Autonomous Stair Climber - Round 3

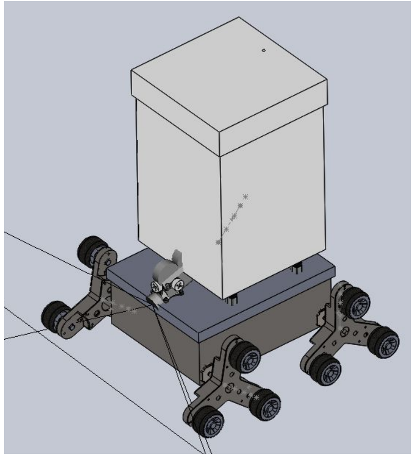
Team Name: LazyBot

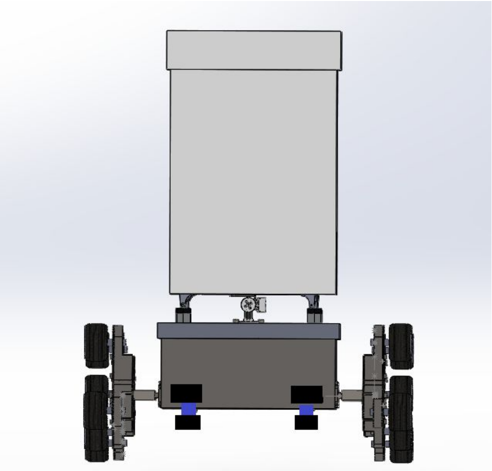
Institute Name: International Institute of Information Technology, Hyderabad

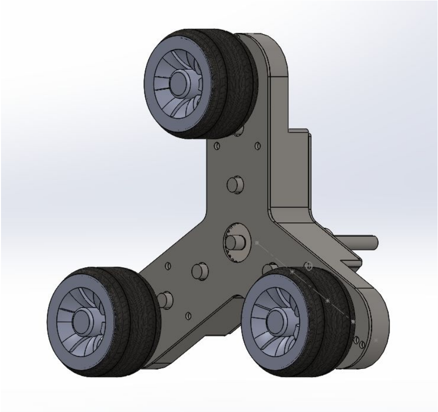
Team Member details

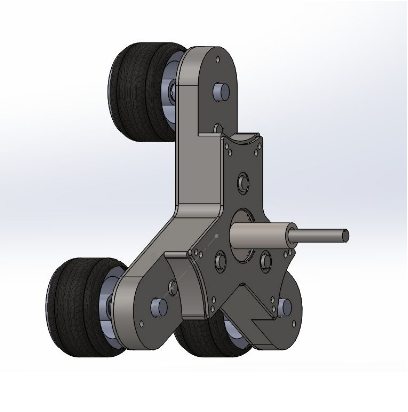
Vedant Mundheda	Karan Mirakhor	Rahul Kashyap Swayampakula	Dipanwita Guhathakurta	Puppala Avinash Prabhu
Undergraduate Researcher at Robotics Research Centre(IIIT Hyderabad)	Undergraduate Researcher at Robotics Research Centre(IIIT Hyderabad)	Undergraduate Researcher at Robotics Research Centre(IIIT Hyderabad)	Undergraduate Researcher at Robotics Research Centre(IIIT Hyderabad)	Undergraduate Researcher at Robotics Research Centre(IIIT Hyderabad)
	Cleared 3 levels in E-yantra Robotics competition	Cleared 3 levels in E-yantra Robotics competition		
Working on Aerial Manipulator Project	Working on Aerial Manipulator Project	Working on Aerial Manipulator Project	Working on Collision Avoidance in Dynamic Environments	Working on estimating free space in a warehouse environment.

Assembly









The outer side designed bot contains:

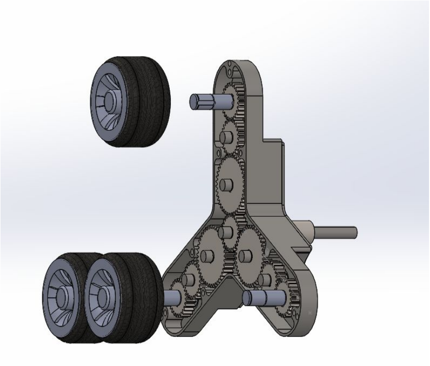
- 4 Tri Star Wheels for the motion of bot.
- A gimbal placed at front to hold and maintain stability of camera.
- A payload compartment with lid is attached to carry the payload.
- Linear actuators lifting and balancing the payload
- The required Ultrasonic sensors and Servo Motors are placed around the bot as shown in the right figure.

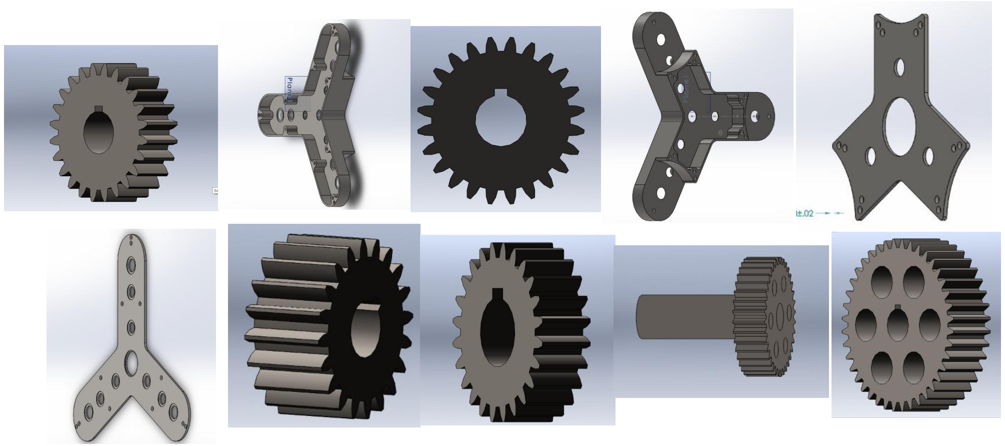
The Tri star wheel is a robust and maneuverable design that has two modes of operation :

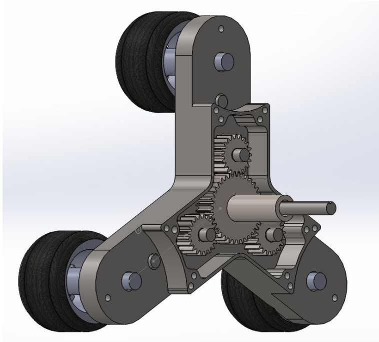
1) Driving Mode for Motion on surfaces.
2) Tumbling Mode for Stair ascent or descent.

