iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 Issue: VII Month of publication: July 2021

DOI: https://doi.org/10.22214/ijraset.2021.36816

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

Autonomous Stair Climbing Robot

Prayansh Shsank Joshi[1], Rithwik Chittereddy[2], Ishan Andola[3], Sreenethi Ganesh[4]
[1]*Vellore Institute Of Technology*

Abstract: An autonomous robot, also known as simply an autorobot or autobot, is a robot that performs behaviors or tasks with a high degree of autonomy. Autonomous robotics is usually considered to be a subfield of artificial intelligence, robotics, and information engineering. Specifically, autonomous robots can help: Increase efficiency and productivity. Reduce error, re-work, and risk rates. Improve safety for employees in high-risk work environments. In this COVID era every 4 out of the 5 Fortune 500 e-commerce companies and especially the e-commerce giants like Walmart , Amazon and Flipkart are focusing even more on contact less delivery. Delivery using automated bots is a concept that can be used even after the Covid situation gets over. The popularity of e-commerce buying is exploding, and the phenomenon has huge implications for shippers and distributors. To paraphrase Charles Darwin, "those companies best prepared to adapt to this new reality are the ones that will flourish while laggards will be left in the dust". Logistics is all about movement from one place to another. This movement happens across facilities and within facilities. While AGVs (AUTOMATED GUIDED VEHICLES) have found their way into applications of material movement inside a warehouse, they are designed to move across flat surfaces with minor undulations (in some cases requiring specialised mezzanine surfaces). This paper comes up with conceptual building and designing an autonomous robot which can work on flat surfaces and even climb up and down standard stairs with a payload of 3-5kg. This paper further will inform you about the conceptual building of an autonomous stair climbing robot involving all the steps like sensor fusion , navigation , tracking , mapping and path planning.

Keywords: Autonomous Robots , Navigation , Path planning , sensors , sensor fusion , optimal path , autonomy , odometry , IMU.

I. INTRODUCTION

A. Problem Statement And Solution

In an e-commerce scenario the last mile delivery is the phase where the customer experiences his desired ordered item. Sophistication in handling and delivery of the items will ensure customer confidence. Robots can be used to assist the "wish masters" make deliveries ergonomical resulting in efficient delivery. This kind of automation can make a really good case for delivery of large items like fridges/ washing machines in multi floor apartments without lifts. Otherwise this can be a very tedious human task which will require a good amount of labour. They can also find applications in goods movement within the warehouse.

Hence , the aim is to come up with an autonomous robot which can work on flat surfaces and even climb up and down standard stairs with a payload of 3-5kg(considering a smaller bot version) and even upto 100 kgs(considering a bigger bot version).

B. Objectives

The robot needs to demonstrate the following capabilities :

1) It must be able to climb up and down a flight of stairs without dropping a package.
2) The robot must be able to turn on a quarter landing and re-align it to the next flight of stairs.
3) There should not be any possibility of toppling.
4) Turning radius should be as small as possible.

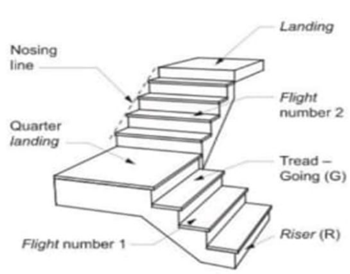

Fig 1. This represents the environment where the bot will start from the bottom of the stairs , then move all the way up and finally will come back to the starting point.

C. Functionalities Of Robot

This bot will be able to function in both remote controlled as well as autonomous ways.

This bot will be able to self lock the box to be delivered , then climb the stairs and upon reaching the target location will be able to unlock the box and the customer can take the delivery.

This robot will be able to remember the path it followed while going to the destination hence makes the bot much faster while returning. Additionally it will map down the environment using the SLAM(SIMULTANEOUS LOCALIZATION AND MAPPING) algorithm.

This bot will be able to interact with the customer using the customer friendly UI in case the customer wants to replace or return the order.

Irrespective of the number of alignment of stairs our bot will be able to deliver the package and come back.

In the case of a crowded stair, the bot will be able to avoid the obstacles coming in the way.

The bot's live location could be assessed by the delivery man as well as the customer in case someone tampers the bot.

II. CONCEPTUAL BUILD

A. Sensor Fusion

Sensor Fusion is basically combining 2 or more sensors in a way that generates a better understanding of the system. By better understanding , we mean more accuracy , better precision , more consistency and hence more dependable.

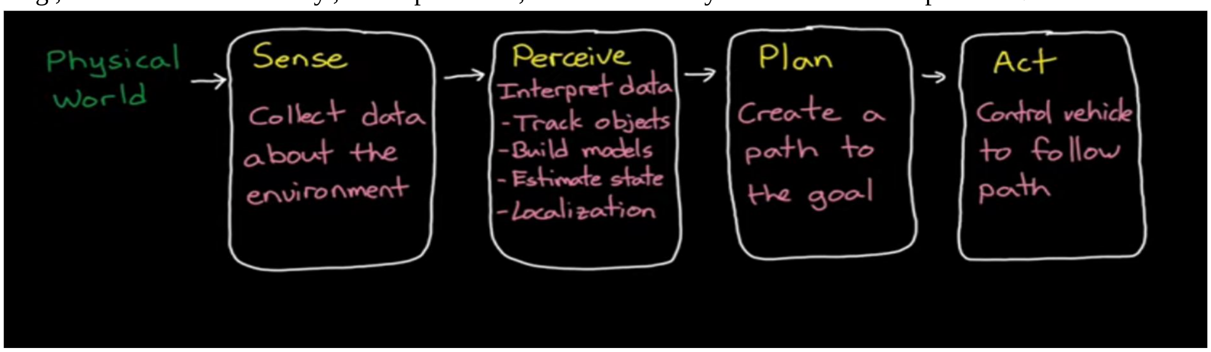

Fig 2. Capabilities of an autonomous system.

Now the above blocks represent the various capabilities of an autonomous system.

The first one being Sense is the ability of the system to sense the surrounding and get data about the surrounding. This can be done using various sensors for example in our bot it is LIDAR , SONAR , camera and many more.

The second one being Perceive is the ability to bring out sense in the sensor sensed data.

Without proper interpretation the data collection is of no use . For example if my bot cannot differentiate between a person and a stationary dustbin then the data collection is of no use.

Sensor Fusion happens at the first 2 steps that are Sense and Perceive.

Now the advantages of Sensor Fusion are **-**

- *It Can Increase the Quality of Data:* When a sensor records data then the data can come out to be noisy. For example in our case we have an accelerometer and if it collects data regarding the vibrations or acceleration or the orientation then it will come out to be noisy. So instead we can use 2 accelerometers and take the average of them(remember they shouldn't be correlated) and hence we will have less noise. So in this way we reduced the noise by using the very simple Sensor Fusion algorithm which is an averaging function. Also fusion of sensors like this reduces the noise by a factor of square root of the number of sensors.

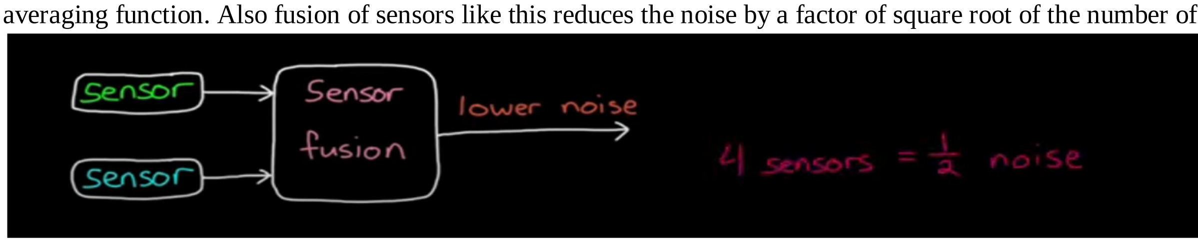

Fig3 . This figure shows how noise can be reduced to half by using a assembly of 4 sensors

Now this is not what we planned to use in our Stair Climbing robot but instead we choose to have an assembly of 3 different sensors which are magnetometer , accelerometer and gyroscope. By selecting 2 different sensors we have completely nullified the probability of noise of these 2 being correlated and these will be used to calibrate each other.

Now when we observe a change in orientation by accelerometer, then the gyroscope can verify whether it came from the bot or the environment and hence gives us an increased accuracy.

