UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS

VISIÓN ELECTRÓNICA

https://doi.org/10.14483/issn.2248-4728

VISIÓN ELECTRÓNICA

A differential drive mobile robot controlled by using the robotics operational system (ROS)

Un robot móvil de accionamiento diferencial controlado mediante el sistema operativo robótico (ROS)

Henry Borrero-Guerrero[1], Juan José López-Aguilar[2], Alejandra Orduz-Otálora[3], Sara Valentina Barrero[4], Andrés Leonardo Jutinico-Alarcón[5]

Abstract

Considering that the working of mobile robots is facilitated by using distributed embedded systems capable of working cooperatively and that managing these systems requires distributed software applications generally written in C++ or Python, it can be argued that ROS (Robotics Operational Systems) can be considered as a useful tool in undertaking robotic projects as it facilitates the integration of software drivers for installed devices. With ROS, software devices can be kept separate but communicate data to achieve distributed control. This paper describes partial results of a currently ongoing research project, which employs a fuzzy controller to maintain a differential drive mobile robot in a path between crops

[1] Electronic Engineer, Universidad de los Llanos, Colombia. PhD in Mechanical Engineering, University of Sao Paulo, Brazil. Professor: Universidad Distrital Francisco José de Caldas, Colombia. E-mail: hbguerrero@udistrital.edu.co ORCID: https://orcid.org/0000-0003-4243-4205

[2] Electronic Engineer, Universidad Distrital Francisco José de Caldas, Colombia. Student, E-mail: jujlopeza@udistrital.edu.co

[3] Electronic Engineer, Universidad Distrital Francisco José de Caldas, Colombia. Student, E-mail: Electronic Engineer, Universidad Distrital Francisco José de Caldas, Colombia. Student, E-mail: aorduzg@udistrital.edu.co

[4] Electronic Engineer, Universidad Distrital Francisco José de Caldas, Colombia. Student, E-mail: Electronic Engineer, Universidad Distrital Francisco José de Caldas, Colombia. Student, E-mail: svbarrerom@udistrital.edu.co

[5] Electronic Engineer, Universidad Distrital Francisco José de Caldas, Colombia. PhD in Mechanical Engineering, University of Sao Paulo, Brazil. Professor: Universidad Distrital Francisco José de Caldas, Colombia. E-mail: hbguerrero@udistrital.edu.co ORCID: https://orcid.org/0000-0001-9146-9637

rows, which are detected using LiDAR (Light Detection and Ranging) and IMU (Inertial Measurement Unit) sensors. Python programs are used to read sensors and control robot DC motors using a fuzzy controller working in a team on the ROS platform. This paper provides a general overview of our robot, including its structure, hardware, software, reached results and future works.

Keywords: Crop rows following, differential drive mobile robot prototype, fuzzy control, IMU, LIDAR, ROS.

Resumen

Teniendo en cuenta que el funcionamiento de los robots móviles se ve facilitado por el uso de sistemas embebidos distribuidos capaces de trabajar de forma cooperativa y que la gestión de estos sistemas requiere aplicaciones de software distribuidas escritas generalmente en C++ o Python, se puede argumentar que ROS (Robotics Operational Systems) puede considerarse una herramienta útil a la hora de emprender proyectos robóticos, ya que facilita la integración de controladores de software para los dispositivos instalados. Con ROS, los dispositivos de software pueden mantenerse separados pero comunicar datos para lograr un control distribuido. Este artículo describe resultados parciales de un proyecto de investigación actualmente en curso, que emplea un controlador difuso para mantener un robot móvil de accionamiento diferencial en una trayectoria entre hileras de cultivos, que se detectan mediante sensores LiDAR (Light Detection and Ranging) e IMU (Inertial Measurement Unit). Se utilizan programas Python para leer los sensores y controlar los motores DC del robot utilizando un controlador difuso trabajando en equipo en la plataforma ROS. Este artículo proporciona una visión general de nuestro robot, incluyendo su estructura, hardware, software, resultados alcanzados y trabajos futuros.

