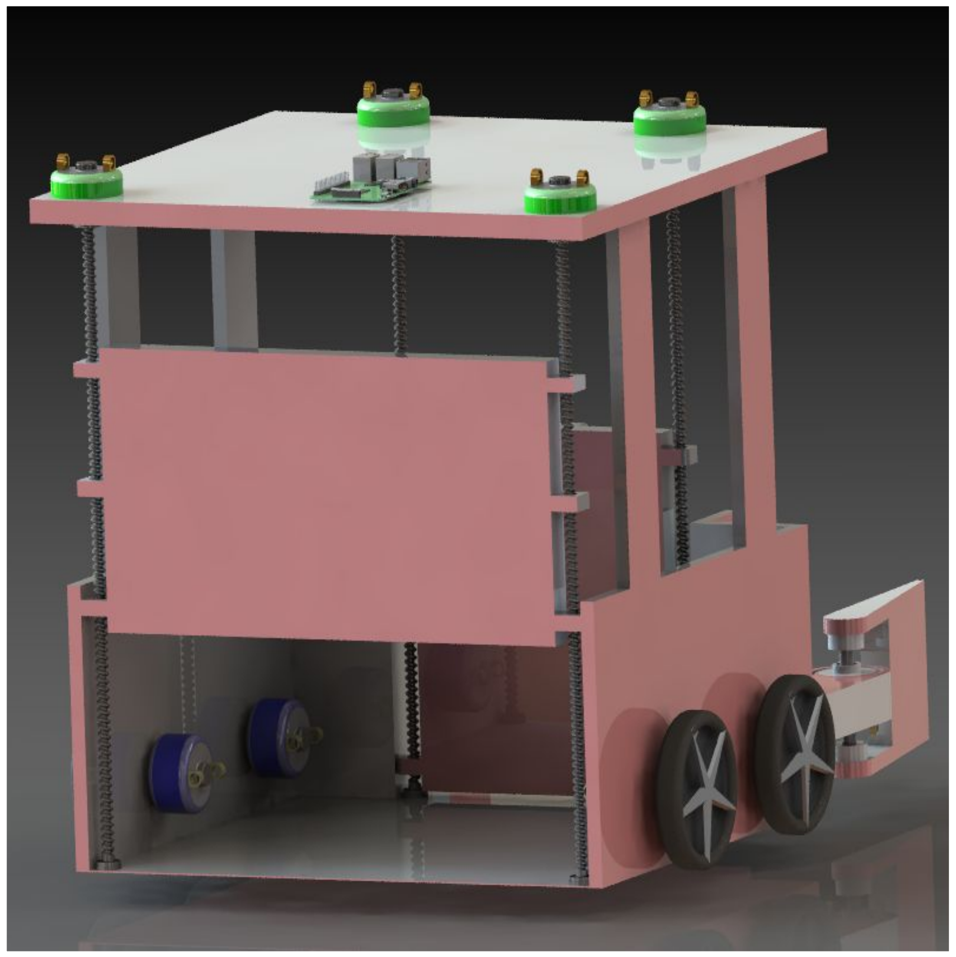
- The feed would resume

- The bot starts to move again.

THE DUMPING PROCESS

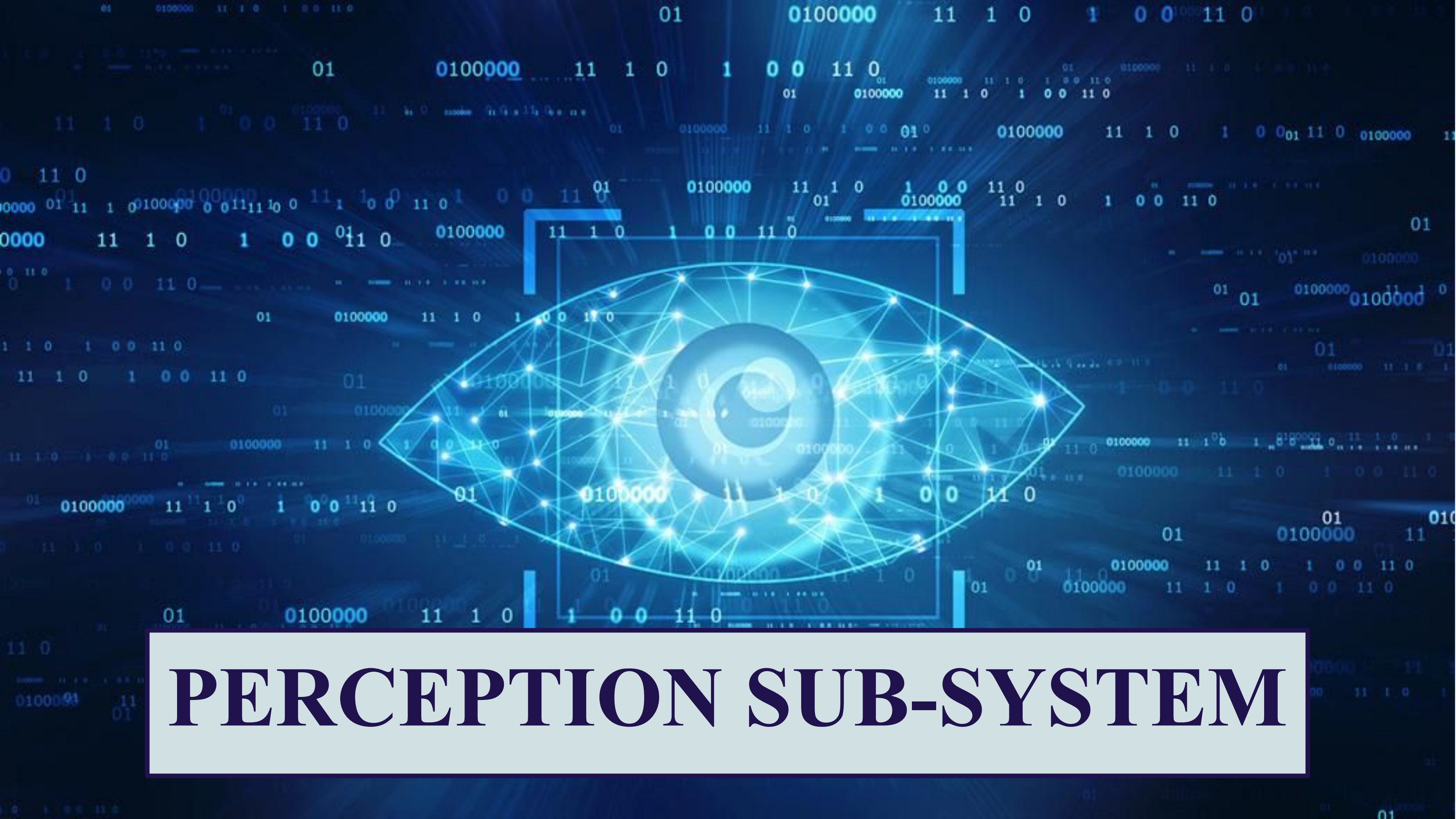
At the end of the simulation when we have collected all the green balls as per our path and we see the dump site, the bot would approach the dumpsite, locate it using the Ultrasonic sensor (Depth measurement) and rotate 180 degrees.

The back wall would rise up and we'll accelerate with max magnitude possible so that the balls fall out of the cart due to inertia. Apart from that we do have a backward slope in the base floor of 2 degrees.