Now there are various algo that can be used to achieve this but what we have used is the kalman filter. Kalman Filter has the mathematical model already built inside the filter and hence we can use both the functionalities of the sensors as well as the environment.

- *It Increases Reliability:* Duplicating sensors is a way in which we can still have the data even if one of the sensors fails , though the data won't be as precise as it earlier was but still we will have some data in hand. Also an assembly of duplicate sensors can be used and data from the best of 2 or 3 can be taken into account to further increase the reliability.
- *It Can Measure Unmeasured States :* Here unmeasured states doesn't mean the states that can't be measured but which can not be measured using the particular sensor. For instance, let's assume I have a camera and I wish to find the distance of a far object using the camera. Now the camera can not differentiate between 2 objects(of similar pixels) one placed near and one far. Now however if we add a second optical sensor and extract 3-D information then the fusion algorithm can compare the scenes from 2 different angles and can measure relative distances between the objects in the 2 images. In this way these 2 sensors can't measure distance individually but they can when combined.
- *It Can Increase Coverage:* Now for our Stair Climbing Robot , it needs to have a larger vision of obstacles around it so just placing one Ultrasonic sensor in the front won't help. Hence , I have used 6 ultrasonic sensors which can provide data from each front.

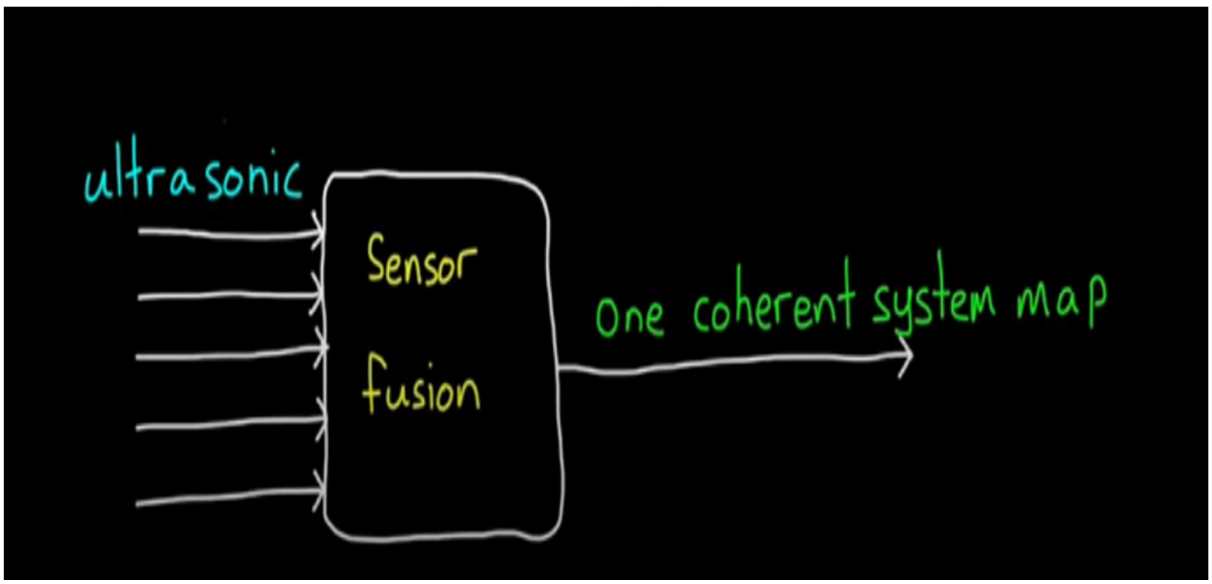

Fig 4. This figure shows how the ultrasonic sensors can be combined to form one coherent map.

1) *Sensor Fusion For Orientation:* As stated earlier we will use a sensor fusion of accelerometer , magnetometer and gyroscope. In general, an accelerometer and magnetometer can be used to determine the orientation of the bot but in our case , we have a moving bot and hence , we also combine the assembly with a gyroscope to increase accuracy and for better precision.

Now to define an orientation we will require 2 things -

a) Reference frame (this is the normal x,y and z axis)

b) Specific rotation frame

Now let's say we have a phone lying on the table.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

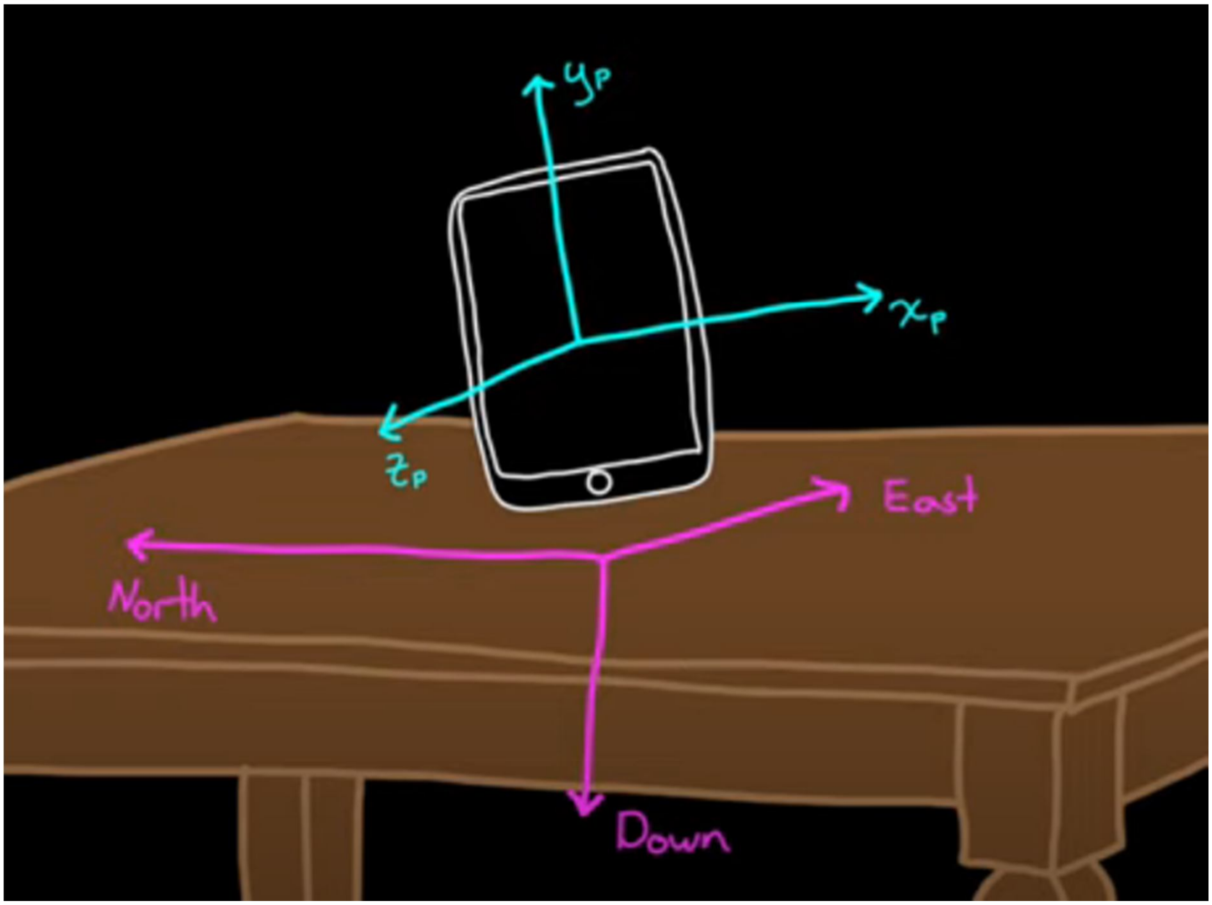

Fig 5. Now the direction with the cyan is the specific rotation axes as these will change when the phone's position changes and the one in pink is the reference frame.

Now from the above figure we can say that the acceleration will be due to gravity and will be opposite to down direction, allso the magnetic field will be in the north direction , the same is shown below.

