ROS: AN AUTONOMOUS ROBOT OPERATING SYSTEM FOR SIMULTANEOUS LOCALIZATION AND MAPPING USING A 2D LIDAR SENSOR

Abstract*– Currently the need and use of autonomous robots is growing exponentially both in the industrial sector and in everyday life, due to this arises the implementation of systems that facilitate the robot to operate autonomously and, in turn, make a graphical model of the environment in which it is mobilized. SLAM systems allow the robot to be located in real time without the need for technologies such as GPS, and at the same time they create a graphic model that is very accurate to reality. There are different ways to implement a SLAM system, one of the most popular is by using LIDAR sensors that through the principle of laser triangulation generate the necessary feedback for the robot to detect surfaces, objects, obstacles and distance between them. The following research project was carried out with the aim of developing an intelligent mobile autonomous robot that allows accurate maps to reality and can be located in real time and accurately, using ROS for communication between actuators, microcontrollers and sensors. The design of the structure and its studies have been carried out with CAD software, the selection of components was made with the use of multicriteria and decision matrices, which helped to select those that best fit the project. It was proposed and demonstrated that the use of this sensor is an excellent option to implement SLAM and thus develop autonomous robots.

Keywords - Autonomous robot, LIDAR, mobile, ROS, SLAM.

I. INTRODUCTION

Autonomous robots are a very common case study since they are required more and more in the industrial sector as well as in everyday life, since they optimize processes, save time, reduce human effort and require minimal or no supervision. That is why the present research tries to implement a SLAM system based on a 2D LIDAR sensor for the development of an autonomous robot. With this system the robot will be able to avoid obstacles and obtain an accurate location within the environment in which it operates, in addition to this, it will be providing information necessary for an accurate mapping indoors, after the mapping done the robot can travel to the required point in a smooth and accurate way. The operation of the robot will be based on a framework that allows the use of robust sensors and a variety of programming languages, because of this it has been decided to implement ROS. A robot with this type of system can be designed for different uses, such as: cargo robot, exploration robot, optimizing robot, measuring robot, etcetera.

Digital Object Identifier (DOI):
http://dx.doi.org/10.18687/LACCEI2021.1.1.520
ISBN: 978-958-52071-8-9 ISSN: 2414-6390

II. PREVIOUS STUDIES

The study of SLAM emerged in 1986 with the first problem suggested by Durrant-Whyte and Bailey, which introduced the main formulas for SLAM including the approximations made to the Kalman filter. In this period, they relied more on understanding the mathematical model that would later lead to the implementation of this type of technology, it was until 2006 that there was an important probabilities approach and the data associated with SLAM emerged. In 2008 it was possible to learn more about the filters required for the use of this technology. By 2011, the middle of the period called "algorithm analysis" was being reached and more was known about the back-end, thus achieving a visual odometry and it was in 2016 when the period of "algorithm analysis" ended with the multiple robots that were developed with this technology and more theoretical aspects that laid the foundation for what we know today as SLAM [1], [2] as shown in figure 1.

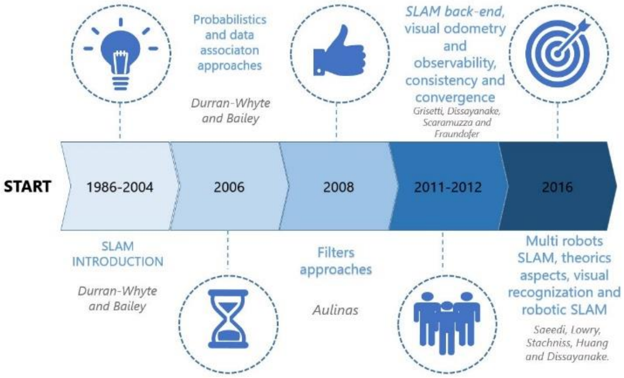


Figure 1. SLAM timeline

III CONTEXT

In recent decades, the use of robots has increased both in factories and in people's homes The proliferation of robotics is accelerating thanks to advances in electronics, mechanics and other similar disciplines. Different devices are being built that are capable of solving all kinds of tasks, from the automation of production lines to robots capable of assisting in medical work. The manufacturing industry according to data obtained in 2019 had reached a world record which was 113 units per 10,000 employees. The country with the highest density is Singapore with a total of around 918 robots, followed by South Korea with a total of 868 units per every 10,000 population including home and industrial robots. [3].

