Autonomous Stair Climbing Robot

Team Name : LazyBot

Institute Name: IIIT Hyderabad

Team members details

Team Name	LazyBot				
Institute Name	International Institute of Information Technology Hyderabad				
Team Members >	1 (Leader)	2	3	4	5
Name	Vedant Mundheda	Karan Mirakhor	Dipanwita Guhathakurta	Rahul Kashyap Swayampakula	Puppala Avinash Prabhu
Batch	2023	2022	2023	2022	2022
Area of expertise	Design & Vision	Design & Mechanism	Robotics Vision & AI	Design & Sensor Placement	Control & Locomotion

Functionalities of the Robot

What all can the robot do?

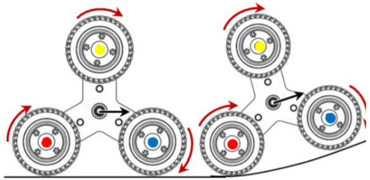
- Autonomously detects Stairs and Plans Path for climbing up or down the stairs
- Fast, Robust, Maneuverable in any terrain, reliable and stable while climbing
- Autonomously detects and changes state b/w Climbing Mode and Driving Mode, independently for front and rear wheels.
- Independently looks for the next Staircase on a landing.
- Keeps payload compartment aligned horizontally always. (using linear actuators and gyro)
- Adjusts bot orientation to remain centrally aligned during Stair Climbing without use of side walls

Are there any things that the robot can do above and beyond the requirement?

- Moves effortlessly on an inclined plane in Driving Mode.
- Fast and stable on Rough and Rocky terrains.
- Carry payload weight upto 25 Kg on a Staircase.
- Server and an App to track the Bot's location, sensor data, battery status etc remotely.
- Climb up or down even on Curved or Spiral Staircase.

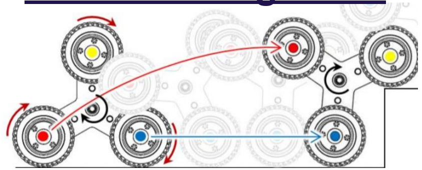
Tri- Wheel Rotational Modes

Driving Mode



1) Only the three wheels are driven to rotate in this mode.
2) Tri Wheel is able to passively pivot about its centroid to adapt for obstacles and slopes.

Stair Climbing Mode



1) Braking mechanism is engaged to cause the entire Tri Wheel to rotate about centroid.
2) Rotation occurs in the same direction as the three spinning wheels which are locked now.

Are there any out of the box functionalities?

- Overcomes scenarios in which only two wheels remain in contact with Ground. By Toppling the wheel
- Passive tumbling of the Tri wheel and Suspensions give extra stability in Driving Mode.
- Simple, efficient and effective Wheel Mechanism for Stair climbing.
- Prevents overload using Separate Batteries for Front and Rear wheels

Robot Specifications

Physical Specs

- **Robot Dimensions :** 50x50x96.59 cm (lxbxh)
  - Ground Clearance : 16.59cm
  - Chassis dimension : 50x50x20 cm
  - Payload box dimension : 40x40x60 cm
- **Wheel design :** Tri-Star wheel design
- **Wheel dimensions :**
  - Each wheel radius: 7.62cm
  - Spoke Length: 13.97cm
- **Material used for robot body :** Aluminium and plastic.
- **Weight** : 50kgs (without payload)

Technical Specs

- **Ideal operation speed of the bot in Driving Mode:** 8 - 10 Km/hr
- **Ideal operation speed of the bot in Stair Climbing Mode:** 2 - 3 Km/hr
- **Max Obstacle Height it can cross over in Driving Mode:** 40 cm
- **Max Obstacle Height it can cross over in Stair Climbing Mode:** 20 cm
- **Power consumption:** 180 - 1415 W
- **Distance the Bot can go upto on one full charge:** 7.5 - 65 Km depending upon the use case
- **Turning radius:** ~0 cm
- **Ideal Working Scenario Endurance:** 22.23 Km (70% time on flat terrain, 15% time on 45° inclined plane, 15% time in Stair Climbing)