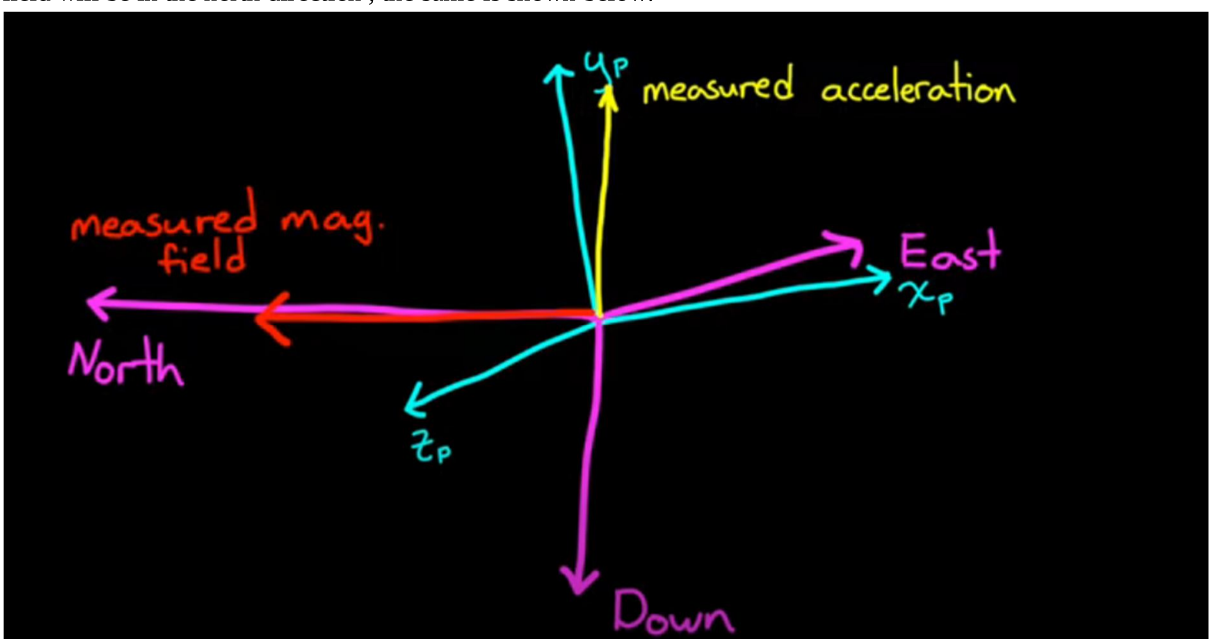

Fig 6. This figure shows how direction can be determined.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

Now suppose we are moving or we are in some other parts of the world then the magnetic field will vary from north to the direction of gravity(typically for North America , it is 60 degrees towards gravity).
So now to determine the directions we first assume just the directions of the magnetic field and the measured acceleration.

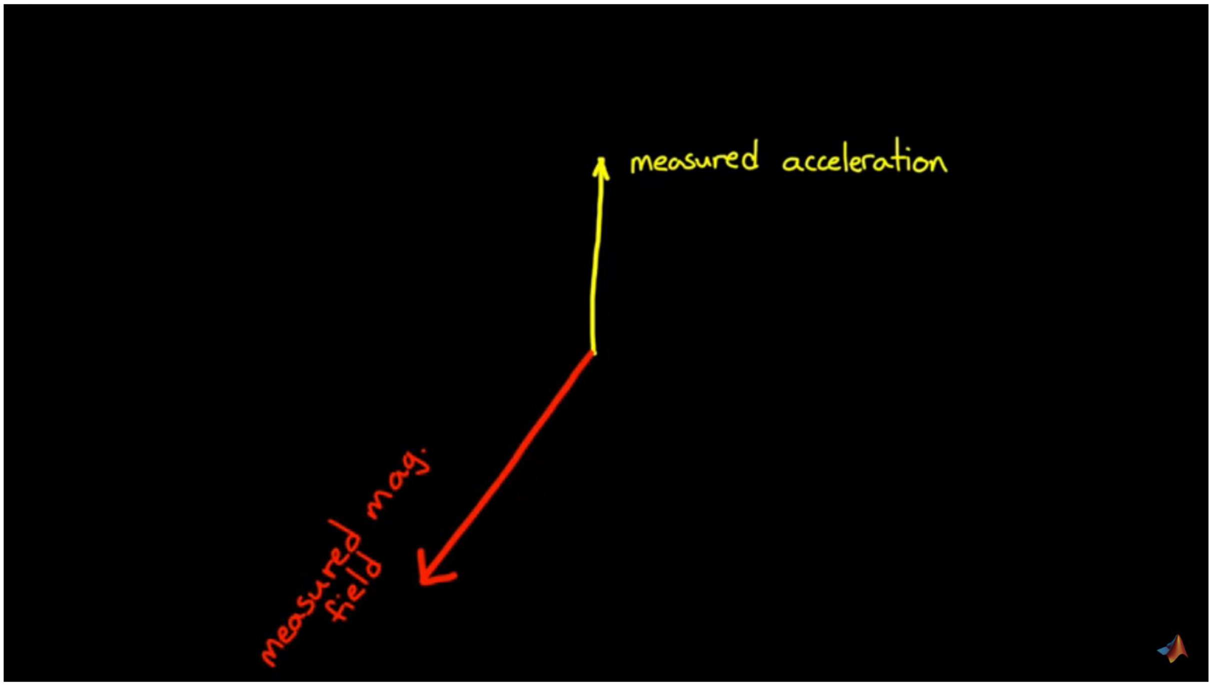

Fig 7. Directions of magnetic field and acceleration.

Now down is the direction opposite to measured acceleration.
East is the cross product of measured magnetic field and down direction.
Finally , north is the cross product of east and down.

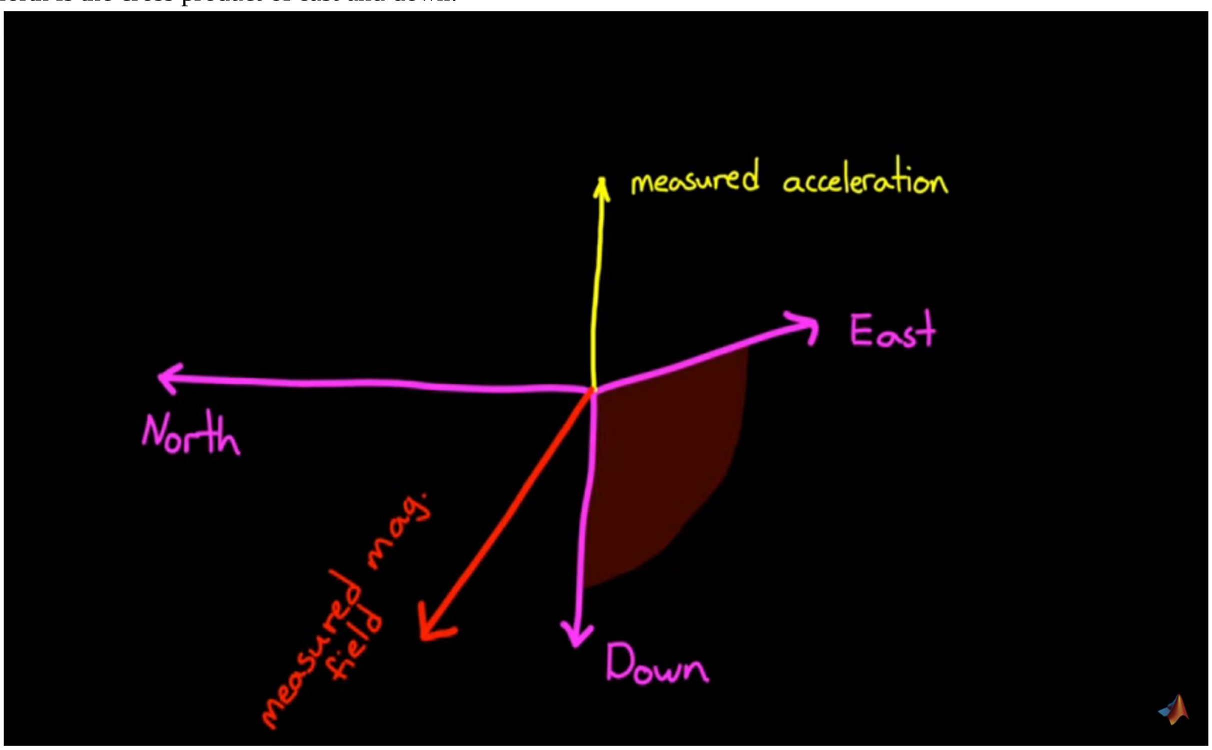

Fig 8. The final predicted axes.