Left top : Isometric View of bot ;
Right top : Front view of bot
Left bottom & right bottom : Tri star wheel

Wheel Design







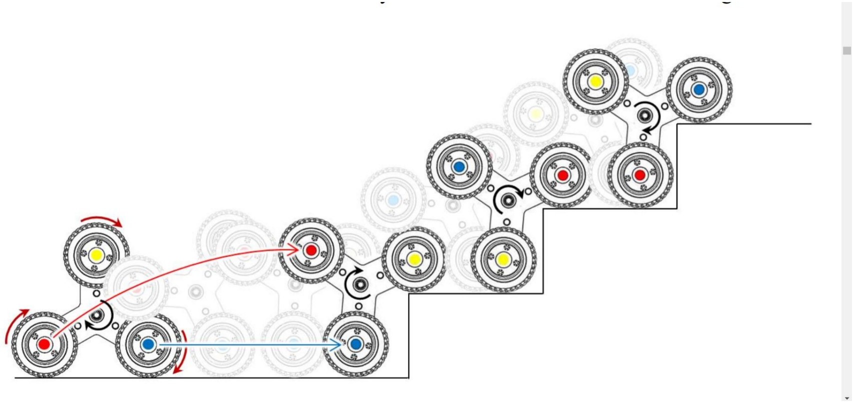
Driving Mode Design

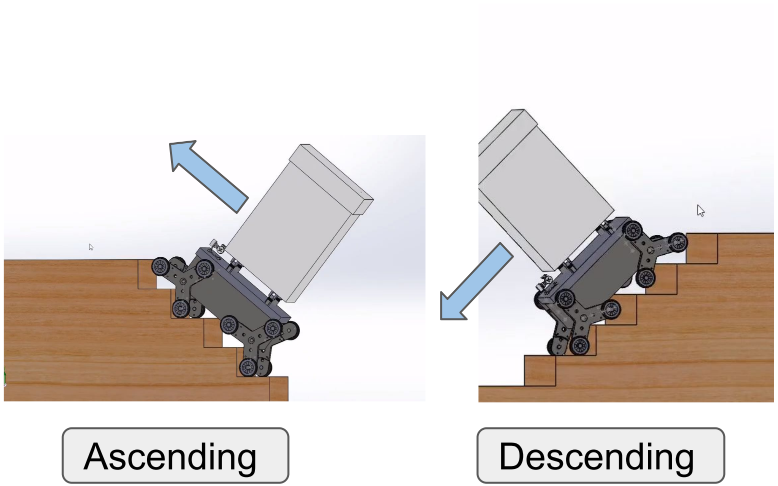
1) The Motor Shaft is directly connected to the central driving gear.
2) Central driving gear is linked to the Large Driving gears (Gear Ratio 1:3).
3) These Large Driving Gears are linked to the Idler gears (Gear Ratio 3:1).
4) Now, these Idler gears are linked to the Main driving gears which are connected to the wheel shaft.
5) This gear design ensures that motor speed and torque is directly delivered to all three wheels simultaneously in this mode..
6) Simulation link : https://youtu.be/Z-jkf5voGJo

Tumbling Mode Design

1) The Central tumbling Gear is **NOT** linked to the motor drive shaft.
2) The Central tumbling gear is linked to the Small Tumbling gear (Gear Ratio 3:1).
3) These small tumbling gear is linked to the Large driving gears.
4) Now when we lock the Central Tumbling Gear using Braking Mechanism, the whole Tri Wheel is constrained to rotate about its Central axis.
5) This Gear Design ensures three times the Motor torque during Tumbling.
6) Simulation link : https://youtu.be/PUGKdtVIjqo

Wheel motion



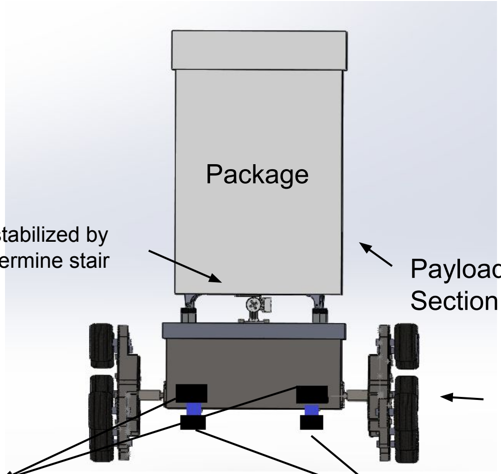


The leftmost Tri-Wheel instance in Figure (left side) begins in Driving Mode with all three wheels spinning in a clockwise direction. The arrow encircling the centroid of the mechanism indicates the initiation of the braking mechanism to induce tumbling.

 As the Tri-Wheel body begins to rotate about its centroid, it simultaneously continues to translate in a horizontal direction, as indicated by the blue horizontal arrow. As the leading (blue) wheel rolls forward on the flat surface, the top THE TRI-WHEEL MECHANISM (yellow) wheel begins to rotate forward and lands on the first step and assumes the position as the new leading wheel. At this point, the leading (yellow) wheel rolls forward. As the leading (yellow) wheel approaches the next step, the top (red) wheel swings over top of the yellow wheel and lands on the next step. The pattern continues as the Tri Wheel mechanism effectively “walks” itself up the set of stairs.

Robot views (Showing Electronic Components)