Robot/Solution Limitations

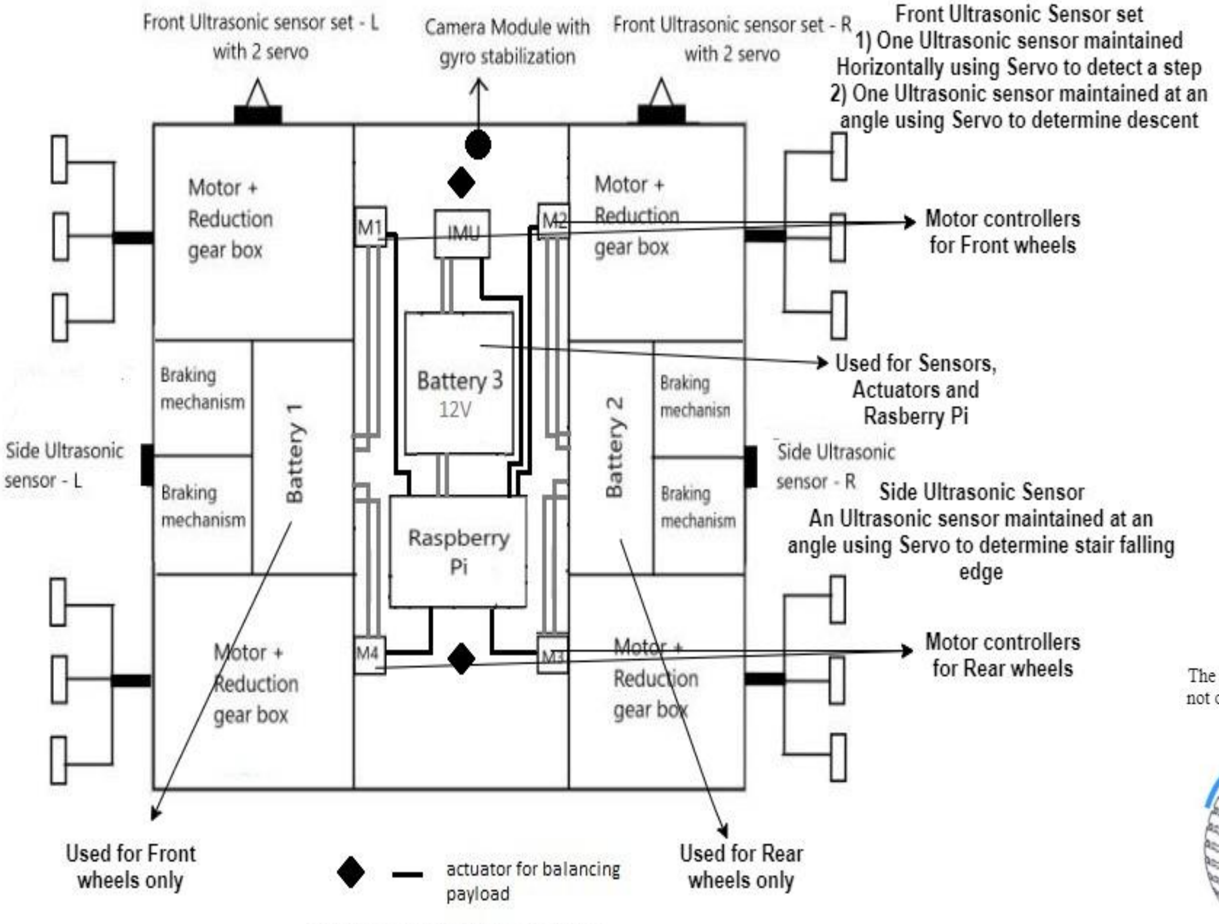
Limitations other than Requirements

- Robot and Tri Wheel system is designed to operate on Staircases of 15-20 cms Height with 24-30 cm of Tread in Stair Climbing Mode.
- Speed restricted to 11 Km/hr due to the choice of Geared DC Motors for excess Torque during Stair climbing.
- It can carry a maximum payload of upto 25Kg only.
- With one full battery charge :
  - Upto 65 Km in Driving Mode on a flat terrain.
  - Upto 7.5 Km in Stair Climbing Mode
- Software Limitation : Vision algorithm works less efficiently in case of curved stairs. We rely on Ultrasonic Sensor Data and IMU to climb Curved stairs.