ACCELEROMETER + MAGNETOMETER		**GYROSCOPE**
Produces absolute measurements		Produces relative measurements.
Corrupted by common disturbances		Needs initial orientation
		Gets drifted over time
	USING KALMAN FILTER	
	ACCELEROMETER + MAGNETOMETER + GYROSCOPE	
	Initialize the orientation	
	Use magnetic field and acceleration to correct the gyro shift.	

The above block shows how finally we can achieve all the objectives using the 3 sensor assembly.

2) *Adding GPS To The Sensor Fusion:* Now for the next part in addition to the orientation , we also need to have information about the velocity and the position of our bot. Now , If we simply place a GPS it will work in the situations where the direction and the velocity is changing relatively slower and hence a few meters off would not make a big difference. However , in our case the bot will change its position and velocity at a faster rate and will need position updates every second. So when the motion is faster then GPS alone can't get the work done and hence it is used with IMU(Inertial Measurement Unit).

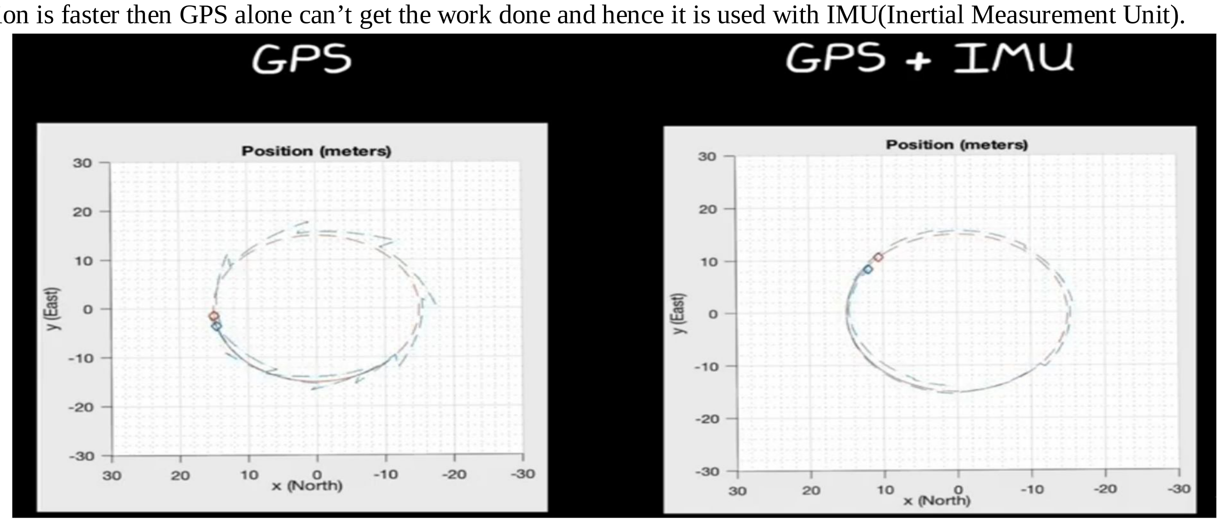

Fig 9. This shows how GPS, if used alone, can lead our stair climbing bot to inaccurate paths and when used with IMU can help out stair climbing to optimally move around the path.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

B. Tracking

Goal in this module is to determine the state of a remote object by fusing results from sensors and models.

A RADAR sensor(this will be replaced with a LADAR sensor in further stages) and a Camera module would be used to track the object(s).

1) *Single Object Tracking:* Generally when we have just one object to track then it is relatively easier as compared to multiple objects tracking. When we just have one object to track then we generally use multiple filters. For our example we need to detect whether it's moving at the same speed or it is taking a turn or it is accelerating.

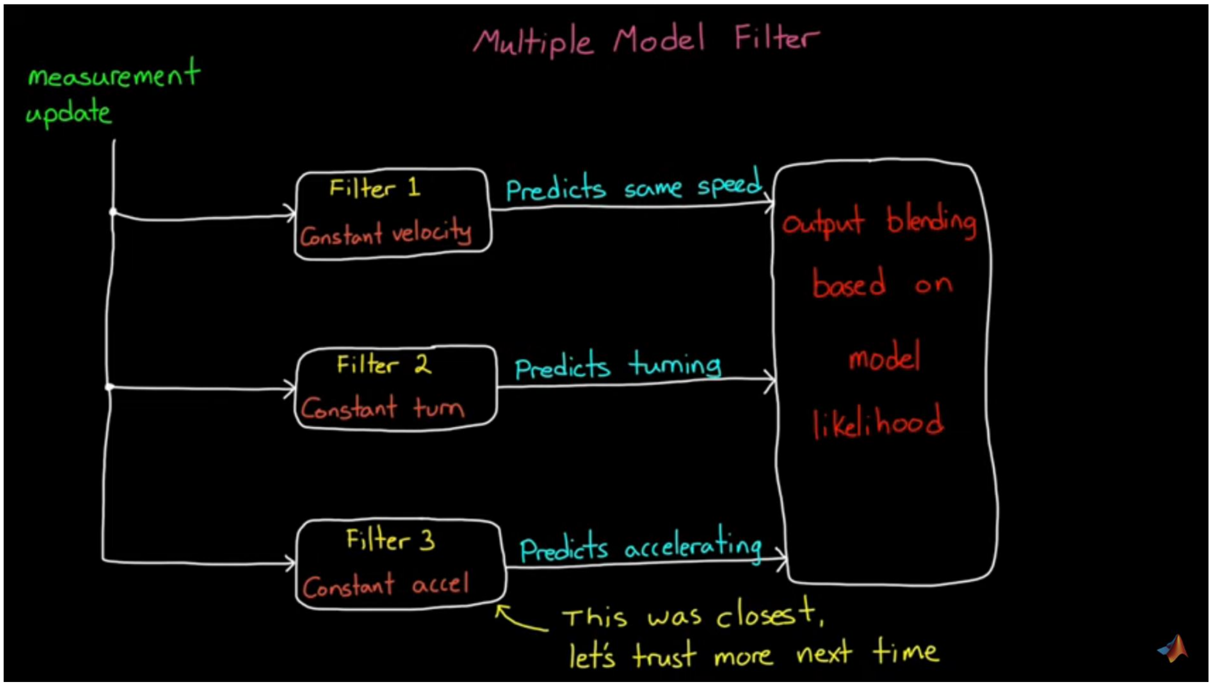

Fig 10 . This shows how multiple filters can be used to achieve the required goal.

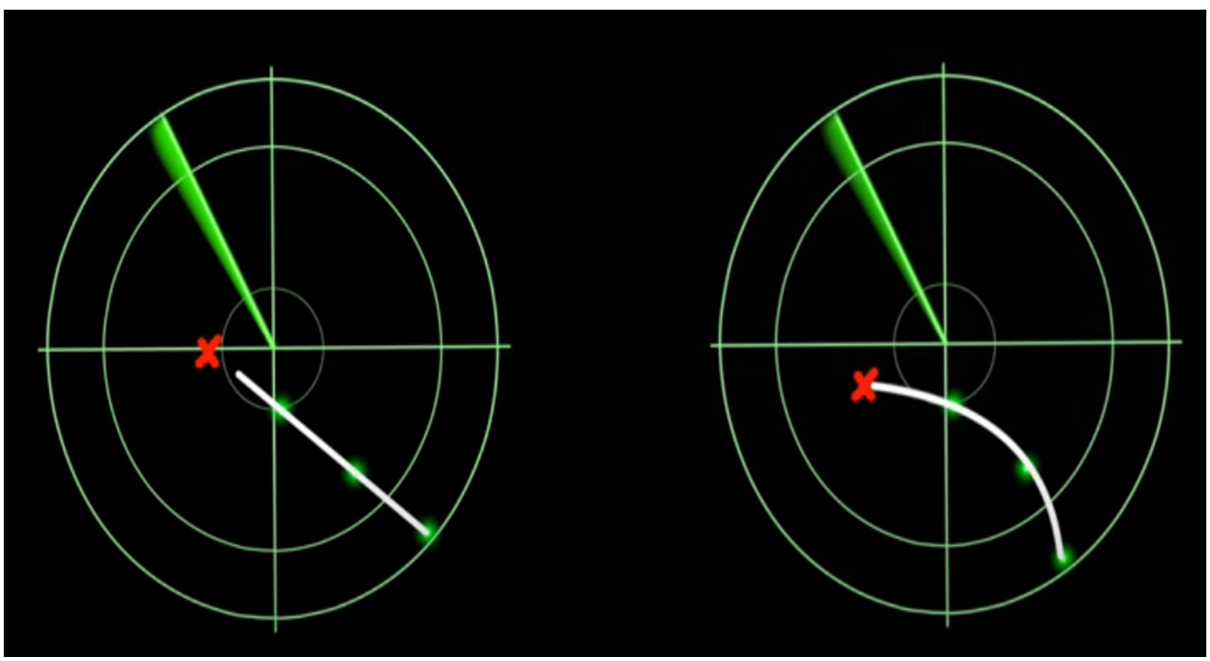

Fig 11. These filters combined can pretty much now think like a human does. If a person is moving forward they will predict that it continues to move forward , if a person is moving with acceleration that it will keep moving with that acceleration.