Front View of the bot



Camera in Front stabilized by Gyro Gimle to determine stair and Bot direction

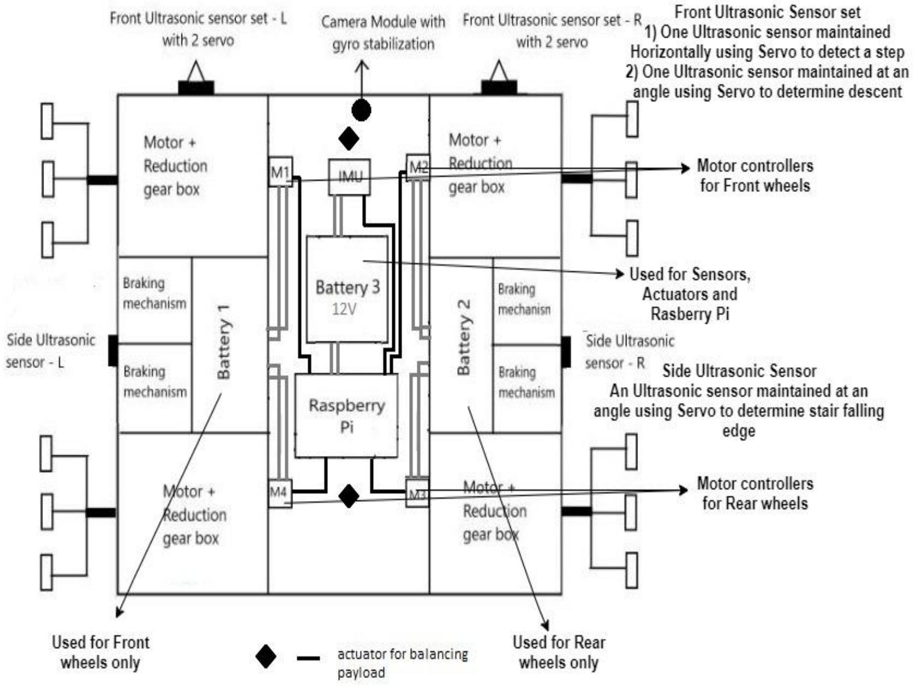
Payload Section

Tri-Wheel

Horizontally Maintained Ultrasonic Sensors

30 degrees to vertical Ultrasonic sensors using servo near two wheels to maintain alignment before stair descent

Top view of the bot

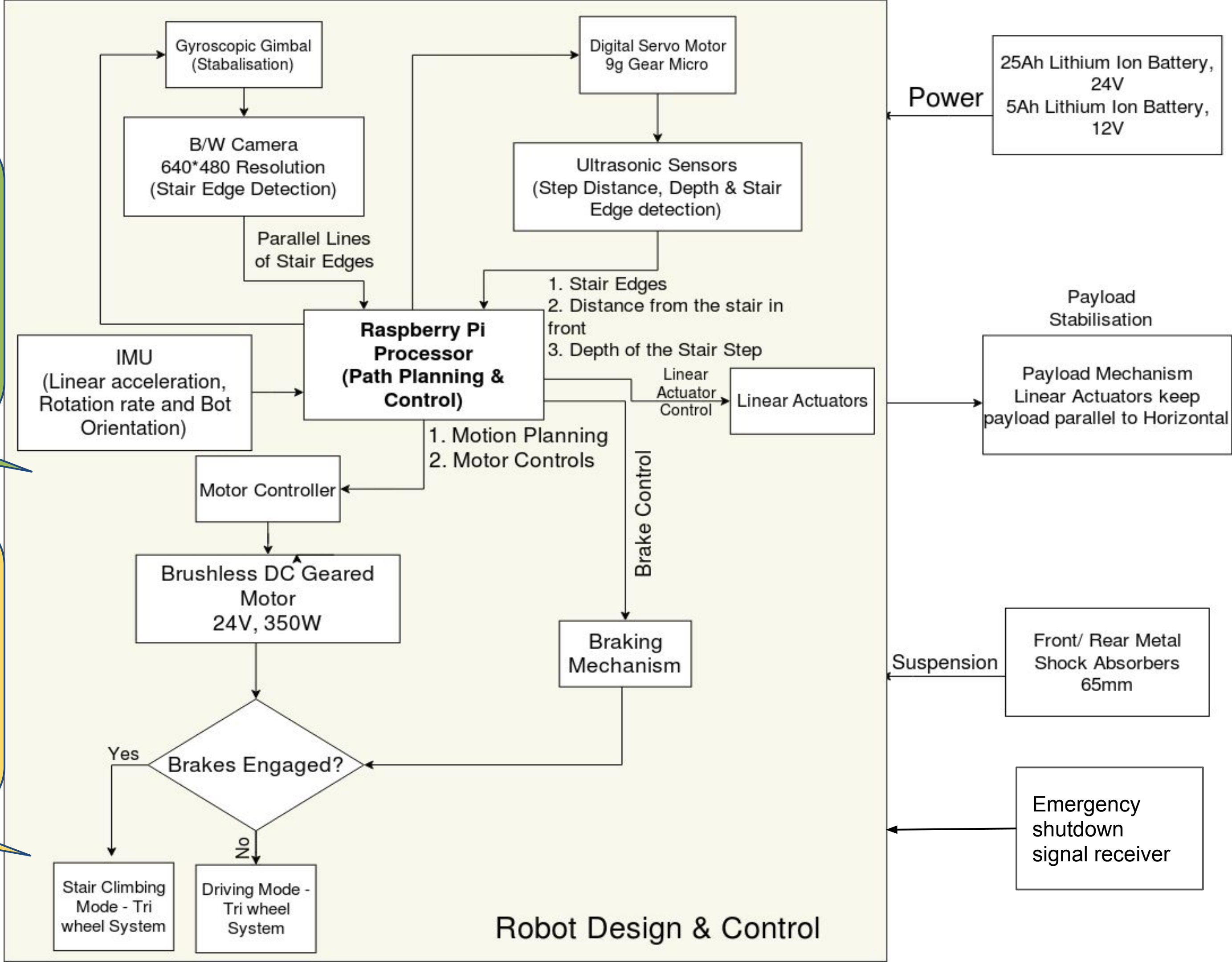


Top View of Component section

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.



Stair detection using YOLO(You Only Look Once)

YOLOv5

We used **YOLOv5** to extract the portion of the image in which the stairs are present.

1488 — Train set Size

288x416 — Training Image Size

0.5 — IOU Threshold

0.4 — Confidence Threshold

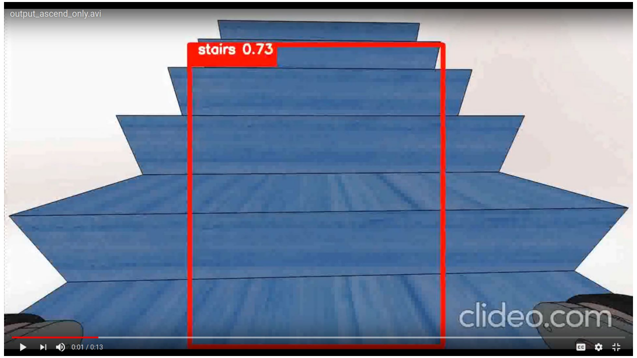
Results on Test

0.044s — Running Time

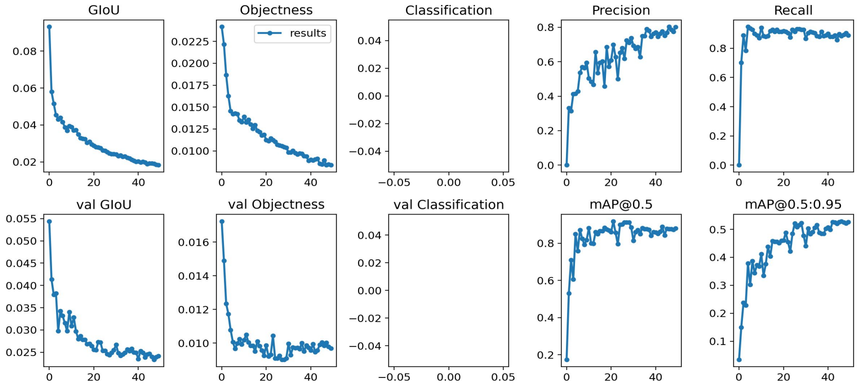
0.810 — Precision

0.841 — Recall

Results after training YOLOv5



Full video at: shorturl.at/pLR01



Vision Algorithm for ascending (Planning and Perception)