No Limitations compared to the Requirements!!

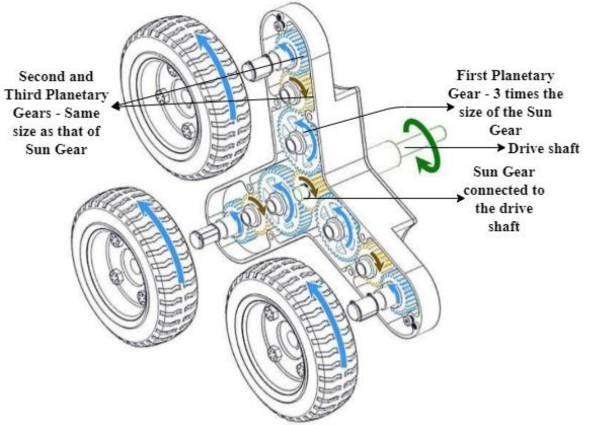
Robot Design

Robot Top View

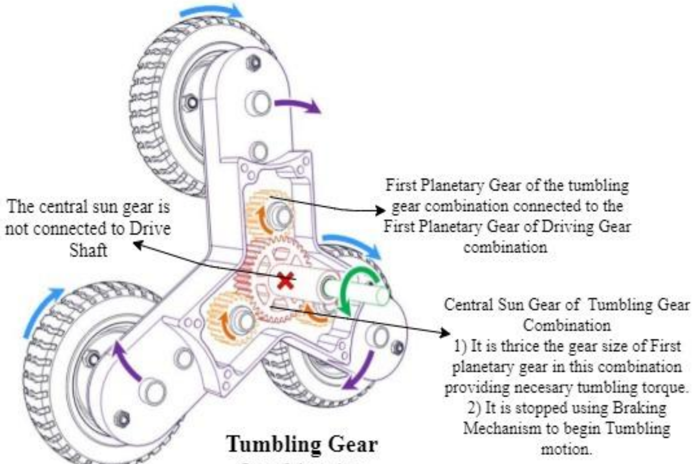


Top View of Component section

Wheel Design

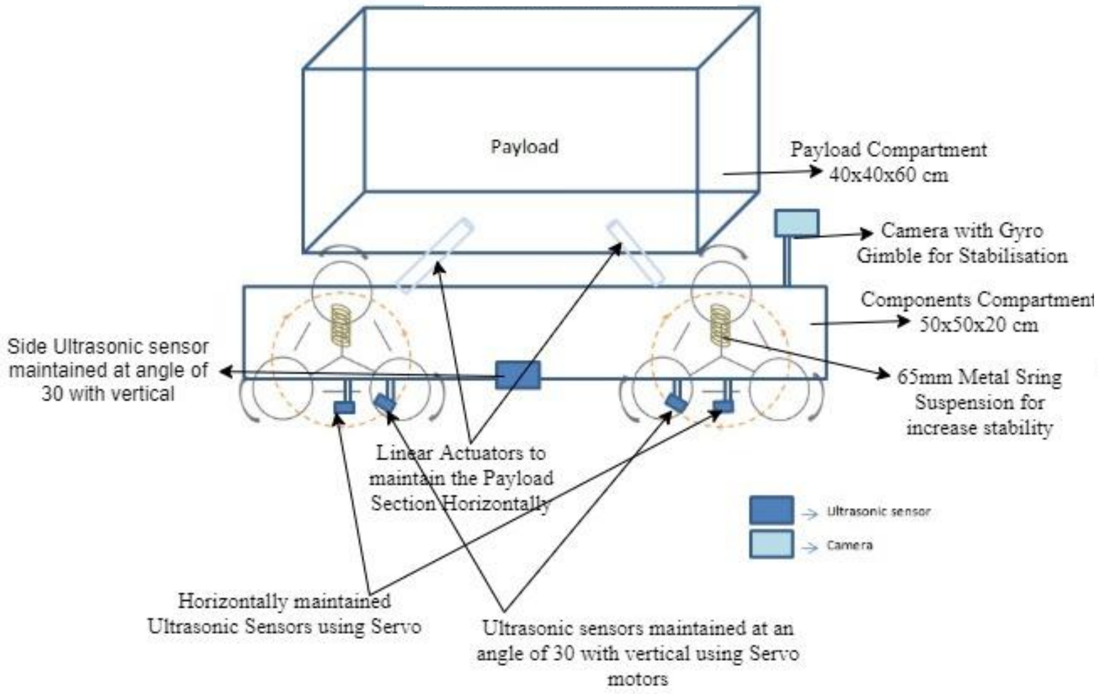


Gear arrangement for Tri Wheel in

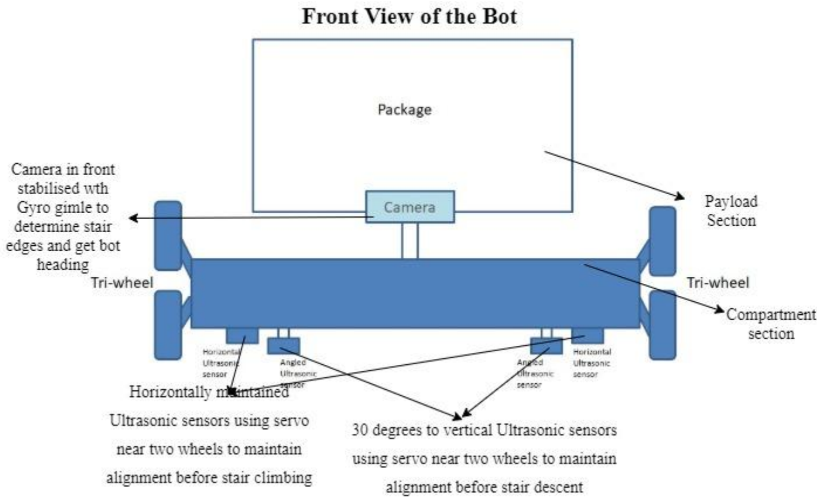


Tumbling Gear Combination

Robot Side View

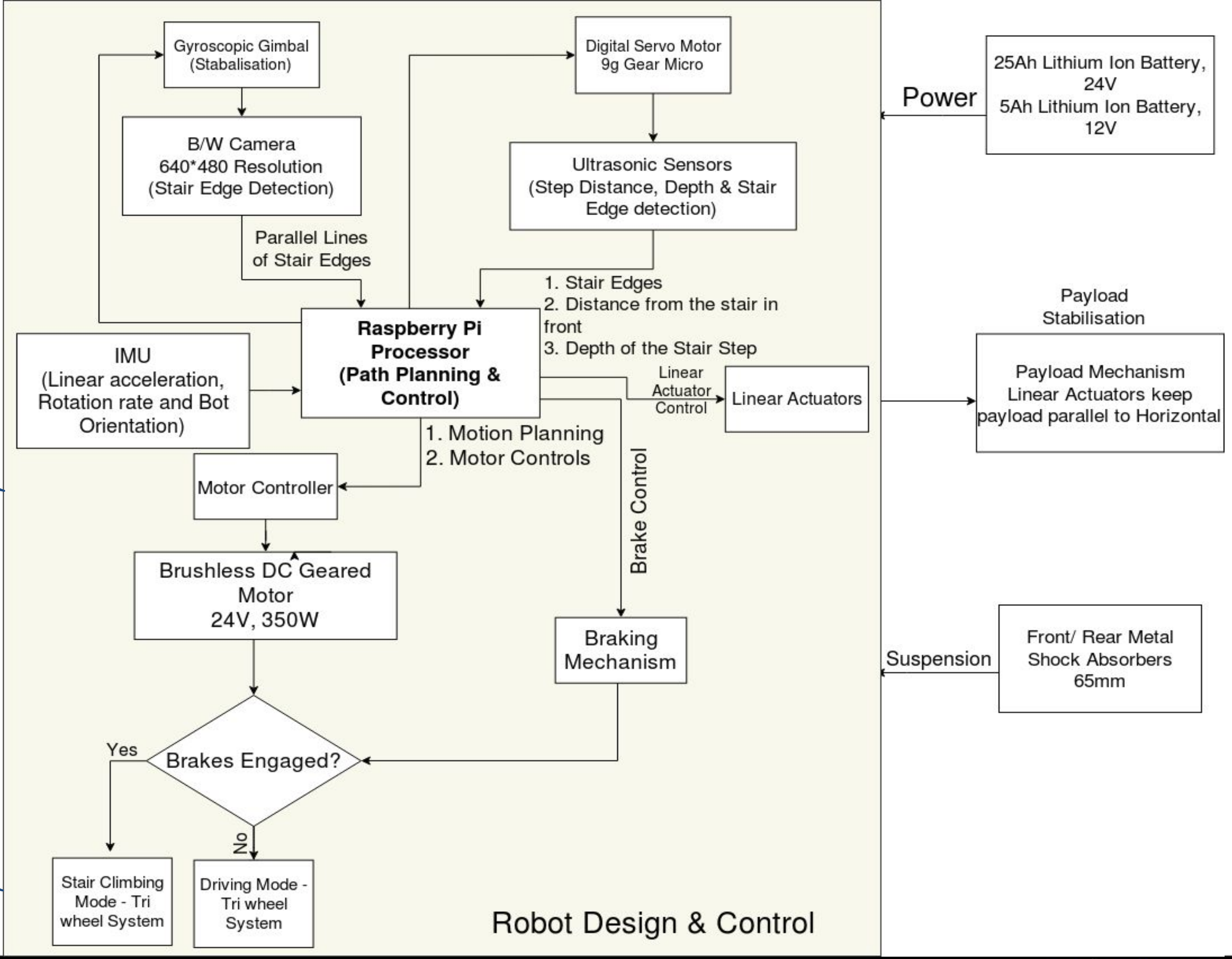


Robot Front View



Architecture

The diagram describes how the electrical modules are connected in the bot. And explains the transfer of different signals among the modules at a higher level.



<u>Note:</u> The Motors and brakes are connected to wheels and linear actuators are used to stabilise payload. The power lining is explained in the previous slide.

Brief on Programming Module

Programming language used:

Python

OpenCV

Raspberry Pi modules to support hardware:

- GPIO support on Raspberry Pi: **RPi.GPIO** for the HC-SR04 ultrasonic sensor.
- Raspberry Pi Camera module: **Picamera** to capture to get a stream of images of the staircase.

Python Modules to be built (for stair detection through image processing):

- **Gabor filter**: using OpenCV's inbuilt `getGaborKernel`, `filter2D`.
- **Edge detector**: using OpenCV 's inbuilt `Canny` edge detector.
- **Edge linking:** to be coded from scratch using Python.
- **RANSAC:** using skimage's inbuilt `ransac`.

Estimated overtime of the system is less than 60 ms

Custom built modules to control the various hardware components:

- Program to control the various states(rolling, ascending, descending) by processing the inputs from the ultrasonic sensor, IMU and gyroscope and controlling the brakes.
- Program in the Microcontroller which will control the wheels (taking turns) based on the current state and path plan.
- Program for the autonomous orientation of robot in the direction of the next stairway by processing distance inputs from lateral ultrasonic sensors at the quarter-landing.

Execution Plan

1. Exploring various solutions based on cost, feasibility, etc. (July 1-2)
2. Making a list of the requirements for a particular solution. (July 3-4)
3. Drawing a rough sketch sketch of the bot and the algorithm flowchart (July 5-7)
4. Making CAD model of proposed Design using Solidworks and developing code. (July 8-17)
5. Testing the robot in simulation (July 18-25)
6. Improving the algorithm and design based on the results (July 26-29)
7. Order the parts required for the robot.
8. Assemble the hardware
9. Testing the Algorithm on the board and verifying the commands without body.
10. Testing the hypothesis on the bot
11. Improving the results
12. Final submission