Palabras clave: Seguimiento de hileras de cultivos, prototipo de robot móvil de accionamiento diferencial, control difuso, IMU, LIDAR, ROS.

1. Introduction

This paper shares results from ongoing research that aims to agricultural mobile robotics autonomous navigation. in which we are into an initial stage in which we configured a mobile robot prototype to experiment on reactive crop rows following, since we are interested in developing strategies to control the navigation of mobile robots between crop paths.

A literature review allows to identify several approaches in which mobile robots reactive navigation can be achieved by using closed loop control systems, for instance, Report documented in [1] describes the development and implementation of a closed-loop fuzzy control system to keep the vehicle orientation; In [2] a differential drive mobile robot which supports its mobility on a fuzzy closed-loop control system configuring then a wall follower is documented; In [2, 3] it is presented a robust control system to keep a car-like mobile robot in the middle of crop paths; Document in [4] shows a low-cost approach mobile robot project in which a closed-loop fuzzy control system is used to configure a wall follower.

In a general review of control systems used to path tracking on mobile robots, shows that this is an area of interest that is still under study. Results shared in [5] presents the implementation of a closed loop control system in order to improve the trajectory tracking performance of a mobile robot using a Linear Quadratic Regulator; Ben Halima Abid et al. in [6], share results about an effective design of a PI (Proportional Integral) controller to solve two tasks 'going to target' and 'following a trajectory' developed for the mobile robot Khepera II; In [7], authors propose the implementation of robust control strategies to treat the problem of mobile robot autonomous navigation, reporting contributions using the Khepera II mobile robot.

ROS is an open-source framework that allows communication data between programs or processes, for instance, if program A wants to send data to program B, and B wants to send data to program A, we can easily implement it using ROS. Although this data communication could be implemented using socket programming directly, it could get complex when several applications are required. ROS arises as an alternative for interprocess communication. Besides the message interface between processes, one advantage of ROS is off-shelf-algorithms, ecosystem/community support, extensive tools and simulators, among others [8].

ROS has been extensively used in robotics projects since its capabilities to distribute control taking advantage of messaging communication, for mobile robotics case, it is possible to find diverse undertakings, for instance In [9] a control system for an agricultural mobile platform that incorporates a large number of sensors is presented, actuators and auxiliary modules for navigation, power management, and others using ROS capabilities; In [10] it is discussed the use of ROS to implement closed-loop control for multi-terrain mobile robots.

In terms of reactive navigation of mobile robots, we are committed to arguing that it is possible to invest time in configuring low-cost mobile robots with embedded digital controllers based on formal control systems engineering, as well as modular programming strategies, for which we believe ROS (Robotics Operational System) will provide valuable alternatives. After describing the prototype and its working in section II, we share our considerations and future work in section III. Finally, we present our acknowledgments and references.

2. General description of mobile robot

A differential drive mobile robot prototype was assembled, and this is shown in Figure 1. It measures approximately 0.19 meters in length, 0.2 meters in width, and 0.29 meters in height. In terms of weight, it is approximately 1.5 kilograms.

Our mobile robot has three floors structured to carry the vehicle elements. The first floor is destined to allocate two DC motors; over the second floor we set up two 12Vdc suitable batteries, one of them is dedicated to supply a mini-PC, Arduino Board, an IMU (Inertial Measurement Unit) and a LIDAR sensor whereas a second battery is dedicated to supply 2 H-bridges and thus correspondingly the two vehicle DC motors. Two H-bridges, an Arduino Mega

2560, and an IMU BNO055 are also allocated on the second floor; Over the third floor we installed a mini personal computer (mini PC) commercialized by the Hyundai brand which is equipped with an Intel Atom processor, 4GB of RAM, 32GB of solid-state memory, 4 USB ports, wireless network connectivity, Linux Ubuntu 20.04 operating system and ROS Noetic. On the last floor, there is a YDLidar that is used to scan the distances around the navigation environment.

In figure 1, two views of the robot are shown, showing some of its devices, which indicate that the LIDAR DC motor and batteries are located at the rear of the structure.

