Mansoura University
Faculty of Engineering
Mechatronics Engineering Program

Autonomous Delivery Robot

Ahmed Nabil Hamed Abdelrahma Badawy
Ahmed Tarek Yehia Abdel Latif
Ahmed Walid Mohamed Bedair
Bishoy Attia Roshdy Attia
Gamal Ahmed Gamal Edin Habib
Mahmoud Alaa Abdelsamea Nasif
Mohamed Alaaeldin Mohamed Ibrahim
Mohamed Badwy Elsaid Badwy
Mohamed Bedier Mohamed Ibrahim
Mohamed Moustafa Aboroya Abdelghafar
Mostafa Basem Badwy Saafan
Rawan Gamal Aboelhasan
Ziad Mohamed Fekry Abdelhaq

UNDER SUPERVISION OF:
Associate Professor Abeer Tawakol
Electronics and Communications Engineering Department
Eng. Ahmed Ramadan
Mechatronics Engineering Department

Acknowledgments

The success and outcome of this project required a lot of guidance and assistance from many people, and we are extremely privileged to have got this all along with the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.
It is a great privilege for us to express our profound gratitude to our respected **Dr. Abeer Tawakol** and **Eng. Ahmed Ramadan** for their constant guidance, valuable suggestions, supervision, and inspiration throughout the project work, which it would have been difficult to complete the work without them. Their inclination for incubating new ideas and delivering the solution was always an important source of inspiration and motivation for us.

Abstract

In our project, our aim is to overcome one of the modern problems of this century, which is autonomous robot navigation and localization. Using Robot Operating System (ROS), we percept the surrounding of the scene around our robot using Lidar and Kinect. By Appling Simultaneous localization and mapping (SLAM) algorithm from the generated Point Cloud, our robot will be able to generate a map of the surroundings and move in it blind folded by generating the required path trajectory in the halls. Our robot will contain a locked box which will contain the delivered objects and by using barcode, it will not open until it reach the location and recognize the desired barcode. Also, Due to Covid-19 precautions, there will be a feature that will be able to guarantee checking the local health authority guidance. By using this tools we aim to use our robot in working in different places with multi-purpose tasks, as when it enters the desired workspace, it will read it and by using the scanned map it will go to the desired place that we choose on our controlling interface web to get something there.

Table of Contents

List of Figures

List of Abbreviations

AMR	Autonomous Mobile Robot
AGV	Autonomous Guided Vehicle
Lidar	Light Detection and Ranging
Radar	Radio Detection and Ranging
ROS	Robot Operating System
SLAM	Simultaneous Localization and Mapping
ACML	Adaptive Monte Carlo Localization
DIY	Do it Yourself
TCP/IP	Transmission Control Protocol / Internet Protocol
UDP	User Datagram Protocol
SDK	Software Development Kit
PPR	Pulse Per Revolution
IMU	Inertial Measurement Unit
AHRS	Attitude and Heading Reference Systems
MEMS	Micro Electromechanical System
VGA	Video Graphics Array
CMOS	Complementary metal-oxide-semiconductor
ML	Machine Learning
EKF	Extended Kalman Filter
RGBD	RGB-Depth Image
RTAB-Map	Real-Time Appearance-Based Mapping
KLD	Kullback-Leibler distance
TF	Transformation Matrix
PV	Process Variable
SP	Set-Point
FCE	Final Control Element
PID	Proportional Integral Derivative
ROI	Region of Interest
HOG	Histogram of Oriented Gradients
PCB	Printed Circuit Board
SMPS	Switched-Mode Power Supply
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
IGBT	Insulated Gate Bipolar Transistor

IC	Integrated Circuit
PWM	Pulse Width Modulation
NO	Normally Open
NC	Normally Closed
USB	Universal Serial Bus
D.O.F	Degrees of Freedom
ICR	Instantaneous Center of Rotation
FEA	Finite Element Analysis
F.O.S	Factor of Safety
FVK	Forward Vehicle Kinematics
FOV	Field of View
SSF	Static Stability Factor
CAD	Computer Aided Drawing

1. Introduction

Mobile robots have the capability to move around in their environment and are not fixed to one physical location. Mobile robots can be "autonomous" (AMR - autonomous mobile robot) which means they are capable of navigating an uncontrolled environment without the need for physical or electro-mechanical guidance devices. Alternatively, mobile robots can rely on guidance devices that allow them to travel a pre-defined navigation route in relatively controlled space (AGV - autonomous guided vehicle). By contrast, industrial robots are usually more-or-less stationary, consisting of a jointed arm (multi-linked manipulator) and gripper assembly (or end effector), attached to a fixed surface. Mobile robots have become more commonplace in commercial and industrial settings. Hospitals have been using autonomous mobile robots to move materials for many years. Warehouses have installed mobile robotic systems to efficiently move materials from stocking shelves to order fulfillment zones. Mobile robots are also a major focus of current research and almost every major university has one or more labs that focus on mobile robot research. Mobile robots are also found in industrial, military and security settings. Domestic robots are consumer products, including entertainment robots and those that perform certain household tasks such as vacuuming or gardening.

Mobile robots have the capability to move around in their environment and are not fixed to one physical location. Mobile robots can be "autonomous" (AMR - autonomous mobile robot) which means they are capable of navigating an uncontrolled environment without the need for physical or electro-mechanical guidance devices. Alternatively, mobile robots can rely on guidance devices that allow them to travel a pre-defined navigation route in relatively controlled space (AGV - autonomous guided vehicle). By contrast, industrial robots are usually more-or-less stationary, consisting of a jointed arm (multi-linked manipulator) and gripper assembly (or end effector), attached to a fixed surface. Mobile robots have become more commonplace in commercial and industrial settings. Hospitals have been using autonomous mobile robots to move materials for many years. Warehouses have installed mobile robotic systems to efficiently move materials from stocking shelves to order fulfillment zones. Mobile robots are also a major focus of current research and almost every major university has one or more labs

that focus on mobile robot research. Mobile robots are also found in industrial, military and security settings. Domestic robots are consumer products, including entertainment robots and those that perform certain household tasks such as vacuuming or gardening.

Wheeled robots are robots that navigate around the ground using motorized wheels to propel themselves. This design is simpler than using treads or legs and by using wheels they are easier to design, build, and program for movement in flat, not-so-rugged terrain. They are also better controlled than other types of robots. Disadvantages of wheeled robots are that they can't navigate well over obstacles, such as rocky terrain, sharp declines, or areas with low friction. Wheeled robots are most popular among the consumer market, their differential steering provides low cost and simplicity. Robots can have any number of wheels, but three wheels are sufficient for static and dynamic balance. Additional wheels can add to balance; however, additional mechanisms will be required to keep all the wheels on the ground when the terrain is not flat. In our project we introduce a robot with good navigation and localization technique to solve a lot of problems mentioned above in this section.

Figure 1-1 Our Autonomous Delivery Robot

1.1. Introduction to Wheeled Robots

Most wheeled robots use differential steering, which uses separately driven wheels for movement. They can change direction by rotating each wheel at a different speed. There may be additional wheels that are not driven by a motor these extra wheels help keep it balanced.

1.1.1. Two Wheeled Robots

Two wheeled robots are harder to balance than other types because they must keep moving to maintain upright. The center of gravity of the robot body is kept below the axle, usually this is accomplished by mounting the batteries below the body. They can have their wheels parallel to each other, these vehicles are called dicycles, or one wheel in front of the other, tandemly placed wheels. Two wheeled robots must keep moving to remain upright and they can do this by driving in the direction the robot is falling. To balance, the base of the robot must stay with under its center of gravity. For a robot that has the left and right wheels, it needs at least two sensors. A tilt sensor that is used to determine tilt angle and wheel encoders which keep track of the position of the platform of the robot.

Figure 1-2 Roomba Robot

Figure 1-3 Segway Robot

1.1.2. Three Wheeled Robots

Three wheeled robots may be of two types: differentially steered (2 powered wheels with an additionalfree rotating wheel to keep the body in balance) or 2 wheels powered by a single source and a poweredsteering for the third wheel. In the case of differentially steered wheels, the robot direction may be changed by varying the relative rate of rotation of the two separately driven wheels. If both the wheelsare driven in the same direction and speed, the robot will go straight. Otherwise,
depending on the speed of rotation and its direction, the center of rotation may fall anywhere in the line joining the two wheels.

Figure 1-4 Three Wheeled Robot.

1.1.3. Four Wheeled Robots

Four wheeled robots may be of two types: 2 powered, 2 free rotating wheels, Same as the Differentiallysteered ones above but with 2 free rotating wheels for extra balance. More stable than the three-wheel version since the center of gravity has to remain inside the rectangleformed by the four wheels instead of a triangle. This leaves a larger useful space. Still, it's advisable to keep the center of gravity to the middle of the rectangle as this is the most stable configuration, especially when taking sharp turns or moving over a non-level surface.

The other type is 2-by-2 powered wheels for tank-like movement This kind of robot
uses 2 pairs of powered wheels. Each pair (connected by a line) turn in the same

direction. The tricky part of this kind of propulsion is getting all the wheels to turn with the same speed. If the wheels in a pair aren't running with the same speed, the slower one will slip (inefficient). If the pairs don't run at the same speed the robot won't be able to drive straight. A good design will have to incorporate some form of car- like steering.

Figure 1-5 Four Wheeled Robot

1.1.4. Omni Wheeled Robots

Another option for wheeled robots that makes it easier for robots with wheels not all mounted on the same axis to have Omni Wheels. An omni wheel is like many smaller wheels making up a large one, thesmaller ones have axis perpendicular to the axis of the core wheel. This allows the wheels to move in two directions, and the ability to move holonomicaly, which means it can instantaneously move in any direction. Unlike a car, which moves non-holnomicaly and has to be in motion to change heading. Omni-wheeled robots can

move in at any angle in any direction, without rotating beforehand. There are three types of omni wheeled robots:

1- Three Omni Wheeled Robots: Omni wheeled robots use a triangular platform, with the three wheelsspaced at 60-degree angles. With 90-degree rollers slope.

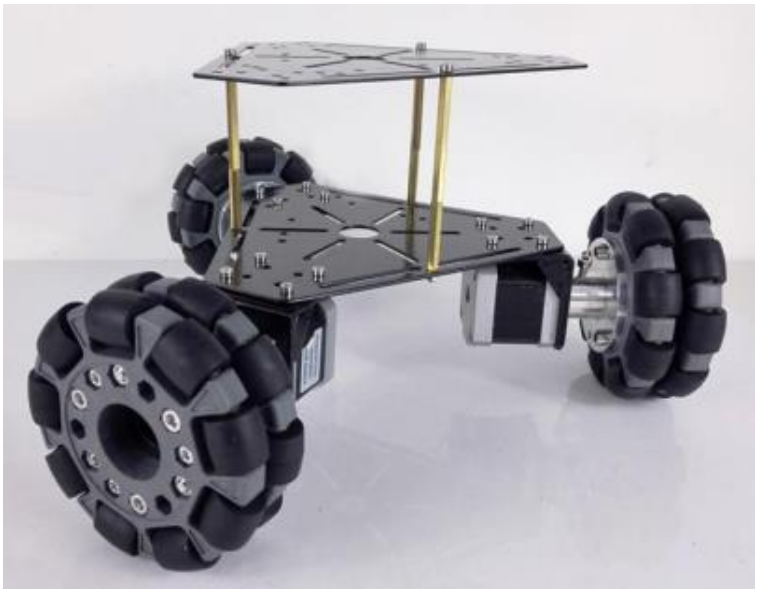

Figure 1-6 Three Omni Wheeled Robot

2- Four Omni Wheeled Robots: Omni wheeled robots use a normal 4 wheeled platform. With 90-degree rollers slope.

Figure 1-7 Four Omni Wheeled Robot

3- Mecanum Wheeled Robots: Mecanum wheeled robots use a normal 4 wheeled platform, with 45- degree rollers slope.

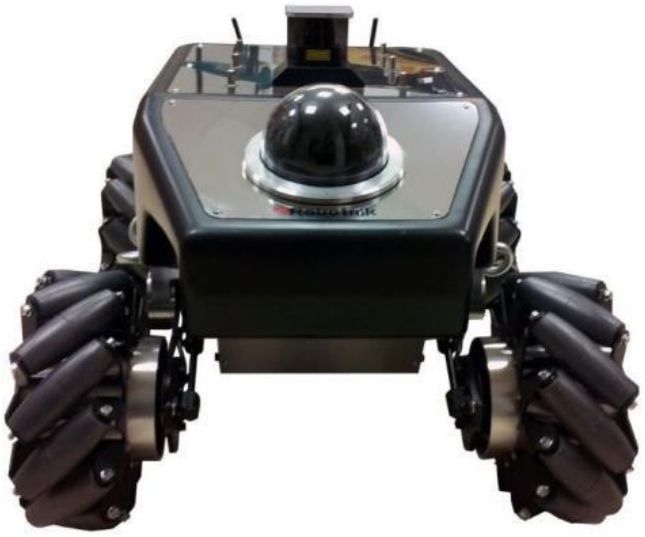

Figure 1-8 Mecanum Wheeled Robot.

1.2. Autonomous system overview

As human intercepts the navigation process through different stages as sensing the road then identifying objects and surrounding, based on these data the human mind starts to plan his path and next move. Later these plans are executed in terms of brake, steering, and acceleration

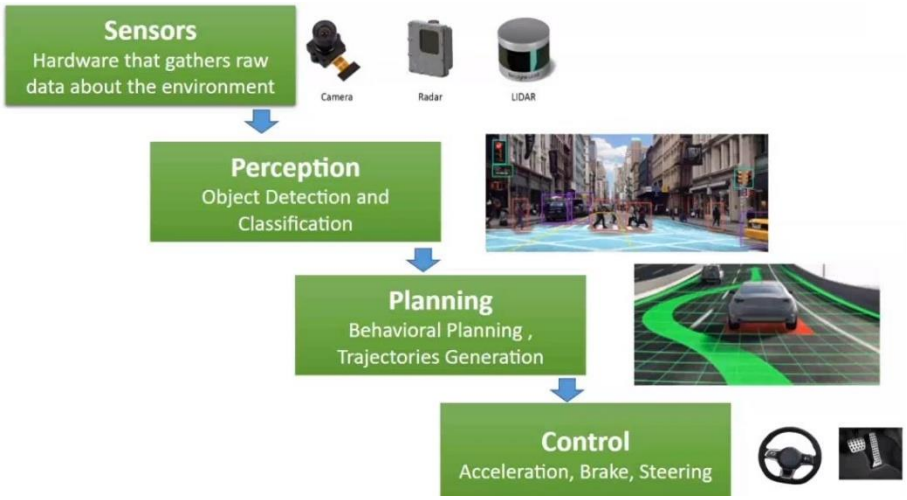

Figure 1-9 Autonomous System Overview.

1.2.1. Sensing

Like people, Autonomous robots must sense their surroundings to safely navigate. People use senses like hearing, sight, taste, smell, and touch to interact with their environments. Autonomous car technology developers' provision self-driving cars with high-tech sensor systems to sense analogously.

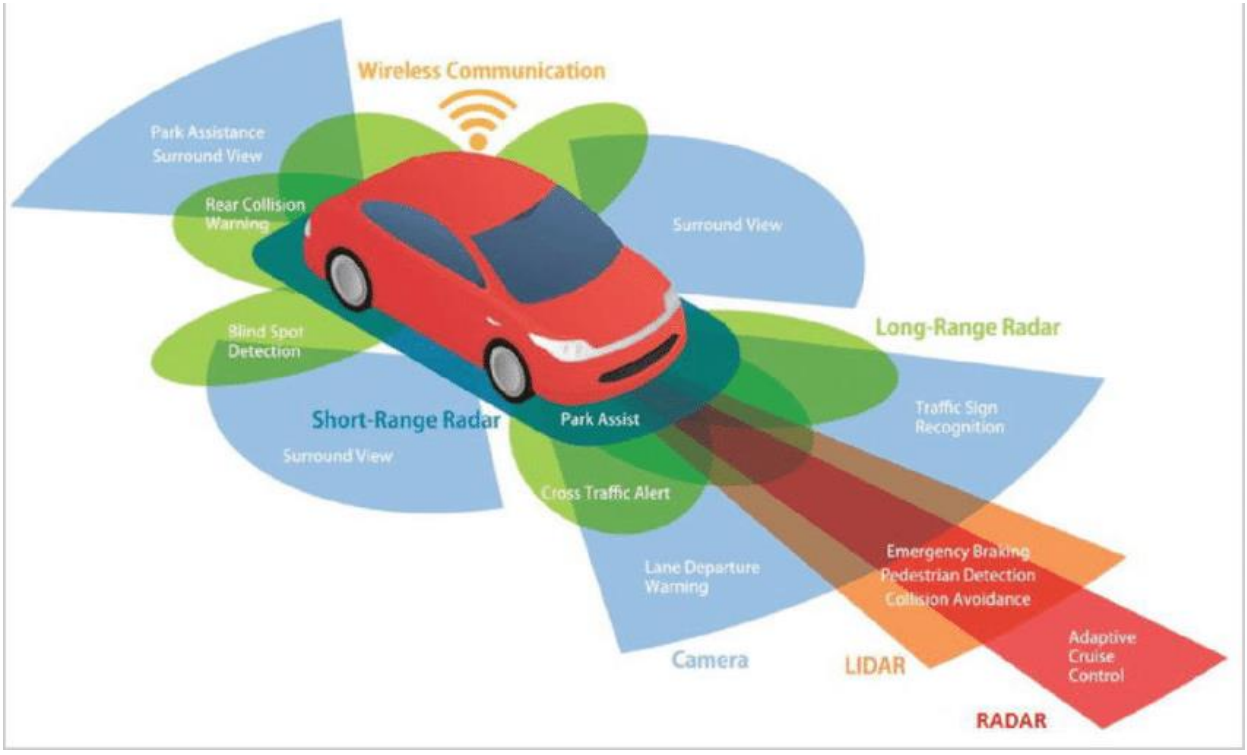

Figure 1-10 Wireless Communication and Sensors in Self Driving Cars

1.2.1.1. Lidar

Lidar (light detection and ranging), also known as 3D laser scanning, is a tool that Autonomous robots use to scan their environments with lasers. A typical lidar sensor pulses thousands of beams of infrared laser light into its surroundings and waits for the beams to reflect off environmental features. Many pulses create point clouds (sets of points representing 3D forms in space) of light.

Lidar systems measure the amount of time it takes to emit a laser signal and sense the same light beam reflected from a physical surface onto its photodetectors. Lidar uses the speed of light to calculate distances to objects. The longer it takes

for a lidar photodetector to receive a return light signal, the farther away an object is.

Lidar systems enable Autonomous robots to detect small objects with high precision. However, lidar is often unreliable at nighttime or in inclement weather.

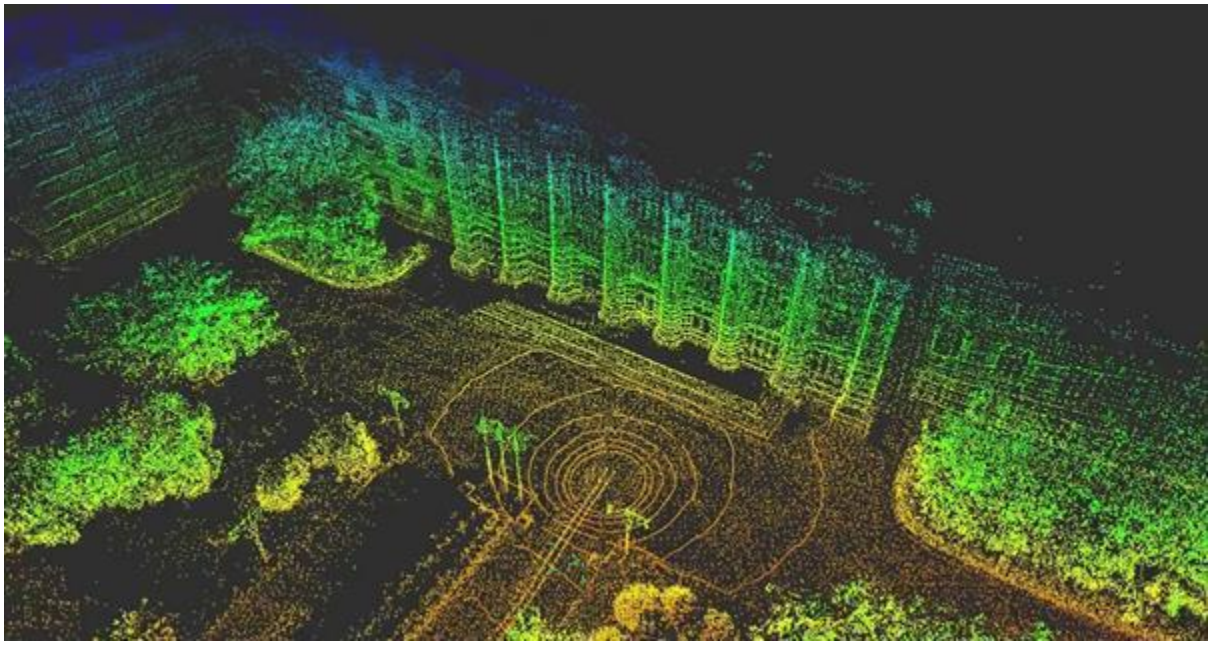

Figure 1-11 Lidar View as a 3d point

1.2.1.2. Camera

Autonomous vehicles can visualize their environments with high-resolution digital camera images. Autonomous robots can use camera images to "see" and interpret environmental details (e.g., signs, traffic lights, animals) in ways that approximate human vision (aka computer vision).

Figure 1-12 Depth camera on autonomous robot.