Perception and Path Planning

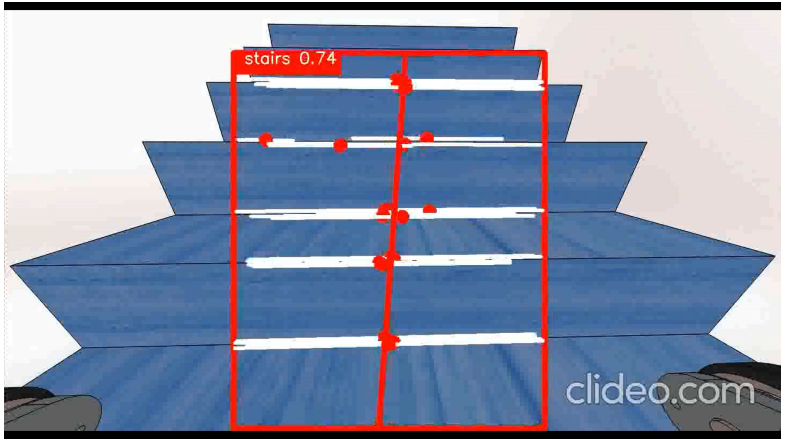
We run edge detection, linking and line fitting algorithm in real-time on the portion of image within the bounding box **(refer slide 10)**. This eliminates the side wheels and the bot body from the camera image.

- We obtain a fitted line denoting the desired orientation of the bot, compare it with the current bot orientation via the IMU and steer the bot towards the desired orientation using a PID controller.

- **If the slope of the planned path is greater than a certain threshold, we use the previous frame's slope**

Final output after running YOLO and this algorithm on our simulation

Full video at: https://youtu.be/ctwr45Y9hrs



- ➔ **The red rectangle denotes the bounding box showing confidence of stair detection by YOLO.**
- ➔ **The white lines denote stair edges.**
- ➔ **The red points are midpoints of obtained edges.**
- ➔ **The central red line denotes the desired heading orientation.**

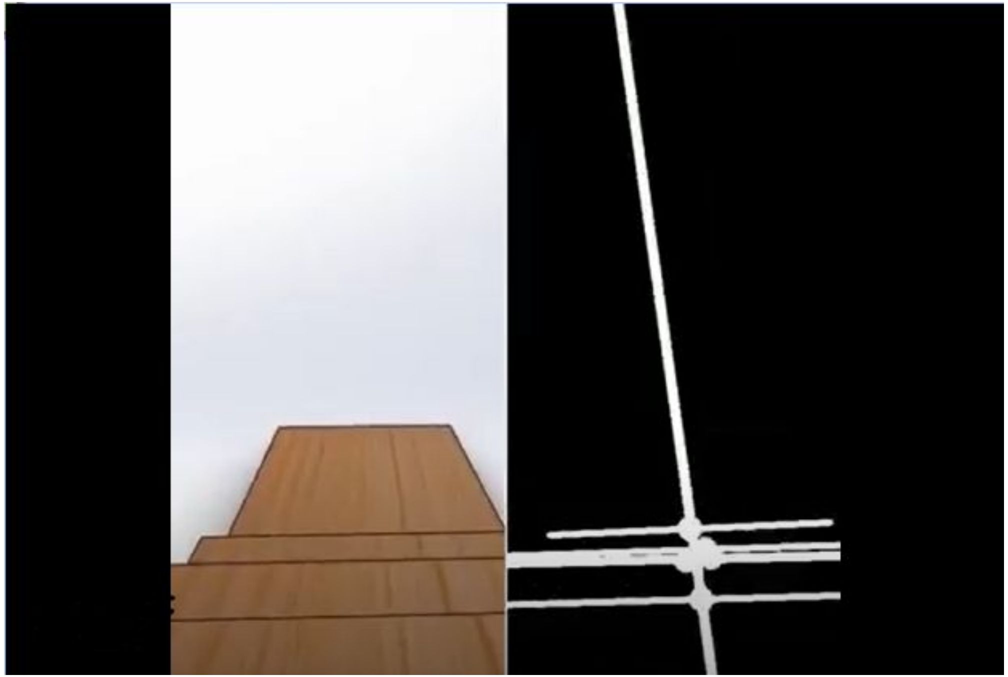
Vision Algorithm while Descending

Perception and Path Planning

We run edge detection, linking and line fitting algorithm in real-time.

We put thresholds **(abs(slope) < 0.05)** for slopes of detected stair edges to eliminate the side edges.

We obtain a fitted line denoting the desired orientation of the bot, compare it with the current bot orientation via the IMU and steer the bot towards the desired orientation using a **PID controller.**