BOT COST ESTIMATION

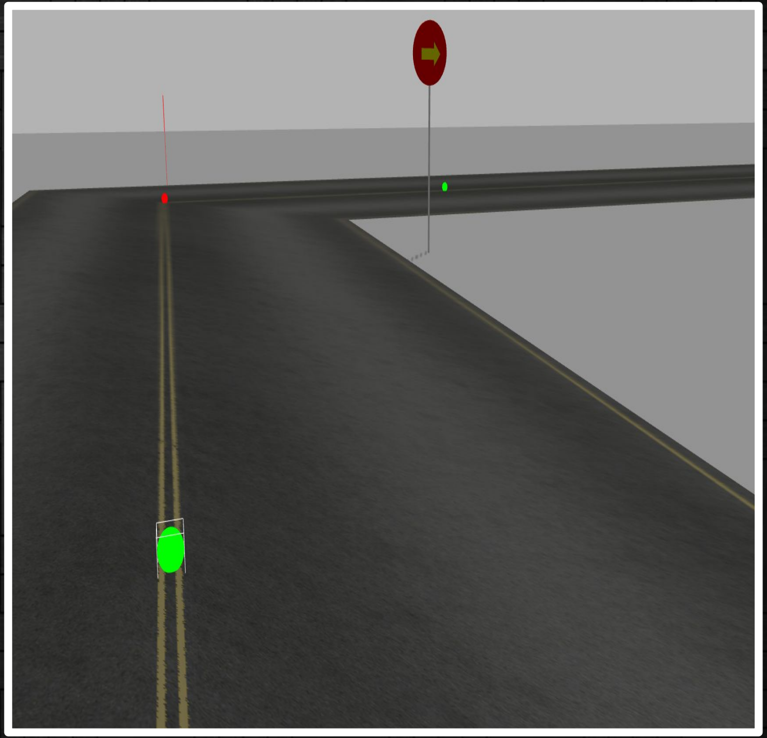
PART	DESCRIPTION	QTY.	PRICE PER UNIT	PRICE(₹)
BODY	RAW MATERIALS AND FABRICATION	1	2,000	2,000
LEAD SCREW	FOR PRISMATIC DOORS	4	250	1,000
WHEELS	HIGH STRENGTH PLASTIC WHEELS	4	125	500
WIRING & CONNECTIONS	FITTING AND CONNECTION OF PARTS	1	500	500
BEARINGS	TO HOLD THE LEAD SCREWS	4	50	200
ULTRASONIC SENSOR	ROBOTICS (MEDIUM RANGE)	1	300	300
CAMERA	HIGH QUALITY ROBOTICS CAMERA	1	1,000	1,000
LIDAR	TF 02 MEDIUM RANGE ROBOTICS LIDAR	1	3,000	3,000
MOTORS AND CONTROLLERS	HIGH TORQUE MOTORS AND COMPATIBLE MOTOR CONTROLLERS TO OPERATE THEM	8	500	4,000
RASPBERRY PI 4	2GB/4GB MICROPROCESSOR	1	4,000	4,000
TOTAL PRICE				**₹16,500**



PERCEPTION SUB-SYSTEM

PROBLEMS SOLVED USING PERCEPTION

1. Detection of lane/middle line of the road so as to follow it
2. Detection and classification of sign boards and arrows beside the road.
3. Detection and classification of ball based on its color
4. Calculating the distance of ball form the bot
5. Detection of the dumping site

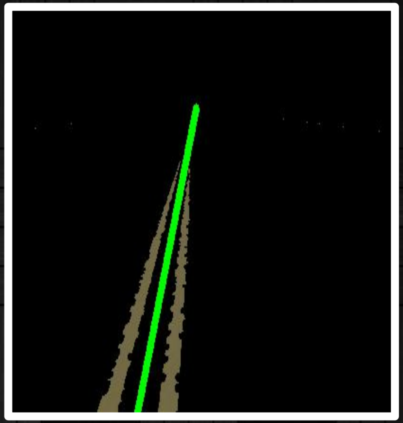


Line detection

<u>*Methods that failed-*</u>

1. Outer road lane detection - Wasn't as practical as the lanes were widely separated and one lane could easily go out of the frame.

2. Road centre-line detection - This worked better than above lane Detection, the output was an equation of line obtained from Hough line transform. However, it was easily affected by noise Which would create abrupt changes in the line equation.



<u>*SOLUTION-*</u> Represent the middle-line as the centroid of the mask.

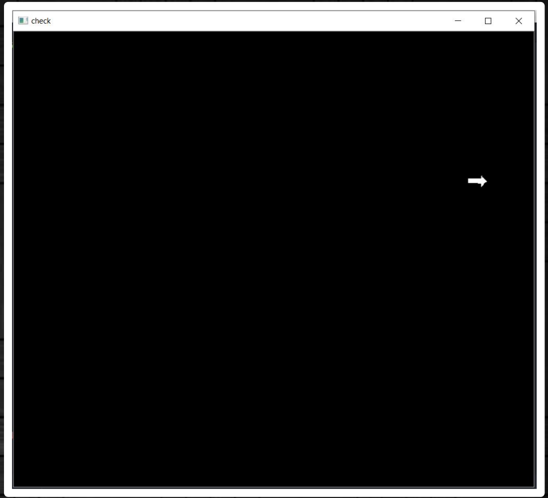
- Define a color range similar to the color of the lane
- Mask out that color from the image
- Find the centroid of all points of the lane using moments.

This eliminates abrupt changes due to noise, due to averaging from centroid

Arrow Detection and Classification

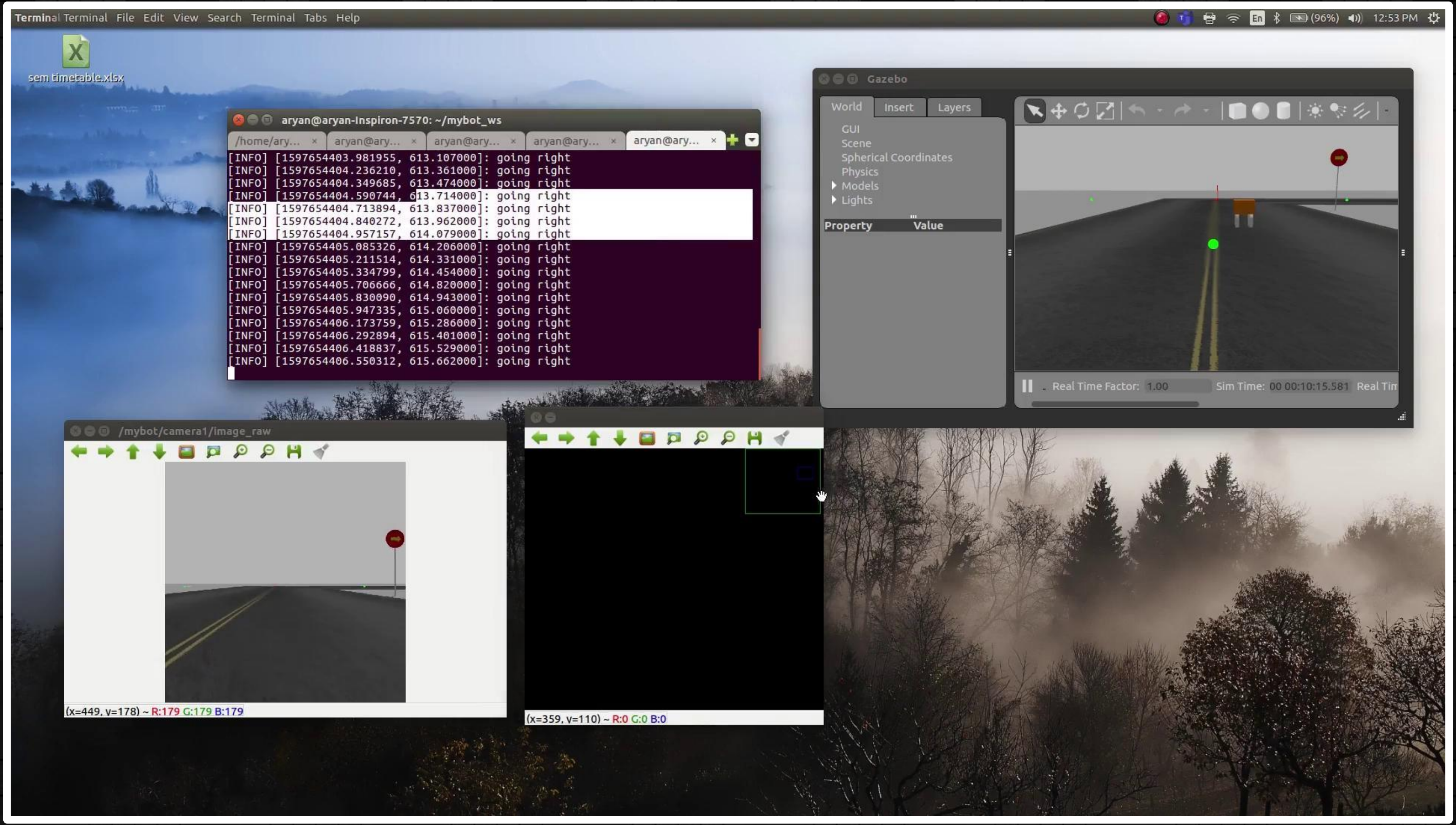
- A vital part of the autonomisation task was to detect and classify arrow markers, placed at intersections of roads
- Methods that failed :-
  - Number of contour points in a shape - Failed because balls were incorrectly being detected as arrows
  - Haar cascade - Failed because the cascade available online was inconsistent
  - Harris corners - There were corner elements in the environment apart from the arrows
  - Hough circles - Too sensitive and would detect circles where there weren't any.
- Solution:
  - Since the arrows are of different colour, we can mask out their colour hence preventing interference from other objects
  - A mask was applied, following which arrows were classified only if their size was greater than a fixed amount to ensure turns are taken at the right time