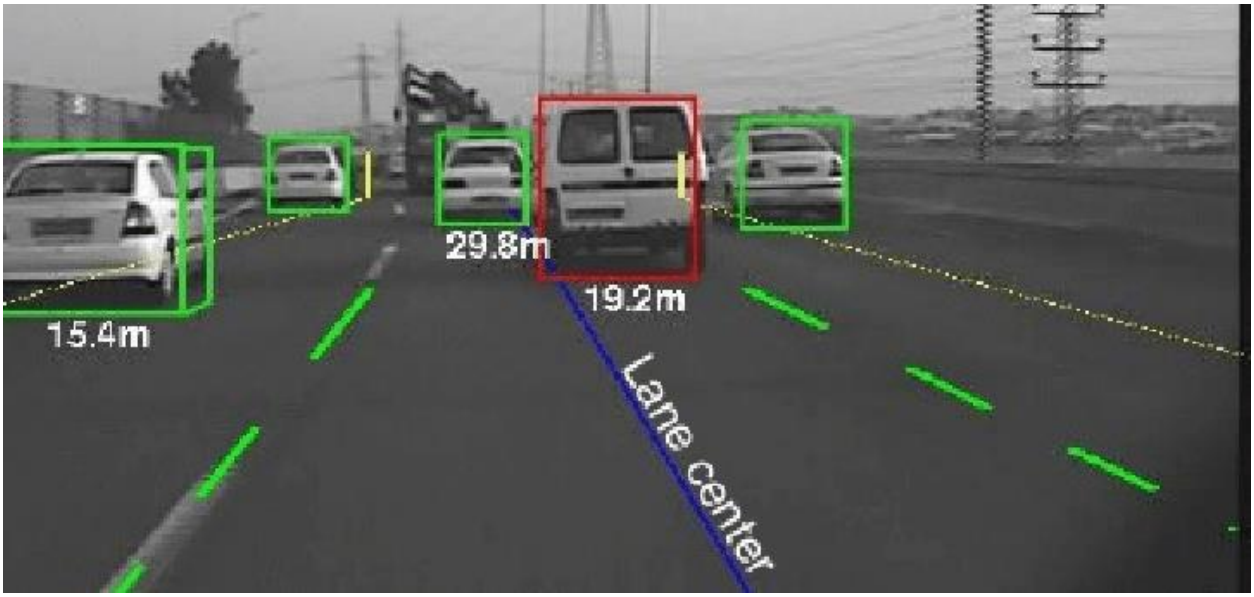

Figure 1-13 A sample of the real time artificial vision camera with detected objects for autonomous vehicles

1.2.1.3. Radar and Ultrasonic

Radar, which stands for (Radio Detection and Ranging), is a device that is comprised of an antenna that emits a radio signal in a specific direction, and a radio receiver, that detects the radio signal once it has bounced off of objects in the environment. The idea is to calculate the time it takes for a specific emitted radio signal to return, and with a constant (approximate) speed, the distance between the antenna and the object the signal bounced off of, can be calculated.

The Autonomous Robot passes through these stages too but with difference in execution of course. The sensing stage was covered in the last section. So, we will take next about the Perception stage and the problems the car has to overcome.

Figure 1-14 numerous targeted, finite directions for radar sensors put

1.2.2. Perception

The car has to make sense of all inputs that came from the sensors.in other words Perception is Autonomous Robots ‘See’ the World. like identifying objects like traffics, signs, lanes, and pedestrians using some Object detection algorithms. And also estimating the depth using Lidar and the speed of the surroundings using Radar. And our main goal in the project which is mapping the surrounded scene is also form the perception process.

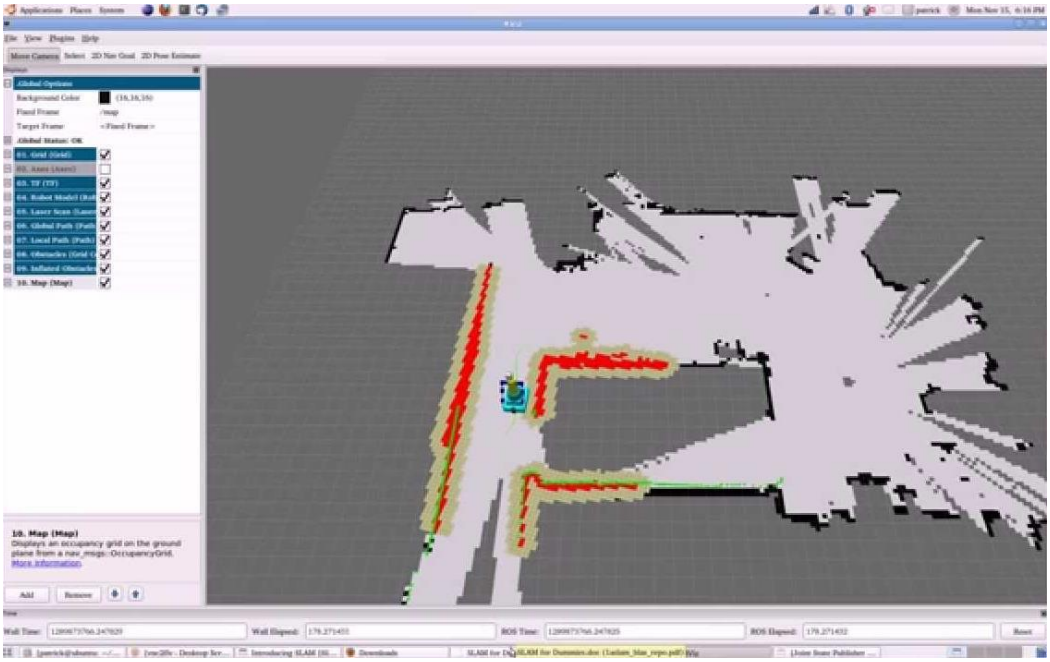

Figure 1-15 Map Generated Using SLAM.

1.2.3. Planning

The path planning problem is one of the most interesting and researched topics. The aim of the robot path planning is to search a safe path for the mobile robot. Also, the path is required to be optimal. several research works tackling the

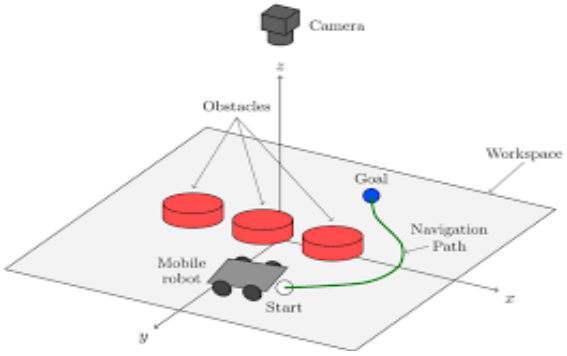

Figure 1-16) Planning a Trajectory Path.

path planning problem have been proposed in the literature. Until now, many methods have been used for path planning of mobile robots.

1.2.4. Control

The last step in the Autonomous driving system is logically the controlling of the vehicle. Based on the planed trajectory path and the detected object. Like maintaining speed and making sure we are on the right track of the planned trajectory.

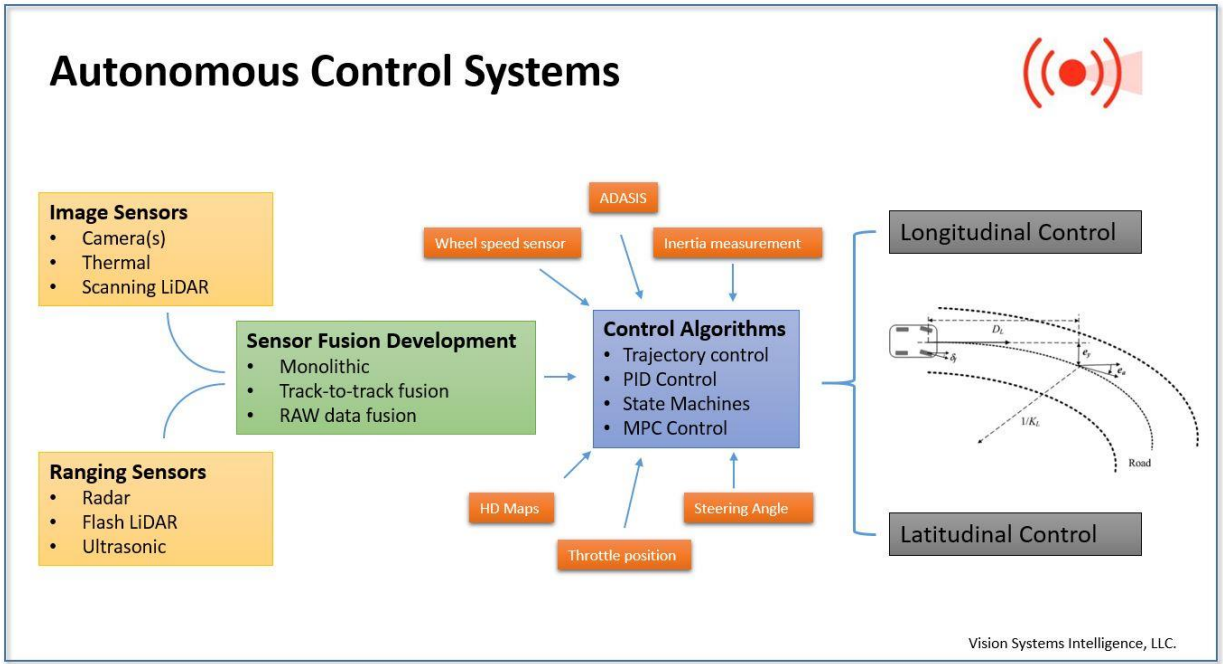

Figure 1-17 Autonomous control system.

1.3. Our Implementation for an Autonomous Robot

Based on our small resources. We chose some of these problems facing Autonomous Robots and tried to tackle it on a small scale. The problems we chose to try implementing were the problem of mapping of the environment around the robot.

The second problem we chose to face object avoidance to make the robot alter its position to avoid different objects.

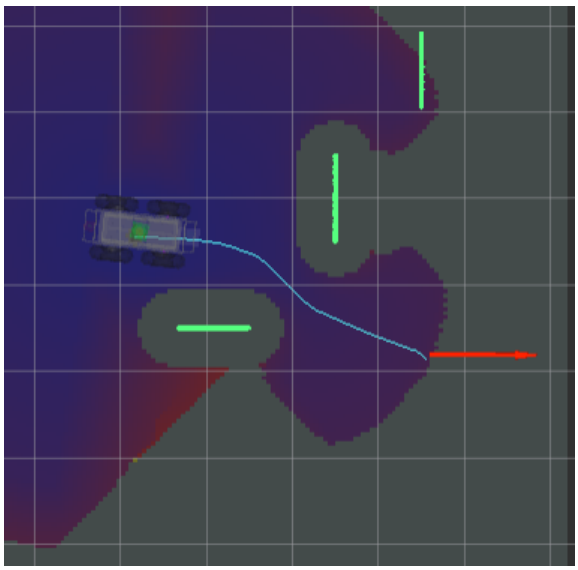

Figure 1-18 Global Path Planner

1.4. Overview

In the next Chapters we will go into details about how we implemented our project, and we will talk about the methodology, techniques, and ideas we used. In Chapter 2 we will talk about the hardware implementation of the project and the Mechanical parts for it. Later in Chapter 3 we will mention overview about Robot Operating System (ROS) which is a robotics framework that uses multithreading concept, and we will talk about the different stages of autonomous motion and how we implemented our work.

2. Hardware Implementation

2.1 Electronics

2.1.1. LM2596HV DC-DC 3-40V to 1.23-37V 3A

The LM2596-XXE5/F5 series of regulators are monolithic ICs that provide all active functions for a step-down (buck) switching regulator, capable of driving 3A load with excellent line and load regulation. These devices are available in fixed output voltage of 3.3V, 5V, 12V and an adjustable output version. Requiring a minimum number of external components, these regulators are simple to use and include internal frequency compensation and a fixed- frequency oscillator.

The LM2596-XXE5/F5 series operates at a switching frequency of 150 kHz, thus allowing smaller sized filter components than what would be needed with lower frequency switching regulators. Available in standard 5-lead TO-220 and TO-263 packages with several different lead bend options. A standard series of inductors are available from several different manufacturers optimized for use with the LM2596-XXE5/F5 series. This feature greatly simplifies the design of switch-mode power supplies. Other features include a guaranteed ±4% tolerance on output voltage under specified input voltage and output load conditions, and ±15% on the oscillator frequency. External shutdown is included, featuring 80µA standby current. Self-protection features include a two stage frequency reducing current limit for the output switch and an over temperature shutdown for complete protection under fault condition

Features:

- 3.3V, 5V, 12V, and adjustable output versions
- Adjustable version output voltage range, 1.3V to 37V±4% max over line and load conditions
- 150kHz±15% fixed switching frequency
- TTL shutdown capability
- Operating voltage can be up to 40V
- Output load current:3A
- TO220-5L and TO263-5L packages
- Low power standby mode.
- Thermal shutdown and current limit protection.
- High efficiency

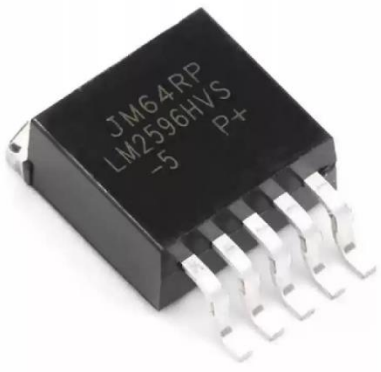

Figure 2-1

- Built-in switching transistor on chip
- Requires only 4 external components
- Use readily available standard inductors

2.1.2. Step-Down Buck Converter

A buck converter (step-down converter) is a DC-to-DC power converter which steps down voltage (while drawing less average current) from its input (supply) to its output (load). It is a class of switched-mode power supply (SMPS) typically containing at least two semiconductors (a diode and a transistor, although modern buck converters frequently replace the diode with a second transistor used for synchronous rectification) and at least one energy storage element, a capacitor, inductor, or the two in combination. To reduce voltage ripple, filters made of capacitors (sometimes in combination with inductors) are normally added to such a converter's output (load-side filter) and input (supply-side filter).

Switching converters (such as buck converters) provide much greater power efficiency as DC-to-DC converters than linear regulators, which are simpler circuits that lower voltages by dissipating power as heat, but do not step up output current. The efficiency of buck converters can be very high, often over 90%, making them useful for tasks such as converting a computer's main supply voltage, which is usually 24 V to 12 V, down to lower voltages needed by USB, DRAM and the CPU, which are usually 5, 3.3 or 1.8 V.

Figure 2-2

DC-DC converters are also known as Choppers. Here we will have a look at the Step-Down Chopper or Buck converter which reduces the input DC voltage to a specified DC output voltage. A typical Buck converter is shown in the following figure:

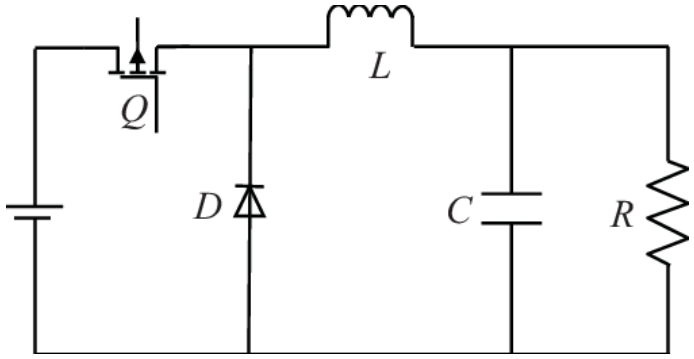

Figure 2-3

The input voltage source is connected to a controllable solid-state device which operates as a switch. The solid-state device can be a Power MOSFET or IGBT. Thyristors are not used generally for DC-DC converters because to turn off a Thyristor in a DC-DC circuit requires another commutation which involves using another Thyristor, whereas Power MOSFET and IGBT can be turned off by simply having the voltage between the GATE and SOURCE terminals of a Power MOSFET, or, the GATE and COLLECTOR terminals of the IGBT go to zero. The second switch used is a diode. The switch and the diode are connected to a low-pass LC filter which is appropriately designed to reduce the current and voltage ripples. The load is a purely resistive load. The input voltage is constant and the current through load is also constant. The load can be seen as current source.

The controlled switch is turned on and off by using Pulse Width Modulation (PWM). PWM can be time based or frequency based. Frequency based modulation has disadvantages like a wide range of frequencies to achieve the desired control of the switch which in turn will give the desired output voltage. This leads to a complicated design for the low-pass LC filter which would be required to handle a large range of frequencies. Time based Modulation is mostly used for DC-DC converters. It is simple to construct and use. The frequency remains constant in this type of PWM modulation. The Buck converter has two modes of operation.

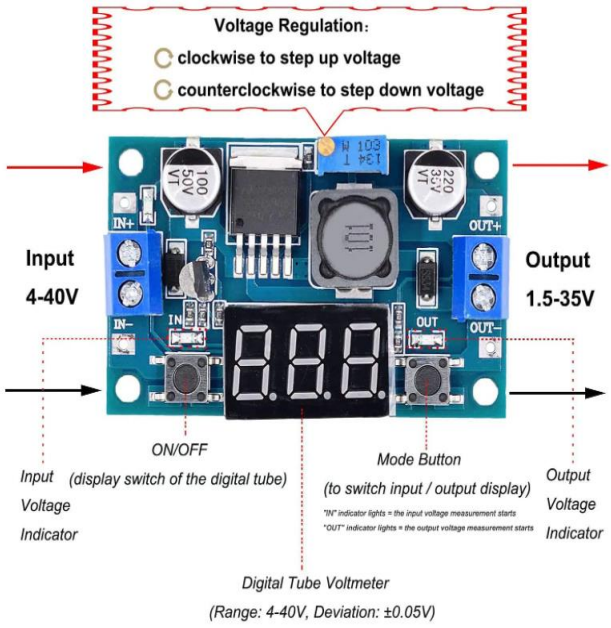

Figure 2-4

2.1.2.1 DC-DC Step Down Converter 8A (4V~40Vdc to 1.25V~36Vdc)

Specification:

- Product name: DC step-down module Converter
- Voltage regulating mode: PWM modulation
- Input voltage: DC4-40V
- Output voltage: DC1.25-36V
- Max output current: 8A (more than 5A for long time use, need Fan if over 5A
- Max power: 200W
- Conversion efficiency: 94%
- Switching frequency: 180KHZ
- Dimensions: 61*41*27mm

Figure 2-5

Chip: switching regulator Regulating chip XL4016 with
over current protection, over temperature protection, Short circuit protection.
When reaching maximum output current 8A, need to add fan. Long term work recommended about 5A

2.1.3. Battery System

Figure 2-6 battery cells

Figure 2-7 Battery system

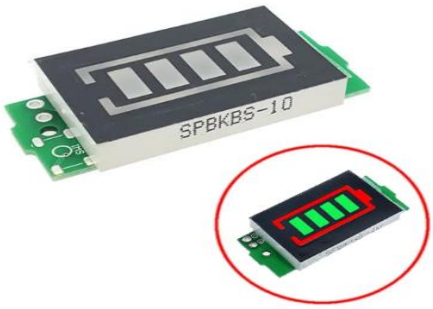

Figure 2-9 battery level indicator

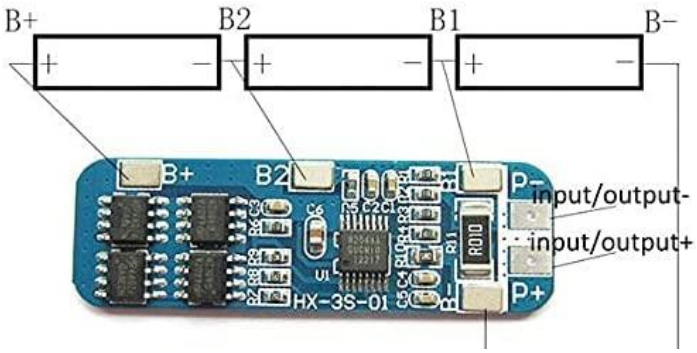

Figure 2-8 Battery Management system

2.1.3.1. Lithium-Ion Batteries

Lithium-ion is the most popular rechargeable battery chemistry used today. Lithium-ion batteries power the devices we use every day, like our mobile phones and electric vehicles.
Lithium-ion batteries consist of single or multiple lithium-ion cells, along with a protective circuit board. They are referred to as batteries once the cell, or cells, are installed inside a device with the protective circuit board.

What are the components of a lithium-ion cell?

- **Electrodes:** The positively and negatively charged ends of a cell. Attached to the current collectors
- **Anode:** The negative electrode
- **Cathode:** The positive electrode
- **Electrolyte:** A liquid or gel that conducts electricity
- **Current collectors:** Conductive foils at each electrode of the battery that are connected to the terminals of the cell. The cell terminals transmit the electric current between the battery, the device and the energy source that powers the battery
- **Separator:** A porous polymeric film that separates the electrodes while enabling the exchange of lithium ions from one side to the other

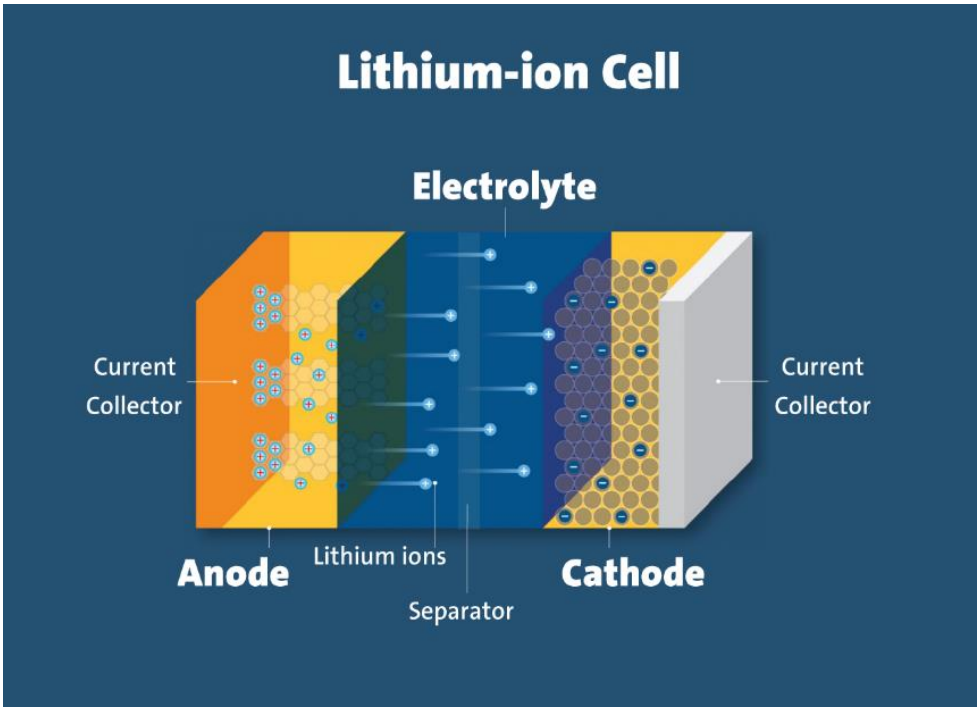

Figure 2-10 Lithium Ion cell

How does a lithium-ion cell work?