LiDAR sensing is basically a remote sensing technology which emits laser light beam with defined intensity and focus, measures the reflected beam arrival time detected by the photodiodes (PD) within the sensor [11]. LIDAR sensors are capable of measuring distances to obstacles around them. As part of the YDLidar user manual provided by its manufacturer, instructions are provided on how to set up this sensor to work with Linux Ubuntu, as well as advice on how to install the drivers to work with ROS. Although this manual specifies that it is only compatible with Ubuntu 18.04, we tested YDLidar on Linux Ubuntu 20.04 obtaining acceptable results.

Figure 1. General view of Robot structure

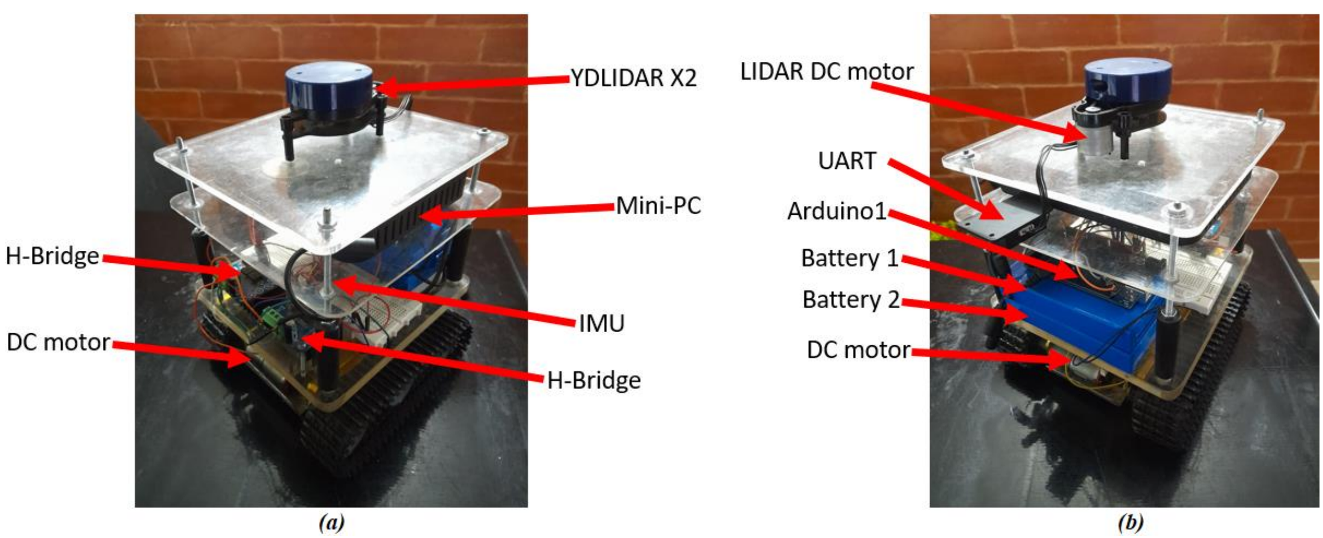
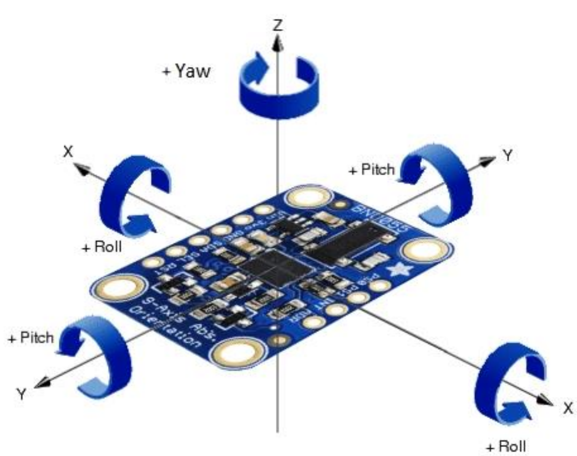


Figure 2. YDLidar X2

Report documented in [12] describe the BNO055 IMU as an intelligent 9-axis absolute orientation sensor that integrates a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer. BNO055 is shown in Figure 3. According to the illustration, it is possible to measure the pitch, yaw, and raw angles of the platform. As part of our approach, we simply use a yaw angle that is related to the vehicle orientation, which is given in radians.