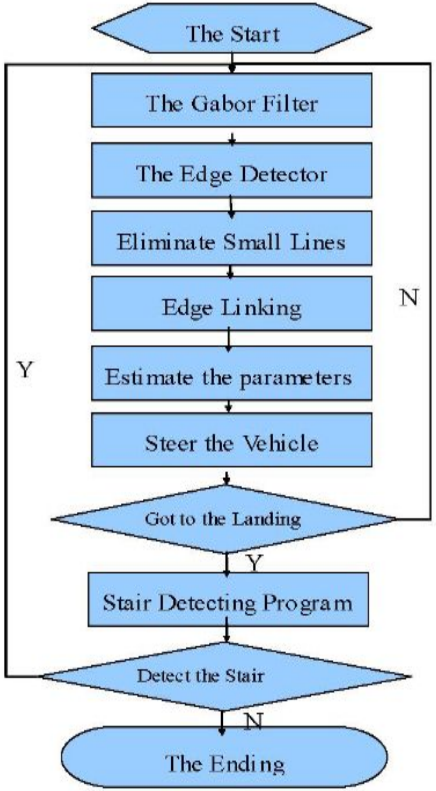
Cost Estimation

1.	Brushless DC Motor 350W	4 x ₹ 3,600
2.	Motor Controller - 350W	4 x ₹ 800
3.	25Ah 24V Li ion Battery	2 x ₹ 7,500
4.	5Ah 12V Li ion Battery	1 x ₹ 500
5.	Chassis Assembly	₹ 1,500
6.	Wheel Assembly	4 x ₹ 2,500
7.	Raspberry Pi 3	₹ 3,000
8.	Camera Module	₹ 600
9.	Sensors and Actuators	₹ 6,000
10.	Suspension system	₹ 2,500
	Total	**₹ 56,200**

Note:
Tentative Timeline

Done
In Progress
Yet to start

Autonomous climbing using Vision Algorithms



- **Gabor Filter** used:
  - Eliminate influence of illumination
  - Keep Stair Edges
- **Generic edge detector** on filtered image:
  - Removes small, vertical edges.
- Remainder adjacent edges linked into horizontal parallel edges (Stair edges)
  - Midpoints of horizontal lines estimated.
- A line is fitted joining centres of detected stair edges by **RANSAC** method (white line in Fig 4) and 2 offset parameters are estimated w.r.t robot orientation (blue line in Fig 4):
  - θ = angle b/w bot orientation and the line joining centres of detected edges
  - Δx = distance b/w bot orientation and line joining centres of detected edges
- If θ and Δx are less than a fixed threshold, maintain previous orientation, else change direction to minimise θ.