In a lithium-ion battery, lithium ions (Li+) move between the cathode and anode internally. Electrons move in the opposite direction in the external circuit. This migration is the reason the battery powers the device—because it creates the electrical current.

While the battery is discharging, the anode releases lithium ions to the cathode, generating a flow of electrons that helps to power the relevant device.

When the battery is charging, the opposite occurs: lithium ions are released by the cathode and received by the anode.

2.1.3.2. Battery Level Indicator

- Working voltage: 3-34V, working current 5mA.
- Working temperature: -20~50 °C
- The voltage error is 2% (requires higher accuracy, do not shoot)
- Size: 43.5 * 20cm

Warm sound tips: This model is not waterproof. If it is used outdoors, please waterproof it, because the electronic components should be used in a dry environment. Lithium-ion battery the battery indicator board that can be used in Ni-MH batteries, as long as the required voltage is within the range of the parameter list. How to use: Connect the positive and negative terminals of the display board to the positive and negative terminals of the battery under test. The digital tube displays the real-time battery power.

Select the corresponding pad on the tin, you can detect the battery voltage corresponding to S1-S8.

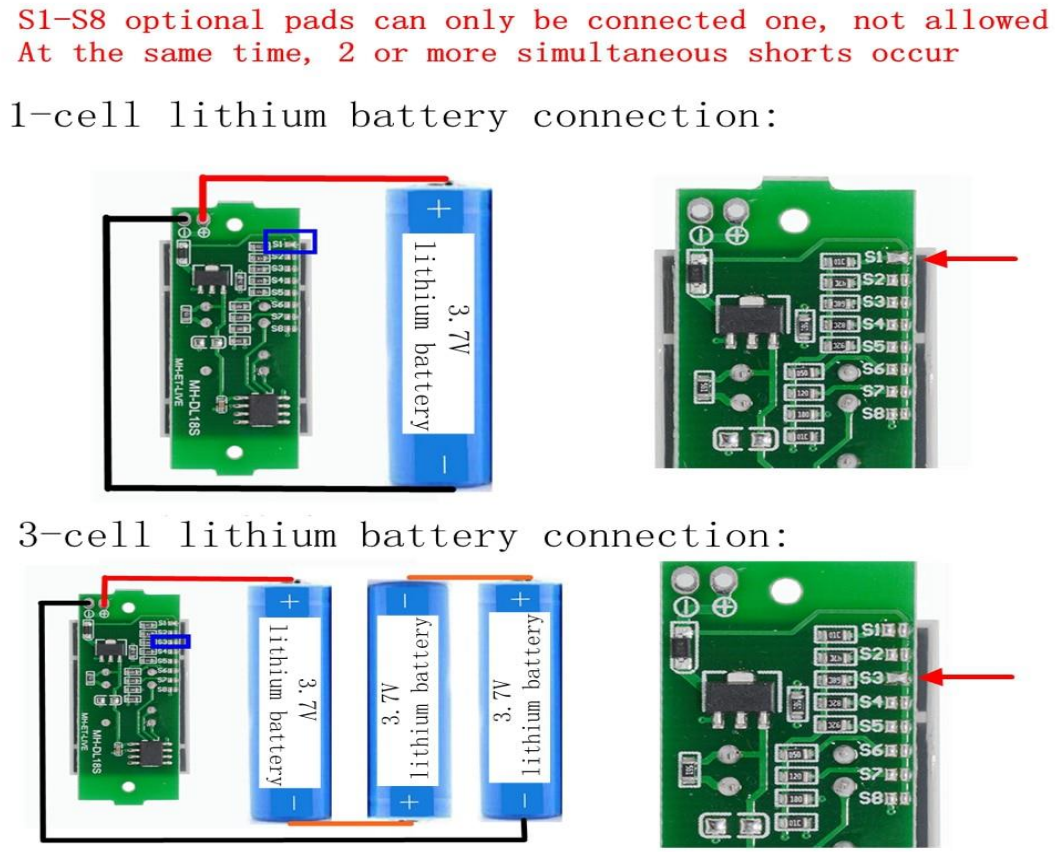

Figure 2-11

2.1.3.3. Battery Management System

Why are battery management systems (BMS) needed and how do they work?

Battery management systems (BMS) are electronic control circuits that monitor and regulate the charging and discharge of batteries. The battery characteristics to be monitored include the detection of battery type, voltages, temperature, capacity, state of charge, power consumption, remaining operating time, charging cycles, and some more characteristics.

Tasks of smart battery management systems (BMS)

The task of battery management systems is to ensure the optimal use of the residual energy present in a battery. In order to avoid loading the batteries, BMS systems protect the batteries from deep discharge and over-voltage, which are results of extreme fast charge and extreme high discharge current. In the case of multi-cell batteries, the battery management system also provides a cell balancing function, to manage that different battery cells have the same charging and discharging requirements.

2.1.3.3.1. DC Power Adapter (24VDC – 1.5A)

Description :

This is industry quality DC power adapter with 24V, 1.5A output. It has barrel jack connector. This adapter works well with many 24V applications.

Specifications

- Input: 100-240V AC 50/60Hz
- Output : 24V, 1.5A
- Cable length: about 110CM
- Output adaptor : Barrel Jack

Figure 2-12 Power adapter

2.1.4. DC Motors

Figure 2-13 DC Motor

The DC motor was chosen for its controllability, and to supply DC power from a portable source. DC 24V - 111RPM - 16Kg.cm Motor with Encoder, High Torque Speed Reduction Motor was suitable for our project specifications. Was chosen based on motor calculation.

Specifications:

Rated Voltage: 24V
No Load Current: 140mA
Load Torque Current (With Load): 800mA
Reduction Ratio: 1:72
Error: ±10%
Torque: 16 Kgf.cm (0.62 N.m)
Output Power: 2.3 W
Stall Current: 3A Motor
Weight: 272 Grams

Figure 2-14

Motor Calculations

To calculate the motor power:
We should first calculate the total forces on the robot
F (total) =F (Rolling) +F (Gradient) +F (Drag)
F (Rolling) =C*m*a
Suppose the robot's mass is 45kg and its max speed is 25km/h and the coefficient of resistance to the robot wheels is 0.02
F for each wheel=45/4=11.25kg
So,
F (Rolling) =0.02*11.25*9.81=2.21N
F (Gradient) =m*a*sinβ
Suppose that the robot will run on a flat surface so β=0 and
F (Gradient) =0
F (Drag) =0.5*(ρ* V2 *c*A)
C=0.82, v=6.94m/s, ρ for air=1.23 and A=0.5*0.6 m2
F (Drag) =0.5*(1.23* 6.942*0.82*0.5*0.6) =7.29N
F (Total) =7.29+2.21=9.5N
Power=F (Total)*Velocity
Power=9.5*6.94=65.93 Say 70 Watt

To Calculate the Torque
Suppose the diameter of the wheel is 13cm so,
D=.13m r=0.065m
The circumstance of the wheel =2π*r
=2* π* 0.065=0.408m
So, it will go 0.408 in each revolution
Suppose the speed is 25km/h which equals 416.67m/min
Max wheel speed =distance per minute at top speed/ distance covered by the wheel in each rev
=416.67/0.408=1022 RPM
Power= (2* π*N*T)/60
70= (2* π*1022*T)/60
So, the torque should be 0.66 N.m
Let's Say we need 4 motors each motor with torque about 1 N.m
Based on calculation our decision was made.

2.1.5. BTS7960 High Current 43A H-Bridge Motor Driver

The BTS7960 is a fully integrated high current H bridge module for motor drive applications. Interfacing to a microcontroller is made easy by the integrated driver IC which features logic level inputs, diagnosis with current sense, slew rate adjustment, dead time generation and protection against over temperature, overvoltage, under voltage, overcurrent and short circuit. The BTS7960 provides a cost optimized solution for protected high current PWM motor drives with very low board space consumption.

Brief Data:

- Input Voltage: 6 ~ 27Vdc.
- Driver: Dual BTS7960 H Bridge Configuration.
- Peak current: 43-Amp.
- PWM capability of up to 25 kHz.
- Control Input Level: 3.3~5V.
- Control Mode: PWM or level
- Working Duty Cycle: 0 ~100%.

Figure 2-15 Motor Driver

- Over-voltage Lock Out.
- Under-voltage Shut Down.
- Board Size (LxWxH): 50mm x 50mm x 43mm.
- Weight: ~66g.

Schematic Diagram:

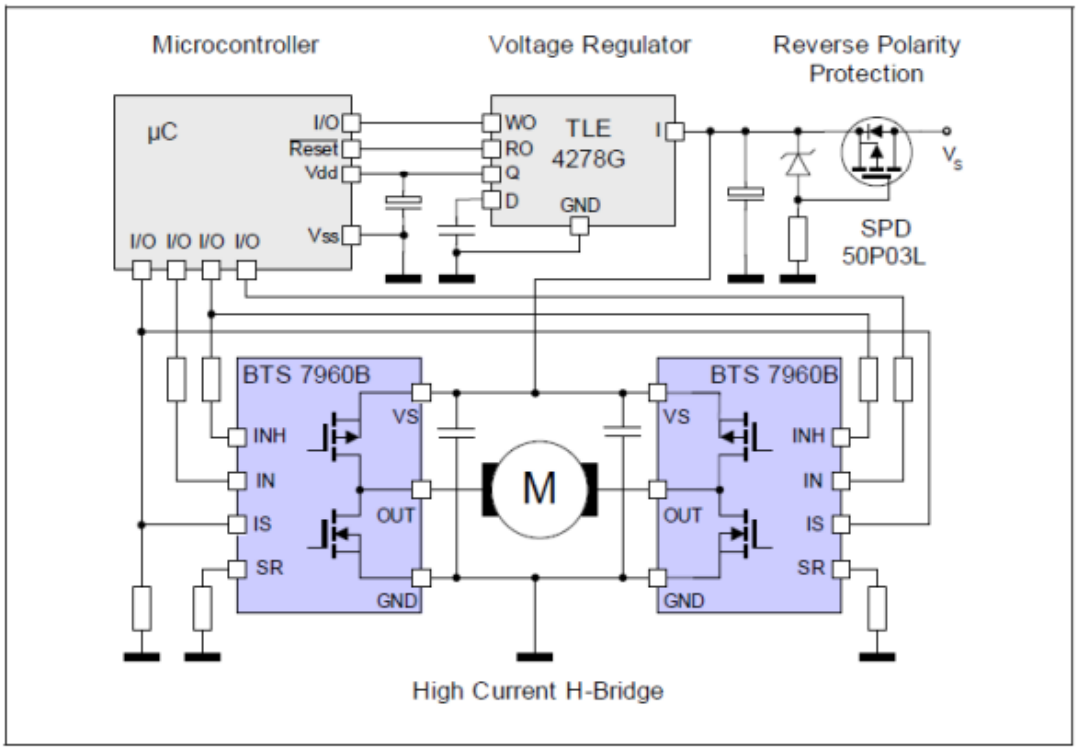

Figure 2-16 schematic Diagram

H-Bridge – For controlling rotation direction

The DC motor's spinning direction can be controlled by changing the polarity of its input voltage. A common technique for doing this is to use an H-Bridge. An H-Bridge circuit contains four switches with the motor at the center forming an H-like arrangement. Closing two particular switches at the same time reverses the polarity of the voltage applied to the motor. This causes a change in spinning direction of the motor.

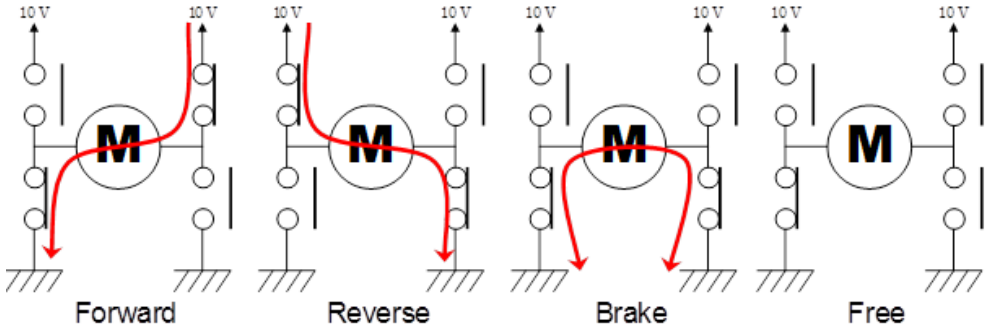

Figure 2-17 Rotation Direction control

PWM – For controlling speed

PWM is a technique where the average value of the input voltage is adjusted by sending a series of ON-OFF pulses. The average voltage is proportional to the width of the pulses known as Duty Cycle. The higher the duty cycle, the greater the average voltage being applied to the dc motor (High Speed) and the lower the duty cycle, the less the average voltage being applied to the dc motor (Low Speed). The speed of a DC motor can be controlled by varying its input voltage. A common technique for doing this is to use PWM (Pulse Width Modulation).

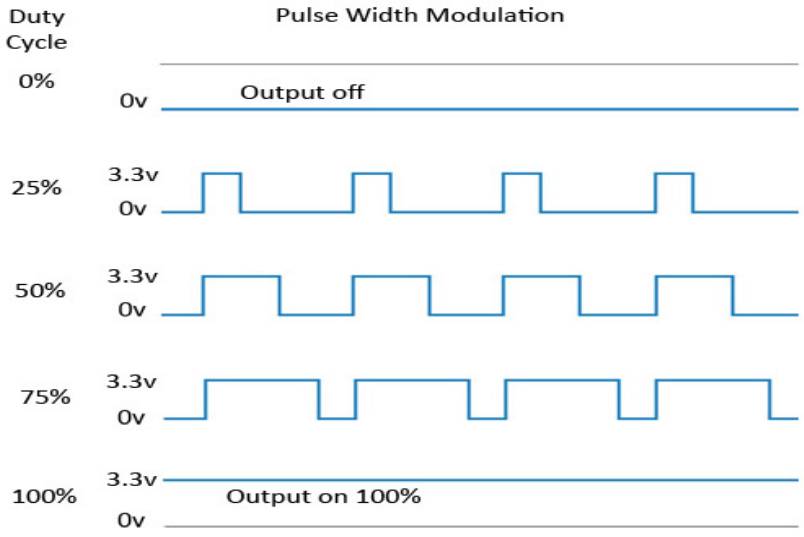

Figure 2-18 PWM

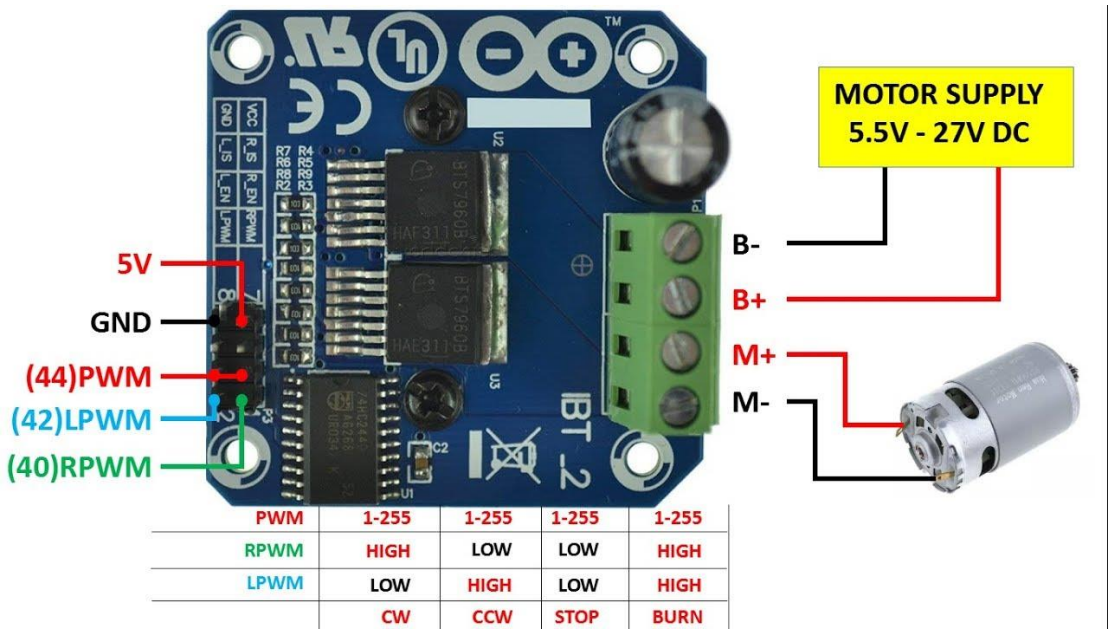

Figure 2-19

2.1.6. Teensy 3.2

Description

The Teensy is a breadboard-friendly development board with loads of features in a, well, teensy package. Each Teensy 3.2 comes pre-flashed with a bootloader so you can program it using the on-board USB connection: No external programmer needed! You can program for the Teensy in your favorite program editor using C *or* you can install the Teensyduino add-on for the Arduino IDE and write Arduino sketches for Teensy!

Figure 2-20 Teensy

The processor on the Teensy also has access to the USB and can emulate any kind of USB device you need it to be, making it great for USB-MIDI and other HID projects. The 32 bit processor brings a few other features to the table as well, such

as multiple channels of Direct Memory Access, several high-resolution ADCs and even an I2S digital audio interface! There are also 4 separate interval timers plus a delay timer! Oh yeah, and all pins have interrupt capability. Also, it can provide system voltage of 3.3V to other devices at up to 250mA.

All of this functionality is jammed into a 1.4 x 0.7 inch board with all headers on a 0.1" grid so you can slap in on a breadboard and get to work! The Teensy 3.2 adds a more powerful 3.3 volt regulator, with the ability to directly power an ESP8266 Wi-Fi, WIZ820io Ethernet, and other 3.3V add-on boards that require a little more power. Additionally, if it is used within the Teensy 3.1 limits of operation, the Teensy 3.2 and 3.1 are interchangeable!

Teensy 3.2 pinout

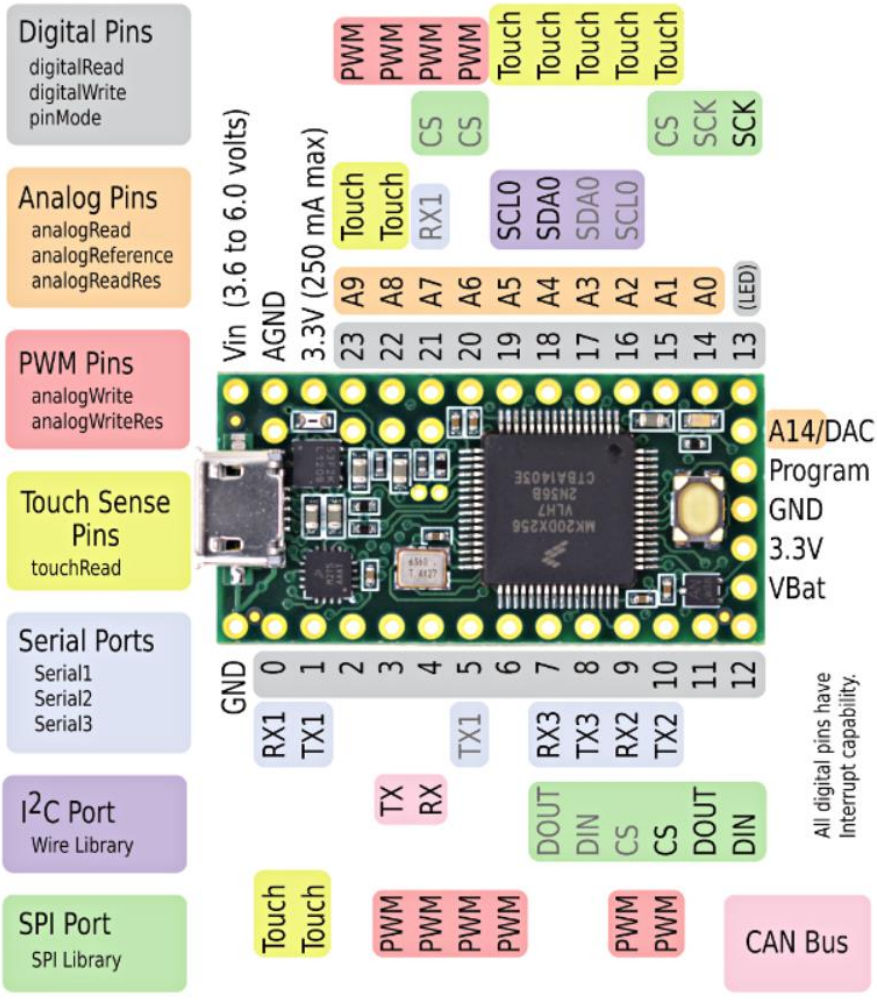

Figure 2-21 Teensy pinout

Features

- 32 bit ARM Cortex-M4 72 MHz CPU (M4 = DSP extensions)
- 256K Flash Memory, 64K RAM, 2K EEPROM
- 21 High Resolution Analog Inputs (13 bits usable, 16 bit hardware)
- 34 Digital I/O Pins (5V tolerance on Digital Inputs)
- 12 PWM outputs
- 7 Timers for intervals/delays, separate from PWM
- USB with dedicated DMA memory transfers
- 3 UARTs (serial ports)
- SPI, I^2C, I2S,CAN Bus, IR modulator
- I2S (for high quality audio interface)
- Real Time Clock (with user-added 32.768 crystal and battery)
- 16 DMA channels (separate from USB)
- Touch Sensor Inputs
- 1.4 x 0.7" (~35 x 18 mm)

2.1.7. RASPBERRY PI 4 Computer MODEL B – 4GB RAM – MADE IN UK

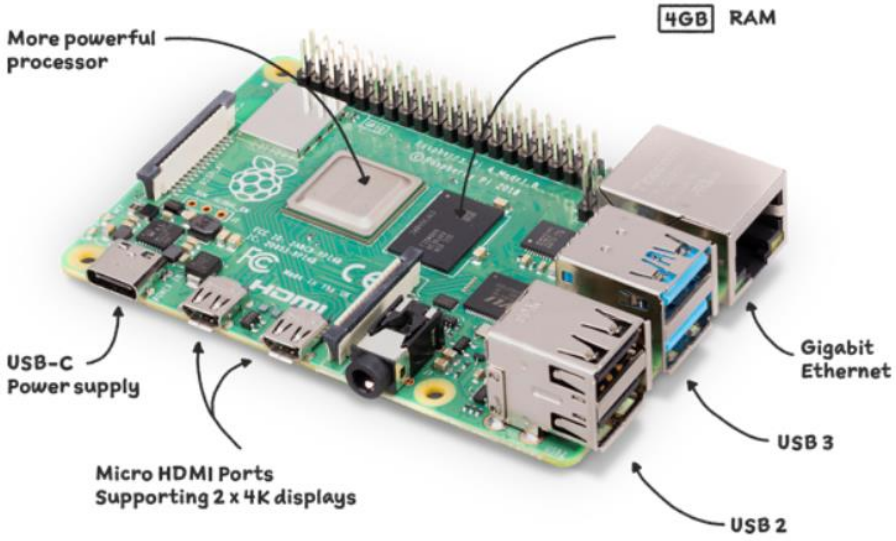

Figure 2-22 Raspberry 4B

Description

The 4GB version of Raspberry Pi 4 Model B is one of the newest and most powerful members of the fourth-generation Raspberry Pi family from the

Raspberry Pi Foundation that is the fastest board released to date. Whilst the Pi 4 looks similar in shape and size to previous boards, it has been re-engineered and enhanced to be more capable and efficient than ever before, allowing it to contend with a desktop PC experience.