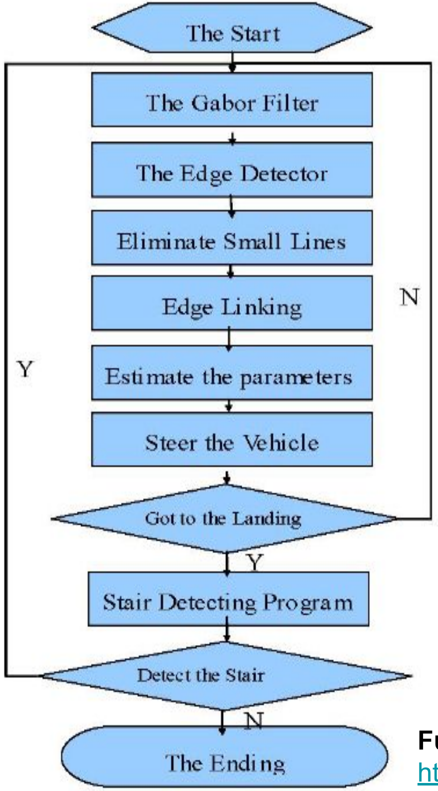


Simulation

Stair edges and fitted line

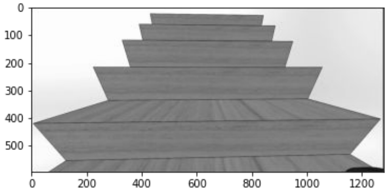
Simulation Video : shorturl.at/hFIUW

Edge detection, linking and desired heading

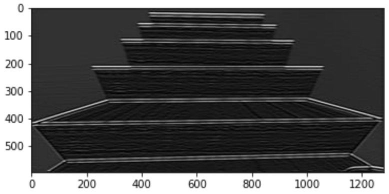


- **Gabor Filter** used to eliminate influence of illumination
- **Canny edge detector** on filtered image gives all edges in the image.
- **Connected Components** used to group nearby edges into components and assign hue values to each.
- Adjacent connected components are linked into long horizontal edges(stair edges)
- Midpoints of horizontal linked stair edges estimated.
- **fitLine** fits a line to the midpoints of stair edges based on the **M-estimator** (http://en.wikipedia.org/wiki/M-estimator) technique that iteratively fits the line using the weighted least-squares algorithm

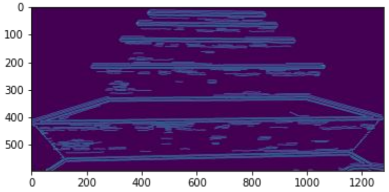
`Full notebook at:` https://github.com/susiejojo/Stair_Climber/blob/master/stairclimber_vision3.ipynb



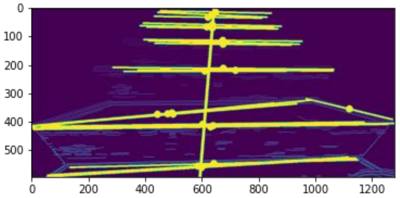
Original Grayscale Image



Filtering through Gabor filter



Canny edge detection



Edge linking and line fitting

Distance and Depth Algorithms

Motion on stairs of varying size is not easy in general.
Proposed design **can accomplish this task with ease !!**

Distance to stair

While climbing, the front wheel of the bot should be less than 5 cm from the starting of the stair. In that case the bot can easily climb 30cm or more. This is verified by placing ultrasonic sensors at the front of the bot with a calibrated distance. If the distance is more than threshold the bot enables driving motion to reduce the distance between stair and bot.

Distance_detected = ultrasonic_measurement

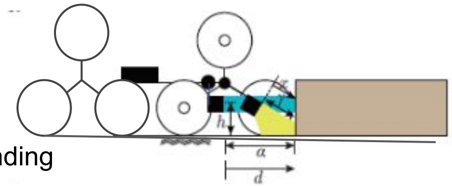
Depth detection

While descent bot needs to know that there are descending stairs and know the depth it is achieved by placing a tilted ultrasonic sensor. As we know the orientation of sensor from the calculated distance of ultrasonic sensor we can get depth of the surface.

Depth_detected = [(distance_detected)*cos(orientation_angle)] - (sensor_height)

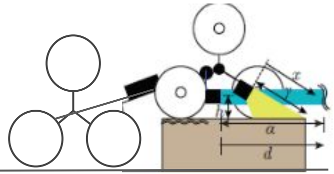
θ= 0 degrees
d <= a
x <=y
Increase power

Rolling → Ascending



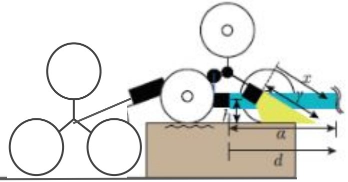
θ= 120 degrees
d > a
x <=y
Normal power

Ascending → Rolling

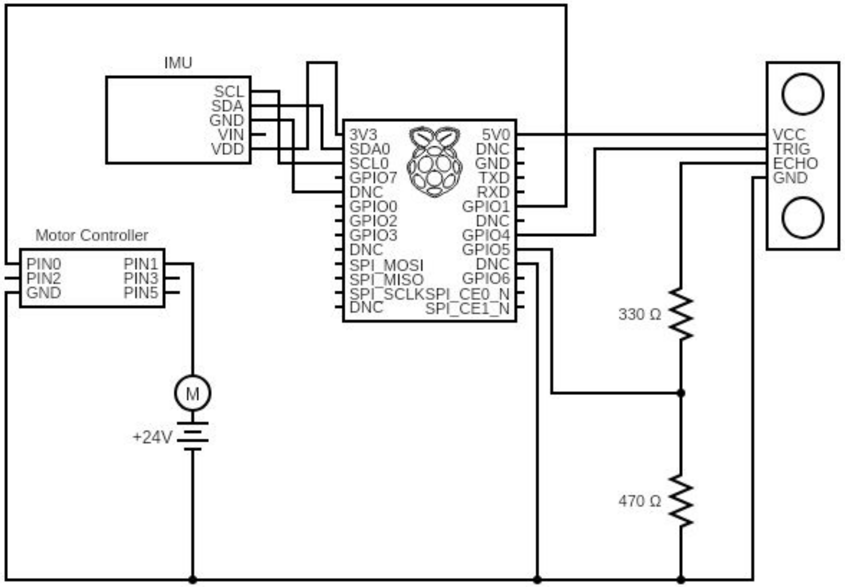


θ= 0 / 120 degrees
d > a
x > y
Decrease power

Rolling → Descending



Electronic Components



Circuit diagram denoting connection of Raspberry Pi with IMU, a single Ultrasonic sensor and DC motor with motor controller