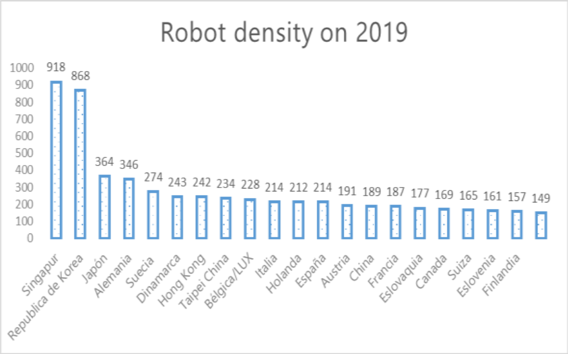


Figure 2. Robot density through the world on 2019 [3]

It is important to recognize the trend of robots at this historical moment. Today the range of possibilities for building robots is very wide. The components and devices needed to build them are becoming more and more affordable. Investments in modern robotic technology will also be driven by the requirement for a lower carbon footprint. SLAM based on LIDAR sensor is very popular, because the LIDAR sensor has the advantages of high accuracy, possesses strong resistance to interference and is robust to illumination variation. The 2D LIDAR sensor enjoys the advantages of high accuracy range measurement and low cost compared to expensive 3D LIDAR scanners, making it a more suitable choice for application scenarios [4]. ROS is a flexible system that provides several tools and libraries for writing robotic software. It offers several powerful functions to assist developers in tasks such as message passing, computer code distribution, reuse and implementation of state-of-the-art algorithms for robotic applications [5]. The versatility offered by ROS is something that is not achieved in other systems.

IV. METHODOLOGY

This research has a quantitative focus, dynamic analysis of movement will be performed in the robot body, static analysis as well to corroborate that the prototype to be made supports all the implemented components, different tests will be performed to the LIDAR sensor including the variation in the frequency of the motor and the rotary encoders to ensure that they generate the feedback correctly.

A. Variables

The variables are: Static analysis, localization, mapping and power supply.
The static analysis is based on weight: the equation for it is as follows [6]:

$$W = m * g \tag{1}$$

Where:
W= object weight
m= mass of the object
g= force of gravity.

The localization will be obtained from the incremental encoders, being important to know the individual velocity and distance which is given by the following equations [7]:

$$w = R\frac{v_r - v_l}{L} \tag{2}$$

Where:

W = angular velocity.

R = radius of the gears.

L = length between wheels.

Vr = Speed of the right wheel.

VL = velocity of the left wheel

$$\dot{x} = v\cos\emptyset \tag{3}$$

Where:

$\dot{x}$ = Position on the x-axis of the robot.

v = Velocity.

∅ = Angle.

$$\dot{y} = v sen\emptyset \tag{4}$$

Where:

$\dot{y}$ = Position in the y-axis of the robot.

v = Velocity.

∅ = Angle.

The mapping depends on the amount of information that the LIDAR will provide, the number of points depends on the scanning frequency which is given by the following equation [8]:

$$D = \left(\frac{R * 2\pi}{ppv}\right)N \tag{5}$$

Where:
- D = Distance traveled.
- R = Radius.
- ppv = Pulses per revolution.
- N = Number of pulses traveled.

Power is the battery autonomy which is given by the following equation [9]:

$$H = \frac{Vb(Ib)}{Vb(Ic)} \tag{6}$$

Where:
H=autonomy in hours
Vb= battery voltage

Ib= battery milliampere hours
Ic= consumption of components.

B. Tools and instruments-based

The following software will be used for the development of the prototype:

- Solidworks, the Solidworks 3D modeling software will allow to perform dynamic motion tests on the robot structure.
- Fritzing will be used to make the connection diagrams of the different components of the prototype.
- Mapper by SLAMTEC is the software that uses the LIDAR sensor.
- Hector SLAM for Linux is a simulation software for mappings.
- Rviz is the robot's real-time visualization software and map [5].
- Arduino IDE will be used to program logic to the microcontrollers and to be able to test the encoders.
- Cloud Shell editor by Google is the software used to program in Python language.
- Matlab will be used to simulate Kalman filters.
- VNC Viewer is a program that allows the remotely access and manipulation of the minicomputer with ROS.

C. Study methodology

For the development of the prototype we chose to use the "V" methodology created by Vasic & Lazarevic. The "V" methodology is focused on the production of mechatronic devices which involve electronic, electrical, mechanical and microcontroller components.