Speedier Processor

The new model is a high-performance general-purpose computer, incorporating an improved quad-core 64-bit ARM Cortex-A72 CPU to rival desktop computers. For the first time, the Pi 4 is available in a choice of memory sizes, so with up to 4GB of RAM (LPDDR4), you will see your Pi being able to carry out much more complicated tasks easily.

Improved Networking

With true Gigabit Ethernet and PoE capability (requires additional PoE HAT); 2.4GHz and 5GHz 802.11b/g/n/ac Wi-Fi and Bluetooth 5.0 with up to four times the range of the old specification. All together these make the Raspberry Pi 4 Model B board an ideal fit for your next SBC-based IoT project.

Upgraded USB Capacity and Power

Pi 4 has four USB ports, two USB 2.0 ports and two of which have been upgraded to USB 3.0. The board has a USB Type-C connector for power input, instead of the Micro USB connector the Raspberry Pi 3 had, so you can consolidate your cable collection. The headphone jack has made a return too!

Dual Video Output

Two micro-HDMI ports, replace the single full-size HDMI connector on previous versions, allowing the Pi 4 to support 2 4k monitors. The Pi's hardware decoding has also been improved to allow it to decode 4K video at 60 frames per second using HEVC encoding.

Backwards Compatible

You don't have to worry about the Pi 4 Model B not working with any of your existing pi projects, as the 40-Pin GPIO header and software is fully backwards-compatible.

Specifications

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 1GB, 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
- 2 × micro-HDMI ports (up to 4kp60 supported)
- 2-lane MIPI DSI display port
- 2-lane MIPI CSI camera port
- 4-pole stereo audio and composite video port
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- OpenGL ES 3.1, Vulkan 1.0
- Micro-SD card slot for loading operating system and data storage
- 5V DC via USB-C connector (minimum 3A*)
- 5V DC via GPIO header (minimum 3A*)
- Power over Ethernet (PoE) enabled (requires separate PoE HAT)
- Operating temperature: 0 – 50 degrees C ambient

* A good quality 2.5A power supply can be used if downstream USB peripherals consume less than 500mA in total.

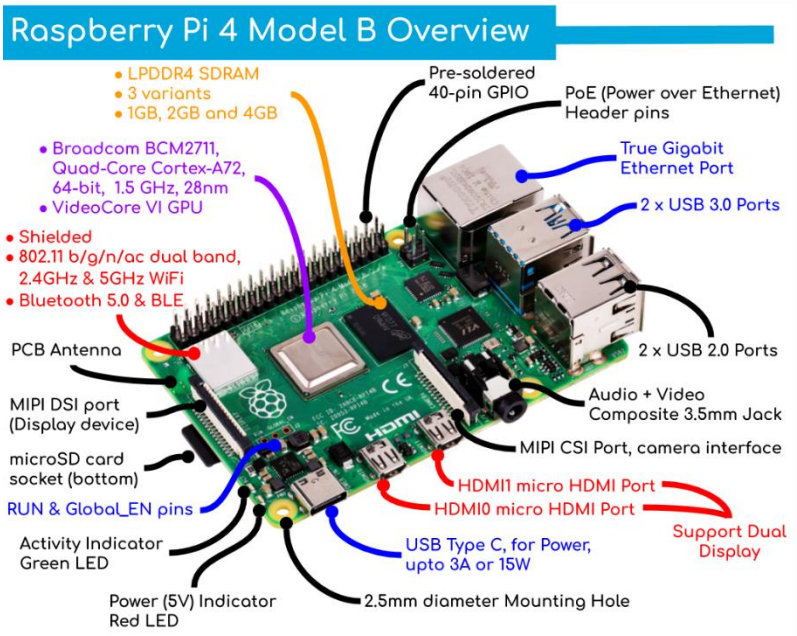

Figure 2-23 Raspberry pi overview

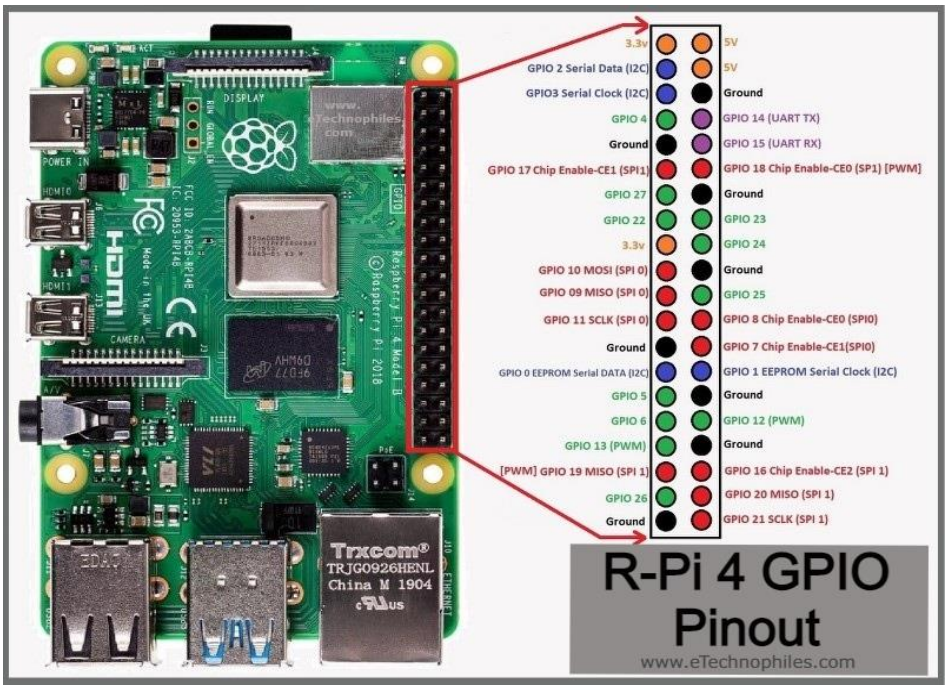

Figure 2-24 R-Pi GPIO pinout

2.1.8. GY-521 – MPU6050 IMU (3Axis Gyro+3Axis Accelerometer)

Description:

The MPU-6050 is the latest product of invensense company products. This sensor is used in Ardupilot autopilot. The sensor contains a 3 axis MEMS accelerometer and a 3 axis MEMS gyro in a single chip. It is very accurate, since it contains 16-bits analog to digital conversion hardware for each channel. Therefore it captures the x, y, and z channel at the same time.

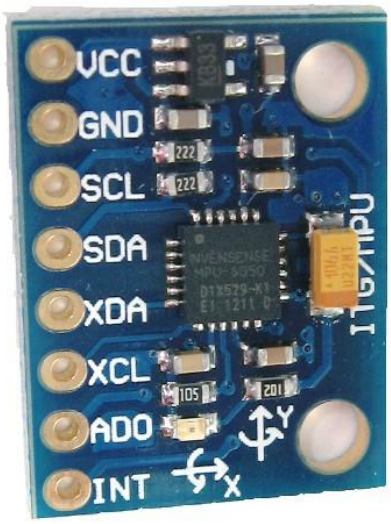

Figure 2-25 MPU6050

Also, it is accurate because you have the 3 axis gyro and 3 axis accelerometers on single chip so you do not need to align them.

The sensor has a “Digital Motion Processor” (DMP), also called a “Digital Motion Processing Unit”. This DMP can be programmed with firmware made by invensense and is able to do complex calculations with the sensor values.

Features:

- I2C Digital-output of 6 or 9-axis Motion Fusion data in rotation matrix, quaternion, Euler Angle, or raw data format
- Input Voltage: 2.3 – 3.4V.
- Tri-Axis angular rate sensor (gyro) with a sensitivity up to 131 LSBs/dps and a full-scale range of ±250, ±500, ±1000, and ±2000dps.
- Tri-Axis accelerometer with a programmable full scale range of ±2g, ±4g, ±8g and ±16g
- Digital Motion Processing™ (DMP™) engine offloads complex Motion Fusion, sensor timing synchronization and gesture detection.
- Digital-output temperature sensor.

GY-521 Pin Layout:

- VCC (The breakout board has a voltage regulator. Therefore, you can connect the board to 3.3V and 5V sources.)
- GND
- SCL (Serial Clock Line of the I2C protocol.)
- SDA (Serial Data Line of the I2C protocol.)
- XDA (Auxiliary data => I2C master serial data for connecting the module to external sensors.)
- XCL (Auxiliary clock => I2C master serial clock for connecting the module to external sensors.)
- AD0 (If this pin is LOW, the I2C address of the board will be 0x68. Otherwise, if the pin is HIGH, the address will be 0x69.)
- INT (Interrupt digital output)

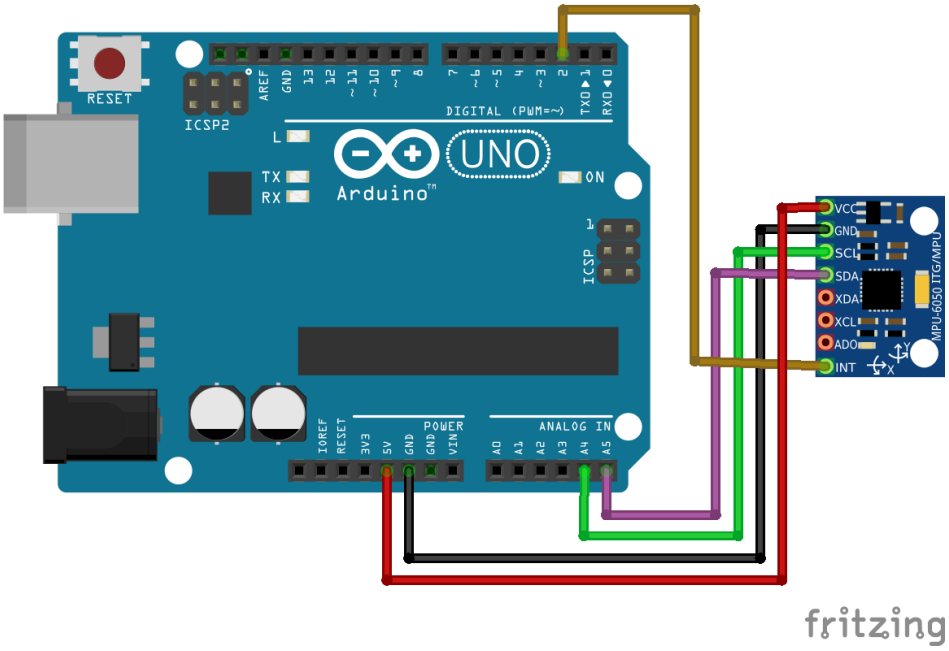

Figure 2-26 Wiring layout of GY-521

2.1.9. Solenoid (Linear Motion) 12V-5N Force

Solenoids are a great way to induce linear motion for pushing, pulling or controlling switches and levers. This high quality solenoid is designed to work directly with 12V which makes it a great match for embedded projects. It has a stroke of about 10mm and 5 N pulling force. Used widely in medical apparatus, home appliances, vending machines, game machine, auto door lock, cabinet and many other applications.

Figure 2-27 Solenoid

Specifications

- Rated Voltage: DC 12V
- Action Form: Pull
- Rated Stroke: 10mm
- Rated Force: 5N
- Current: 1A
- Body Size: 3 x 1.7 x 1.4cm
- Net Weight: 38g

2.1.10. RPLIDAR A1M8-R6 - 360 Degree Laser Scanner

Figure 2-28 Lazer Range Scanner

Description

RPLIDAR A1 is a low cost 360 degree 2D laser scanner (LIDAR) solution developed by SLAMTEC. The system can perform 360 degree scan within 12-meter range (6-meter range of A1M8-R4 and the belowing models). The produced 2D point cloud data can be used in mapping, localization and object/environment modeling.

RPLIDAR A1's scanning frequency reached 5.5 hz when sampling 1450 points each round. And it can be configured up to 10 hz maximum. RPLIDAR A1 is basically a laser triangulation measurement system. It can work excellent in all kinds of indoor environment and outdoor environment without direct sunlight exposure.

RPLIDAR A1 is based on laser triangulation ranging principle and uses high-speed vision acquisition and processing hardware developed by Slamtec. The system measures distance data in more than 8000 times per second.

The sample rate of LIDAR directly decides whether the robot can map quickly and accurately.

RPLIDAR improves the internal optical design and algorithm system to make the sample rate up to 8000 times,

Comparison under different conditions

360 Degree Omnidirectional Laser Range Scanning

The core of RPLIDAR A1 runs clockwise to perform a 360-degree omnidirectional laser range scanning for its surrounding environment and then generate an outline map for the environment.

Figure 2-29 360 Degree laser

OPTMAG Original Design

Most traditional non-solid LIDARs use slip ring to transfer power and data information, however, they only have thousands of hours of life due to mechanical wearing out. Slamtec has integrated the wireless power and optical communication technology to self-design the OPTMAG technology, which breakouts the life limitation of traditional LIDAR system. It fixes the electrical connection failure caused by the physical wearing out so as to prolong the life-span.

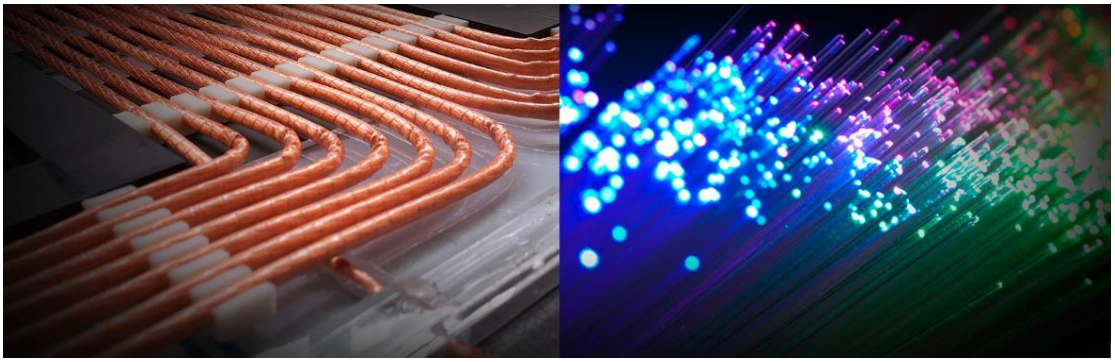

Figure 2-30

Configurable Scan Rate from 2-10Hz

Users can adjust the scan rate by alternating the motor PWM signal

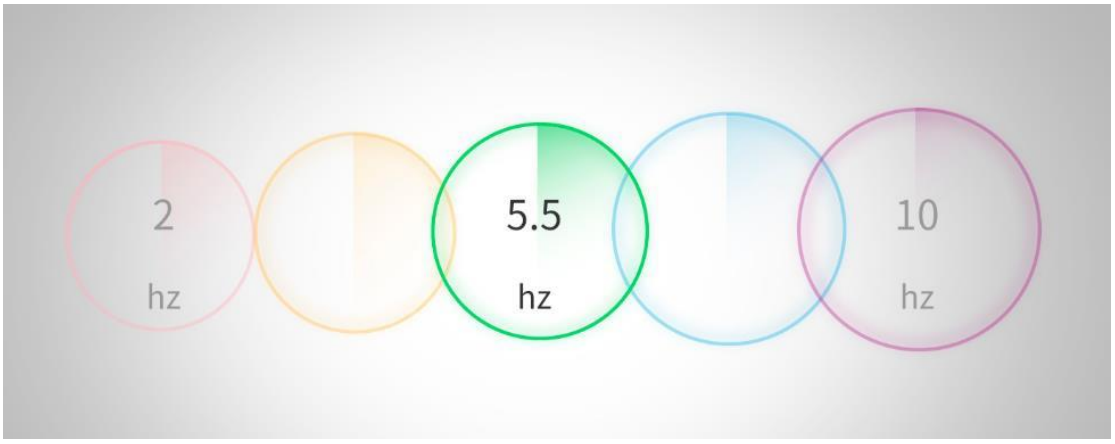

Figure 2-31

Ideal for Robot Navigation and Localization

RPLIDAR is the designed sensor for applying SLAM algorithm

Figure 2-32

Plug and Play

Just connecting the RPLIDAR and a computer via a micro USB cable, users can use the RPLIDAR without any coding job

Figure 2-33

SPECIFICATION

Model: RPLIDAR A1M8-R6

Distance Range: A1M8-R4 and the bellowing models 0.15 - 6 m A1M8-R5 0.15-12m

Angular Range: 0-360 degree

Distance Resolution: ＜0.5mm ＜1% of the distance (All distance range)

Angular Resolution: ≤1 degree

Sample Duration: 0.125 millisecond

Sample Frequency: ≥ 8000Hz

Scan Rate: 5.5Hz

Weight: 170g

2.2 Kinematics and Mechanical Design

3. Software

3.1 Robot Operation System (ROS)

ROS is an open-source, meta-operating system for your robot. It provides the services you would expect from an operating system, including hardware abstraction, low-level device control, implementation of commonly- used functionality, message-passing between processes, and package management. It is a software framework for programming robots. Prototypes originated from Stanford AI research, officially created and developed by Willow Garage starting in 2007. Currently maintained by Open-Source Robotics Foundation. Consists of infrastructure, tools, capabilities, and ecosystem.

Distro	Release date	Poster	*Tuturtle*, turtle in tutorial	EOL date
ROS Noetic Ninjemys (**Recommended**)	May 23rd, 2020			May, 2025 (Focal EOL)
ROS Melodic Morenia	May 23rd, 2018			May, 2023 (Bionic EOL)
ROS Lunar Loggerhead	May 23rd, 2017			May, 2019
ROS Kinetic Kame	May 23rd, 2016			April, 2021 (Xenial EOL)
ROS Jade Turtle	May 23rd, 2015			May, 2017
ROS Indigo Igloo	July 22nd, 2014			April, 2019 (Trusty EOL)
ROS Hydro Medusa	September 4th, 2013			May, 2015

Figure 3-1 Different Distribution of ROS.

3.1.1 ROS

3.1.1.1 ROS NODES

Form the main advantages in ROS is that Multiple nodes can publish to the same topic and also Multiple nodes can subscribe to the same topic, A node can publish to multiple topics, A node can subscribe to multiple topics. Nodes are executables that can communicate with other processes using topics, services, or the Parameter Server. Using nodes in ROS provides us with fault tolerance and separates the code and functionalities making the system simpler. A node must

have a unique name in the system. This name is used to permit the node to communicate with another node using its name without ambiguity. A node can be written using different libraries such as roscpp and rospy; roscpp is for C++ and rospy is for Python.

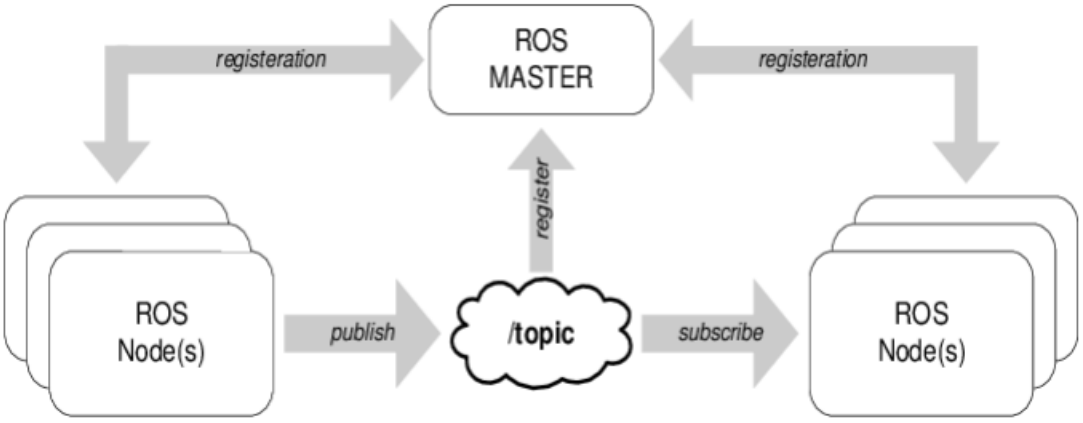

Figure 3-2 ROS Communication Graph.

3.1.1.2 ROS package

The main reason people in research and industry use ROS is the availability of multiple packages that can perform many robotic operations like basic motion to drawing a map for the scene around the robot. Packages form the atomic level of ROS. A package has the minimum structure and content to create a program within ROS.