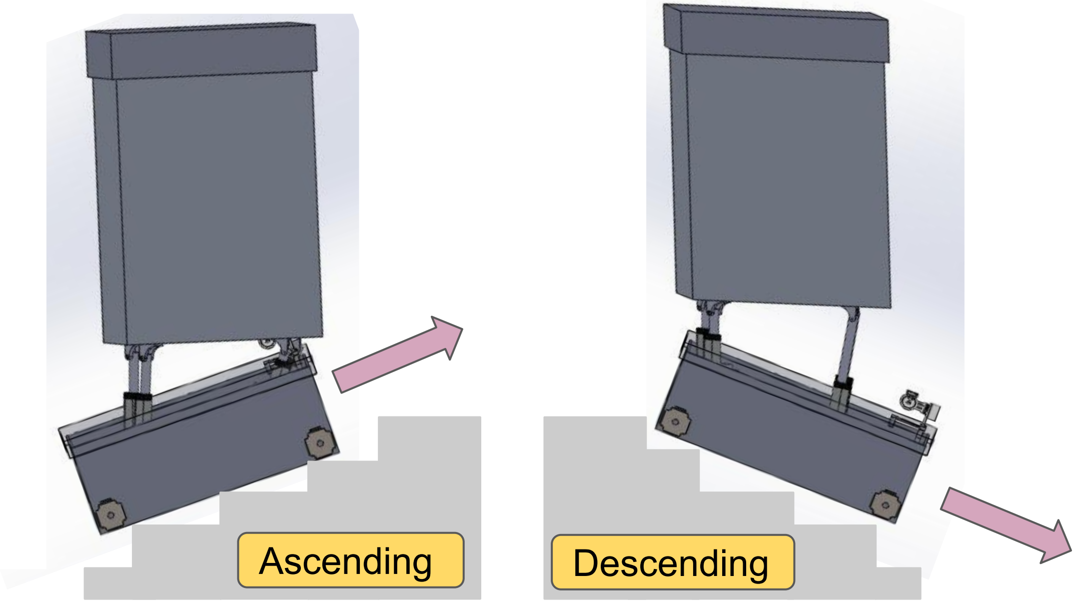
Component	Name	Purpose
Microcontroller	Raspberry Pi 3	Image Processing, Path Planning and Control
Ultrasonic Sensors	HC-SR04 connected via voltage dividers to Raspberry Pi	Depth and distance estimation
DC GEARED Motor	EBIKE MY1016Z3 Brushless DC geared motor(24 V, 350 W)	Regulating wheel motion
Gyroscopic Gimbal		Stabilize the camera in case of jerky motion
Camera	Raspberry Pi camera module	
Motor controller	24V for MY1016 350W	
RF circuit		Emergency remote

Load Stability

The Load Stabilization is done using **Linear Actuators**. The orientation of the payload is aimed to maintain payload orientation parallel to the ground.

Reason:

- **Reduces the torque about the wheel**
- **Prevents unnecessary movement of the load inside the box.**



Ref_orientation(0 deg)

+

–

Bot_orientation (IMU value)

Orientation - motor algo
M1 = K*(diff)
M2 = K*(diff)
M3 = -K*(diff)
M4 = -K*(diff)

Controller

Actuator

WIth all these mechanisms the bot can carry a maximum payload of 25kg!!

Payload Calculations

Torque calculations f bot without payload(50-55kg)	
Mode	Torque on each motor
On plane ground	
Driving	>=0.37 N-m
Tumbling	>= 18 N-m
45 degree inclination	
Driving	>=8.08 N-m
Tumbling	>=18 N-m

Torque calculations of bot with maximum payload(75-80 kg)	
Mode	Torque on each motor
On plane ground	
Driving	>=0.5 N-m
Tumbling	>= 24 N-m
45 degree inclination	
Driving	>= 9.899 N-m
Tumbling	>=24 N-m

Torque vs speed calculations (Non -ideal)	
Torque (N-m)	Speed (kmph)
14	>=5.833
13	>=6.28
12	>=7.6
11	>=8.618
8	>= 9.6
4	>= 19.2

The operating torque without payload : 14-16 N-m.The operating torque without payload : 18-20 N-m.

The calculations are based on the assumption that coefficient of friction between wheel and ground : 0.025 and gravity : 9.8 m/s^2

The torque vs speed calculations are done for 24v 350W motors (mentioned in components) assuming 80% gear efficiency.

Power consumptions details

Motor consumption

24V 350W geared DC motor rated @ 324 RPM @ 11Nm

Working Scenario :

1) 90% Stair Climbing + 10% on Flat Surface = 0.9x11x324/60 + 0.1x2x324/60 = 57.51W
2) 70% Flat Surface + 15% @45 Deg incline plane + 15% Stair Climbing = 0.7x2x324/60 + 0.15x9x324/60 + 0.15x11x324/60 = 23.76W