The making of the classifier



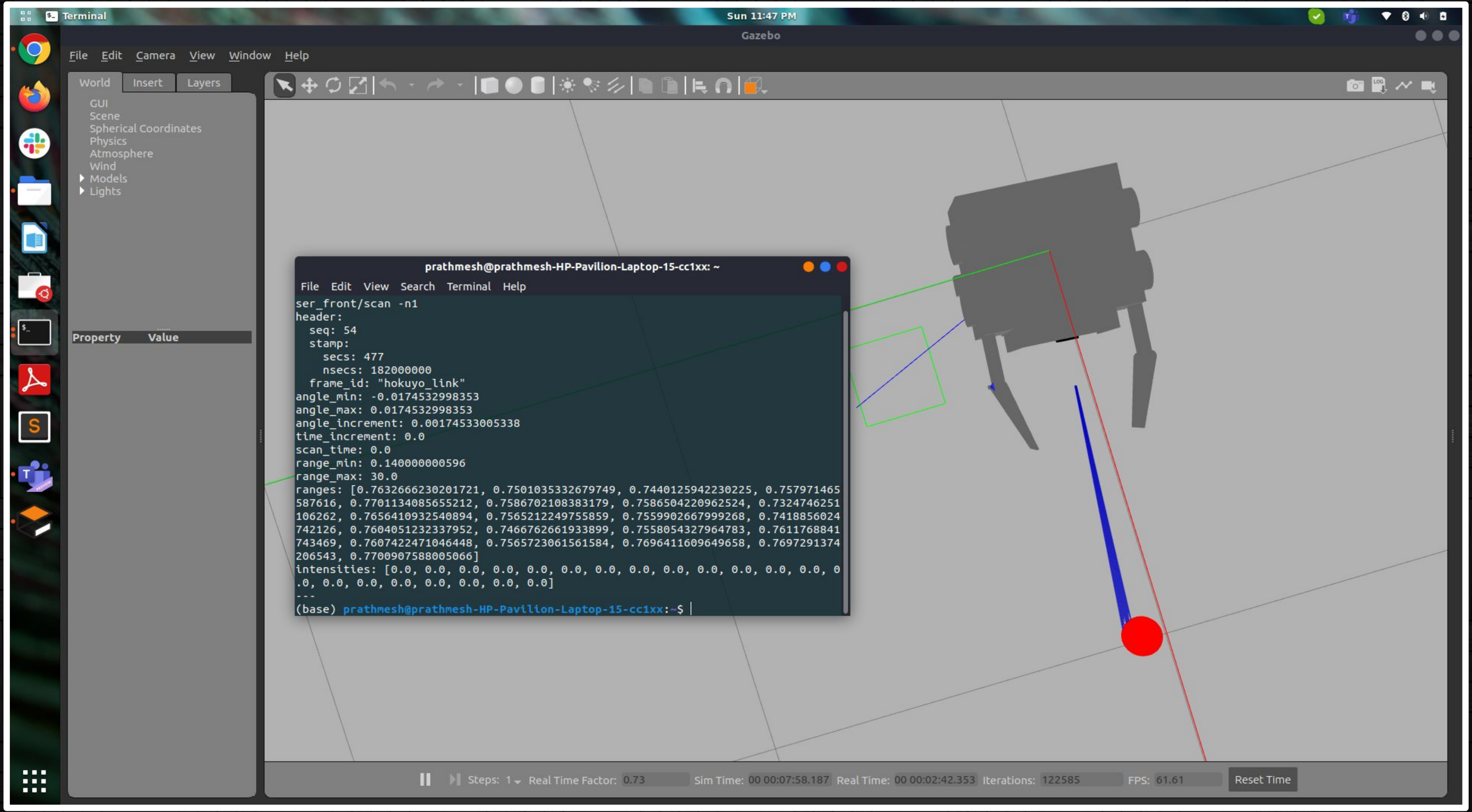

```
C:\Users\HOME\Desktop\UMIC shtuff\arrow_classification and steering>py basic_arrow.py
186.0
going right
230.0
going right
136.0
going right
37.5
going right
35.0
going right
```

Arrow Classification in Use



LiDAR

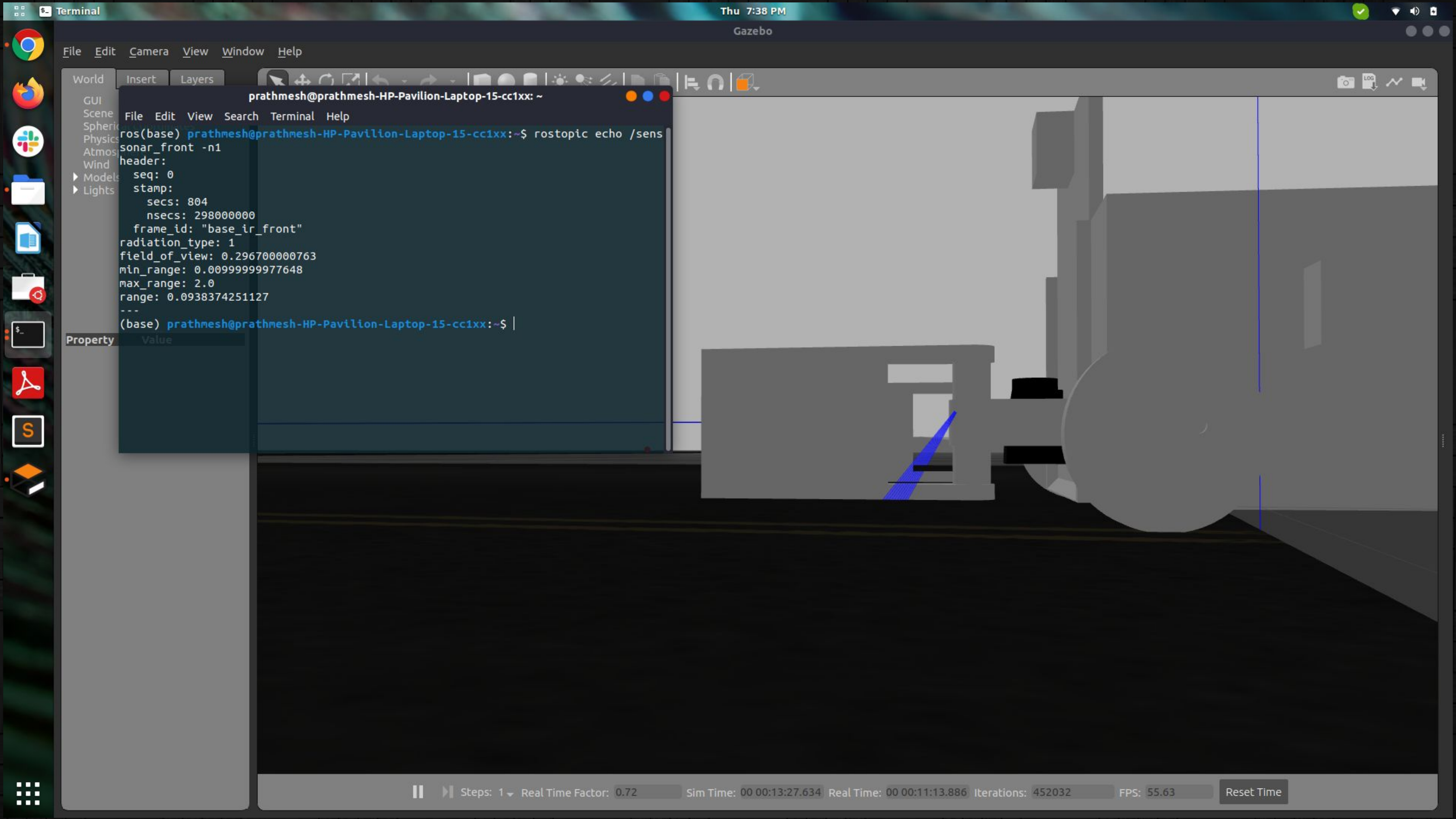
- The movement of flaps which are on the front of bot is dependent on the distance between the bot and the ball. Here we have used a Laser sensor to calculate the distance between these two.
- We used the Laser plugin in gazebo to add the sensor to the bot and attached it at the bottom of the front gate.
- The LiDAR is set to measure the distance between the bot and object in front in the range of 0.2m to 3m