2) Multiple Object Tracking

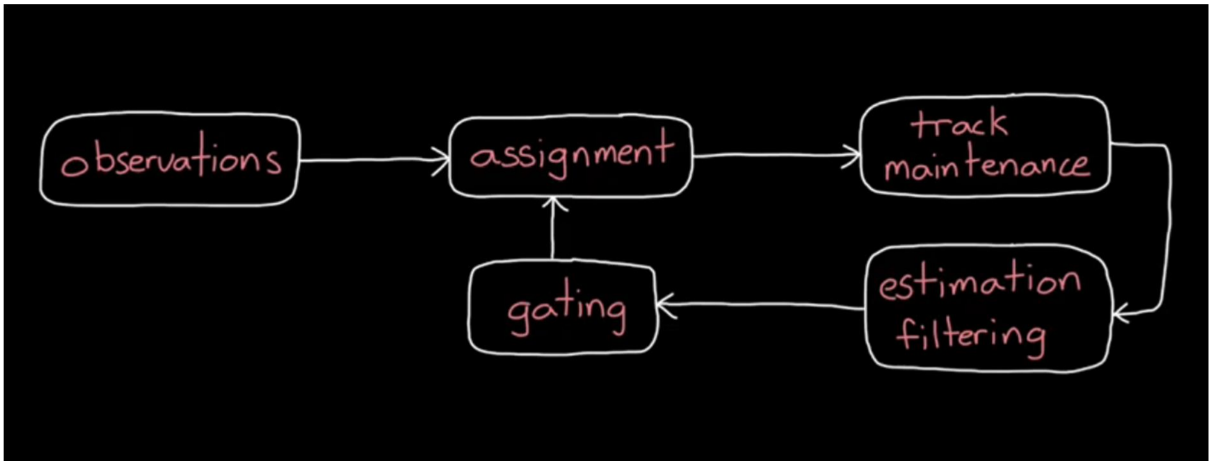

Fig 12. Steps for multiple object

a) *Observation:* The first step is observation where we check how many objects are there in the surrounding and take them into consideration using the sensors.
b) *Assignment:* Now the second part is handling the observations . Now, many algorithms can be taken into account one of them being the JPDA (JOINT PROBABILITY DATA ASSOCIATION). In JPDA it makes a weighted combination of the various objects while giving the nearby object a higher weight and hence ignores the ones which are at very far distance as they would hardly contribute to the system.
c) *Track Maintenance:* Now once we have the objects to track in our hand we take their trajectory into account and the tracks followed by them. The track is created once the object enters the range and is deleted once the object crosses the range.
d) *Estimate Filtering:* This one is the same as we did in the case of single object tracking.
e) *Gating:* This is the process in which we ignore the far placed objects as they would hardly contribute to our environment and the system.

Now these all need to be accompanied by a Sensor which can provide them all the inputs.
Hence , for our system we will use a LADAR sensor. Now this is very close to a LIDAR sensor but more useful for less complex surroundings.
We prefer LIDAR over RADAR , although Radar sensors being highly reliable they aren’t really capable of telling where the radar target actually is.
Now like the LIDAR sensor, the LADAR sensor keeps rotating and emits thousands of lasers in one second which gets reflected back by the obstacles and the objects.
These light reflections are then used by the on board computer to create a 3-D cloud of the surrounding which is then converted into an animation and using the speed of the light and the distance converted by it to know the exact location.
Pre-Scan - A pre-scan is a process where the LADAR sensor laser scans the road surface several hundred times in a second and readings are then passed on to the on-board computer which in fraction of seconds performs the computing and the suspensions of the wheels are then adjusted to maintain a safe distance or to stop or to keep moving or accelerating.

C. Navigation

Navigation is the ability to determine your location and plan a path to some goal.
Navigation in the wilderness may require you to have a GPS to know my location and a map to locate the optimal path through the lakes and the mountains to my camp side.
Now autonomous navigation is the same but without humans. Autonomous navigation is the ability of a vehicle or a robot to determine its location using a set of sensors and plan a path on to it’s goal.
Now there are various types of autonomy given to the mobile vehicles or robots.
It can be controlled by using some humans from a remote location and that case is tele-operations and can range upto fully autonomous robots like the one in our case.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

Now full autonomy can be achieved broadly in 2 ways -

1) *Heuristic Approach:* In this way we set some predefined rules that our bot will follow.
Say for example we give instruction to a maze bot that needs to keep the wall to its left and proceed. This way doesn't need much information about the environment and is generally not the best option.

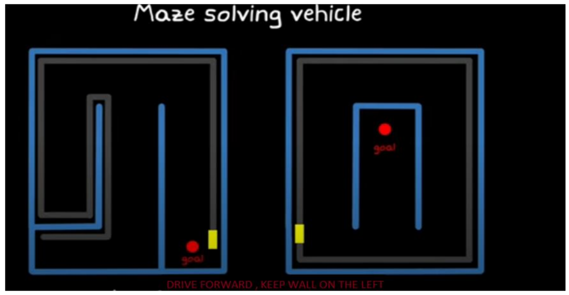

Fig. 13 : This figure demonstrates how the bot(in yellow colour) is choosing the path(in blue colour) when given a rigid rule to keep the wall on the left and move forward.

2) *Optimal Approach:* This way is completely dynamic. In this method we will need information about the surrounding and in this approach planning is achieved using optimization.

In this approach the bot models out the area or modify the given model to find the optimal path and proceed to its goal.
Now an example of this is Self Driving Cars , unlike the maze solving bot we can't give these a solid instruction like keep moving forward and keep the curb on right.
So these are the 2 main approaches used but in some cases accompanying either of these can sometimes put us in trouble say for example assume you are travelling in a car and there is a slower car in front of you and by optimal approach we need to give the instruction to change the lane but for changing the lane your car requires more information other than the slow car's position. So here only using the optimal approach won't work. Instead we can use a combination of both heuristical as well as the optimal approach. Like for this example, if you give a heuristic approach like to always pass the slow vehicle whenever it is safe to do so, it automatically takes into account the possibility of changing lanes like in the case of optimal approach.
Combination of these 2 approaches has been increasingly used in the case of warehouse bots , search and rescue bot , UAV , spacecrafts like OSIRIS - REX used for space missions like collecting samples from the asteroid venue that were not covered before and various others.
Hence for our application too we will be using a combination of both. Remember Autonomous Navigation isn't easy because environments aren't known. For example designing an autonomous spacecraft orbiting the earth is an easier task than an autonomous aircraft at least in terms of environment complexity because space is much more predictable than air and then autonomous aircraft is a simpler problem as compared to an autonomous vehicle.

D. Mapping

Now for navigation we need to map out the optimal path.
We will proceed with 2 possibilities:

1) *Map Being Given:* Now we assume that we already have a map in hand.

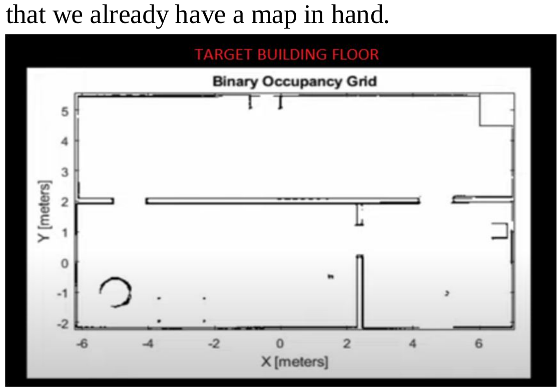

Fig . Assume this is the map we have in hand

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

Now as told in previous modules we are using a LADAR sensor which will emit the laser beams and will the data about what's in the front.