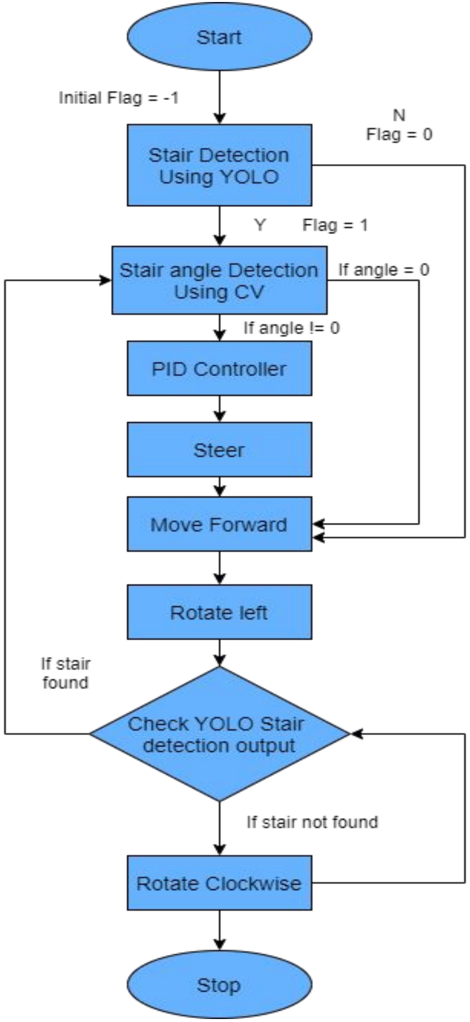
Sensors, Actuators & Camera

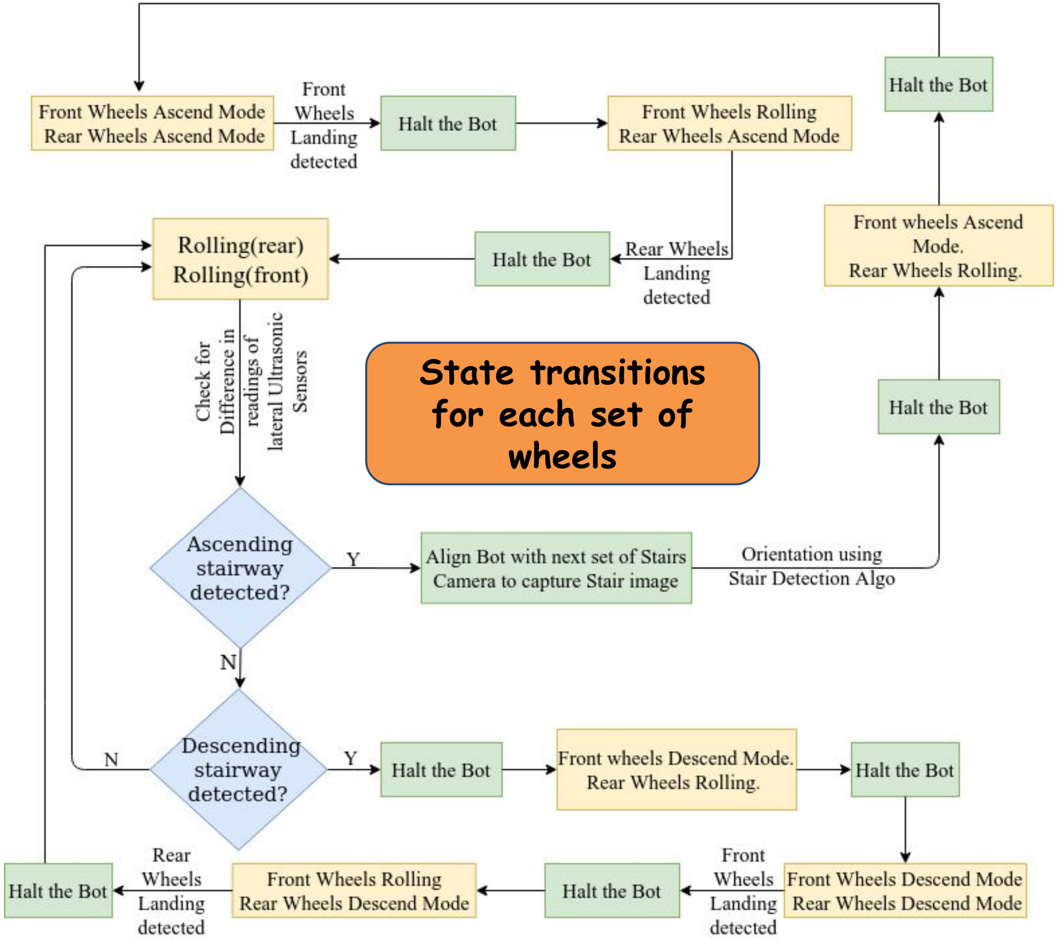
1) 8 - Ultrasonic Sensors 3.3V - 5V @ Operating Current 5 mA, Standby Current @ 14uA = 0.2W.
2) Raspberry Pi 3 250mA @ 5V= 1.25
3) 4 - Linear Actuators (for 5Kg Payload) - 12V @ 2A = 96W
4) 6 - Servo Motors 5V @200mA = 6W.
5) Raspberry Pi 3B - 5V @ 250mA 1.25W

Total Sensors, Actuator & Camera Power = 104.7W

- **Maximum Power consumption: 334.74W**
- **Distance the Bot can go upto on one full charge:** 7.5 - 65 Km depending upon the use case

Detailed Activity timeline





User Interface

Payload placement : A separate compartment is made for payload, user should place the payload in the compartment provided and close the compartment with the lid. A clip mechanism or more advanced locking system will be provided to close the lid of the compartment.

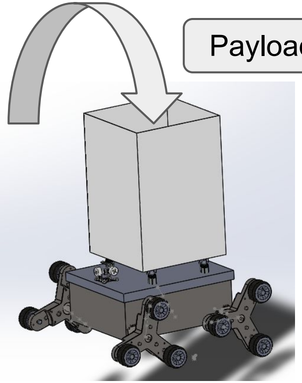
Charging the bot: A charging port is provided connecting all batteries inside the bot. A 48V charger can be used to charge the bot.

Setting up the bot :

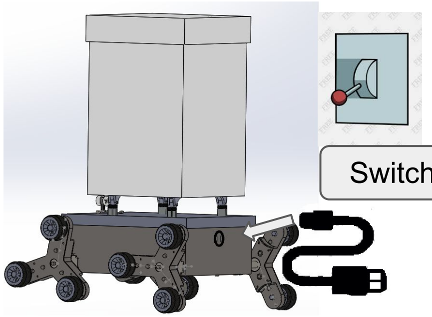
- A power switch will be provide to ON/OFF the robot.
- It is advised to place bot where it has a good view of the stairs
- In case of long journey one needs to provide starting and ending locations

Recovery mode: If the bot malfunctions it can be shut down using the emergency remote by standing at safe distance (max 500m) or it can be manually shut down using power switch of the bot.

If Bot detects any system failure then it alarms user, the user may shut it down using power or emergency shut down.



Payload Placement



Switch

Charging Port

Emergency Shutdown

Recovery Mode Design

If there is an obstacle in the path of travel the robot the robot comes to rest in no time by applying brakes so that collision is avoided.

If the bot is running out of charge while climbing stairs it uses its backup energy to hold the bot at the state to prevents falling and also alarms user about the charge.

From the taken measures and design of the bot there is very low chance of falling of the bot, but if it accidentally falls:

- The designed compartment contains shock absorbers will give good protection for the payload and avoid any damage to it.
- In case if the bot falls down, it detects it according to reading of IMU and immediately system failure will be alarmed to user.
- If the bot malfunctions it can be immediately shutdown using the **Emergency shutdown remote** provided or by turning off the power switch on the bot. This system is designed using 500+m LOS range with 2.4Ghz radio
- Additional fire detection also can be done by deploying temp sensors

Simulation/Videos

The motion simulations are done in Solidworks 2016. We chose Solidworks because of its robust physics engine which helped us to simulate accurately.

Links showing gears rotation in rolling and tumbling mode of bot:

- **Rolling mode :** https://youtu.be/Z-jkf5voGJo
- **Tumbling mode :** https://youtu.be/PUGKdtVIjqo

Links of the Bot ascending and descending on stairs and switching between states based on the situation:

- **Rolling on plane ground :** https://www.youtube.com/watch?v=KaJuw24jhKU
- **Tumbling on stairs :** https://youtu.be/Me_sMcBY8to

Output of the Stair Detection Algorithm

- **Camera output while ascending stairs :** https://youtu.be/ctwr45Y9hrs
- **Camera output while descending stairs** : https://youtu.be/GUMBIVCMhro

Cad models

- https://github.com/karan-13-hub/Tri-Wheel-Design
- https://github.com/karan-13-hub/Stair-Climber

Explain ease of portability of the bot

Distributed weight

The bot has **advantage of eight wheels**. As it has eight wheels the total force is distributed across eight wheels so this makes bot move in any terrain.

Robustness to terrains

Indoors the bot can easily avoid any climb any small obstacle to big steps in the way of its transport because of its tumbling mode of motion. It can also be carried outdoors in rocky, muddy or sandy terrain.

Support for large weights

Even if the weight of bot is bit high it is easily movable because of the amazing wheel design.