3.1.1.3 ROS Topics and Messages

Topics are buses used by nodes to transmit data. Topics can be transmitted without a direct connection between nodes, meaning the production and consumption of data are decoupled. A topic can have various subscribers. Each topic is strongly typed by the ROS message type used to publish it, and node scan only receive messages from a matching type. A node can subscribe to a topic only if it has the same message type. The topics in ROS can be transmitted using TCP/IP and UDP. The TCP/IP-based transport is known as TCPROS and uses the persistent TCP/IP connection. This is the default transport used in ROS. The UDP-based transport is known as UDPROS and is a low-latency, lossy transport. So, it is best suited for tasks such as teleoperation. ROS has a tool to work with topics called rostopic. It is a command-line tool that gives us information about the topic or publishes data directly on the network

A message must have two principal parts: fields and constants, Fields define the type of data to be transmitted in the message, for example, int32, float32, and string, or new types that you have created before, such as type1 and type2. Constants define the name of the field.
In ROS you can find a lot of standard types to use in messages as shown in the following figure

Primitive type	Serialization	C++	Python
bool	Unsigned 8-bit int	uint8_t	bool
int8	Signed 8-bit int	int8_t	int
uint8	Unsigned 8-bit int	uint8_t	int
int16	Signed 16-bit int	int16_t	int
uint16	Unsigned 16-bit int	uint16_t	int
int32	Signed 32-bit int	int32_t	int
uint32	Unsigned 32-bit int	uint32_t	int
int64	Signed 64-bit int	int64_t	long
uint64	Unsigned 64-bit int	uint64_t	long
float32	32-bit IEEE float	float	float
float64	64-bit IEEE float	double	float
string	ASCII string (4-bit)	std::string	string
time	Secs/nsecs signed 32-bit ints	ros::Time	rospy.Time
duration	Secs/nsecs signed 32-bit ints	ros::Duration	rospy.Duration

Figure 3-4 ROS Common messages.

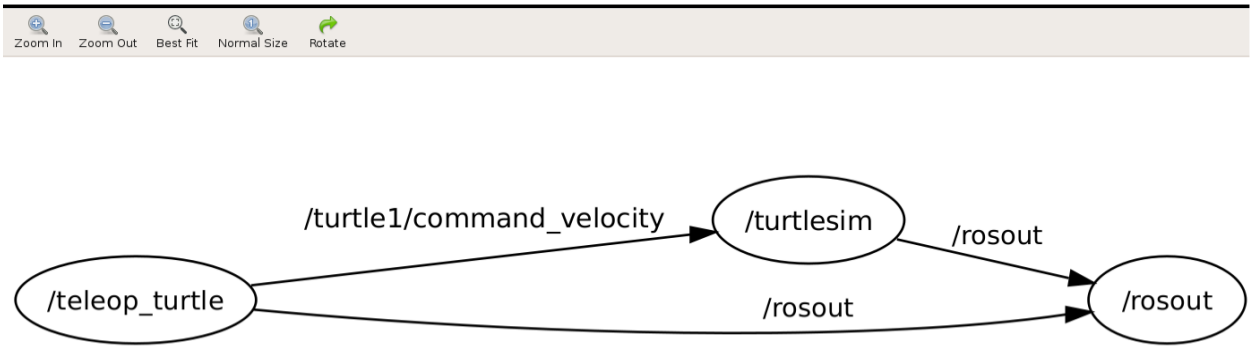

Figure 3-3 Example of topic exchange in ROS.

3.1.2 Ubuntu

Ubuntu is a Linux distribution based on Debian and composed mostly of free and open-source software. Ubuntu is officially released in three editions: Desktop, Server, and Core for Internet of things devices and robots. All the editions can run on the computer alone, or in a virtual machine. Ubuntu is a popular operating system for cloud computing, with support for OpenStack.

Figure 3-5 Ubuntu Distribution

We used Ubuntu as our operating system on jetson nano to host ROS. The version we used for Ubuntu is 18.04 and ROS Melodic.

3.1.3 Gazebo

Gazebo is an open-source 3D robotics simulator. It integrated the ODE physics engine, OpenGL rendering, and support code for sensor simulation and actuator control.

Gazebo can use multiple high-performance physics engines, such as ODE, Bullet, etc. (the default is ODE). It provides realistic rendering of environments including high-quality lighting, shadows, and textures. It can model sensors that "see" the simulated environment, such as laser range finders, cameras (including wide-angle), Kinect style sensors, etc. For 3D rendering, Gazebo uses the OGRE engine.

With Gazebo you are able to create a 3D scenario on your computer with robots, obstacles and many other objects. Gazebo also uses a physical engine for illumination, gravity, inertia, etc. You can evaluate and test your robot in difficult

or dangerous scenarios without any harm to your robot. Most of the time it is faster to run a simulator instead of starting the whole scenario on your real robot.

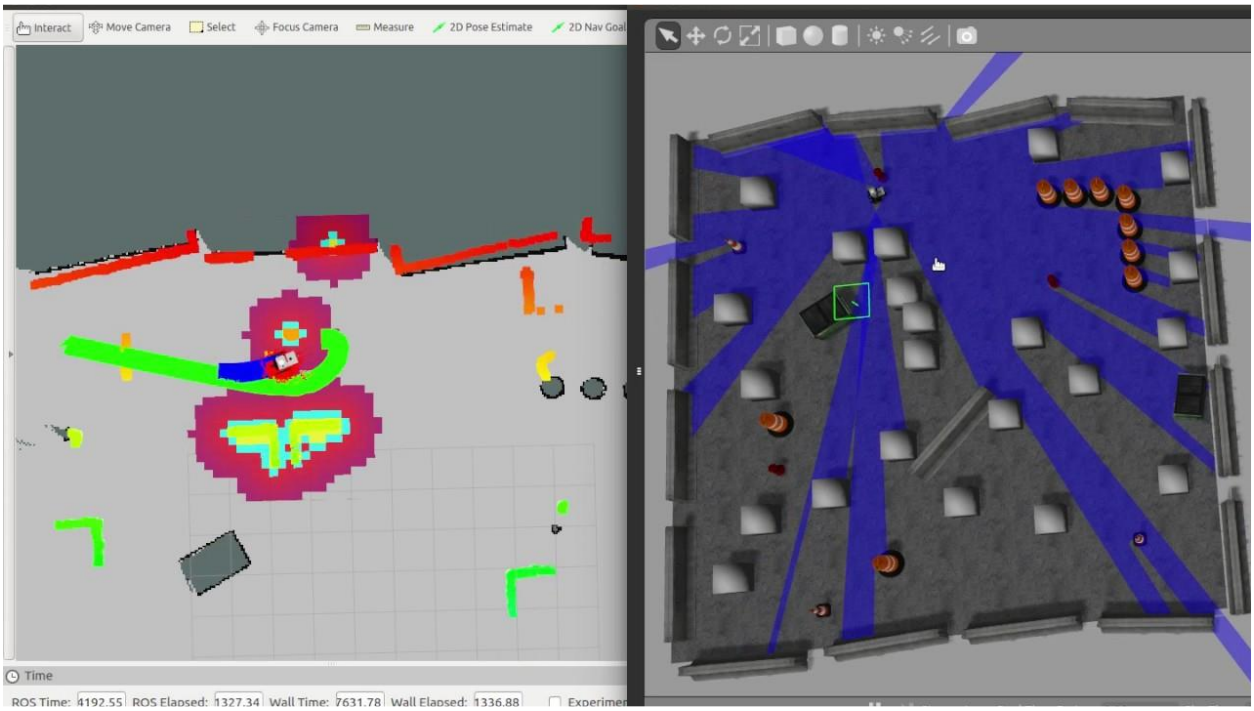

Figure 3-6 ROS Navigation with own Robot on Gazebo.

Originally Gazebo was designed to evaluate algorithms for robots. For many applications it is essential to test your robot application, like error handling, battery life, localization, navigation and grasping. As there was a need for a multi-robot simulator Gazebo was developed and improved.

3.1.4 What is rviz?

rviz (short for "ROS visualization") is a 3D visualization software tool for robots, sensors, and algorithms. It enables you to see the robot's perception of its world (real or simulated).
The purpose of **rviz** is to enable you to visualize the state of a robot. It uses sensor data to try to create an accurate depiction of what is going on in the robot's environment.
What is the Difference between **rviz** and **Gazebo**?

The difference between the two can be summed up in the following excerpt from Morgan Quigley (one of the original developers of ROS) in his book Programming Robots with ROS:

“rviz shows you what the robot thinks is happening, while Gazebo shows you what is really happening.”

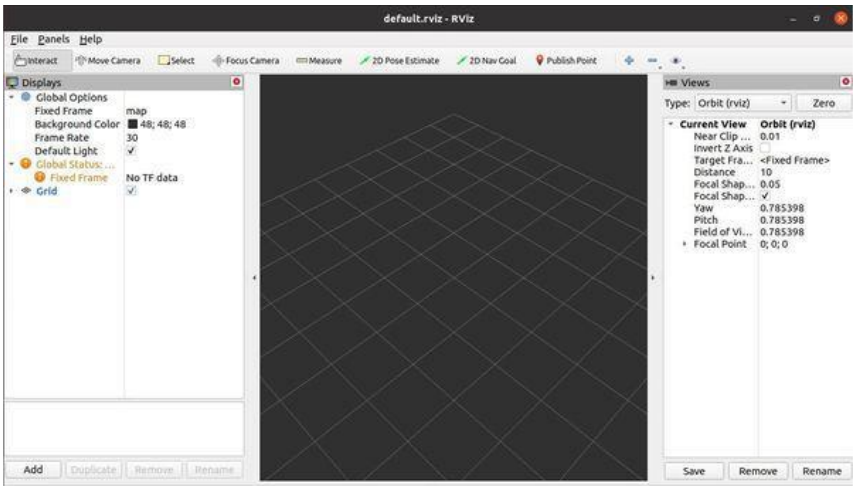

Figure 3-7 rviz Interface.

3.2. Sensors

First stage in every autonomous robot cycle is sensing. We made sure we have covered many aspects to capture all data we can collect from the surroundings environment. In this section we will discuss in depth about the sensors we used and some preprocessing for the sensory data.

3.2.1. Positional Sensors

3.2.1.1. Encoders

Encoder is a sensing device that provides feedback. It converts motion to an electrical signal that can be read by some type of control device in a motion control system. The encoder sends a feedback signal that can be used to determine position, count, speed, or direction. A control device can use this information to send a command for a particular function. Encoders used in cut-to-length

applications, plotters, robotics, packaging, conveying, automation, sorting, filling,

Figure 3-8 Motor Encoder Sensor.

imaging.

We used encoder for tracking the robot position by calculating the number of revolutions each motor made. This way is called Odometry feedback. Each encoder has a property of number of pulses generated per revolution. This is an indication of how much distance change the robot can feel and encounter. The type we used in our project is 3 PPR. But our motor is geared with a gear head of ratio 1:60.That makes the exact value 180 pulses generated for each wheel revolution. Moreover, This Encoder has 2 channels each generates 3 pulses per revolution with phase shift 90 degrees. The logic circuitry in motor driver kit combines the two channels' pulses by an XOR gate resulting in an interrupt signal of 180 PPR. This gives a resolution of 0.21 cm per encoder interrupt pulse.

Specification

- Two channel Hall Effect encoder.
- Operation Voltage 4.5~24V
- Operation Current 14~20mA
- 3PPR

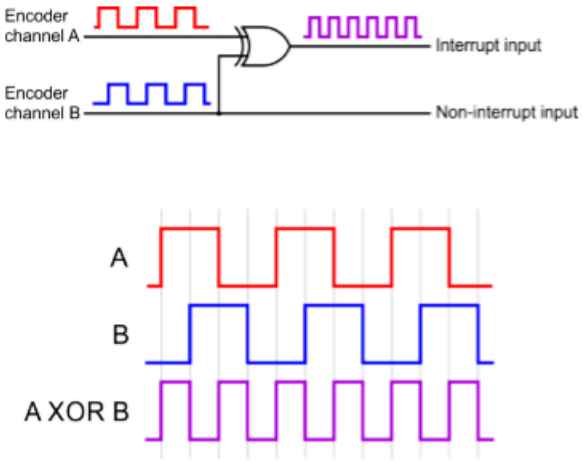

Figure 3-9 Encoder Pulse Output Signals.

3.2.1.2. IMU

An Inertial Measurement Unit (IMU) is a device that can measure and report specific gravity and angular rate of an object to which it is attached. An IMU typically consists of:

- Gyroscopes: providing a measure angular rate
- Accelerometers: providing a measure specific force/acceleration
- Magnetometers (optional): measurement of the magnetic field surrounding the system.

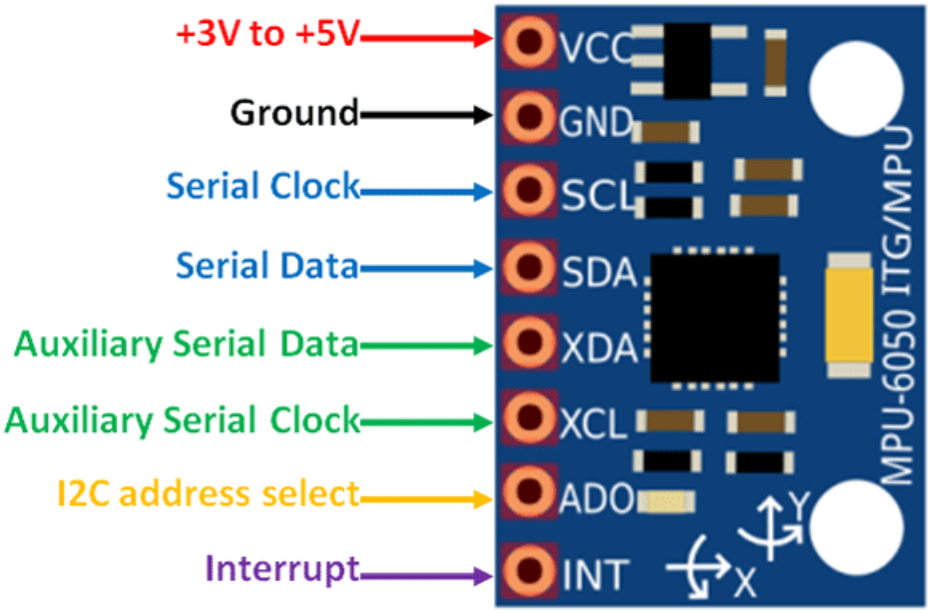

Figure 3-10 MPU-6050 IMU.

The addition of a magnetometer and filtering algorithms to determine orientation information results in a device known as an Attitude and Heading Reference Systems (AHRS).
In our robot, we used MPU 6050 which contains a MEMS accelerometer and a MEMS gyro in a single chip.
We used IMU to measure acceleration and angular velocity. So, it can help in the estimation process as we know that displacement is the integral of integral of acceleration. And on the other hand, the heading angle of robot can be calculated from integrating the angular velocity measured by gyroscope. But the integration operation in such case leads to accumulated errors as we will show in next sections.

3.2.2. Laser Sensor

3.2.2.1. Lidar Sensor

Lidar is an acronym for “light detection and ranging.” It is sometimes called “laser scanning” or “3D scanning.” The technology uses eye-safe laser beams to create a 3D representation of the surveyed environment. A typical lidar sensor emits pulsed light waves into the surrounding environment. These pulses bounce off surrounding objects and return to the sensor. The sensor uses the time it took for each pulse to return to the sensor to calculate the distance it traveled. In our robot we used RPLIDAR A1M8.

Figure 3-11 Lidar Sensor.

Features:

- Range Radius: 12 meters; Power Supply: 5V; The size of scews fit into the mounting holes on the bottom of Lidar will be M2.5.
- 360 Degree Omnidirectional Laser Range Scanning Configurable Scan Rate from 2-10Hz; Plug and Play
- 8000 Times Sample Rate, the Highest in the Current Economical LIDAR industry
- OPTMAG Original Design, prolong the life-span, Ideal for Robot Navigation and Localization

3.2.3. Raspberry pi Camera v2

The Raspberry Pi Camera Module v2 is a high quality 8-megapixel Sony IMX219 image sensor custom designed add-on board for the Raspberry Pi, featuring a fixed focus lens. The Camera Module 2 can be used to take high-definition video, as well as stills photographs. It's easy to use for beginners but has plenty to offer advanced users if you're looking to expand your knowledge. It can used for time-lapse, slow-motion, and other video cleverness. we used it in our project for remote navigation and interaction.

Figure 3-12 Raspberry pi camera

3.2.4. Frame transformation

We can't use the sensory data without using knowing where it is referenced to. And we may have the sensor connect with a little shift or in a position different form we must read data from. So, we perform Frame transformation to this data to the position we want the data to be read form.

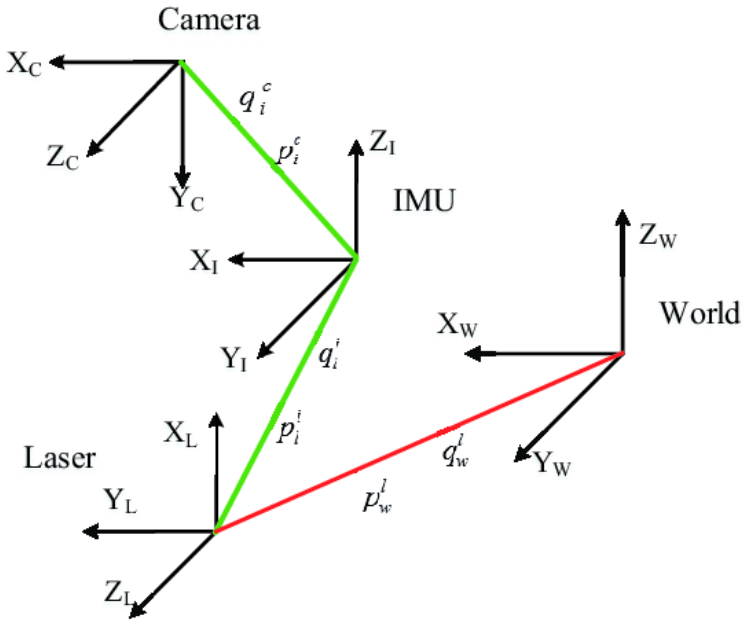

Figure 3-13 Sensors Frame Transformation to the Reference.

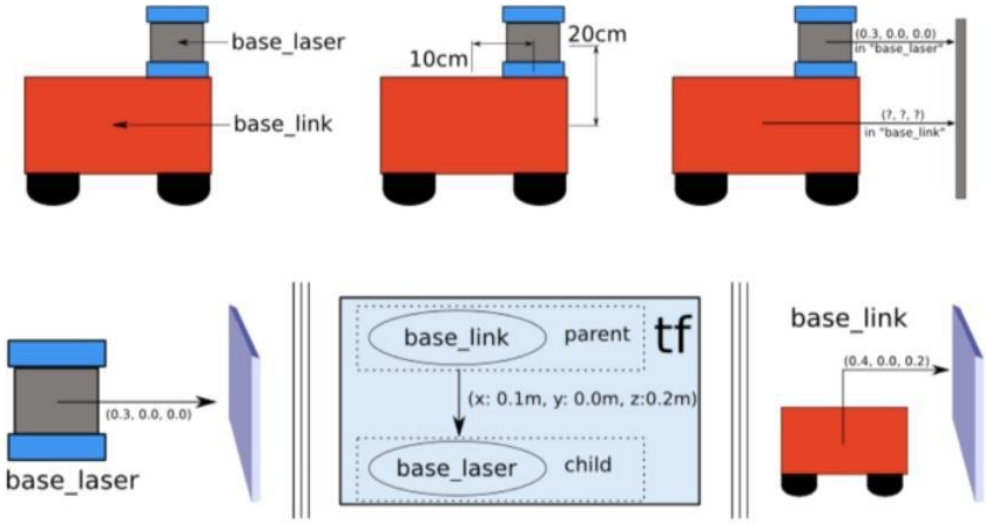

Figure 3-14 Lidar Frame Transformation to base link.

3.2.4.1. Rotation

A rotating frame of reference is a special case of a non-inertial reference frame. That is rotating relative to an inertial reference frame. An everyday example of a rotating reference frame is the surface of the Earth.

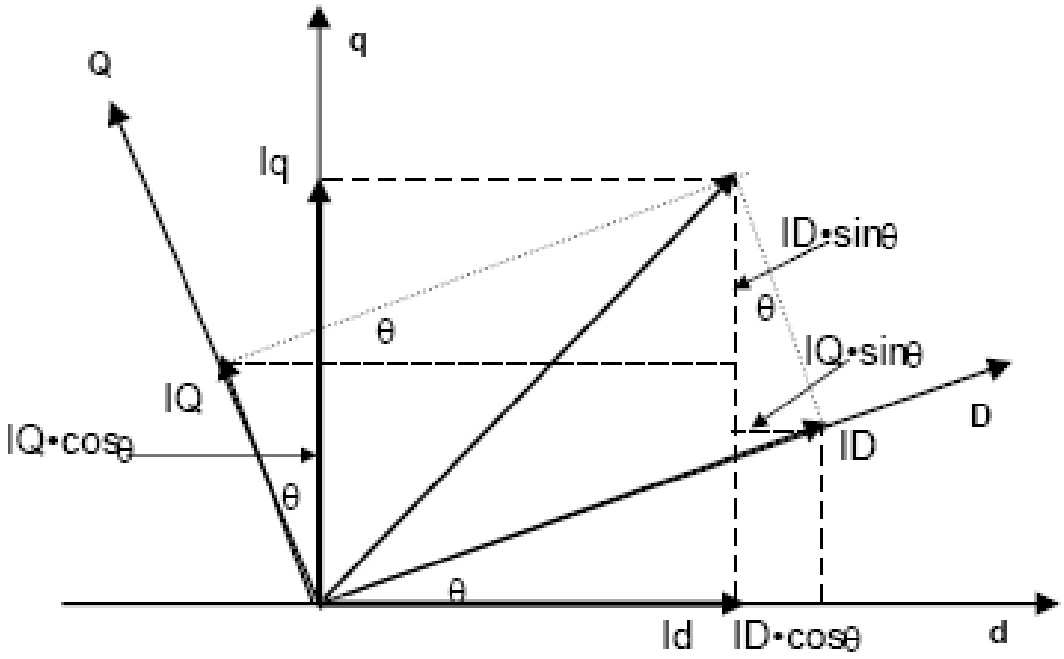

Figure 3-15 Frame Rotation.

The following equations is used to rotate the readings of the sensors around the Principal axes.

$$Rot(\mathrm{y}, \Phi) = \begin{bmatrix} C\Phi & 0 & S\Phi \\ 0 & 1 & 0 \\ -S\Phi & 0 & C\Phi \end{bmatrix} \qquad (3.1)$$

$$Rot(z, \theta) = \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (3.2)$$

$$Rot(\mathrm{x}, \alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta & -S\alpha \\ 0 & S\alpha & C\alpha \end{bmatrix} \qquad (3.3)$$

3.2.4.2. Transformation

Transformation is the rotation and also translation add to it. Using these two operations we can move all sensory data to any position reference to another position.

$$\begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix} = \begin{bmatrix} a_{xx} & a_{xy} & a_{xz} & a_{xt} \\ a_{yx} & a_{yy} & a_{yz} & a_{yt} \\ a_{zx} & a_{zy} & a_{zz} & a_{zt} \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \tag{3.4}$$

3.3 Perception

In robotics, perception is understood as a system that endows the robot with the ability to perceive, comprehend, and reason about the surrounding environment. The key components of a perception system are essentially sensory data processing, data representation (environment modeling), and ML-based algorithms.