Figure 3. BNO055 IMU [12]



Because we needed to control a mobile robot via ROS from an x64 mini-PC with only four USB ports, USB connectivity is not an issue with the YDLidar X2 since it is equipped with an attached UART (see Figure 1(b)) that allows this type of connection between the LIDAR and the mini-PC. It is necessary for the IMU sensor to have an I2C connection, which is not available in the mini-PC. Therefore, we decided to use an Arduino Mega 2560 board to connect it to the mini-PC. The Arduino Mega 2560 board reads the IMU data from the sensor and sends it via USB to the mini-PC as a bridge between the sensor and the mini-PC. Previously, the Arduino had been configured to work in accordance with ROS requirements.

The block diagram shown in Figure 4 provides an overview of the general operation of the proposed general system. By following the block diagram, we can see that LIDAR measurements are sent directly to the mini-PC while BNO055 utilizes an Arduino board as a bridge to send IMU data to the mini-PC as well. A mini-PC processes information derived from LIDAR and BNO055 to calculate control actions which are sent to an Arduino board, which converts these signals into PWM (Pulse Wide Modulation signals) signals for application to two H-bridges that transfer energy to DC motors using PWM signals.

Please refer to Figure 1(b) again to understand that the rear region of our robot coincides with the LIDAR DC motor, and using Figure 5 also coincides with +0/0 radians on the sensor, and the front region coincides with + π/-π radians. According to the depicted left side corresponds to an interval from +0 to *+π* radians whereas the right side corresponds to an interval from *-π to* -0 radians, both intervals being clockwise. Since LIDAR X2 reports distances for obstacles around it, we are using the reported distances and angles in an array that can be accessed.

We chose the LIDAR to be positioned over the robot structure due to our interest in navigating autonomously between crop rows, as described in the previous paragraph, and it was necessary to develop a methodology for determining lateral distances to crop rows, justifying the use of angular intervals. As shown in Figure 6(a), some LIDAR measurements actually reflect from crop rows, while others pass through current separations, which indicates that it is not appropriate to use just one beam angle when measuring, which is why a region of measurement between low and high thresholds was defined.

In the first experiment, the distance to a wall is measured as depicted by Figure 6(b). Some measurements are included jointly with the corresponding angle of radians so we can determine for each measurement its corresponding projection that is then accumulated to arrive at the average, which is the lateral distance.

In terms of LIDAR usage, we used similar modes as described in [2, 13, 14], this was done in order to follow walls describing the importance of vehicle orientation. Figure 7 illustrates how YDLidar X2 and BNO were used. In three different circumstances, a red arrow pointing forward was placed to indicate that the vehicle was moving forward. In situation (a) the vehicle orientation is $\psi = 0$, since in our approach we desired ($\psi_d = 0$) then in this case the orientation error ($e_\psi = \psi_d - \psi$) is 0; In situation (b) $\psi < 0$ and according to the IMU working, then $e_\psi > 0$; In situation (c) $\psi < 0$, then $e_\psi < 0$.Similarly, Figure 7 illustrates how LIDAR was used to determine the distance to the crop row. In the depicted situations, we are attempting to measure the left side lateral distance by delimiting a region between low and high thresholds to obtain LIDAR measurements as described for Figure 6, again following Figure 7, in all situations included, a partial lateral distance ($\boldsymbol{lx}$) is depicted at π radians. In situations (a, b and c) measurement region is defined as shown and therefore lateral distance ($\boldsymbol{l}$) corresponds to its projection using the absolute value of e_ψ($l = l_x \cos(|e_\psi|)$). Note that in situation (a) $\boldsymbol{lx}$ is the same lateral distance.

Figure 4. Block diagram about used devices.