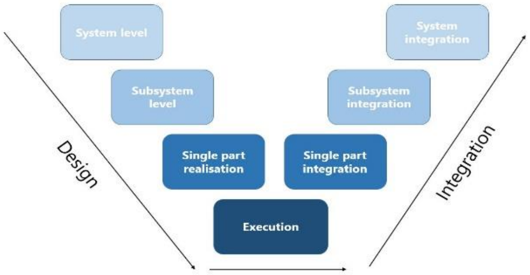


Figure 3. “V” methodology

The "V" methodology is composed of two cycles: cycle A and cycle B. In cycle A, the design and analysis of the design is performed, once the prototype is manufactured, cycle B is performed, in which tests and corrections are made to the prototype in order to obtain a final prototype with the least amount of errors [10].

IV ANALYSIS AND RESULTS

4.1 Structural system

The structural system is where all the components that make up the robot are coupled, for this system the design was made in Solidworks as well as the static and dynamic analysis.

4.1.1 Static analysis

The static analysis was based on the calculations performed using equation 1, the structure is subdivided into two parts, upper and lower part, in the upper part is coupled the 2D LIDAR sensor and the battery. The result is a weight of 4.56 N. For the lower part the arduinos, the jetson nano, the H-bridge and the voltage regulator are coupled.
A weight of 5 N was used in the simulations to calculate the displacements in [mm].

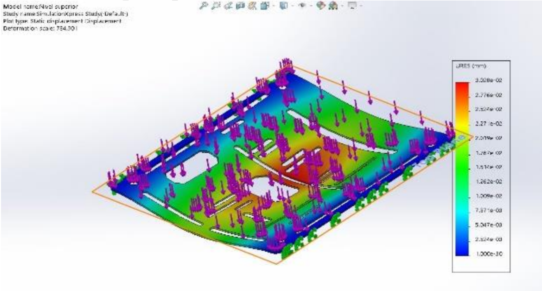


Figure 4. Displacements on top side

As can be seen in Figure #4, the displacement is very small and no rupture point is found in the structure. Its maximum value occurs in the center giving a result of $3.02{\times}10^{-2}$ mm and its minimum displacement value occurs at the edges with a value of $2.543{\times}10^{-3}$ mm.

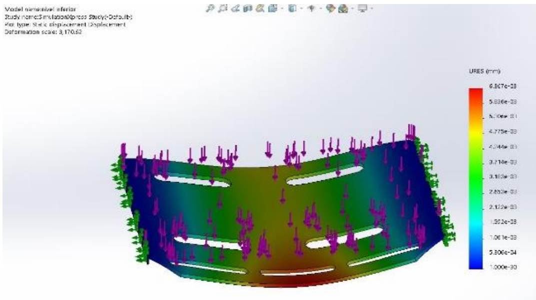


Figure 5. Displacements on bottom side

As can be seen in Figure 5, the displacement at the bottom is very small and no rupture point is found in this part of the structure. Its maximum value is $6.357{\times}10^{-3}$ mm, such displacement occurs in a small part of the front half of the piece and its minimum value of displacement occurring at the edge is $5.306{\times}10^{-4}$ mm. Knowing the displacements is important as well as the stresses to which the parts of the structure are subjected, and that is why a Von Misses test was performed on each part to which the components will be attached. 3003 aluminum alloy was used for the simulation.

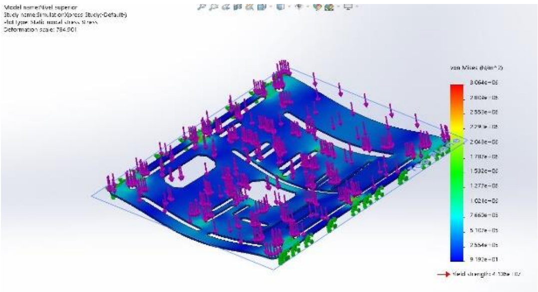


Figure 6. Von misses on top side

Figure 6 shows the results of the von misses tests on the part that forms the upper part of the robot. Its maximum value is 3.064×10-6 N/m^2 and the minimum value is 2.554×10^{-6} N/m^2.

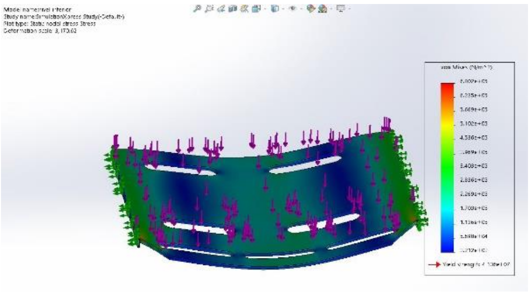


Figure 7. Von misses on bottom side

Figure 7 shows the results of the von misses tests on the part that forms the lower part of the robot. Its maximum value is 6.802×10-5 N/m2 and the minimum value is 5.698×10^{-4} N/m^2

4.1.2 Dynamic analysis

For the dynamic analysis, the Solidworks tool for motion studies was used. With the data obtained in the simulations, the calculations were made in equation 2 to know the individual velocity and equation 5 to calculate the distance traveled. Radius 1.5 cm, length between wheels or gears 20 cm and averaged linear velocities of 1.43 mm/sec. Using equation 2 we have an individual angular velocity of 0.0107 deg/sec or 38.61 deg/hour. For the distance traveled, equation #5 is used with the following data: radius 1.5 cm, 30 pulses per turn and 20 number of pulses traveled, with these data a distance of 6.28 cm is obtained, all at a theoretical level.