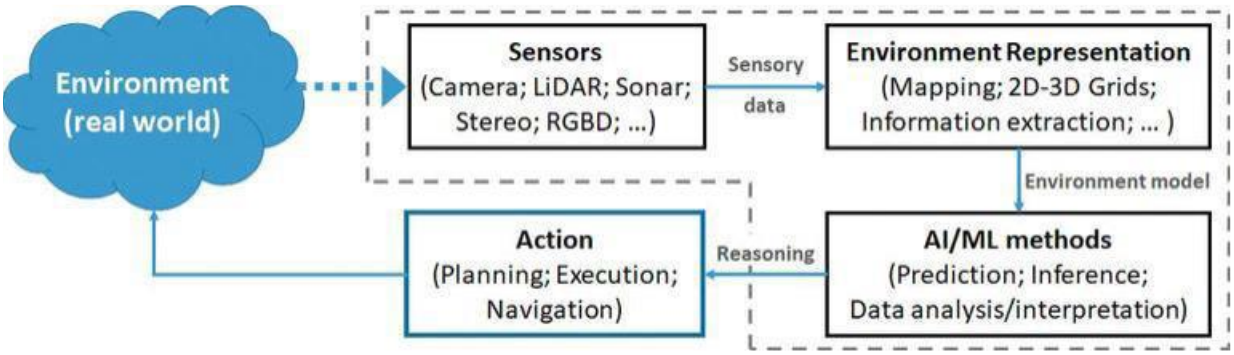

Figure 3-16 Typical robotic perception system.

Sensor-based environment representation/mapping is a very important part of a robotic perception system. Mapping here encompasses both the acquisition of a metric model and its semantic interpretation and is therefore a synonym of environment/scene representation. This semantic mapping process uses ML at various levels, e.g., reasoning on volumetric occupancy and occlusions, or identifying, describing, and matching optimally the local regions from different timestamps/models, i.e., not only higher-level interpretations. However, in most applications, the primary role of environment mapping is to model data from

exteroceptive sensors, mounted onboard the robot, in order to enable reasoning and inference regarding the real-world environment where the robot operates.

3.3.1 Data

In this topic we will talk about the sensory data our robot precept and how filter it is using famous techniques like Extended Kalman Filter (EKF), Particle Filter and many other gaussians based error handling ways.

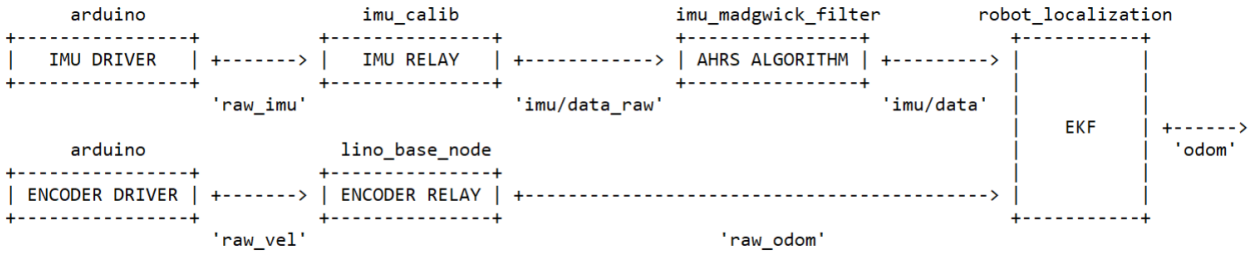

Figure 3-17 Data flow from each node and packages used before it reaches the filter.

Odometry information is used to estimate the robot's position relative to its origin. Primarily, Our Robot's linear and angular velocity found in the odometry data, published in "raw_odom", is calculated by counting the change in number of ticks over time. However, dead-reckoning that is solely based on motor encoders could be prone to errors due to system noise and wheel slippage. As a remedy, these velocities are fused with the orientation data produced by the Inertial Measurement Unit (IMU), published in "imu/data" topic, using an Extended Kalman Filter (EKF) through robot_localization package. This provides the robot an educated guess of its current pose when a sensor gets too noisy or when data starts missing during estimation.

3.3.1.1. Filtering

As mention we deal with many sensors to percept our environment but each sensor has its own uncertainty and mixing them together will lead to more unnecessary uncertainty and errors that will be hard to handle. So, we pass these data on some stages before fusion this sensory data.

In this section we will talk about these data and how handle them.

3.3.1.1.1. Encoder Data

Encoder sensor produces pulses as the robot moves. If we counted these pulses, we can estimate the position of the robot reference to the starting point. We count these pulses in a given time interval to estimate the current speed of the motor. But as we established, we need odometry as feedback to our motion and it's a Quaternion frame not a speed or pulses, so we pass these raw speed values to a node to apply numerical integration to the raw velocity, this will result in the desired frame.

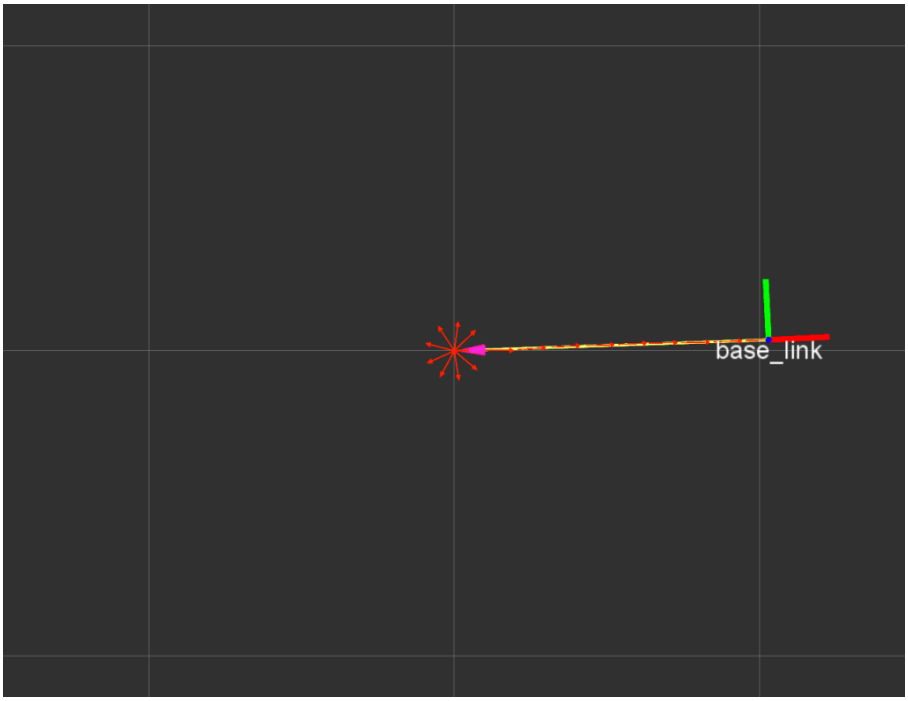

Figure 3-18 Odometry frame reference to the Robot.

$$\omega = [0, \omega_1, \omega_2, \omega_3] \tag{3.5}$$

$$\dot{\mathbf{q}} = \tfrac{1}{2}\,\Omega\mathbf{q} \tag{3.6}$$

$$\Omega = \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} \tag{3.7}$$

3.3.1.1.2. IMU Data

As we mentioned in the previous, IMU sensor generates 9 Values: 3 for angular rotation speed and 3 for linear acceleration. And finally, 3 values indicating the magnetic field in the linear direction.

As we did with the encoder data, we want to filter this data and convert it to Quaternion frame of the odometry. So, we use two stages of filtering. First calibration bias that we add to the reading due to the shifts and variance we may add due to miss fixing the sensor on the robot with a shifted angle.

And the second and the most important is the "IMU Madgwick Filter" which apply quaternion transformation to result with the odometry frame.

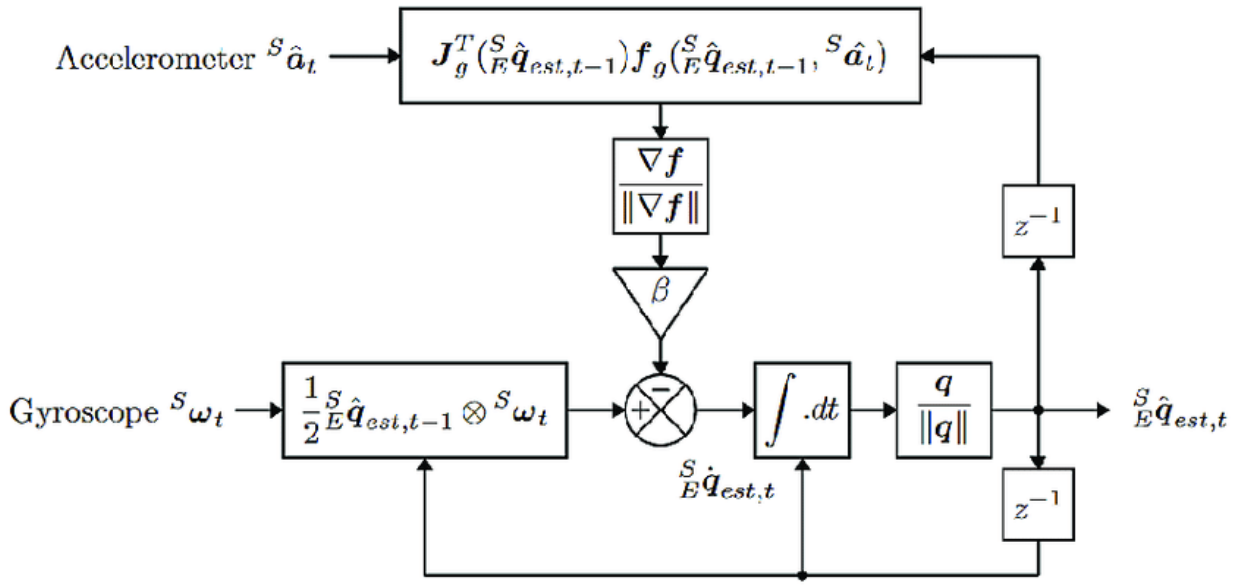

Figure 3-19 IMU Madgwick Filter Block Diagram.

3.3.1.2. Fusion

Now we have two sources of odometry but off course they have their own uncertainty and we need to get rid of this error or at least minimize it. In doing so we will use a technique called Extended Kalman filter (EFK).

3.3.1.2.1. Extended Kalman filter (EFK)

In the extended Kalman filter, the state transition and observation models don't need to be linear functions of the state but may instead be differentiable functions.

$$\boldsymbol{x}k = f(\boldsymbol{x}k-1, \boldsymbol{u}k\ ) + \boldsymbol{w}k \qquad (3.8)$$

$$\boldsymbol{z}k = h(\boldsymbol{x}k\ ) + \boldsymbol{v}k \qquad (3.9)$$

Here wk and vk are the process and observation noises which are both assumed to be zero mean multivariate Gaussian noises with covariance Qk and Rk respectively. uk is the control vector. The function f can be used to compute the predicted state from the previous estimate and similarly the function h can be used to compute the predicted measurement from the predicted state.

However, f and h cannot be applied to the covariance directly. Instead, a matrix of partial derivatives (the Jacobian) is computed.

At each time step, the Jacobian is evaluated with current predicted states. These matrices can be used in the Kalman filter equations. This process essentially linearizes the non-linear function around the current estimate.

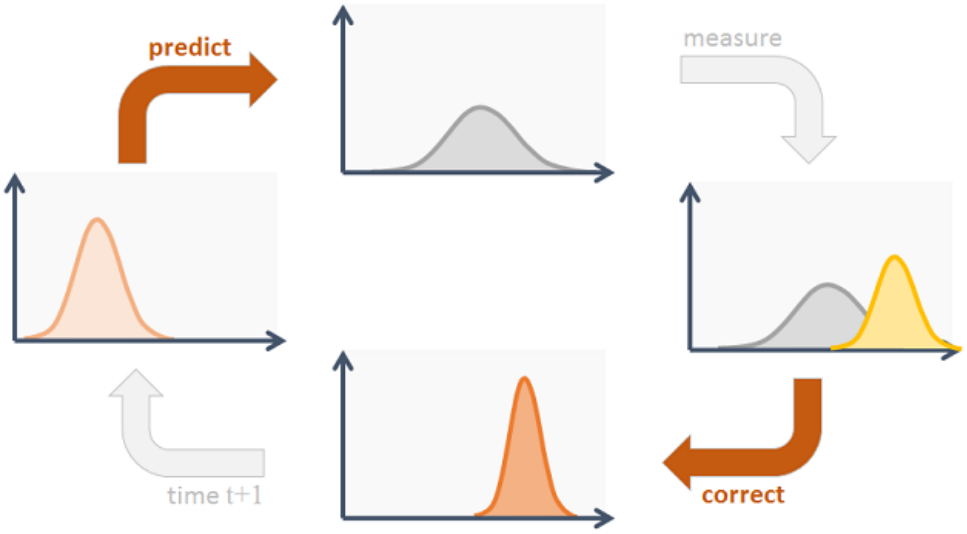

Figure 3-20 Kalman Filter predict, measure, and update process.

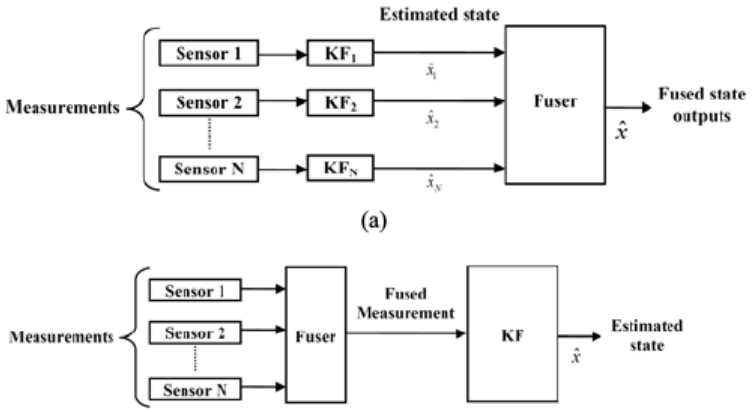

Figure 3-21 Sensor fusion block Diagram Using Kalman Filter.

3.3.2. Navigation

In robotic mapping and navigation, simultaneous localization and mapping (SLAM) is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it. While this initially appears to be a chicken-and-egg problem there are several algorithms known for solving it, at least approximately, in tractable time for certain environments. Popular approximate solution methods include the particle filter, extended Kalman filter, and Graph SLAM.

SLAM algorithms are tailored to the available resources, hence not aimed at perfection, but at operational compliance. Published approaches are employed in self-driving cars, unmanned aerial vehicles, autonomous underwater vehicles, planetary rovers, newer domestic robots and even inside the human body.

3.3.2.1. Problem Formation

Given a series of sensor observations Ot over discrete time steps t, the SLAM problem is to compute an estimate of the agent's location Xt and a map of the environment mt. All quantities are usually probabilistic, so the objective is to compute:

$$P\ (mt,\ xt \mid o1{:}t) \qquad (3.10)$$

Applying Bayes' rule gives a framework for sequentially updating the location posteriors, given a map and a transition function $P(xt \mid xt{-}1)$.

$$P(x_t \mid o_{1:t}, m_t) = \sum_{m_{t-1}} P(o_t \mid x_t, m_t) \sum_{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1} \mid m_t, o_{1:t-1}) / Z \tag{3.11}$$

Similarly, the map can be updated sequentially by:

$$P(m_t \mid x_t, o_{1:t}) = \sum_{x_t} \sum_{m_t} P(m_t \mid x_t, m_{t-1}, o_t) P(m_{t-1}, x_t \mid o_{1:t-1}, m_{t-1}) \tag{3.12}$$

Like many inference problems, the solutions to inferring the two variables together can be found, to a local optimum solution, by alternating updates of the two beliefs in a form of EM algorithm.

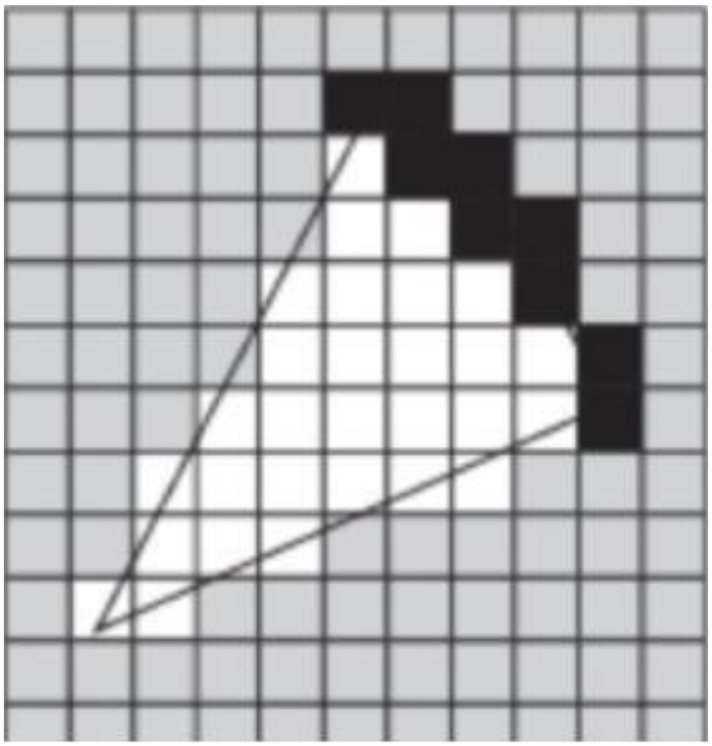

Figure 3-22 Map Grid Cell Occupancy.

So, we represent occupant grid cell with black and white grid cell as free and for unexplored areas they are colored with gray.

3.3.2.2. GPS

GPS is a constellation of satellites that provides a user with an accurate position on the surface of the earth. This satellite based navigation system was developed by the U.S. Department of Defense (DoD) in early 1970s. It was first intended for military use but later it was made available to civilian users. GPS can provide precise position and time information to a user anywhere in the world. It is a one way system i.e. a user can only receive signals but cannot send signals to the satellite. This type of configuration is needed due to security reasons as wells as to serve unlimited number of users.

The GPS system consists of three main segments:

1) Space segment
2) Control segment
3) User segment

The space segment consists of a constellation of 24 satellites in fixed orbits around the earth. Each satellite continuously transmits GPS signals to the earth, which consist of two carrier frequencies, digital codes, and navigation message. The carrier frequency and codes are used in determining distance of the satellite from the receiver. The navigation message consists of information like satellite location and clock compensation

The control segment includes a network of tracking stations. These tracking stations continuously monitor the satellite orbit; checks satellite clock, atmospheric conditions, satellite almanac, provide necessary compensation for clock error, and upload data to satellites. The OCS (Operational Control System) consists of one master control station (MCS), several monitor stations, and ground control stations. The MCS is located in the United States at Shriever Air Force Base, Colorado Springs, Colorado.

The user segment consists of all civilian and military users. A GPS receiver can lock on to any visible satellites and determines its location on earth's surface. GPS system was first conceived for military usage such as navigating in remote terrains and coordinating military activities. With addition of civilian access to GPS signals, it is used in various applications such as surveying, mapping, hiking, and vehicular navigation.

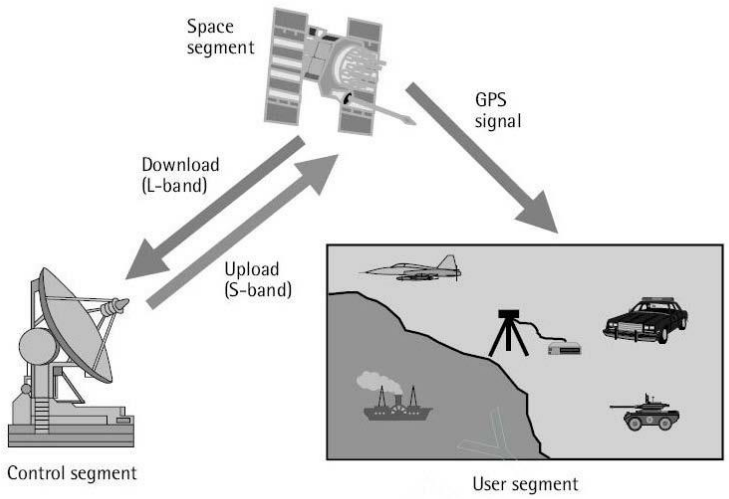

Figure 3-23 GPS segments

An autonomous car navigation system based on Global Positioning System (GPS) is a new and promising technology, which uses real time geographical data received from several GPS satellites to calculate longitude, latitude, speed and course to help navigate a car. The goal of the project is to make an auto navigational car model that can route through known or pre-programmed co-ordinates autonomously without any control by human. It is designed to demonstrate the feasibility of using a low-cost strap-down inertial measurement unit (IMU) to navigate between intermittent GPS fixes. The present hardware consists of a GPS receiver, IMU, compass and a data processing computer.

This research paper demonstrates a prototype development of autonomous car to enable the aerial vehicle accomplish the required autonomy and maintain satisfactory automated driving operation. Design objectives include simplicity, robustness, versatility and improved operational utility. The driverless car features an advanced, highly autonomous drive control system with an anti-drift from its course algorithms. The discovery of interacting hardware and software had led to

breakthrough improvements in the efficiency of maintaining planned route on the track.

3.3.3. Localization

Now that we have constructed the map. And we want to know where is our robot this map. This is known as Localization. Referring to locating the robot reference to the map center.