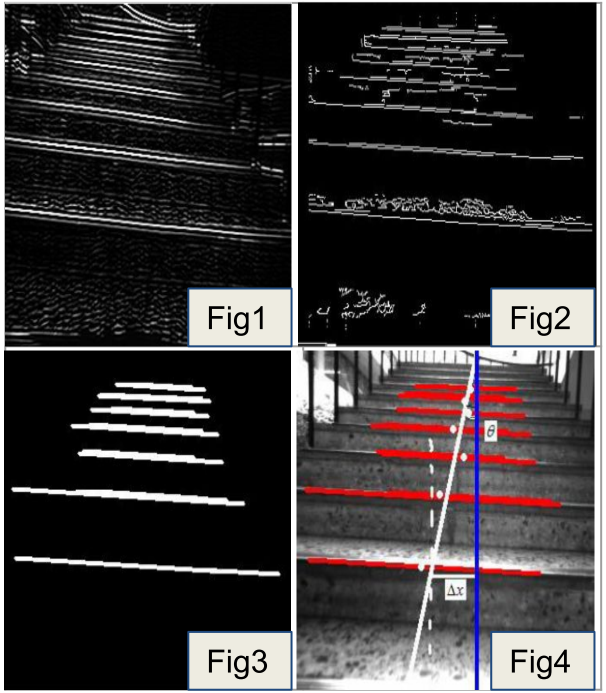
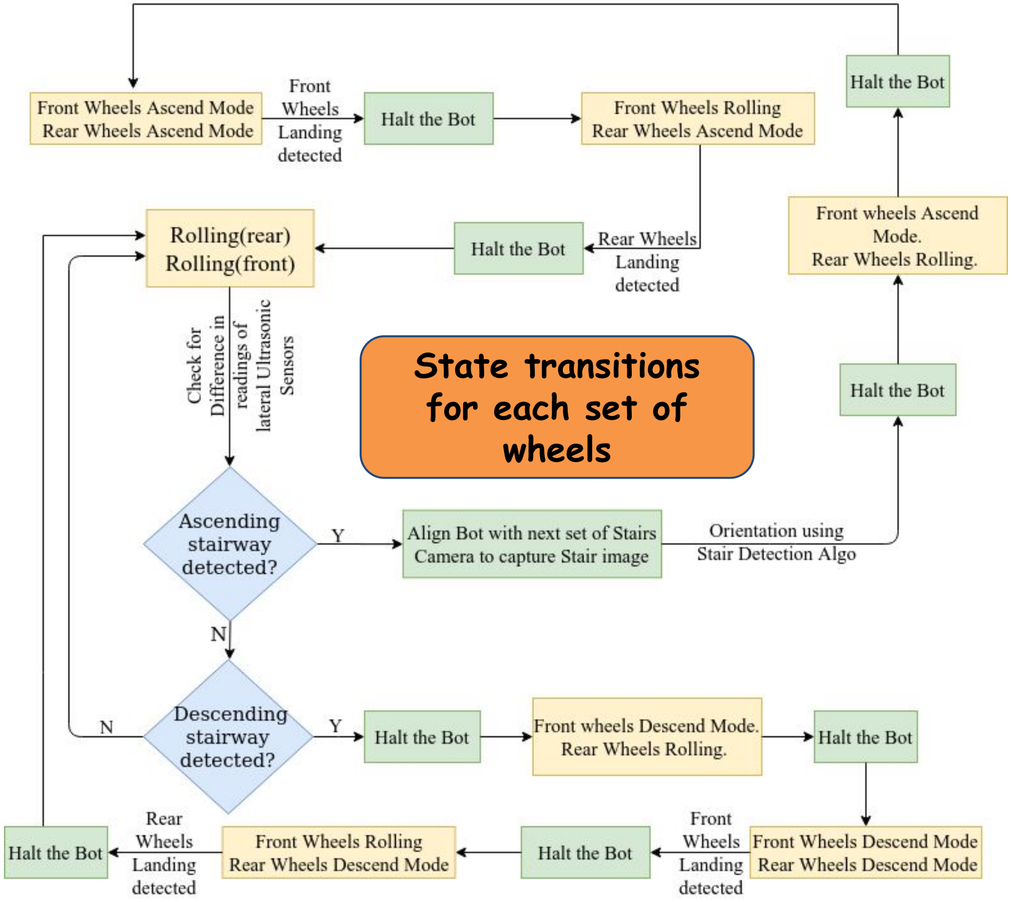


Fig 1: Output of Gabor filter
Fig 2: Output after eliminating small vertical edges
Fig 3: Output after edge-linking
Fig 4: Demonstration of calculating offset parameters

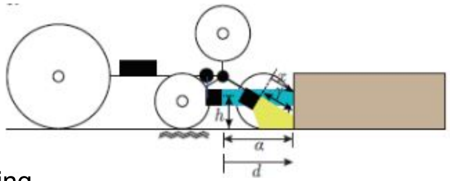
These are references [ref_1](), [ref_2](), [ref_3]()

Control flow



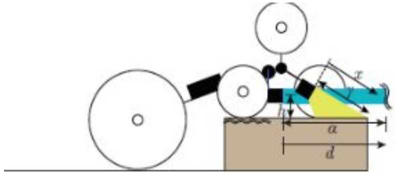
State transitions for each set of wheels

θ= 0 degrees
d <= a
x <=y
Increase power



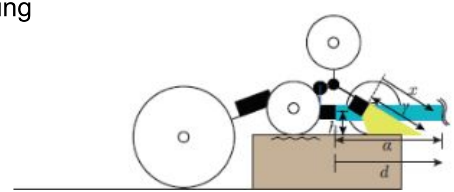
Rolling ➡ Ascending

θ= 120 degrees
d > a
x <=y
Normal power



Ascending ➡ Rolling

θ= 0/120 degrees
d > a
x > y
Decrease power



Rolling ➡ Descending

θ: gyroscope reading to get angle above horizontal
d: Distance in front through ultrasonic sensor
x: Inclined distance through ultrasonic sensor

State switching logic for wheels

Flipkart

GRiD 2.0