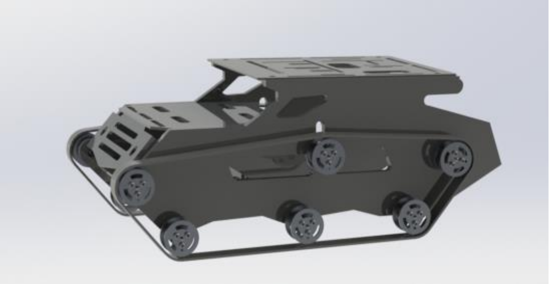


Figure 8. Final structural system of the prototype

Figure 8 shows a rendering of the final design of the structural system of the prototype on which all the analyses and calculations described above were based.

4.2 2D LIDAR Sensor and limitations

The LIDAR sensor is one of the most important components of the prototype, its performance has been more than correct and, despite its limitations, it has allowed the realization of graphic models very accurate to reality. Different tests were performed to the 2D LIDAR sensor which was implemented in the research, according to the parameters on which this sensor is based, the following results were obtained in the different tests performed:

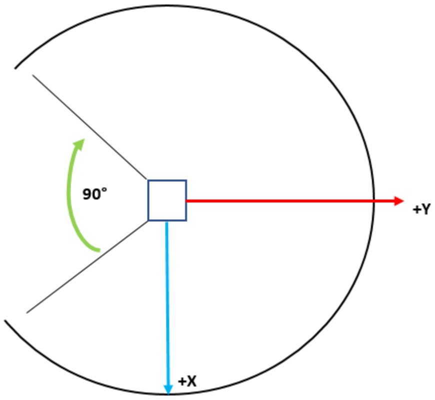


Figure 9. 2D LIDAR sensor range

The 2D LIDAR sensor used in the research makes use of a single plane of lasers to capture the X and Y dimensions. This limits it to be mostly used for sensing and ranging tasks, but this does not mean that it cannot be used in more complex tasks.

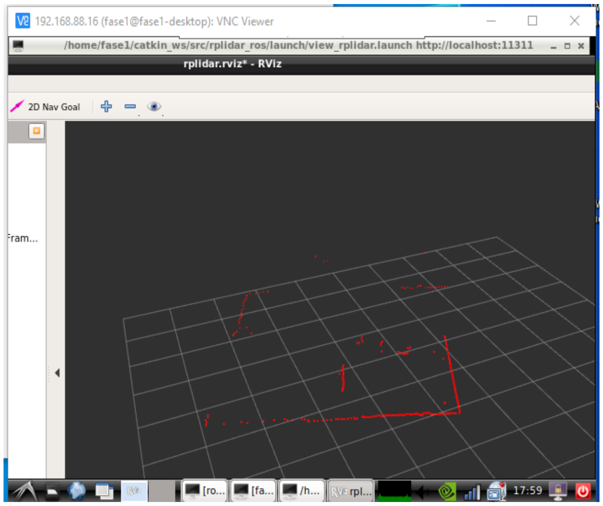


Figure 10. 2D LIDAR sensor using ROS to obtain a real-time vectorization of the site.

As can be seen in the figure 10 the 2D LIDAR sensor is limited to vectorize only the area closes to it, so it is necessary to move it to the new areas to be vectorized. This ensures that it will not have many problems when moving the robot.