Terminal
Sun 11:47 PM
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property Value
prathmesh@prathmesh-HP-Pavilion-Laptop-15-cc1xx: ~
File Edit View Search Terminal Help
ser_front/scan -n1
header:
  seq: 54
  stamp:
    secs: 477
    nsecs: 182000000
  frame_id: "hokuyo_link"
angle_min: -0.0174532998353
angle_max: 0.0174532998353
angle_increment: 0.00174533005338
time_increment: 0.0
scan_time: 0.0
range_min: 0.140000000596
range_max: 30.0
ranges: [0.7632666230201721, 0.7501035332679749, 0.7440125942230225, 0.757971465587616, 0.7701134085655212, 0.7586702108383179, 0.7586504220962524, 0.7324746251106262, 0.7656410932540894, 0.7565212249755859, 0.7559902667999268, 0.7418856024742126, 0.7604051232337952, 0.7466762661933899, 0.7558054327964783, 0.7611768841743469, 0.7607422471046448, 0.7565723061561584, 0.7696411609649658, 0.7697291374206543, 0.7700907588005066]
intensities: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]

(base) prathmesh@prathmesh-HP-Pavilion-Laptop-15-cc1xx:~$
Steps: 1 Real Time Factor: 0.73 Sim Time: 00 00:07:58.187 Real Time: 00 00:02:42.353 Iterations: 122585 FPS: 61.61 Reset Time

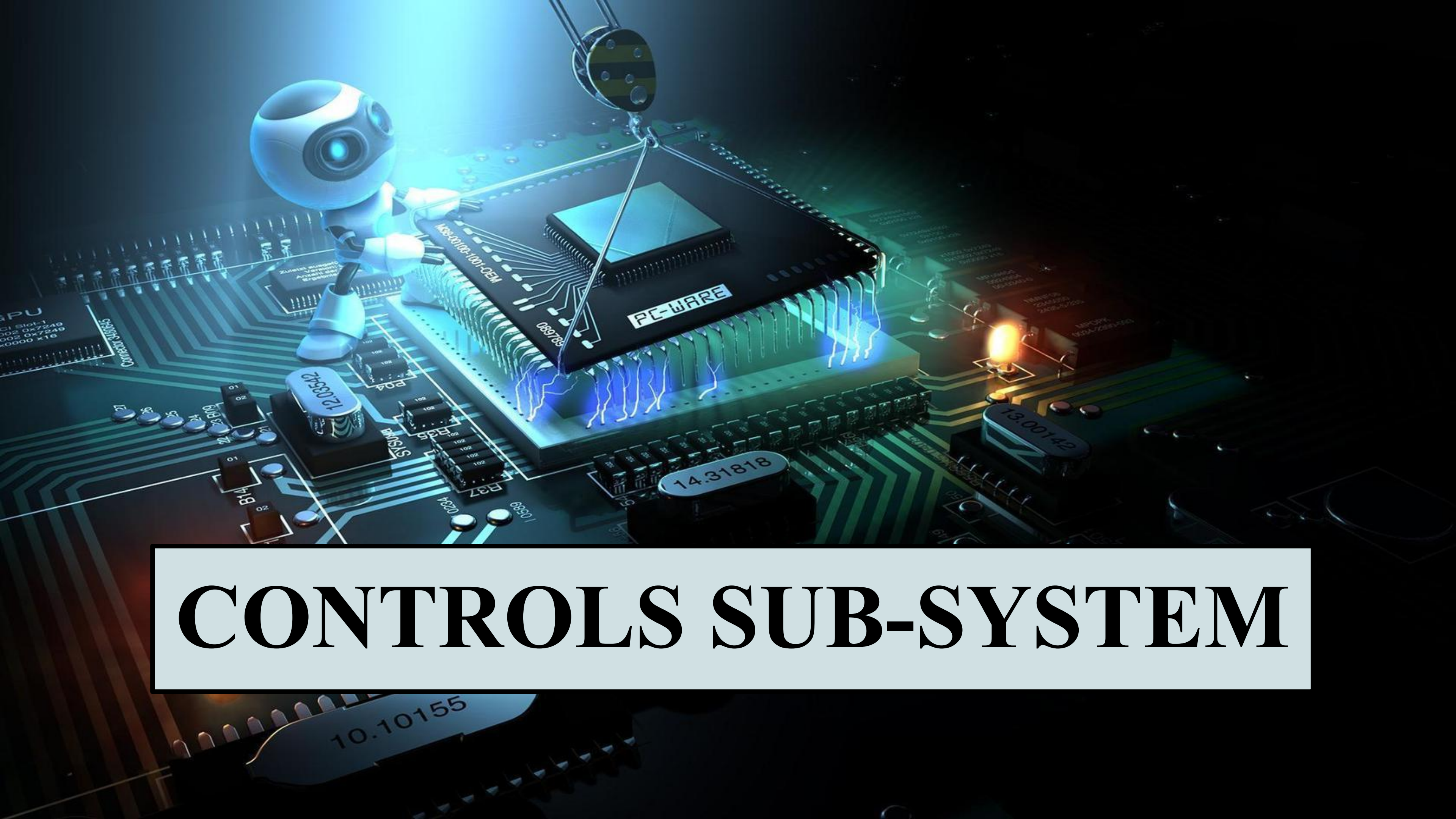
Ultrasonic Sensor (Sonar)

- We are using ultrasonic sensor to detect the dumping site. The sensor is attached to left arm of the bot and it measures its distance from the ground.
- As given in the problem statement, the dumping site will be a pit (khadda) after the last ball. The ultrasonic sensor will detect the pit by abrupt and large changes in output values.
- We were trying to use a laser for this task too, but the laser was unstable with the given minimum value of range. Laser was giving numbers which were very different from each other.
- In the case of ultrasonic sensor we were able to get stable outputs for small ranges. The range for outputs of the ultrasonic sensor being used in our bot is 0.01m to 2m, which is enough to detect the pit.
- We attached the sensor in an inclined way so that we shall be able to detect the pit in advance and have enough space for bot to do a 180° turn.