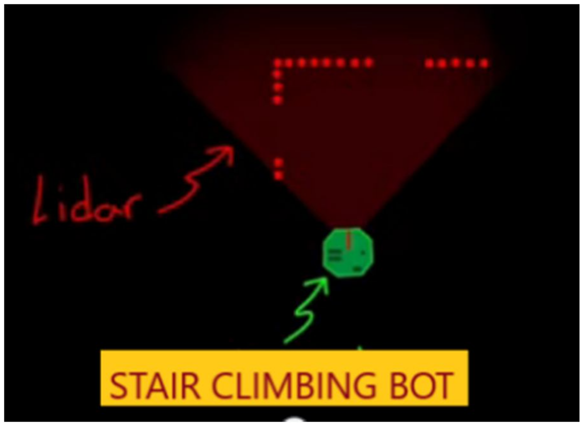

Fig 14. This represents what the LADAR Sensor has detected.

Now we can decide the position of the robot or the place by matching the observation within the given map.
DeadReacon can also be used to predict the position by odometry to predict the final position.
As mentioned earlier, to have better accuracy a combination of both can be used.

2) Not Being Provided The Map

Now the robot is on it's own to sense the environment and create a map of the surrounding on its own.
We are still available with the LADAR sensor and the odometry achieved using wheel encoders or IMU. Now let's assume we have an ideal situation in hand and both the LADAR as well as odometry are 100% accurate. In this case the task of the robot is pretty simple as it just has to move around and map the whole place.
The below figure states how the robot is able to check for the obstacle using the LADAR readings and is able to store the obtained images on the global map or the overall map.

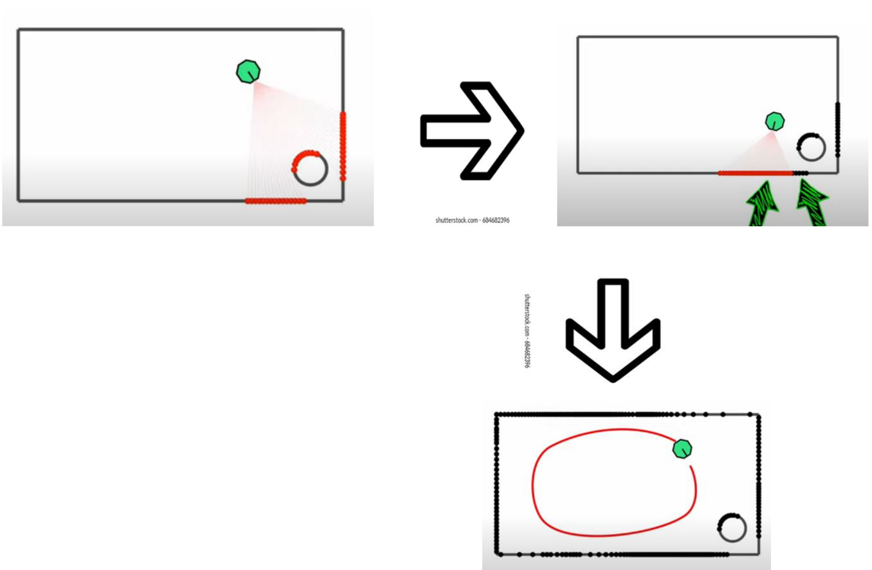

Fig 15. Stages for determining the map and the path using LADAR and Odometry.

However , the 100% accuracy situations are far from reality. It is generally observed that there is some randomness in the reading from the LADAR as well as odometry.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

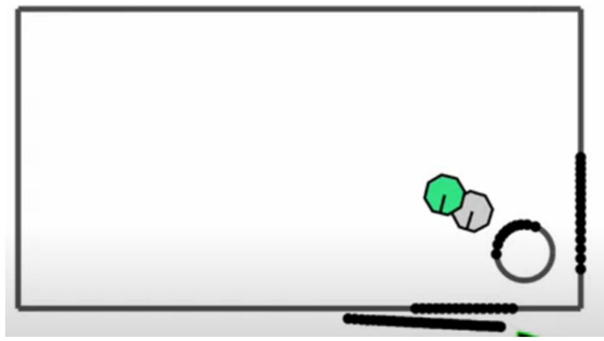

Fig 16. This shows how the ideal boundary detection should be and how the false boundary is being detected.

So the task at hand is to estimate our bot's location and orientation and simultaneously map the environment with better accuracy. This can be achieved using a process called Simultaneous Localization and Mapping. Out of the various algorithms possible inside this we will take into consideration :

a) *Filtering Algorithm:* Filtering algorithm is like the EKF(EXTENDED KALMAN FILTER) in which the state is estimated on the go with only the latest measurements.
b) *Smoothing Algorithm:* This uses the Post Graph Optimization method.

Full trajectories are estimated using the complete set of measurements.

SLAM basically helps in aligning the mis-aligned poses taken by the lidar sensors and is done using the combination of IMU sensors which keeps recording the relative distances and finally when the bot returns at the initial position , we get the randomly taken LADAR images aligned to give a map. This map is also called a pose graph.

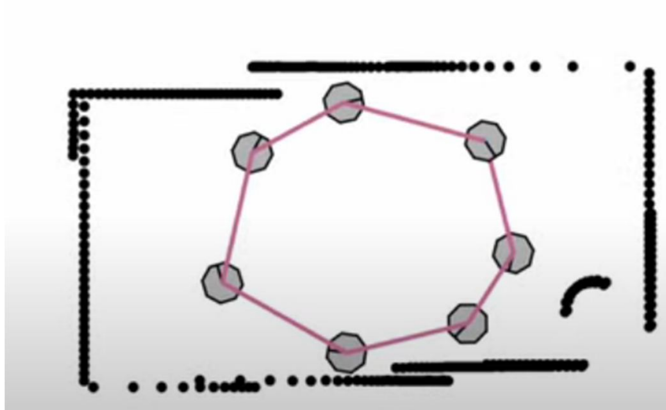

Fig 17. Pose Graph

E. Path Planning

This is the subset of motion planning. In the case of motion planning we are concerned about the path , velocity , acceleration and various other factors.

The only goal here is to achieve an optimal path and this can be done using various search based and sampling based algorithms.

SEARCH BASED ALGORITHMS	SAMPLING BASED ALGORITHMS
A*	RRT
JUMP SEARCH	RRT*

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

1) *Graph Based Solution:* Assume our bot is on the ground floor and the current location is marked as the **start pose** for the bot. Now we wish to reach some location marked as a goal **pose**. Now we have left our bot to reach the location , every move the bot makes takes some cost and the cost depends on how long the move or the step is. Suppose our bot has reached the final goal pose and the total cost is 20 units. Now we will again start this from the start pose and calculate the cost and this process is repeated until we find the path where the cost is the lowest.
2) *Search Based Solution:* In this method we make a grid based map and go cell by cell and determine the cost which the robot will take when moving.

For example, in the assumed grid the horizontal as well as the verticals cells are at distance of 10 units and the diagonal cells are at distance of 14 units.

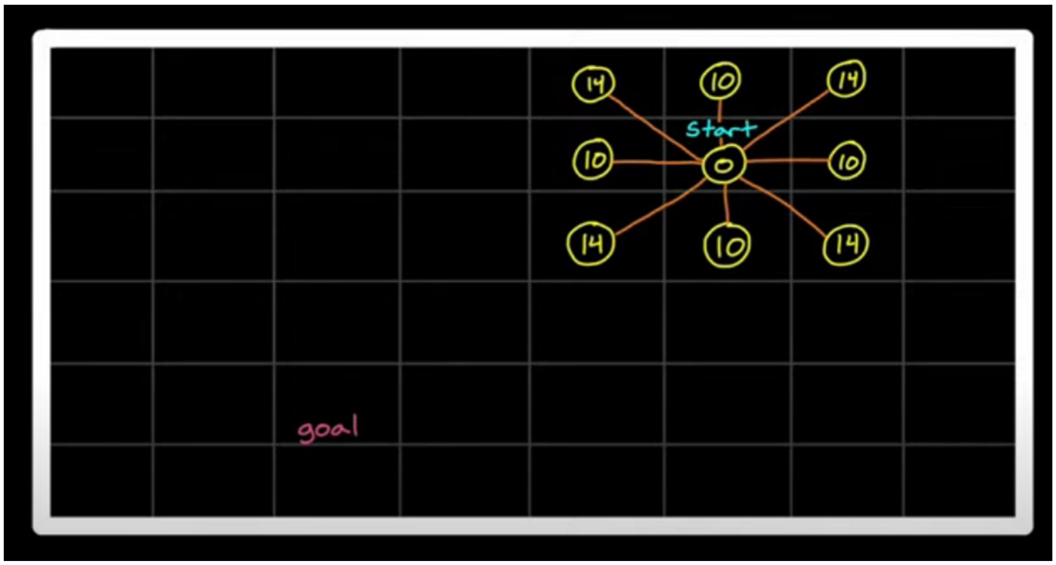

Fig 18. This shows the unit distance assumptions.