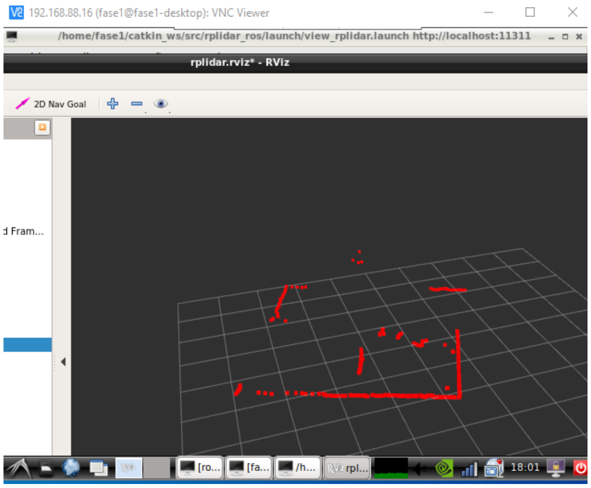
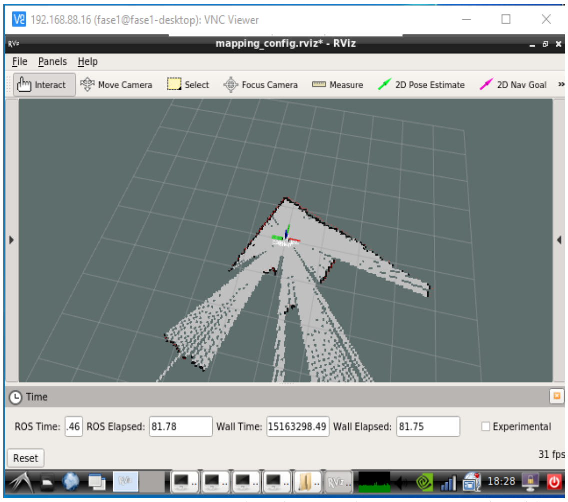


Figure 11. 2D LIDAR sensor Using ROS to obtain real-time location vectorization with increased scan density.

But the 2D LIDAD sensor, by varying the density in its scanning mode, the environment becomes clearer because the sensor's laser picks up fewer points, but with a higher density as we can see in the figure 11.



ROS Time: .46 ROS Elapsed: 81.78 Wall Time: 15163298.49 Wall Elapsed: 81.75

Figure 12. 2D LIDAR sensor using the mapping program and making use of ROS for data interpretation.

As can be seen thanks to the use of the mapping program and running the ROS library at the same time we can create a more detailed environment, then, an image will be shown where you can see the position of the sensor and how it has interpreted the environment.

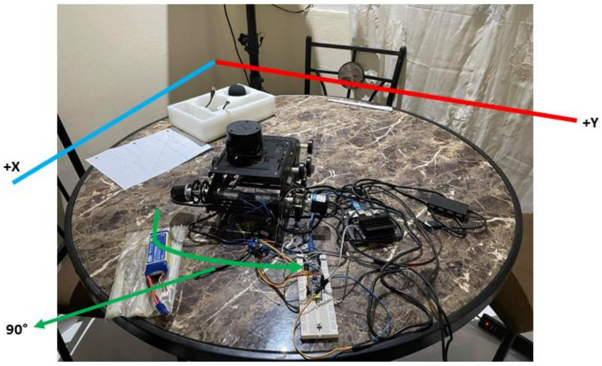
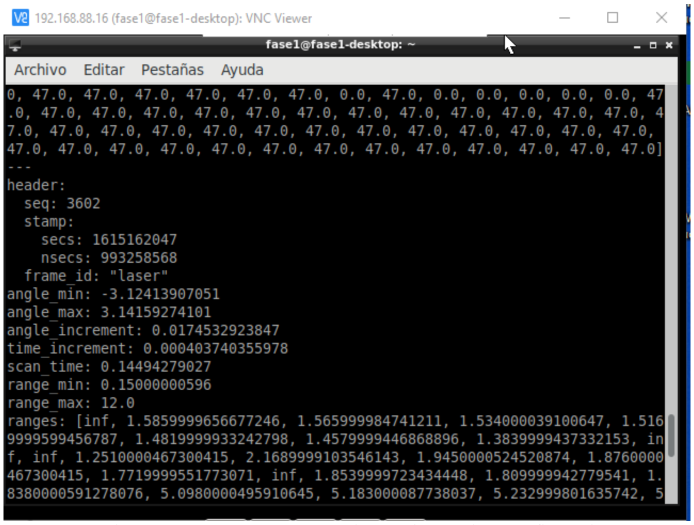