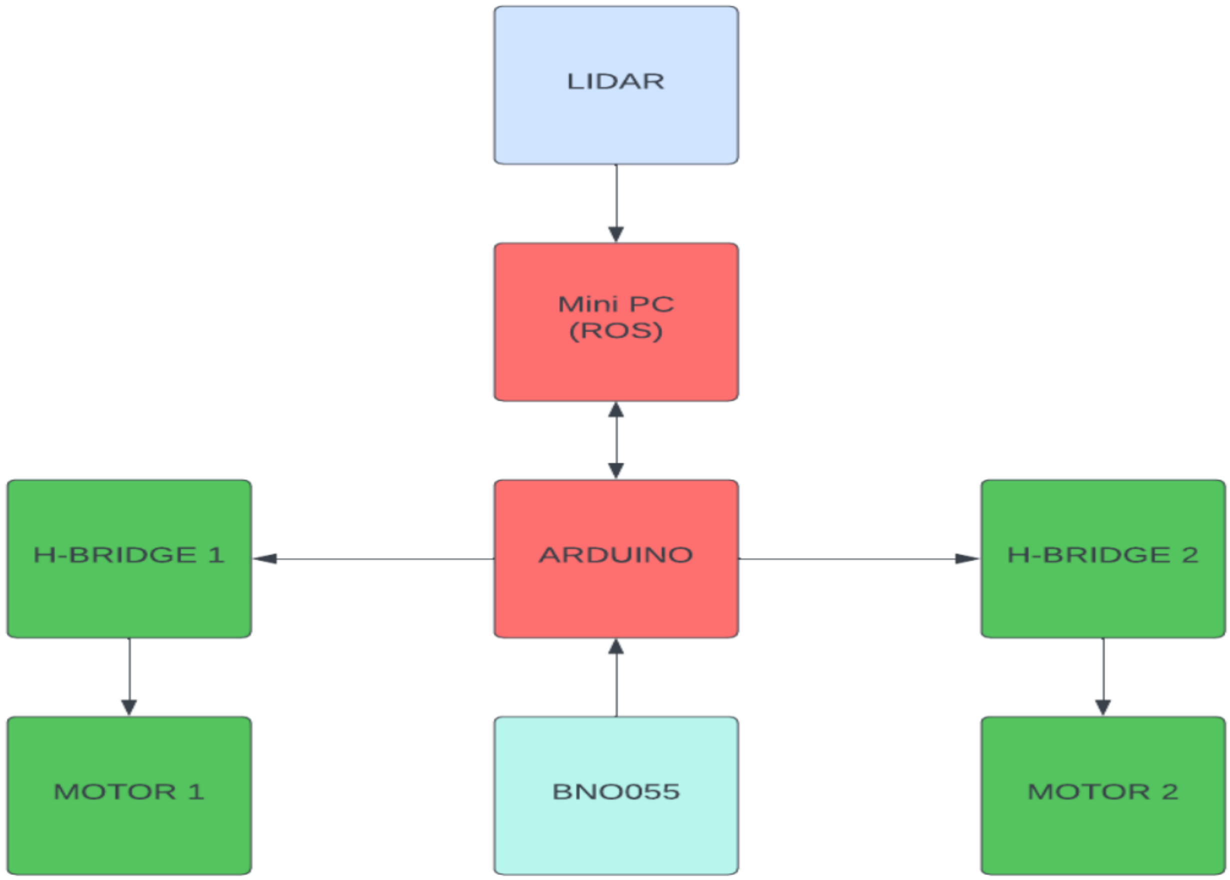


Figure 5. LIDAR X2 measurements angles

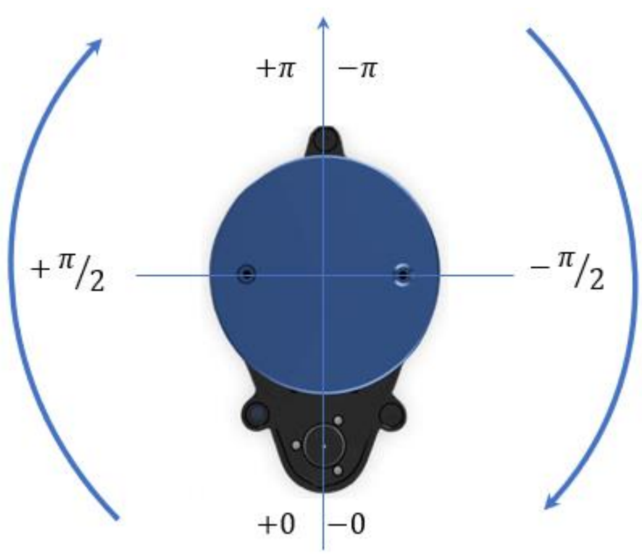