Now the bot is made to move from the start to the stop and the path for which the cost is minimum will be considered as the optimal path.

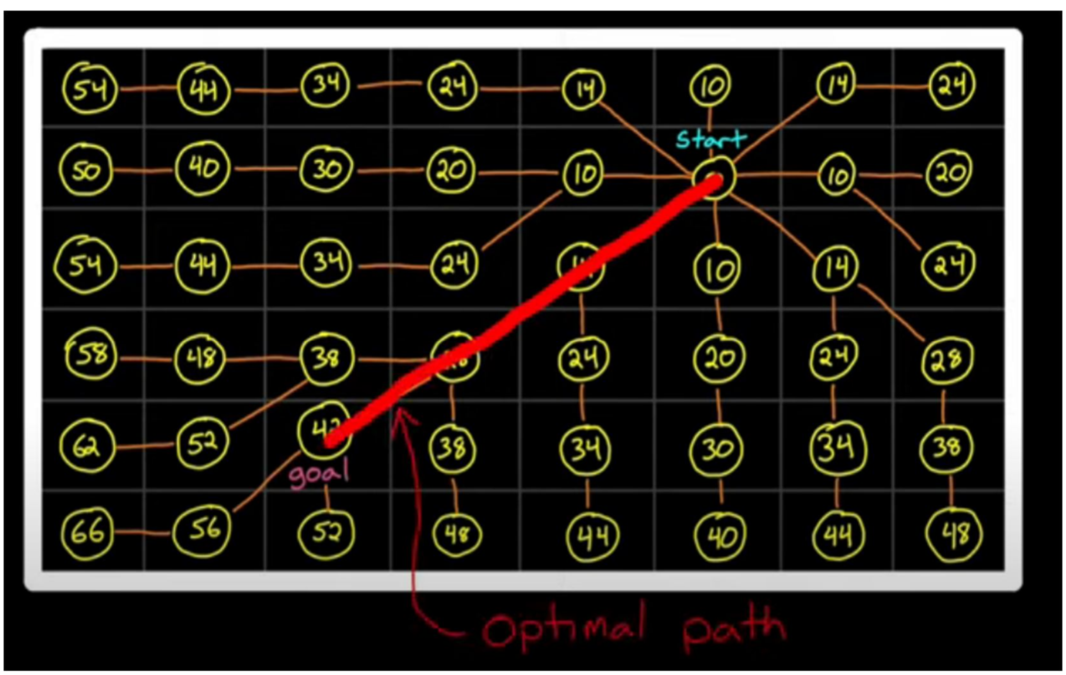

Fig 19. This shows the path with minimum cost and the optimal path.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

III. DESIGN

A. Components

Sr. No.	Name	Quantity
1.	Cytron	2
2.	Raspberry Pi 4	1
3.	Arduino Mega	1
4.	LADAR	1
5.	Camera	1
6.	*Gyroscope	1
7.	*Accelerometer	1
8.	*Magnetometer	1
9.	GPS	1
10.	UltraSonic Sensor	2
11.	Motors(Rated Torque: 0.8 kg-cm and 600-800 RPM)	4
12.	Stepper Motor	4
13.	SONAR Sensor(for testing purpose)	1

*(One also also use IMU sensor which is basically a combination of accelerometer , magnetometer and gyroscope but as explained in 2.1 Sensor Fusion(Pg.5.6))

Sr.No	Name	Quantity
1.	Cytron	2
2.	Raspberry Pi 4	1
3.	Arduino Mega	1
4.	LIDAR	1
5.	Camera	3
6.	Gyroscope	1
7.	Ultra Sonic	8
8.	Infra-Red	2
9.	Stepper Motor	4
10.	Motors(High Torque and High RPM)	4
11.	SONAR	1

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

B. Dimensions

S.NO	NAME	LENGTH	BREATH	HEIGHT
1	BASE CHASSIS	500		100
2	UPPER CHASSIS	640	440	110
3	ELECTICAL BOX	250	150	40
	SLIDING MECHANISM			
4	SLIDING BOX	30	30	40
5	SLIDING PLATE	270	20	10
6	SUCTION PLATE	200	50	5
S.NO	**NAME**	**DIAMETER**	**HEIGHT/LENGTH**	
1	MOTOR HOLDER	37	250	
	PNEUMATICS			
2	CENTRE CYLINDER	38	150	
3	PISTON	12	200	
4	SUPPORT ROD	9	200	
5	PNEUMATIC INTERCONNECTING ROD	30	410	
6	PNEUMATIC BALANCING ROD	45	210	
			WIDTH	**SHAFT DIAMETER**
	GEAR (WHEEL)			
1	MAIN WHEEL	100	25	6
2	CENTRE GEAR (WHICH IS CONNECTED TO MOTOR)	50	8	6
3	2nd GEAR (WHICH IS CONNECTED TO CENTRE GEAR)	50	8	6
4	3rd GEAR (WHICH IS CONNECTED TO MAIN WHEEL AND 3RD WHEEL)	50	8	6

C. Stair Climbing Robot Design

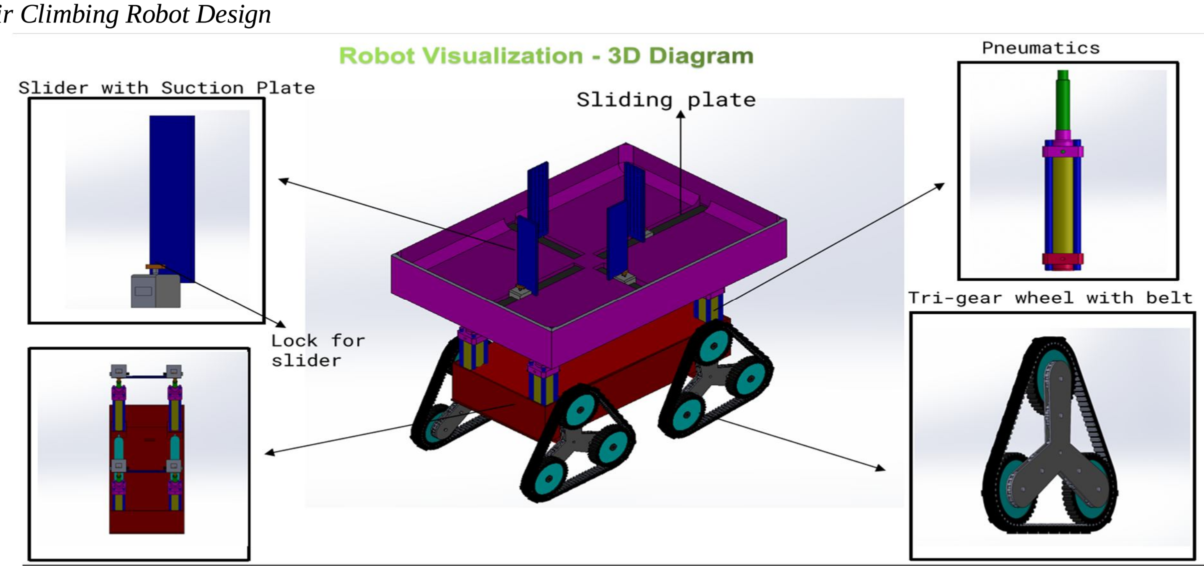

Fig. This is the 3-D visualization of our stair climbing robot.