Figure 13. Environment in which mapping is being performed

The coordinates that the sensor is capturing at that moment are described, resulting in the map of the environment shown in figure 13, which can be seen above, as described at the beginning, the sensor does not have access to information 90° behind its main module, even though the laser is rotating 360°.



```
0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 0.0, 47.0, 0.0, 0.0, 0.0, 0.0, 0.0, 47
.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 4
7.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0,
47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0]
---
header:
  seq: 3602
  stamp:
    secs: 1615162047
    nsecs: 993258568
  frame_id: "laser"
angle_min: -3.12413907051
angle_max: 3.14159274101
angle_increment: 0.0174532923847
time_increment: 0.000403740355978
scan_time: 0.14494279027
range_min: 0.15000000596
range_max: 12.0
ranges: [inf, 1.5859999656677246, 1.565999984741211, 1.534000039100647, 1.516
9999599456787, 1.4819999933242798, 1.4579999446868896, 1.383999437332153, in
f, inf, 1.2510000467300415, 2.1689999103546143, 1.9450000524520874, 1.8760000
467300415, 1.7719999551773071, inf, 1.8539999723434448, 1.809999942779541, 1.
8380000591278076, 5.0980000495910645, 5.183000087738037, 5.232999801635742, 5
```

Figure 14. 2D LIDAR sensor internal scanning configuration

Figure 14 shows the minimum and maximum angles that the sensor can support as well as the increments that it can have in the established time, and also shows the other values that belong to the data read at that moment.

4.3 Control System

As for the control system, the jetson nano was implemented supported by nanos Arduinos, its operation will depend on the feedback and information provided by the sensors and actuators of the robot. All this is connected so that the jetson nano is the master of the prototype.

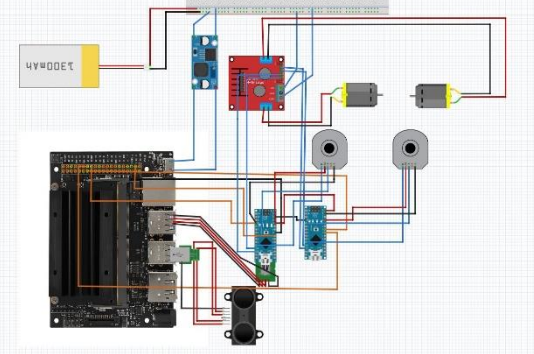


Figure 15. Connection diagram.

Figure 15 shows the connection diagram of the prototype, as can be seen the prototype has a jetson nano 2gb, two nano arduinos, two rotary encoders, two 12v dc motors, an H-bridge, a voltage down regulator and the 2D LIDAR sensor. As for the connections, the nano arduinos are USB powered, but send the information provided by the encoders through the GIO ports of the jetson nano using the ports with the I2C communication protocol. As for the LIDAR sensor, it is powered and communicates via USB.

4.4 Power supply system

Theoretical calculations of the autonomy of the power supply present in the robot were carried out. The power supply is a 12v lipo type 2,200mah battery.

4.4.1 Charging time

The charging time of the 2,200mah battery with the charger provided that charges the battery cells at 1 amp, gives a result that the battery is charged in 2 hours and 12 minutes.

4.4.2 Autonomy

Regarding the autonomy of the robot, it is obtained through equation #6, the total consumption of the robot is 1,233 mah considering the consumption of each component and summing it up. Having this data, the battery autonomy gives us a result of 1 hour and 47 minutes. This may vary depending on the use of the robot.

4.5 Framework implemented and filtering

For the development of the prototype the framework is an essential part since it is in charge of interpreting all the signals and executing the nodes in charge of the logic of the prototype as well as the type of filtering of the signals to obtain the purest signal.

4.5.1 ROS

For the development of the prototype the framework is ROS, due to the operating system version that the jetson nano has, Melodic ROS was implemented based on Ubuntu 18.02. For the operation of the prototype the nodes, topics, messages sent and interpreted in the ROS workspace and compatibilities with the different actuators and sensors that make up the system must be defined.

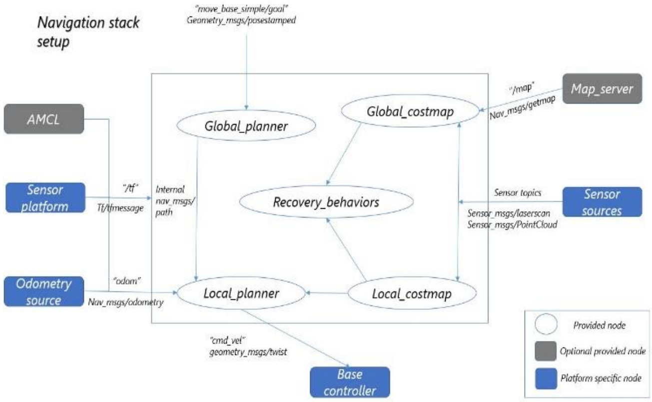
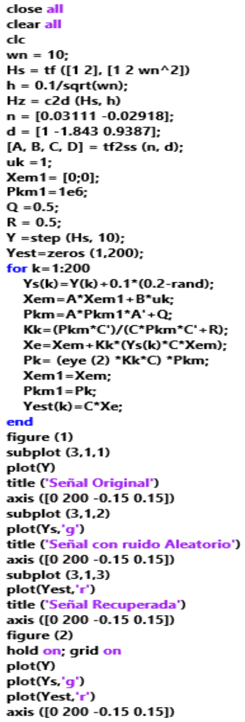