Terminal
Thu 7:38 PM
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Physics
Wind
Models
Lights
Property
prathmesh@prathmesh-HP-Pavilion-Laptop-15-cc1xx: ~
File Edit View Search Terminal Help
ros(base) prathmesh@prathmesh-HP-Pavilion-Laptop-15-cc1xx:~$ rostopic echo /sens
sonar_front -n1
header:
seq: 0
stamp:
secs: 804
nsecs: 298000000
frame_id: "base_ir_front"
radiation_type: 1
field_of_view: 0.296700000763
min_range: 0.00999999977648
max_range: 2.0
range: 0.0938374251127

(base) prathmesh@prathmesh-HP-Pavilion-Laptop-15-cc1xx:~$
Steps: 1
Real Time Factor: 0.72
Sim Time: 00 00:13:27.634
Real Time: 00 00:11:13.886
Iterations: 452032
FPS: 55.63
Reset Time

CONTROLS SUB-SYSTEM



Key tasks of this subsystem :

1. Provide appropriate controls to each of the joints
2. Lane follower
3. Algorithm for controlling the path based on the signs of the board detected and other image data
4. Algorithm for getting and storing the green ball inside and pushing aside the red balls
5. Dumping all the collected green balls in a pit

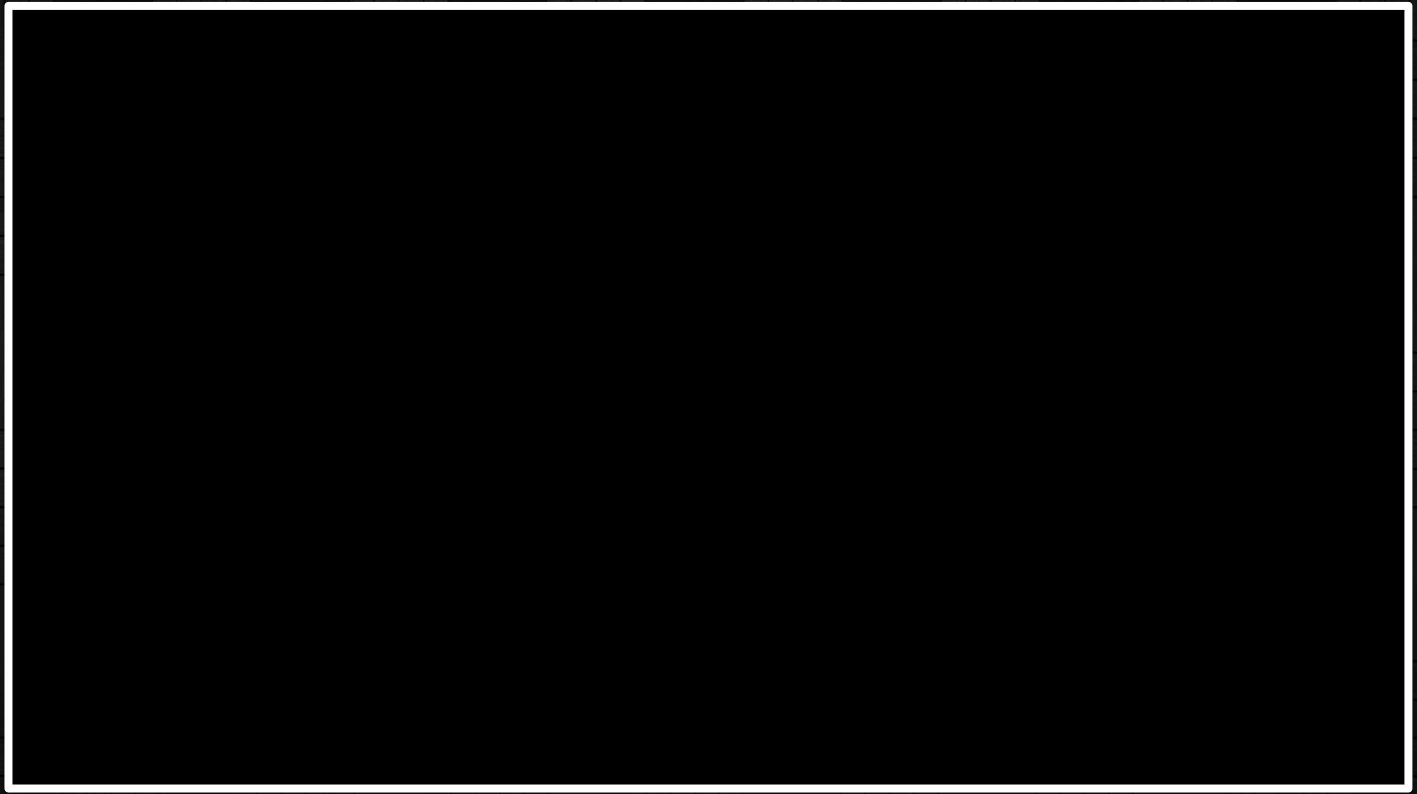
Lane Follower Algorithm

To ensure that the bot stays in the middle to grasp the balls, a mid-lane following algorithm is implemented.

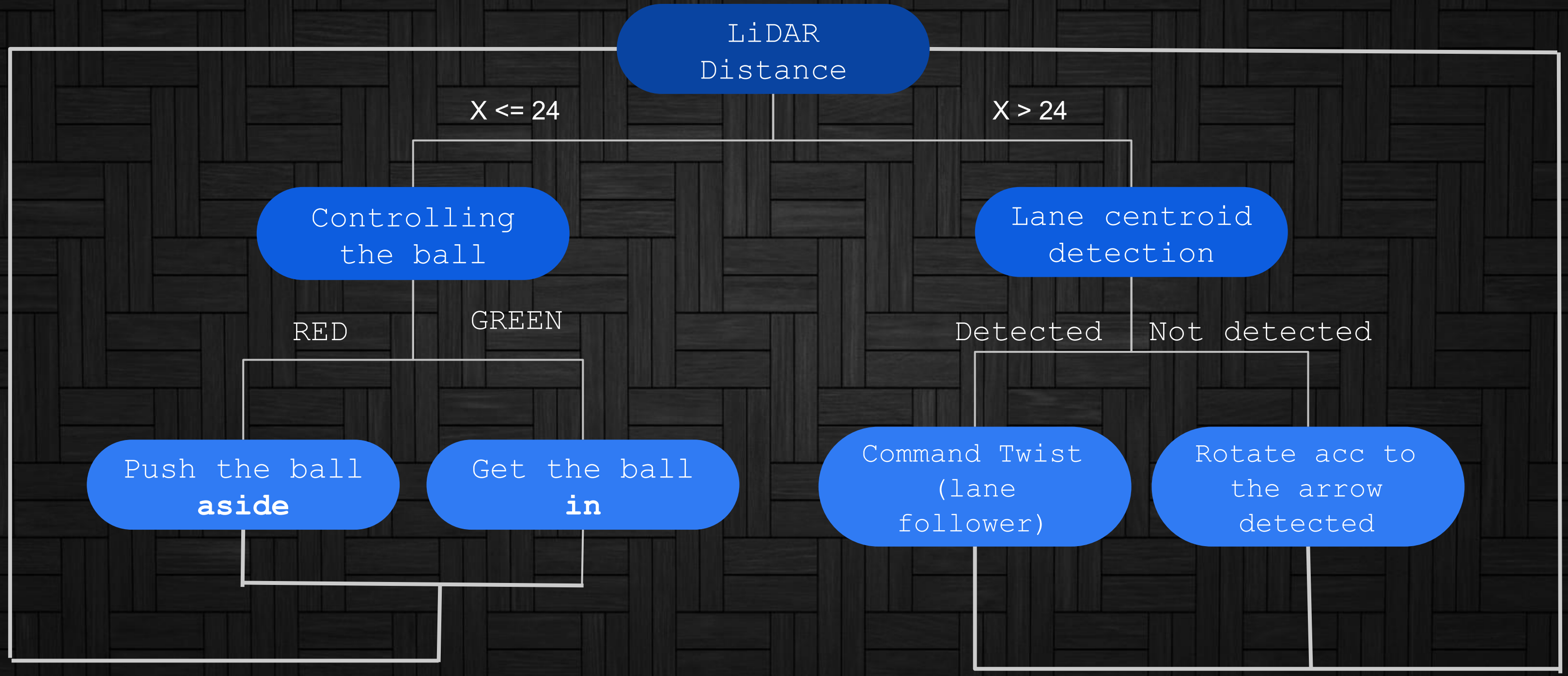
- The image captured by the front cam is cropped to lower portion to reduce computation.
- The centroid of the lane detected is used for calculating the deviation.
- We have currently implemented a simple Proportional based error control law to ensure our bot follows the line. We have defined our error as :

  error_x = (x_coordinate of centroid) - (image_width/2)
  Z_angular velocity = -(error_x)*constant
  Linear velocity_x = 0.1 m/s