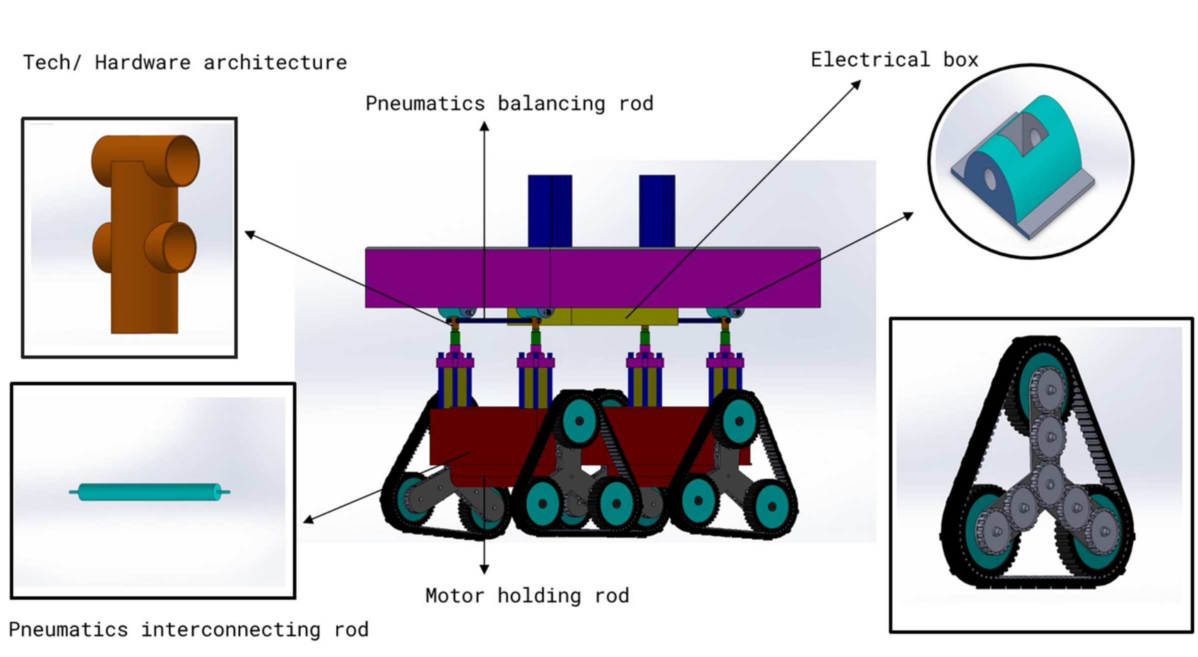

Fig. This shows the architecture of our stair climbing robot.

IV. EXECUTION PLAN

The delivery man places the box to be delivered on the upper plate of the bot. On each of the four suction plates, an ultrasonic sensor is placed which detects the box and starts moving towards the box which will be achieved using linear actuators which moves the plates placed above the sliders. Now the plates have suction pumps spread uniformly on it. When the distance between the box and plates as calculated with the help of Ultrasonic sensors becomes 0 then the suctions stick the box to it from all the four sides. Now a locking system is kept behind all the plates which are called “one-touch lock systems”. This is a manual locking system but with the help of a stepper motor we will automate the locking task as when the motor turns 90 degrees, the plates will be locked and when rotated through the back this will be unlocked. This is done to eliminate any chances of the item getting stolen. The lock could be unlocked by the customer by entering the right OTP provided to them at the time of ordering.

We have used pneumatics to balance the box kept in the upper body. A gyroscope will be used to detect the angle by which the bot is climbing and according to the angle we use DCV(Directional control valve) to control the pneumatics so that the pneumatics would stretch keeping the center of mass static and balancing the box kept a. The piston can rise to 30-35cm so that it can ensure that even when a refrigerator or heavier object is placed, it can balance and keep the center of mass static and at the center. We are also planning to add flippers that will ensure smooth climbing, coming down, and landing.

Our bot will have 3 cameras, 2 for looking at the extreme ends, and one in the center to align our bot towards the center of the stairs. LIDAR sensors of a range around 30cm-200cm will be placed at the top and towards the back as it serves as an eye for the autonomous bots. The algorithm which we will be using is an integration of the RANSAC(Random Sample Con algorithm to detect our desired plane in point cloud data and MDL(Minimum Description Length). This will help to avoid the wrong detection of planes due to the complex geometry that 3D data possesses.

This algorithm will be used to detect the stairs irrespective of the width and height and once detected it will find a straight line of points located towards the middle of the stairs and would align itself towards the selected line of points using the camera in the middle.

This algorithm can be applied to the whole set of stairs at once as well as to each coming stair. Further to assure a more precise movement PID algorithm would be used. Once the bot is climbing it will save in memory the data points for the covered path so that while returning it follows the same data points which will save time. To identify that it has reached the base ultrasonic sensors are placed at the bottom in the four corners.

Also, 2 IR sensors would be placed at both the right side as well as the left side to detect the next set of stairs and will help it to turn with a minimum radius. Further, a SONAR sensor would be used to detect if any person or object is approaching the bot and will be able to detect it.

V. BRIEF ON PROGRAMMING MODULE

There are two main sections in the system, the Autonomous Movement Section and the Self Stabilizing Selection. When the slider with suction plate touches the box, it is detected by an Ultrasonic sensor(when the distance becomes 0) the slider is locked using a stepper motor and this gives a signal to both our systems and the suction plates are activated. The Self Stabilizing Section contains a gyroscope and pneumatics. Using the gyroscope, we get the angle of the base when we are climbing the stairs and with the help of DCV, we control the pneumatics. So that it automatically pushes up the back of the base with the box and pulls down the front of the base so that the base with the box is always stable and parallel to flat ground. Here we are using Arduino Mega, hence we use Arduino Embedded C/C++.

The Autonomous Movement Section consists of a few sub-sections. First, the LIDAR sensor senses the area around it. We put this data through the algorithm where RANSAC removes the outliers and is used to detect and extract the planes in each plane and the MDL principle is used to segregate and match how many planes are there in each block. So, RANSAC identifies the planes and by integrating it with MDL, we avoid detecting wrong planes in the complex geometry of the 3D data. Now using these planes we compute the midpoint of each plane and this is used by the cameras, at both the extreme ends of the front side of the bot, to align the midpoint of the bot to the midpoint of stair by keeping both the cameras equidistant to the midpoint of the stair. This way we are aligning our bot to the middle of the staircase and then we climb the stairs.

We also keep two sensors on the left and right side of the bot to detect where the next stair is going to be detected and the bot is going to turn towards that side and the RANSAC plane detection algorithm continues the rest. This system will be activated when the 4 ultrasonic sensors at all the corners facing towards the ground had the same distance.

This Autonomous Movement Section has a lot of computing as it is going to continuously run the whole trip, so we are using a Raspberry Pi and will be programming it in Python as we will be using concepts like OpenCV.

If needed we can also incorporate a feed for the delivery guy to observe the bot. This will have two modes. There will be a camera with the lidar sensor which will give feed for the first mode. The second mode entails the plane detection overlapped with the camera feed which will give an augmented reality feed for us to see where the bot is detecting the planes of the staircase. This will help when the bot is being remote controlled and we need to know where the stairs and their midpoints are. There can be another mode which entails night vision to see in dark places.

VI. FUTURE SCOPE

As a safety measure, the box carried by the bot can only be unlocked when the customer enters a specific OTP allotted at the time of ordering and one can also apply 2 step verification at the time of delivery.

This bot can be made to interact with the customer using a UI(user interface) in case the customer wants to replace or return the order. The bot movements can be made even faster by using modern algorithms. Pick and Place mechanisms can also be designed which can automatically place the delivery package on the bot.

VII. LIMITATIONS

The limitation can be on the speed of the bot while climbing up when carrying heavy loads such as refrigerators or washing machines. Our bot currently won't be able to pick up the box by itself. Our bot can quickly come back as it stores the path it covered. But once some obstacle comes to it's path then it needs to deviate it's path hence it again needs to repeat the same process as it has done while climbing which can take time.

REFERENCES

[1] https://www-users.cs.umn.edu/~stergios/papers/stair_climbing_IROS02.pdf
[2] https://ris.utwente.nl/ws/files/103953896/Yang2010Plane.pdf
[3] Kaucic, R., R. Hartley, and N. Dano (2001). Plane-based projective recon_x0002_struction. Computer Vision, IEEE International Conference on 1, 420.
[4] https://www.youtube.com/watch?v=9D5rrtCC_E0
[5] J. D. Martens and W. S. Newman, "Stabilization of a mobile robot climbing stairs," in Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2501–2507, May 1994.
[6] Biosca, J. and J. Lerma (2008). Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. ISPRS Journal of Photogrammetry and Remote Sensing 63 (1), 84–98.
[7] Brenner, C., N. Haala, and D. Fritsch (2001). Towards fully automated 3D city model generation. In In Automatic Extraction of Man-Made Objects from Aerial and Space Images III.
[8] https://www.youtube.com/watch?v=9D5rrtCC_E0