Figure 16. Navigation stack setup

4.5.2 Kalman filter

Regarding signal filtering, ROS implements the Kalman mathematical algorithm to generate the purest and most desirable signal from the LIDAR sensor. In order to understand this algorithm, a Matlab simulation has been performed to observe the filtering in a signal with random noise.



```
close all
clear all
clc
wn = 10;
Hs = tf ([1 2], [1 2 wn^2])
h = 0.1/sqrt(wn);
Hz = c2d (Hs, h)
n = [0.03111 -0.02918];
d = [1 -1.843 0.9387];
[A, B, C, D] = tf2ss (n, d);
uk =1;
Xem1= [0;0];
Pkm1=1e6;
Q =0.5;
R = 0.5;
Y =step (Hs, 10);
Yest=zeros (1,200);
for k=1:200
   Ys(k)=Y(k)+0.1*(0.2-rand);
   Xem=A*Xem1+B*uk;
   Pkm=A*Pkm1*A'+Q;
   Kk=(Pkm*C')/(C*Pkm*C'+R);
   Xe=Xem+Kk*(Ys(k)*C*Xem);
   Pk= (eye (2) *Kk*C) *Pkm;
   Xem1=Xem;
   Pkm1=Pk;
   Yest(k)=C*Xe;
end
figure (1)
subplot (3,1,1)
plot(Y)
title ('Señal Original')
axis ([0 200 -0.15 0.15])
subplot (3,1,2)
plot(Ys,'g')
title ('Señal con ruido Aleatorio')
axis ([0 200 -0.15 0.15])
subplot (3,1,3)
plot(Yest,'r')
title ('Señal Recuperada')
axis ([0 200 -0.15 0.15])
figure (2)
hold on; grid on
plot(Y)
plot(Ys,'g')
plot(Yest,'r')
axis ([0 200 -0.15 0.15])
```

Figure 17. Code of a Kalman filter simulation

As can be seen in the illustration 18 the signal recovered by means of a Kalman filter is practically identical to the original one, without such filtering the noise would have generated distortion in the original signal to the point of leaving it incomprehensible.

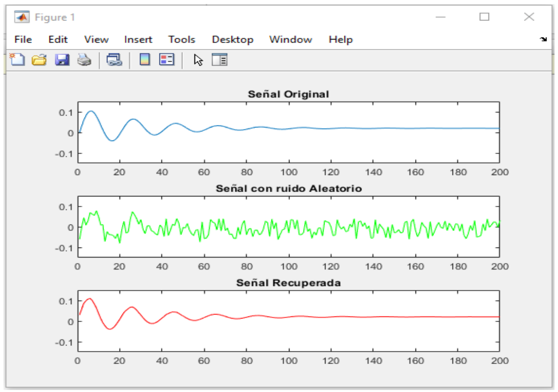


Figure 18. Kalman filter results

4.6 Final prototype and map created by the robot

The final prototype has a maximum height of 18.8 cm, width of 25.2 cm including the encoders assembled on the side of the band and a length of 32 cm.

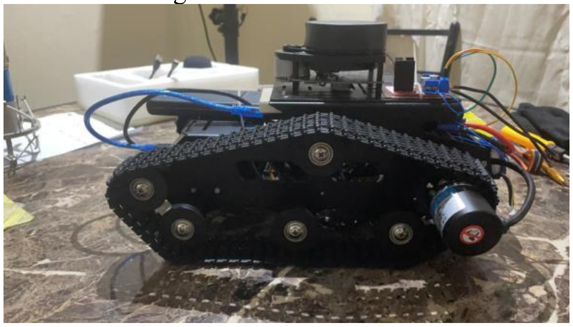


Figure 19. Final prototype.

The maps made are according to the real environment in which the robot operates, using the standard frequency dictated by the manufacturer of the LIDAR 2D sensor which is 6 hz.