Speed and time of travel

- **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
- **Turning radius:** ~0 cm

- **With full speed the bot can climb up the 15 stairs of thread 30 cm and height 20 cm within 10 sec (ideally)**
- **The vision algorithm has latency of 0.104 sec per frame**

With 20 frames per second the bot takes 30-40 sec to cimb up the stairs. And for the descent it takes 40-50 sec.

Details of Electrical Components				
Component	**Price/unit**	**No of units**	**Source**	**Total Price**
Brushless DC Motor 350W	₹3500	4	Roland_electronics(HYD)	₹14000
Motor Controller - 350W	₹897	4	Controller	₹3596
25Ah 24V Li ion Battery	₹7500	2	Battery	₹15000
5Ah 12V Li ion Battery	₹667	1	small_battery	₹667
Raspberry Pi 3 Board	₹2575	1	Board	₹2575
Camera Module	₹410	1	Camera	₹410
Ultrasonic Sensor (HC-SR04)	₹99	6	Sensor	₹594
Servo Motors-1	₹329	2	Servo	₹658
Servo Motors Mini	₹99	6	Servo_mini	₹594
Linear Actuators	₹1000	4	Roland_electronics(HYD)	₹4000
			Total Cost:	₹42,490

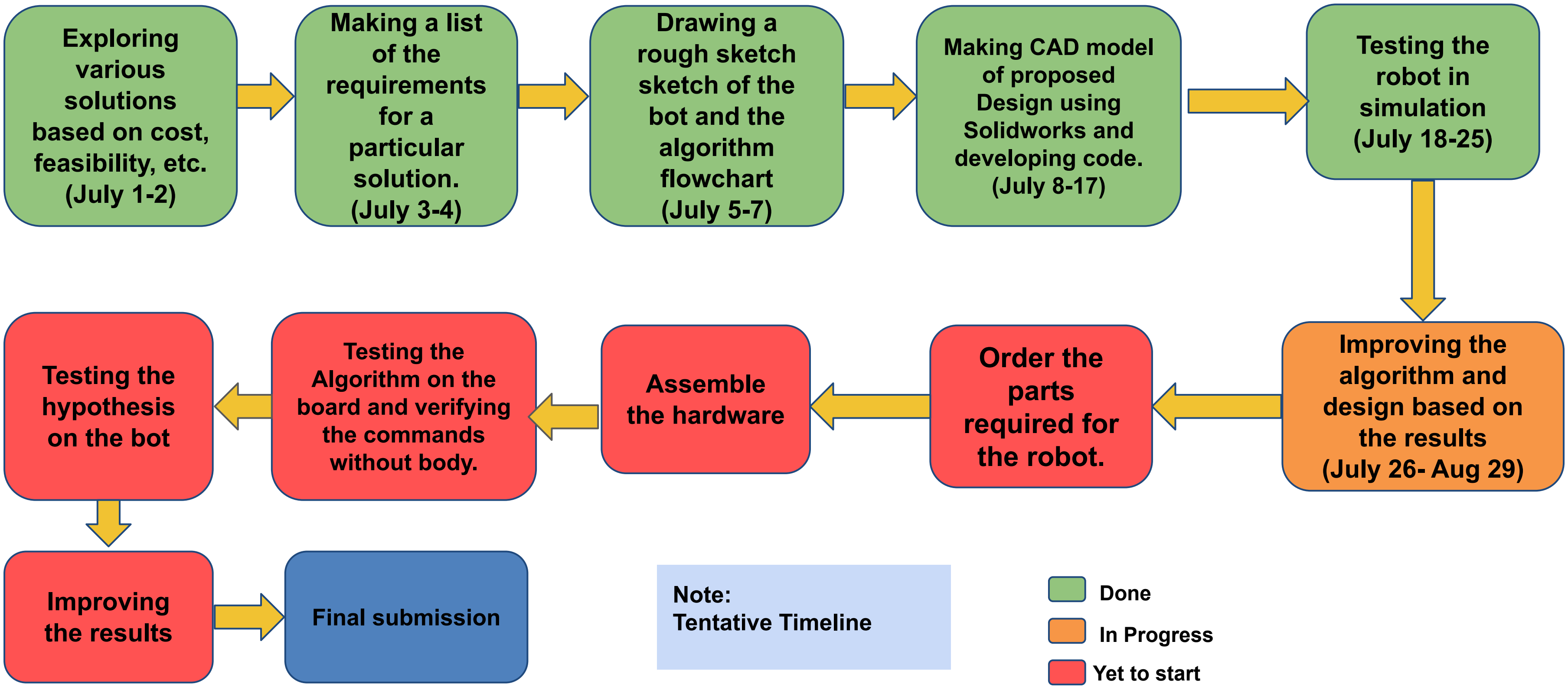
Details of Mechanical Components				
Item	Price/unit	No.of units	Source	Total price
Tri-star wheel	₹2500	4	Makers Lab IIITH	₹10000
Chassis with payload box	₹4000	1	Makers Lab IIITH	₹4000
			Total:	₹14,000

Total Estimated Cost

Electrical Components: ₹42,490
Mechanical Components: ₹14000

Net Total : ₹56,490

Execution Plan



References

- https://ieeexplore.ieee.org/document/8675676
- https://ieeexplore.ieee.org/document/4525517
- https://www-users.cs.umn.edu/~stergios/papers/stair_climbing_IROS02.pdf
- https://ieeexplore.ieee.org/document/4525517
- https://www.semanticscholar.org/paper/Stairs-Detection-Algorithm-for-Tri-Star-Wheeled-and-Thu/edd12c81adde5f042a3efe1b880fa853c86bb46e
- https://www.researchgate.net/publication/256942417_Mechanical_design_and_development_of_tri-star_wheel_system_for_stair_climbing_robot?enrichId=rgreq-94c88eb549b0f51bb614797dcb815d39-XXX&enrichSource=Y292ZXJQYWdlOzI1Njk0MjQxNztBUzoyODkzNzMwNTM1NzEwNzJAMTQ0NjAwMzMyMDI1OA%3D%3D&el=1_x_3&_esc=publicationCoverPdf
- https://ieeexplore.ieee.org/document/6385776
- https://pdfs.semanticscholar.org/08da/dc36b96498615f18864662fd2184e193a7bb.pdf?_ga=2.206971355.998524130.1593947392-395077216.1593947392
- https://www.semanticscholar.org/paper/The-Tri-Wheel%3A-A-Novel-Robot-Locomotion-Concept-the-Smith/2bb16204150b9640cbf8016b86c908ccab08f634