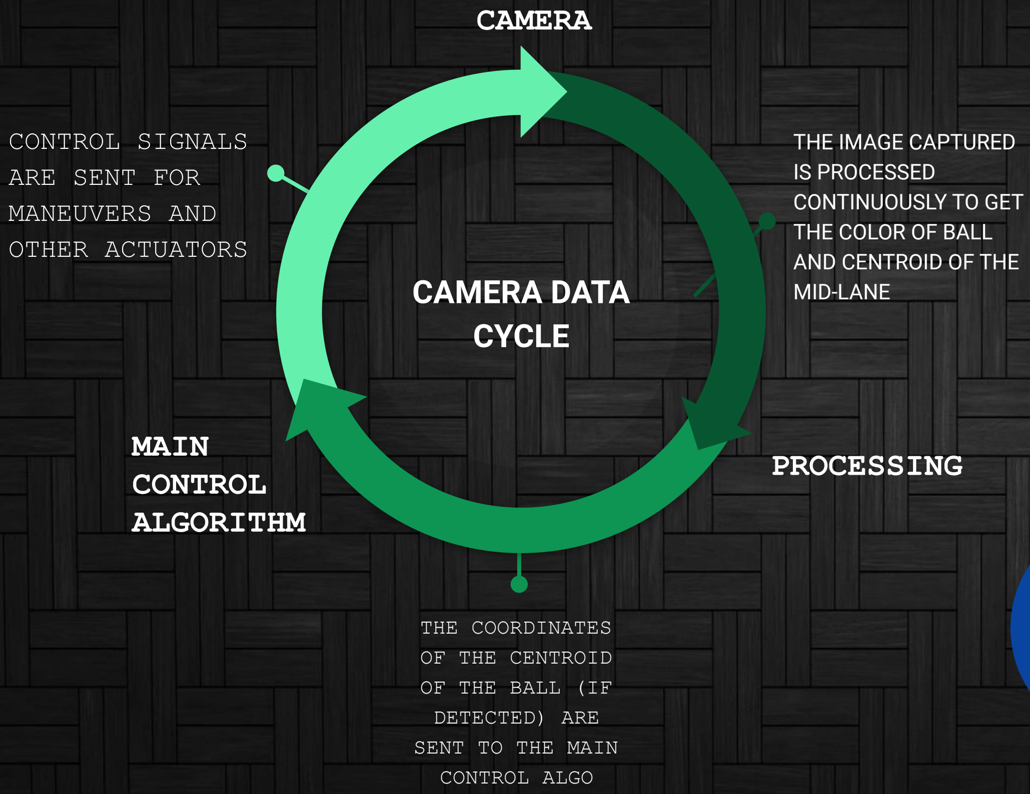
CONTROL OF BOT BASED ON LINE FOLLOWER ALGORITHM



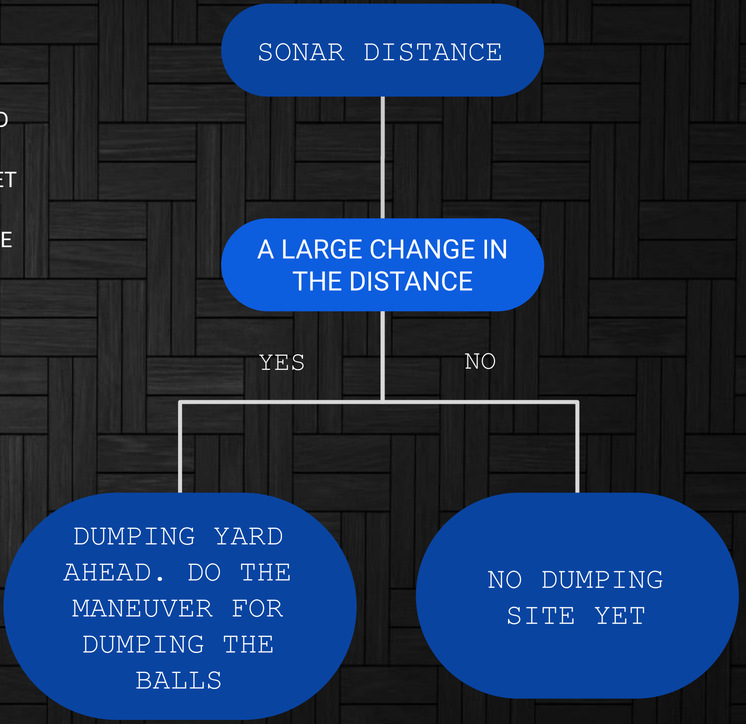
MAIN CONTROL ALGORITHM

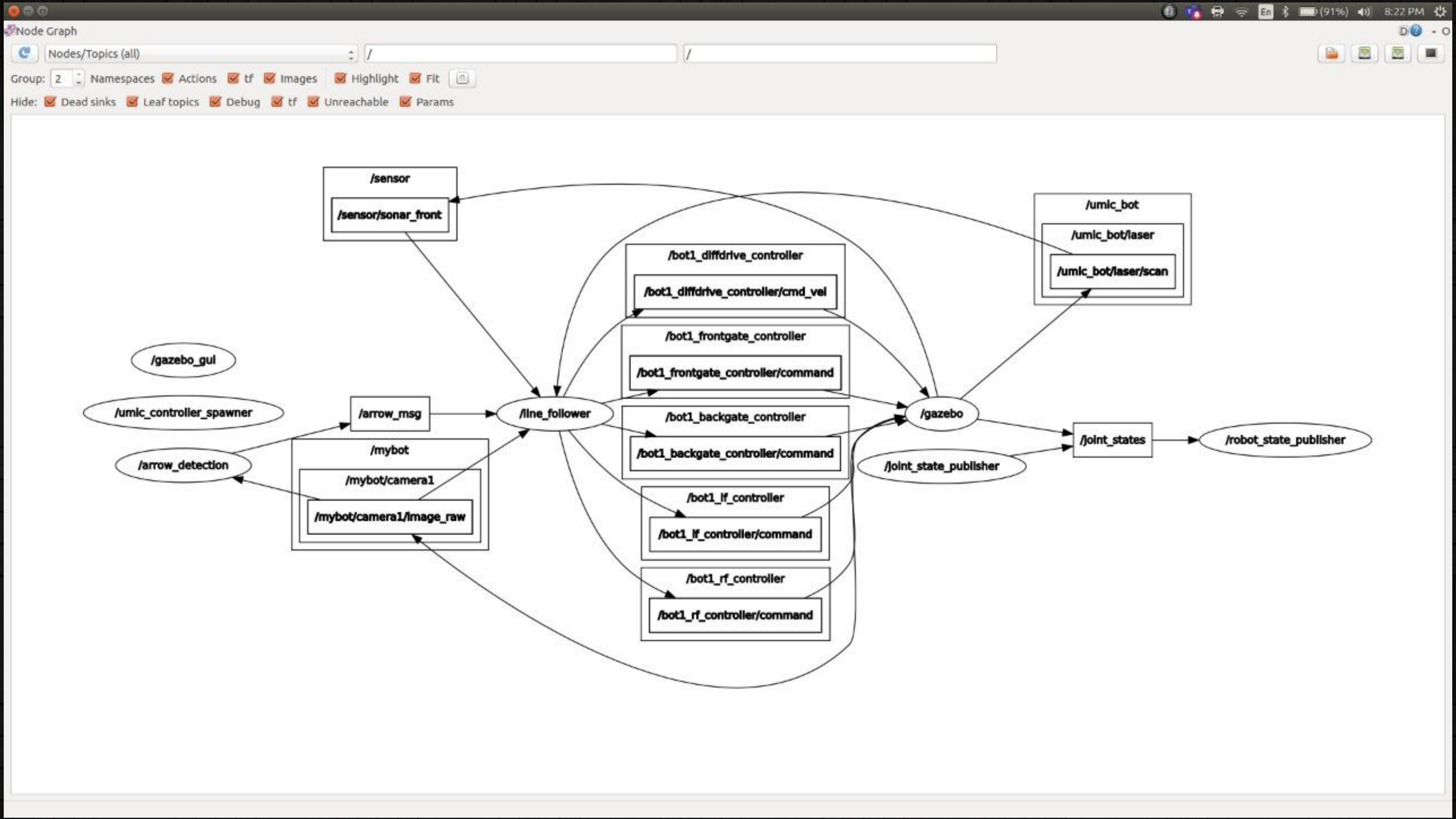


MID-LANE DETECTION AND COLOR OF THE BALL



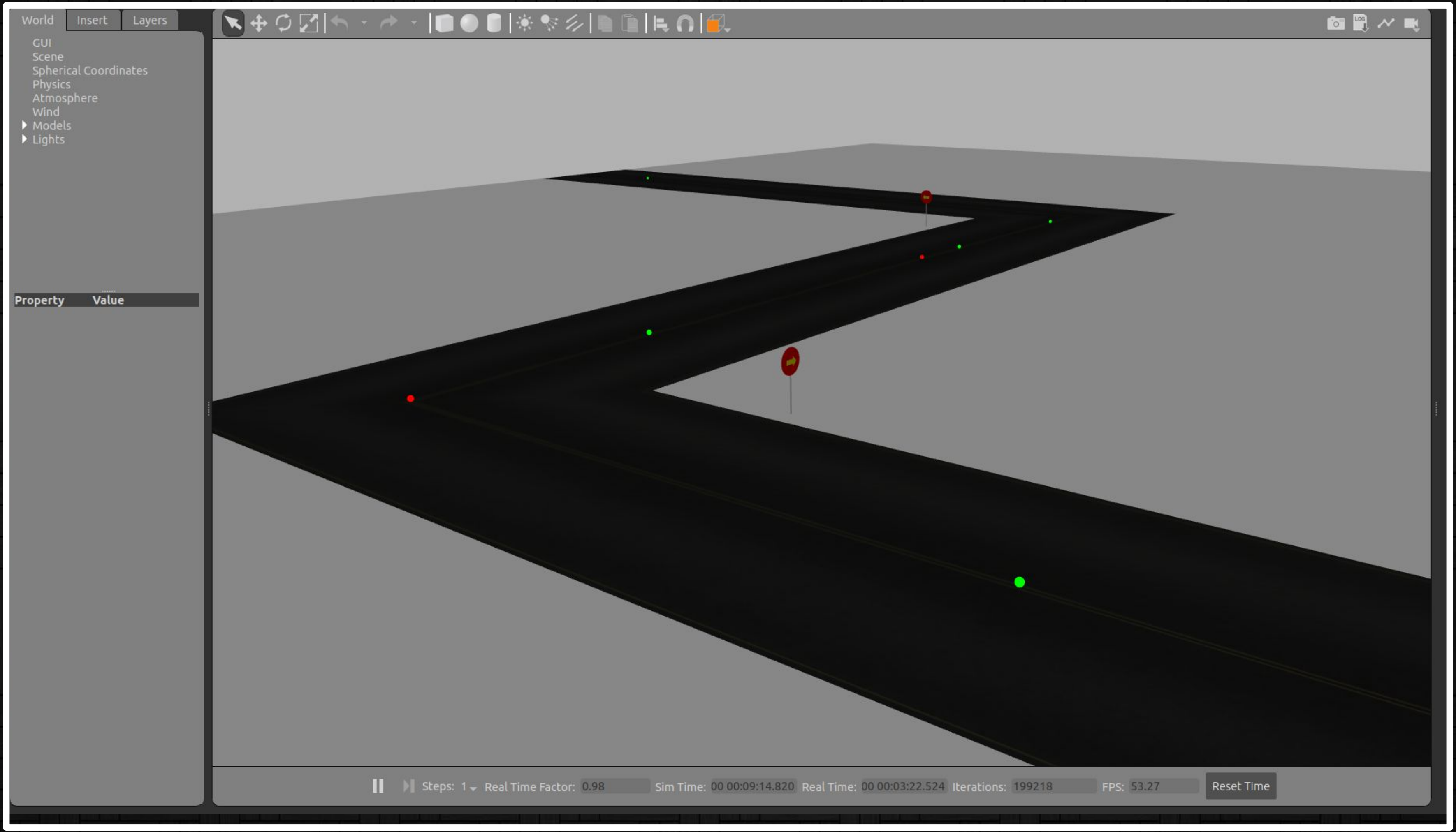
DETECTING THE DUMPING SITE

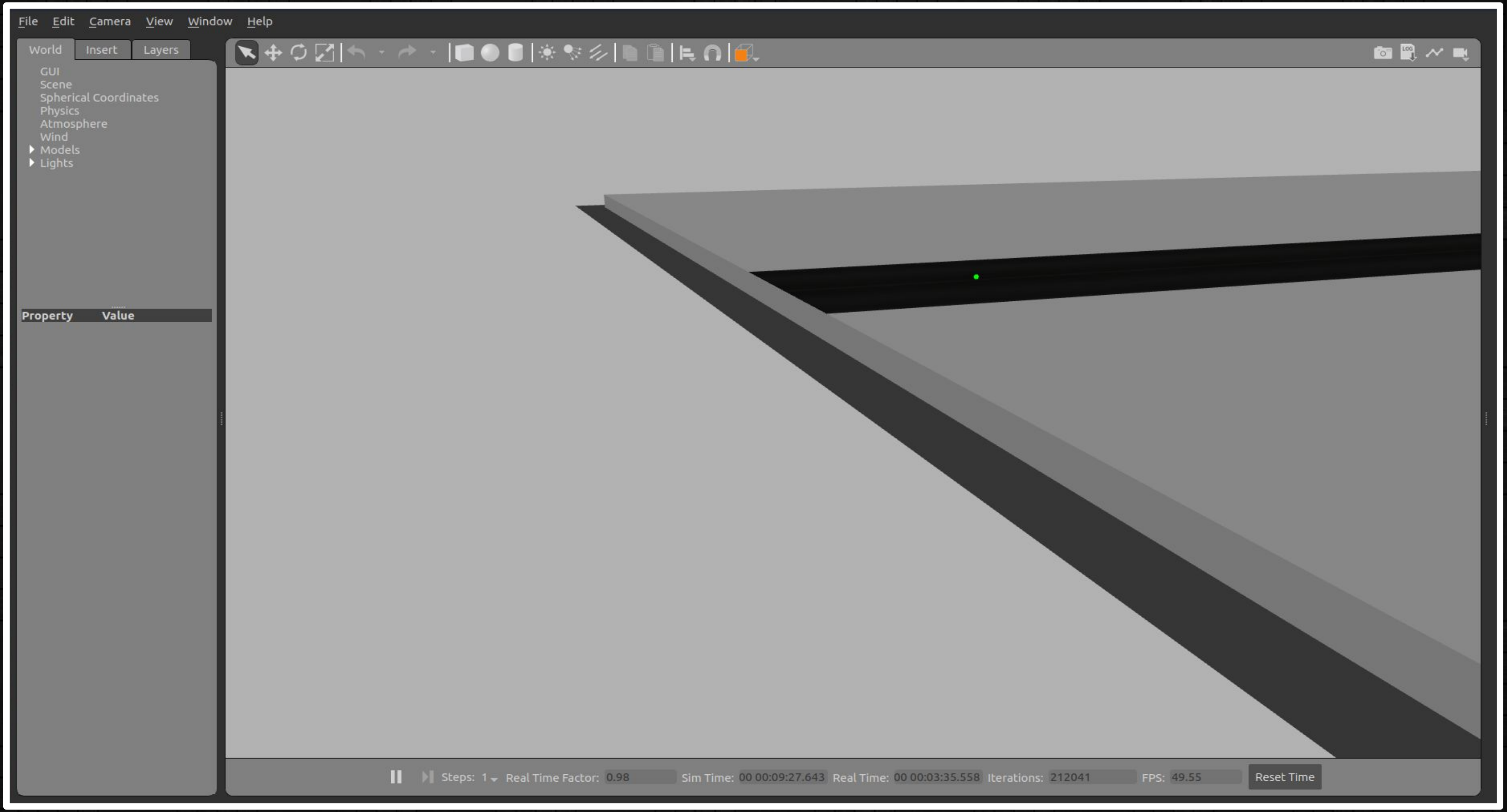


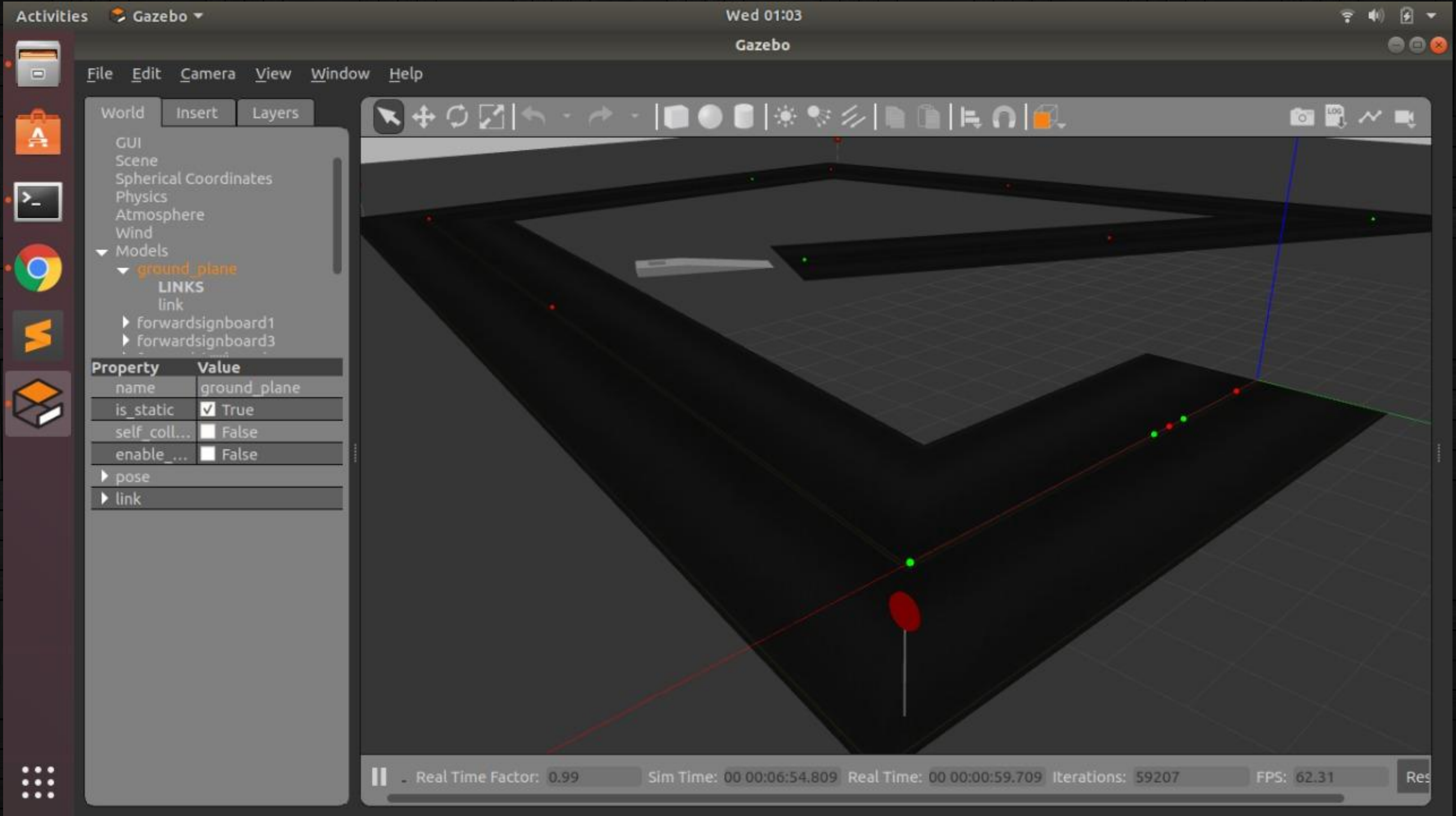

En
(91%)
8:22 PM
Node Graph
Nodes/Topics (all)
/
/
Group: 2
Namespaces
Actions
tf
Images
Highlight
Fit
Hide:
Dead sinks
Leaf topics
Debug
tf
Unreachable
Params
/sensor
/sensor/sonar_front
/bot1_diffdrive_controller
/bot1_diffdrive_controller/cmd_vel
/umic_bot
/umic_bot/laser
/umic_bot/laser/scan
/bot1_frontgate_controller
/bot1_frontgate_controller/command
/gazebo_gui
/umic_controller_spawner
/arrow_msg
/line_follower
/gazebo
/bot1_backgate_controller
/bot1_backgate_controller/command
/joint_states
/robot_state_publisher
/arrow_detection
/mybot
/mybot/camera1
/mybot/camera1/image_raw
/joint_state_publisher
/bot1_lf_controller
/bot1_lf_controller/command