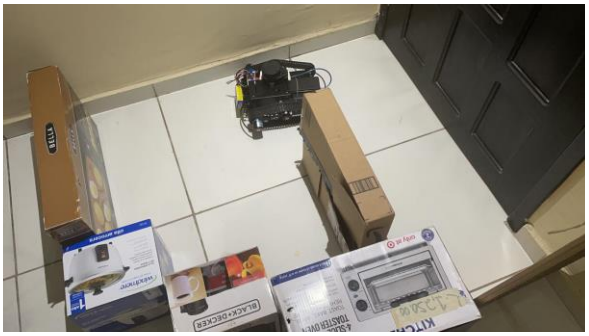
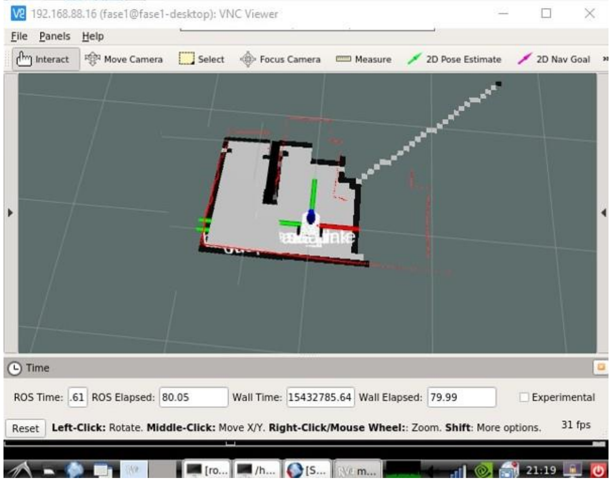


Figure 20. Final prototype in a controlled environment.



ROS Time: .61 ROS Elapsed: 80.05 Wall Time: 15432785.64 Wall Elapsed: 79.99

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options.

Figure 21. Map created by the prototype

As can be seen in Figure #21 the map created by the robot is very accurate to reality, correctly interpreting the environment in which it operates and generating a graphical model that can be useful for many cases and allowing the autonomous navigation.

V CONCLUSIONS

- A mobile robot capable of simultaneously locating and mapping unknown environments using a 2D LIDAR sensor was developed based on the framework for the development of ROS robots.
- The limitations and ways of operation of the 2D LIDAR sensor when implemented in a SLAM system were identified, as well as the advantages of this sensor.
- The characteristics for the development of a prototype considered as an intelligent autonomous robot that, in turn, can be tele operated to improve its versatility, were enumerated and implemented.
- Rotary encoders were selected to support the robot in real time localization because these devices quantify the amount of pulses through the movement of the axis and then transform it into a measurable signal that feeds back to the system with the exact position of the robot.
- ROS was implemented as a framework due to its flexibility, communication capacity and heterogeneous data collection provided by each component that makes up the SLAM autonomous robot.

VI REFERENCES

[1] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., … Leonard, J. J. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *Transactions on Robotics*, 1309-1332

[2] Ashish, R., Sujay, B. M., Caushik, R., Jayashree, S., & Nischal, H. P. (2018). 2D Based Indoor Slam and Autonomous Navigation Using a Terrain Robot. *2018 3rd IEEE International Conference on Recent*

Trends in Electronics, Information & Communication Technology (RTEICT)

[3] IFR (International Federation of Robotics). (27 de Enero de 2021). *Internacional Federation of Robotics. Obtenido de* https://ifr.org/ifr-press-releases/news/robot-race-the-worlds-top-10-automated-countries

[4] Ren R., Fu H., & Wu M.,. (2020). Large-Scale Outdoor SLAM Based on 2D Lidar. *TELKOMNIKA Telecommunications, Computing, Electronics and Control Vol. 18*.

[5] Joseph, L., & Cacace, J. (2018). Mastering ROS for Robotics Programming - Second Edition: Design, build, and simulate complex robots using the Robot Operating System . *Packt Publishing*.

[6] Mott, R. (2006). *Mecánica de fluidos*. PRENTICE-HALL INC.

[7] L. E. Solaque, M. A. Molina, E. L. Rodríguez. (2014). *Seguimiento de trayectorias con un robot móvil de configuración diferencial*. *Ing. USBMed, Vol. 5*, 26-34.

[8] VALENCIA V., JHONNY A., & MONTOYA O., ALEJANDRO, & RIOS, LUIS HERNANDO. (2009). *MODELO CINEMÁTICO DE UN ROBOT MÓVIL TIPO DIFERENCIAL Y NAVEGACIÓN A PARTIR DE LA ESTIMACIÓN ODOMÉTRICA*. *Scientia Et Technica, XV*, 191-196.

[9] Coelectrix. (2019). *Calcular la Autonomía de una Batería*. Coelectrix

[10] Vasilije S. Vasić, M. P. (2008). Standard Industrial Guideline for Mechatronic Product Design. *FME transactions*, 6.