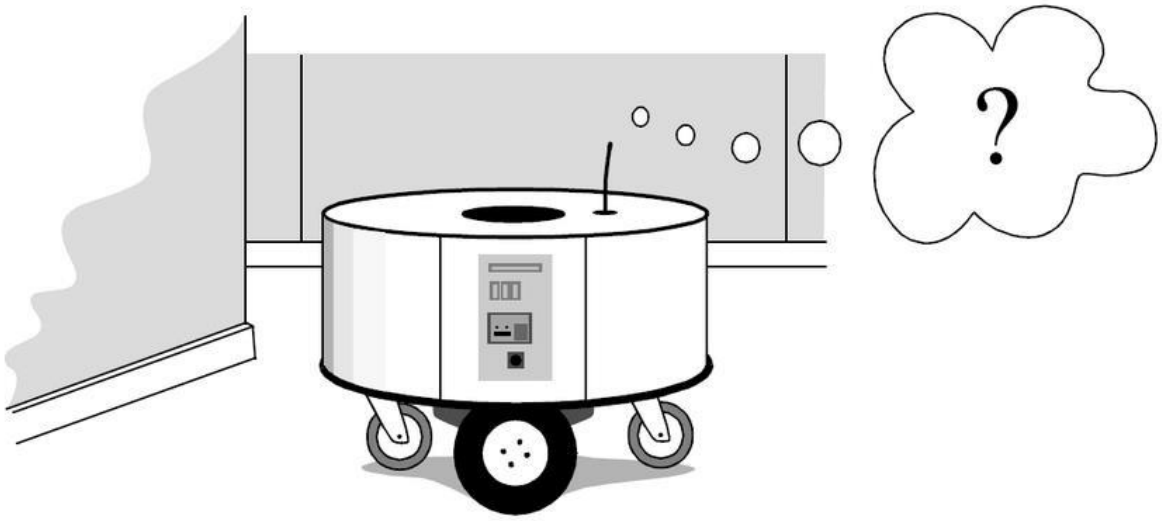

Figure 3-24 Robot doesn't know where he is

3.3.3.1. Adaptive Monte Carlo Localization

AMCL is a probabilistic localization system for a robot moving in 2D. It implements the adaptive (or KLD-sampling) Monte Carlo localization approach (As described by Dieter Fox), which uses a particle filter to track the pose of a robot against a known map.

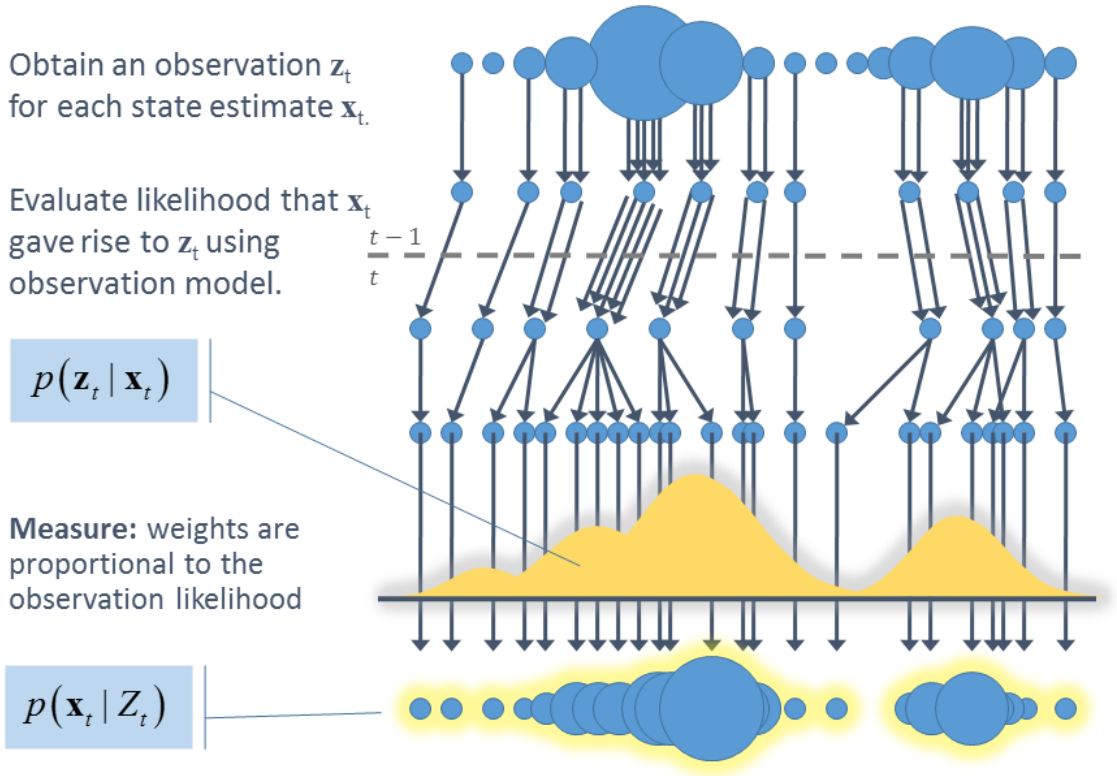

Figure 3-25 Particle Filter sample Matching.

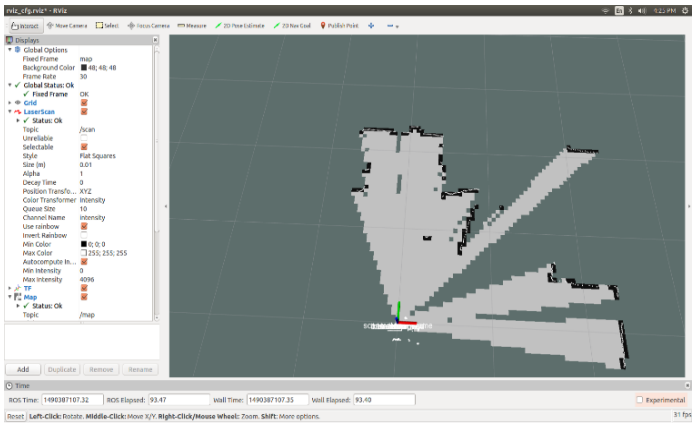

Figure 3-26 Corrected Localization by AMCL.

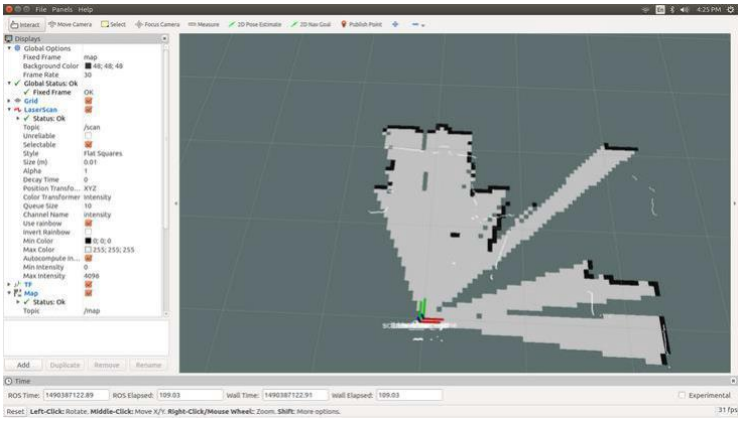

Figure 3-27 Wrong Localization before Using AMCL

The AMCL algorithm:

1) Initialize particle poses and weights according to known start-state information

2) Randomly re-sample particles based on previously computed weights

3) Update particle positions based on motion model and available motion sensor information

4) Update particle weights based on a sensor model by comparing hypotheses to the occupancy grid map and sensor readings

5) Compute expected value of robot pose based on the particle distribution

6) Repeat from step 2

3.4. Path Planning

The path planning problem of mobile robots is a hot spot in the field of mobile robot navigation research: mobile robots can find an optimal or near-optimal path from the starting state to the target state that avoids obstacles based on one or some performance indicators (such as the lowest working cost, the shortest walking route, the shortest walking time, etc.) in the motion space. Generally speaking, the path planning selects the shortest distance of the path: that is, the shortest length of the path from the starting point to the target point as a performance indicator. The mobile robot path planning method can be divided into two types according to the known degree of environmental information: path planning based on global map information or local map information. These two path planning methods are referred to as global path planning and local path planning. The global path planning method can generate the path under the completely known environment (the position and shape of the obstacle are predetermined). With the global map model of the environment where the mobile robots are located, the search is performed on the established global map model. The optimal algorithm can obtain the optimal path. Therefore, global path planning involves two parts: establishment of the environmental model and the path planning strategy.

The Navigation Stack is fairly simple on a conceptual level. It takes in information from odometry and sensor streams and outputs velocity commands to send to a mobile base. Use of the Navigation Stack on an arbitrary robot, however, is a bit more complicated. As a pre-requisite for navigation stack use, the robot must be running ROS, have a TF transform tree in place, and publish sensor data using the correct ROS Message types. Also, the Navigation Stack needs to be configured for the shape and dynamics of a robot to perform at a high level. To help with this

process, this manual is meant to serve as a guide to typical Navigation Stack setup and configuration.

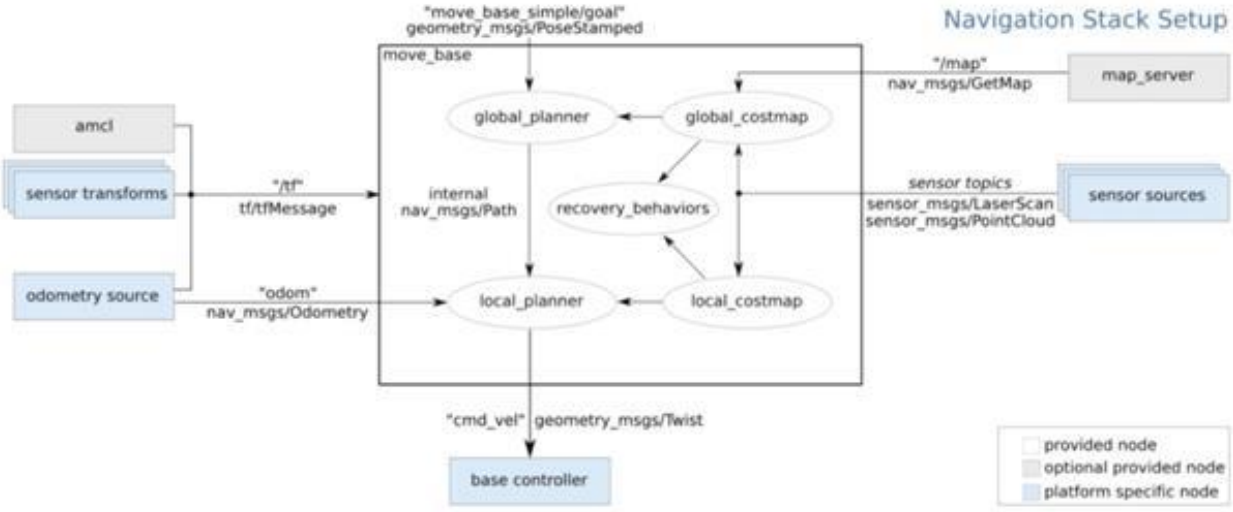

Figure 3-28 ROS Navigation Stack Setup.

3.4.1. Global Planner

As said before, the global planner requires a map of the environment to calculate the best route. Depending on the analysis of the map, some methods are based on Roadmaps like Silhouette proposed by Canny in 1987 or Voronoi in 2007. Some of them solve the problem by assigning a value to each region of the roadmap in order to find the path with minimum cost. Some examples are the Dijkstra algorithm, Best First, and. Another approach is by dividing the map into small regions (cells) called cell decomposition as. A similar approach by using potential fields, the most extended algorithm used few years ago rapidly exploring random trees or the new approach based on neural networks. Some solutions combine the aforementioned algorithms improving the outcome at the cost of high.

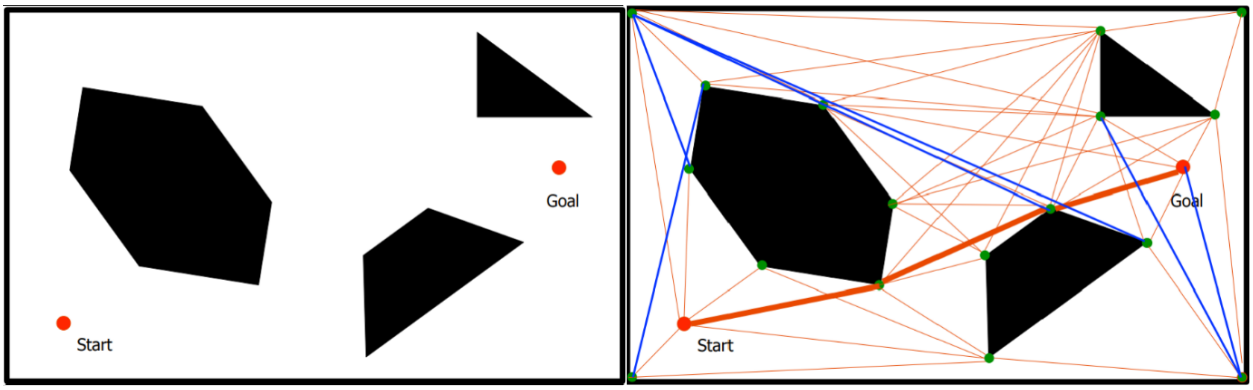

Figure 3-29 Graph Search representation.

3.4.1.1. A* Graph Search Algorithm

A* Search attempts to improve upon the performance of grassfire and Dijkstra by incorporating a heuristic function that guides the path planner Heuristic function H(n), takes a node n and returns a non-negative real number that is an estimation of the path cost from node n to the goa

```
For each node n in the graph
n.f = Infinity, n.g = Infinity
Create an empty list
start.g = 0 , start.f = H(start) add start to list
While list not empty
Let current = node in the list with the smallest f value, remove current from list
If (current == goal node) report success
For each node, n that is adjacent to current
If (n.g > (current.g + cost of edge from n to current))
n.g = current.g + cost of edge from n to current
n.f = n.g + H(n)
n.parent = current add n to list if it isn't there already
```

Figure 3-30 A Graph Search Algorithm.*

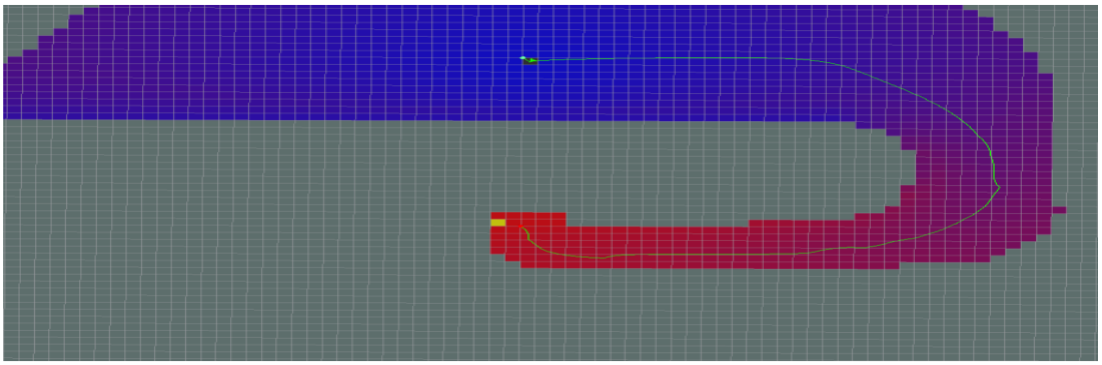

Figure 3-31 Global path planning using A.*

3.4.2. Local Planner (Obstacle Avoidance)

In order to transform the global path into suitable waypoints, the local planner creates new waypoints taking into consideration the dynamic obstacles and the vehicle constraints. So, to recalculate the path at a specific rate, the map is reduced to the surroundings of the vehicle and is updated as the vehicle is moving around. It is not possible to use the whole map because the sensors are unable to update the map in all regions and a large number of cells would raise the computational cost. Therefore, with the updated local map and the global waypoints, the local planning generates avoidance strategies for dynamic obstacles and tries to match the trajectory as much as possible to the provided waypoints from the global planner. Different approaches are based on trajectory generation using Clothoids lines, Bezier lines, arcs and segments, or splines. All of them create intermediate waypoints following the generated trajectory..

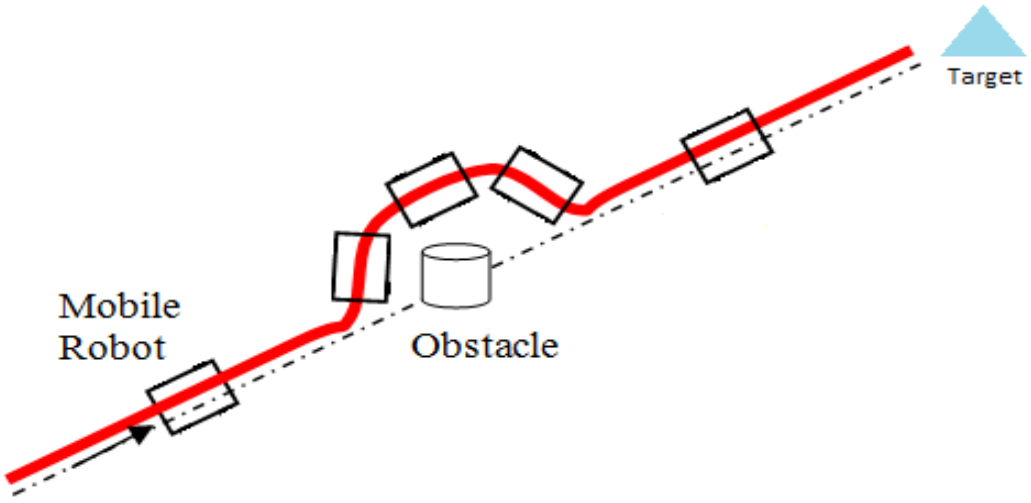

Figure 3-32 Local planning a path to avoid the obstacle.

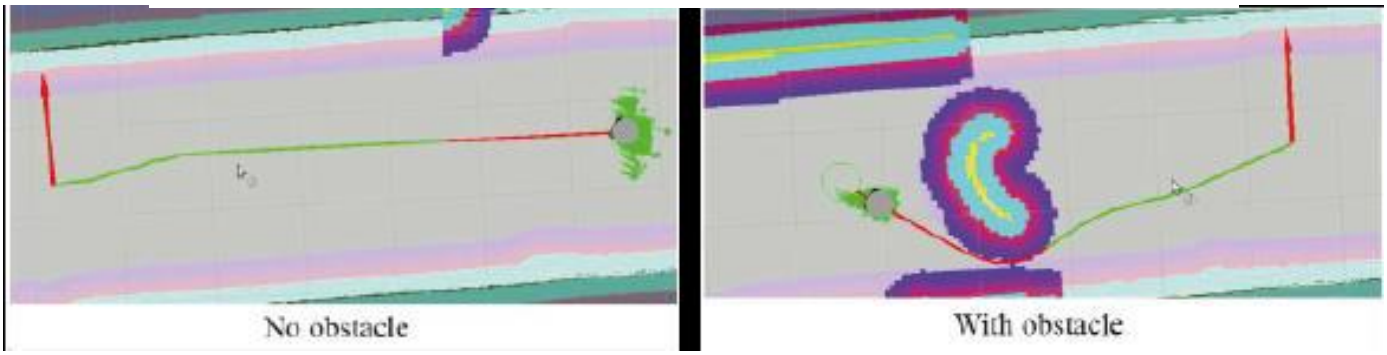

Figure 3-33 ROS Local planning using Cost Map.

3.4.3. Recovery behavior

It is a package provides a recovery behavior for the navigation stack that attempts to clear space by performing a 360-degree rotation of the robot. It is activated when surrounded by obstacle that is preventing the robot to track the global planar path or the given goal is invalid due to robot geometry.

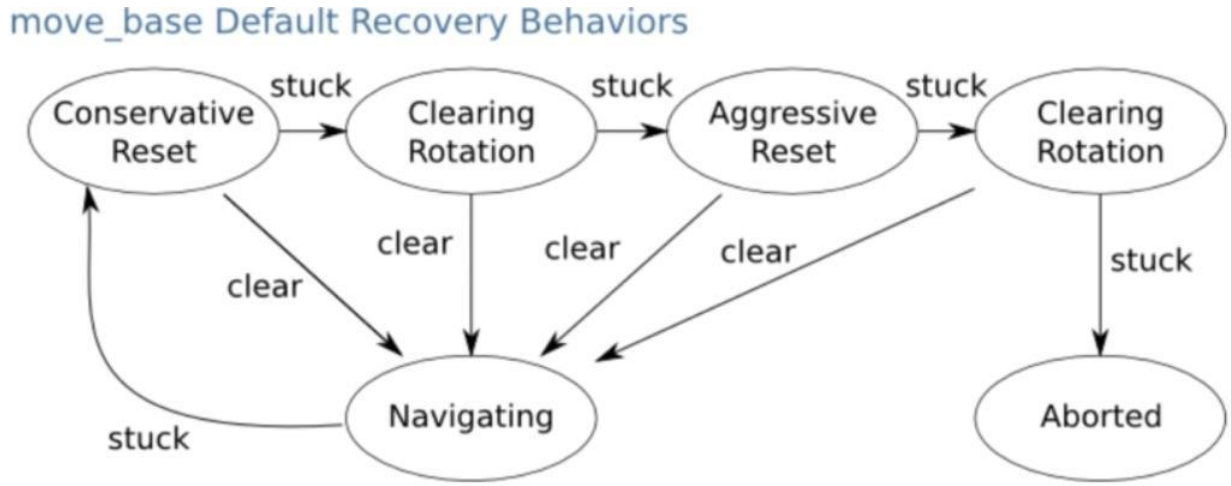

Figure 3-34 Recovery Behavior Flowchart.

3.5. Control

A control loop is the fundamental building block of industrial control systems. It consists of all the physical components and control functions necessary to automatically adjust the value of a measured process variable (PV) to equal the value of a desired set-point (SP). It includes the process sensor, the controller function, and the final control element (FCE) which are all required for automatic control.

3.5.1. Proportional Integral Derivative Controller (PID)

A proportional–integral–derivative controller (PID controller or three term controller) is a control loop feedback mechanism widely used in industrial control systems and a variety of other applications requiring continuously modulated control. A PID controller continuously calculates an error value e (t) as the difference between a desired set point (SP) and a measured process variable (PV) and applies a correction based on proportional, integral, and derivative terms (Denoted P, I, and D respectively) which give the controller its name.

In practical terms it automatically applies accurate and responsive correction to a control function. An everyday example is the cruise control on a road vehicle; where external influences such as gradients would cause speed changes, and the driver has the ability to alter the desired set speed. The PID algorithm restores the actual speed to the desired speed in the optimum way, without delay or overshoot, by controlling the power output of the vehicle's engine.
The first theoretical analysis and practical application was in the field of automatic steering systems for ships, developed from the early 1920s onwards. It was then used for automatic process control in manufacturing industry, where it was widely implemented in pneumatic, and then electronic, controllers. Today there is universal use of the PID concept in applications requiring accurate and optimized automatic control.