/bot1_rf_controller
/bot1_rf_controller/command

Environment

- Created an environment in Gazebo for testing of bot
- Made sign board in solidworks, exported them to urdf and placed at appropriate locations in environment
- Placed balls of red and green colour at random locations at the centre of the road
- To create a dumping site, we placed the roads and balls on a cuboid of height 0.5m. Here we can test the dumping mechanism at the edge of cube.


World
Insert
Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property
Value
Steps: 1
Real Time Factor: 0.98
Sim Time: 00 00:09:14.820
Real Time: 00 00:03:22.524
Iterations: 199218
FPS: 53.27
Reset Time


File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property Value
Steps: 1
Real Time Factor: 0.98
Sim Time: 00 00:09:27.643
Real Time: 00 00:03:35.558
Iterations: 212041
FPS: 49.55
Reset Time


Activities
Gazebo
Wed 01:03
Gazebo
File
Edit
Camera
View
Window
Help
World
Insert
Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
ground_plane
LINKS
link
forwardsignboard1
forwardsignboard3
Property
Value
name
ground_plane
is_static
True
self_coll...
False
enable_...
False
pose
link
Real Time Factor:
0.99
Sim Time:
00 00:06:54.809
Real Time:
00 00:00:59.709
Iterations:
59207
FPS:
62.31

Github link for our Codes, Model files and other files

https://github.com/Bishop-11/Team_4_UMIC.git

https://github.com/PrathmeshBele/TEAM-4_UMIC

Shortcomings :

- Couldn’t implement a perfect dumping maneuver
- Couldn’t do exhaustive research on hardware parts to be used
- Went a little bit problem statement specific and failed on a few real world aspects.

Successes:

- Everything in the original problem statement was implemented with excellent accuracy.
- Learnt time-management and teamwork skills

(Yet to learn how to give good presentations)

SPECIAL THANKS
TO

Our wonderful mentors for the help they have provided to us, even in the middle of the night, hours of discussion and meetings.

All the seniors of UMIC for the help they have provided to us during the training phase, because of which we were able to learn all the required skills for this project.

THANK YOU!!!