3.6. Computer Vision

Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs — and take actions or make recommendations based on that information. If AI enables computers to think, computer vision enables them to see, observe and understand.

Computer vision works much the same as human vision, except humans have a head start. Human sight has the advantage of lifetimes of context to train how to tell objects apart, how far away they are, whether they are moving and whether there is something wrong in an image.

Computer vision trains machines to perform these functions, but it has to do it in much less time with cameras, data and algorithms rather than retinas, optic nerves and a visual cortex. Because a system trained to inspect products or watch a production asset can analyze thousands of products or processes a minute, noticing imperceptible defects or issues, it can quickly surpass human capabilities.

3.6.1 Barcode

A **barcode** or **bar code** is a method of representing data in a visual, machine readable form. Initially, barcodes represented data by varying the widths, spacings and sizes of parallel lines. These barcodes, now commonly referred to as linear or one-dimensional (1D), can be scanned by optical scanners, called barcode readers, of which there are several types. Later, two-dimensional (2D) variants were developed, using rectangles, dots, hexagons and other patterns, called *matrix codes* or *2D barcodes*, although they do not use bars as such. 2D barcodes can be read using purpose-built 2D optical scanners, which exist in a few different forms. 2D barcodes can also be read by a digital camera connected to a microcomputer running software that takes a photographic image of the barcode and analyzes the image to deconstruct and decode the 2D barcode. A mobile device with an inbuilt camera, such as smartphone, can function as the latter type of 2D barcode reader using specialized application software (The same sort of mobile device could also read 1D barcodes, depending on the application software).

3.6.2 Barcode Verification

The process of checking the quality of a photograph includes a barcode before and after printing. Computer vision software sets up standard-based support for barcode verification utilizing parameters related to ISO or AIM barcode quality standards.

Each time a code is entered into a barcode scanner, the scanner's read mechanism will interpret the bars and spaces in the code as the correct symbol. Then, depending on the need of each standard, codes can be visually compared to other printed codes using a 'camera-based' device or an optical character recognition engine.

When using a camera-based device, typically a flatbed scanner or standalone camera, the camera only 'hears' the sound of an audible beep when it reads a symbol. An OCR engine generally requires more even lighting settings and better barcode positioning to obtain a legible image for analysis.

There are three significant barcodes: 1D barcodes consisting of lines, 2D barcodes containing areas, and hybrid barcodes that encode data using both methods. Depending on their width, they scanned with either a narrow or a broad laser.

For Instance, Scandit has created the world's first barcode scanning platform that is 100% mobile, requires no manual input from the user, and uses computer vision augmented reality. The barcode scanning solution is available for iOS and Android users worldwide.

3.6.3 Why Verify?

Barcode quality is integral to the success of an automated system. In a process where quality barcodes accurately store and communicate data – from code to reader to central system – little manual intervention is required. Thanks to quality barcodes, the unique benefits of an automated system are realized: lower costs, higher productivity, and fewer errors. Poor quality barcodes, however, render the system almost as inefficient as using no automation at all. Unreadable barcodes may require re-labeling, re-scanning, or even manual entry of critical information by a human operator – disrupting the productivity of the process and causing a significant loss of time. Bad barcodes may prevent error-tracking, causing a domino effect of failures down the line and resulting in costly scrap and rework. All told, these effects completely counteract the benefits of implementing an automated system, the result being inflated cost, loss of productivity, and increased errors.

The purpose of barcode verification is to prevent this outcome and preserve the intended benefits of the automated system. Verification systems evaluate a barcode's quality against published quality standards for 1D and 2D barcodes using precision instruments such as barcode verifiers or machine vision systems. A verified barcode ensures consistent readability, supporting 100% accurate automated data capture.

3.6.4 When Should You Verify?

To ensure that errors are prevented as early in the automated system as possible, verification must occur before a part enters the system. A verification

step should occur after a part is marked or labeled with a barcode and before the part reaches the station where the barcode is first read.

Figure 3-35 verification must occur before a part enters the system.

Proper verification ensures that every part is processed and shipped with a high-

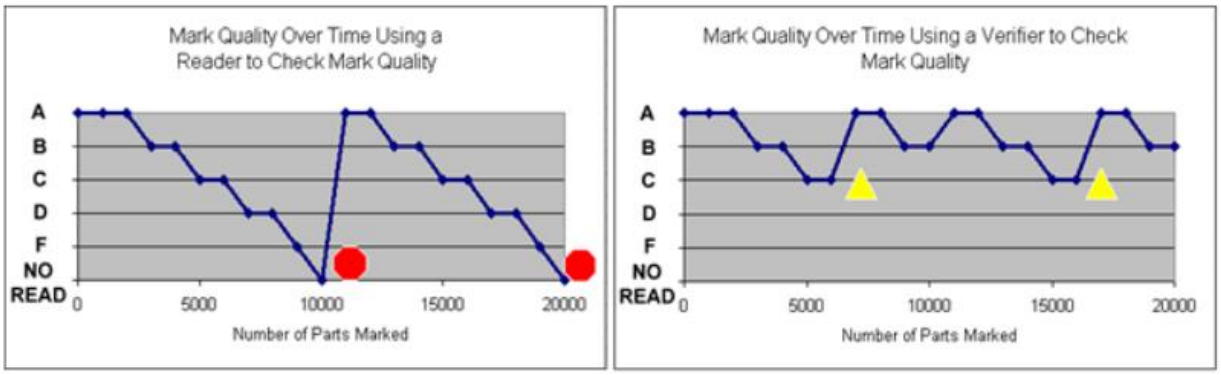

Figure 3-36 Difference between using verifier and reader to check mark quality

quality barcode, despite the fact that marking and labeling systems will degrade over time. A verification system is much more accurate than a standard barcode reader at identifying low-quality barcodes early in the process, before parts with bad barcodes make it through the line and are shipped to end customers. When barcode quality degradation is identified early, the marking or labeling system may be adjusted or replaced before unreadable barcodes are ever produced.

3.6.5 Validation vs. Verification

Depending on the requirements of a particular process, industry, company, or customer, there are two levels of quality grading for ensuring barcode readability: validation (sometimes called process control) and true verification.

Validation/Process Control: Process control is a means of ensuring that barcodes are readable throughout a particular internal or internal/external process. Process

control does not check barcodes for compliance to a published barcode quality standard. Instead, it provides objective measurements for barcode quality when verification to a standard is not possible or not desired. If you are not concerned with meeting published barcode quality standards in your application, you may opt to use a subset of the default verification parameters in the verification system as the criteria for passing codes.

Verification: Verification ensures that a barcode complies with published barcode quality standards, such as ISO 15415, ISO 15416, and AIM DPM. To ensure compliance, all evaluation parameters in the machine vision system must be enabled during the verification process. Fully-conforming verification systems provide reports as evidence of barcode compliance, which can be sent to customers or other invested parties to provide the highest assurance of barcode quality and consistency.

3.6.6 Verification Evaluation Parameters

There are a number of verification evaluation parameters that determine barcode quality and they may be used for either true verification or validation/process control. Published barcode quality standards, such as ISO 15415, ISO 15416, and AIM DPM, require that a designated set of these parameters be met to ensure that a barcode is verified to the standard, while process control grading may require that a barcode meet only a subset of these parameters. Parameters for 1D and 2D barcode evaluation are shown below.

Figure 3-37 1D Verification Evaluation Parameters

Figure 3-38 1D Verification Evaluation

4. References

1. M. B. Alatise and G. P. Hancke, (2020), "A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods," in IEEE Access, vol. 8, pp. 39830-39846, doi: 10.1109/ACCESS.2020.2975643.

2. Kamel, Mohamed & Zhang, Youmin. (2014). Developments and Challenges in Wheeled Mobile Robot Control. 10.13140/RG.2.1.3595.3363.

3. Margarita Martínez-Díaz, (2018), Francesc Soriguera,Autonomous vehicles: theoretical and practical challenges,Transportation Research Procedia,Volume 33 ,Pages 275-282,ISSN 2352-1465

4. Premebida, Cristiano & Ambrus, Rares & Marton, Zoltan. (2018). Intelligent Robotic Perception Systems. 10.5772/intechopen.79742.

5. Robot Path Planning with Avoiding Obstacles in Known Environment Using Free Segments and Turning Points Algorithm Imen Hassani , Imen Maalej , and Chokri Rekik ,Control and Energy Management Laboratory (CEM Lab), National Engineering School of Sfax, University of Sfax, Sfax, Tunisia

6. Bensaci, Chaima & Youcef, Zennir & Pomorski, Denis. (2017). Turtlebot 2 Autonomous Navigation under the Virtual World Gazebo.

7. Zhu, Jianjun & Xu, Li. (2019). Design and Implementation of ROS-Based Autonomous Mobile Robot Positioning and Navigation System. 214-217. 10.1109/DCABES48411.2019.00060.

8. Lentin, J. ed. (2018) Robot Operating System (ROS) for Absolute Beginners. Birmingham: Packt.

9. Lentin, J. ed. (2017) ROS Robotics Projects. Birmingham: Packt.

10. Quigley, Morgan & Conley, Ken & Gerkey, Brian & Faust, Josh & Foote, Tully & Leibs, Jeremy & Wheeler, Rob & Ng, Andrew. (2009). ROS: an open-source Robot Operating System. ICRA Workshop on Open Source Software. 3.

11. ROS for Human-Robot Interaction Youssef Mohamed, Séverin Lemaignan

12. Ubuntu 18.04.6 (2022), LTS (Bionic Beaver). https://releases.ubuntu.com/18.04/. Accessed 15 Jun.

13. Gramescu, Bogdan & Nitu, Constantin. (2014). Block diagram modelling of a DC motor encoder. 185-188.

14. Ahmad, Norhafizan & Raja Ghazilla, Raja Ariffin & Khairi, Nazirah & Kasi, Vijayabaskar. (2013). Reviews on Various Inertial Measurement Unit (IMU) Sensor Applications. International Journal of Signal Processing Systems. 1. 256-262. 10.12720/ijsps.1.2.256-262.

15. KinectFusion: Real-Time Dense Surface Mapping and Tracking Richard A. Newcombe , Shahram Izadi , Otmar Hilliges , David Molyneaux

16. Guzman, Roberto & Navarro, Roman & Cantero, Miquel & Arino, Jorge. (2017). Robotnik—Professional Service Robotics Applications with ROS (2). 10.1007/978-3-319-54927-9_13.

17. Toroslu, I., & Doğan, M. (2018). Effective sensor fusion of a mobile robot for SLAM implementation. 4th International Conference on Control, Automation and Robotics (ICCAR), 76-81.

18. SLAM and Navigation in Indoor Environments Lin, Shang-Yen and Chen, Yung-Chang (2012), Springer Berlin Heidelberg",Berlin, Heidelberg,48--60,

19. Simultaneous Localization and Mapping (SLAM) using RTAB-Map, Sagarnil Das (2018).

20. Monte Carlo Localization: Efficient Position Estimation for Mobile Robots Dieter Fox, Wolfram Burgardy, Frank Dellaert, Sebastian Thrun

21. Gayathri Rajendran, Uma V, Bettina O'Brien, (2021), Unified robot task and motion planning with extended planner using ROS simulator, Journal of King Saud University - Computer and Information Sciences.

22. Kiam Heong Ang, G. Chong and Yun Li, (2005), "PID control system analysis, design, and technology," in IEEE Transactions on Control Systems Technology, vol. 13, no. 4, pp. 559-576, doi: 10.1109/TCST.2005.847331.

23. Pfister Cuno, ed. (2011) Getting Started with the Internet of Things, Santa Rosa: Make: Community.

24. An Efficient and Portable Web Server, Vivek S. Pai, Peter Druschel, and Willy Zwaenepoel Rice University

25. Web Frameworks. https://www.fullstackpython.com/web-frameworks.html. Accessed (2022).

26. DuPlain, Ron. (2013). Flask Web Development.

27. S. Dong, C. Cheng and Y. Zhou, (2011), "Research on AJAX technology application in web development," International Conference on E-Business and E-Government (ICEE), 2011, pp. 1-3, doi: 10.1109/ICEBEG.2011.5881693.

28. Z. Haishui, W. Dahu, Z. Tong and H. Keming, (2010), "Design on a DC Motor Speed Control," International Conference on Intelligent Computation Technology and Automation, 2010, pp. 59-63, doi: 10.1109/ICICTA.2010.795.

29. B. A. Gregory, ed. (1981) An Introduction to ELECTRICAL INSTRUMENTATION AND MEASUREMENT SYSTEMS. 2nd ed. London: Macmillan Education.

30. Seder, Marija & Petrovic, Ivan & Macek, Kristijan. (2005). Motion Control of Mobile Robots in Indoor Dynamic Environments.

31. Robot Kinematics and Dynamics Herman Bruyninckx Katholieke Universiteit Leuven Department of Mechanical Engineering Leuven, Belgium c, (2010).

32. CS.Columbia W4733 NOTES - Differential Drive Robots

33. Gracia, Luis & Tornero, Josep. (2002). Kinematic Control System for Car-Like Vehicles. 882-892. 10.1007/3-540-36131-6_90.

34. Huston, Ron & Kelly, Fred. (2014). Another look at the static stability factor (SSF) in predicting vehicle rollover. International Journal of Crashworthiness. 19. 10.1080/13588265.2014.919730.

35. Rollover / Stability'. Safety Research & Strategies, Inc., https://www.safetyresearch.net/safety-issues/rollover-stability/. Accessed (2022).

36. Sanjay L; Kiran Kumar; K. Kruthika. '' Design and development of a robotic arm '' (2016).

37. Mohammed, Amin & Sunar, Mehmet. (2015). Kinematics Modeling of a 4-DOF Robotic Arm. 10.1109/ICCAR.2015.7166008.

38. Golnazarian, Wanek & Hall, Ernest. (2002). Intelligent Industrial Robots. 10.1201/9780203908587.ch6.5.

39. Evans, Philip. (2001). Rotations and rotation matrices. Acta crystallographica. Section D, Biological crystallography. 57. 1355-9. 10.1107/S0907444901012410.

40. Masood, Muhammd Umar & Haghshenas-Jaryani, Mahdi. (2021). A Study on the Feasibility of Robotic Harvesting for Chile Pepper. Robotics. 10. 1-22. 10.3390/robotics.

41. Chitta, Sachin & Sucan, Ioan & Cousins, Steve. (2012). Moveit![ROS topics]. IEEE Robotics & Automation Magazine - IEEE ROBOT AUTOMAT. 19. 18-19. 10.1109/MRA.2011.2181749.

42. Yasa, Yusuf & Sahin, Ergin & Acar, Cilem & Gözütok Bilir, Rahmet & Firat, Ecem & Mese, Erkan. (2013). Servo motor driver design for high performance applications. 1-6. 10.1109/EPECS.2013.6713062.

43. 'Adafruit 16-Channel PWM/Servo Shield'. Adafruit Learning System, https://learn.adafruit.com/adafruit-16-channel-pwm-slash-servo-shield/overview. Accessed (2022).

44. M. Gupta , S. P. Phulambrikar, (2014), Design and Analysis of Buck Converter, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 03, Issue 03 (March 2014).

5. Appendix

5.1. Budget

No.	Components	Quantity	price	total	Reference
1	Raw Materials	6280 g	2.30 for gram	14444	Lampatronics
2	Raspberry pi4	1	3950	3950	Raspberry Pi 4B - 4GB - MADE IN UK - RAM Electronics (ram-e-shop.com)
3	Motors with encoder	4	1260	5040	uxcell DC 24V 74RPM 25Kg.cm Self-Locking Worm Gear Motor with Encoder and Cable, High Torque Speed Reduction Motor: Buy Online at Best Price in UAE - Amazon.ae
4	Lidar Sensor	1	3276	3276	Slamtec RPLIDAR A1 2D 360 Degree 12 Meters Scanning

					Radius LIDAR Sensor Scanner for Bstacle Avoidance and Navigation of Robots: Buy Online at Best Price in UAE - Amazon.ae
5	**Camera**	1	800	800	رازبيري 8 v2.1 بي ميجابيكسل وحدة كاميرا1080بكسل: اشتري اون لاين بأفضل الاسعار في مصر -سوق.كوم الان اصبحت امازون مصر (amazon.eg)
6	Microcontroller(teensy)	1	375	375	Teensy++ 2.0 Development Board - RAM Electronics (ram-e-shop.com)
7	Motor Driver (BTS7960)	4	275	1100	High Power Motor Driver (BTS7960)

					43Ampere For Robot Smart Car – Lampatronics
8	lights	4	113	452	فلاشر خلفي لمبات ضوءSMD LED قابل لإعادةUSB الشحن ضوءLED إضاءة -للدراجة خلفية للدراجة فائقة السطوع تناسب أي دراجات جبلية، دراجة طريق -وخوذ مة سهلة التركيب لسلا الدراجات: :Amazon.com المستلزمات الرياضية
9	battery(24V)	1	650	650	LI Battery

10	Adapter	1	55	55	DC Power Adapter (7.5VDC – 2A) – Lampatronics
11	DC-DC Step Down Converter LM2596	1	45	45	DC-DC Step Down Converter LM2596 (3A) – Lampatronics
12	DC-DC Step Down 8A Converter XL4016	1	125	125	DC-DC Step Down 8A Converter XL4016 – Lampatronics
13	Locker	1	400	400	قفل الملف اللولبي فولت12الكهربائي اشتر :تيار مستمر اون لاين بأفضل الاسعار في مصر كوم الان.سوق اصبحت امازون مصر (amazon.eg)
14	3Axis Gyro+3Axis (Accelorometer)	1	90	90	GY-521 – MPU6050 IMU (3Axis Gyro+3Axis Accelorometer) – Lampatronics
15	LED STRIP LIGHT	1	190	190	LED STRIP LIGHT RGB 5050 – Lampatronics
Total price				**30992**	

6.2 Datasheet

6.2.1 DC-Motor Datasheet

Voltage		No load		Rated Load				Stall		(L)Length
Range	Rated	Speed	Current	Speed	Current	Torque	Power	Torque	Current	
		r/min	mA	r/min	mA	kg.cm	W	kg.cm	mA	mm
3-9V	6V	1363	80	954	380	0.1	1.05	0.35	1300	18
3-9V	6V	646	80	454	380	0.22	1.05	0.76	1300	17.5
3-9V	6V	281	80	196	380	0.5	1.05	1.7	1300	19
3-9V	6V	171	80	120	380	0.8	1.05	2.8	1300	21
3-9V	6V	133	80	93	380	1	1.05	3.6	1300	21
3-9V	6V	77	80	54	380	1.8	1.05	6.2	1300	23
3-9V	6V	58	80	40	380	2.4	1.05	8.2	1300	23
3-9V	6V	35	80	25	380	4	1.05	13.6	1300	25
3-9V	6V	26	80	18	380	5.2	1.05	18	1300	25
3-9V	6V	22	80	16	380	8.7	1.05	20	1300	27
3-9V	6V	12	80	8.5	380	11.5	1.05	20	1300	27
6-18V	12V	1931	100	1370	330	0.14	2	0.5	1500	18
6-18V	12V	915	100	640	330	0.3	2	1.06	1500	17.5
6-18V	12V	400	100	280	330	0.67	2	2.35	1500	19
6-18V	12V	250	100	175	330	1.1	2	3.86	1500	21
6-18V	12V	188	100	130	330	1.43	2	5	1500	21
6-18V	12V	108	100	76	330	2.5	2	8.6	1500	23
6-18V	12V	82	100	62	330	3.3	2	11.4	1500	23
6-18V	12V	50	100	35	330	5.4	2	18.8	1500	25
6-18V	12V	37	100	26	330	7.2	2	20	1500	25
6-18V	12V	22	100	16	330	12	2	20	1500	27
6-18V	12V	17	100	12	330	16	2	20	1500	27
12-30V	24V	863	10	604	60	0.12	0.7	0.41	210	18
12-30V	24V	400	10	280	60	0.27	0.7	0.9	210	17.5
12-30V	24V	178	10	125	60	0.6	0.7	2	210	19
12-30V	24V	110	10	80	60	1	0.7	3.26	210	21
12-30V	24V	85	10	60	60	1.26	0.7	4.2	210	21
12-30V	24V	49	10	35	60	2.2	0.7	7.3	210	23
12-30V	24V	36	10	25	60	2.8	0.7	9.6	210	23
12-30V	24V	22	10	15.5	60	4.8	0.7	16	210	25
12-30V	24V	14	10	12	60	6.3	0.7	20	210	25
12-30V	24V	10	10	7	60	10.6	0.7	20	210	27
12-30V	24V	7.5	10	5	60	14	0.7	20	210	27

6.2.1 BTS7960 High Current 43A H-Bridge Motor Driver Datasheet

-40 °C < T_j < 150 °C (unless otherwise specified)

Pos	Parameter	Symbol	Limits		Unit	Test Condition
			min	max		
Electrical Maximum Ratings						
3.0.1	Supply voltage	V_{VS}	-0.3	45	V	
3.0.2	Logic Input Voltage	V_{IN} V_{INH}	-0.3	5.3	V	
3.0.3	HS/LS continuous drain current	$I_{D(HS)}$ $I_{D(LS)}$	-40	40[1)]	A	T_C < 85°C switch active
3.0.4	HS pulsed drain current	$I_{D(HS)}$	-60	60[1)]	A	T_C < 85°C t_{pulse} = 10ms
3.0.5	LS pulsed drain current	$I_{D(LS)}$	-60	60[1)]	A	
3.0.6	Voltage at SR pin	V_{SR}	-0.3	1.0	V	
3.0.7	Voltage between VS and IS pin	V_{VS} - V_{IS}	-0.3	45	V	
3.0.8	Voltage at IS pin	V_{IS}	-20	45	V	
Thermal Maximum Ratings						
3.0.9	Junction temperature	T_j	-40	150	°C	
3.0.10	Storage temperature	T_{stg}	-55	150	°C	
ESD Susceptibility						
3.0.11	ESD susceptibility HBM	V_{ESD}			kV	according to EIA/ JESD 22-A 114B
	IN, INH, SR, IS		-2	2		
	OUT, GND, VS		-6	6		