Figure 6. (a) Considerations measuring distance to a crop row, (b) approach to measure distance to a wall.

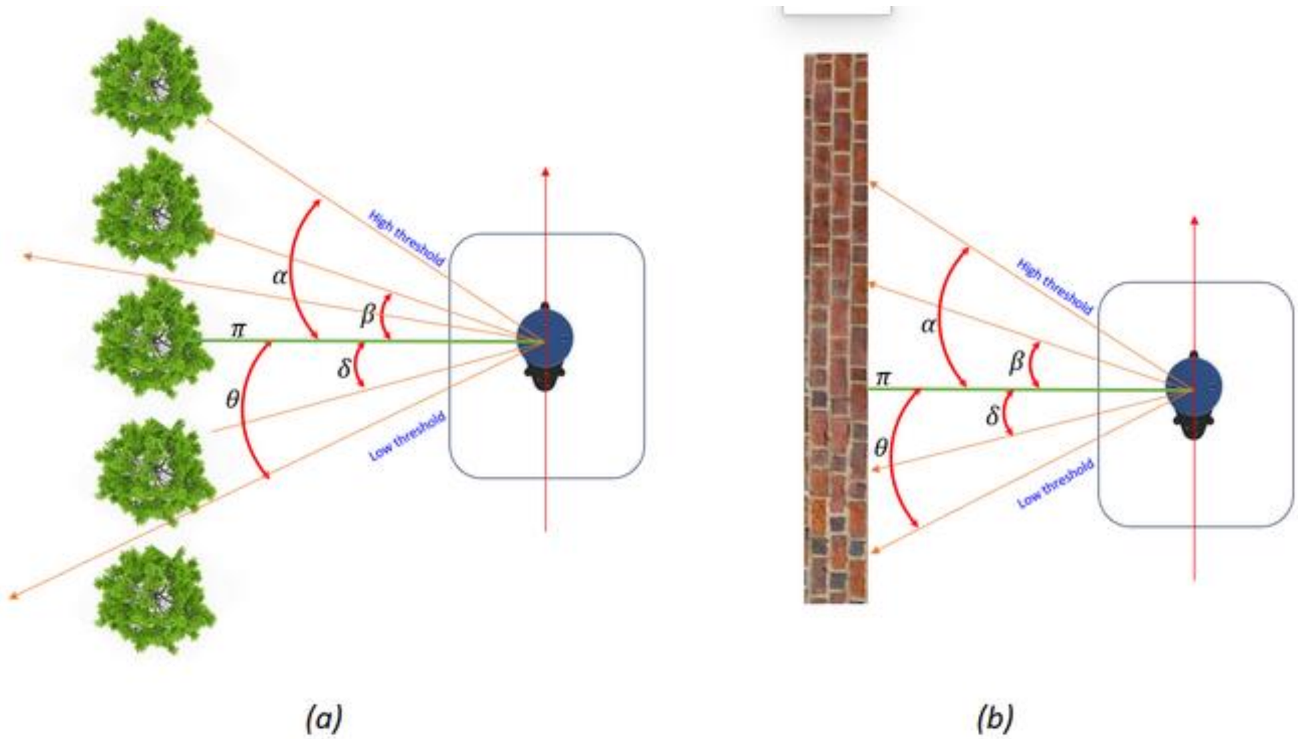
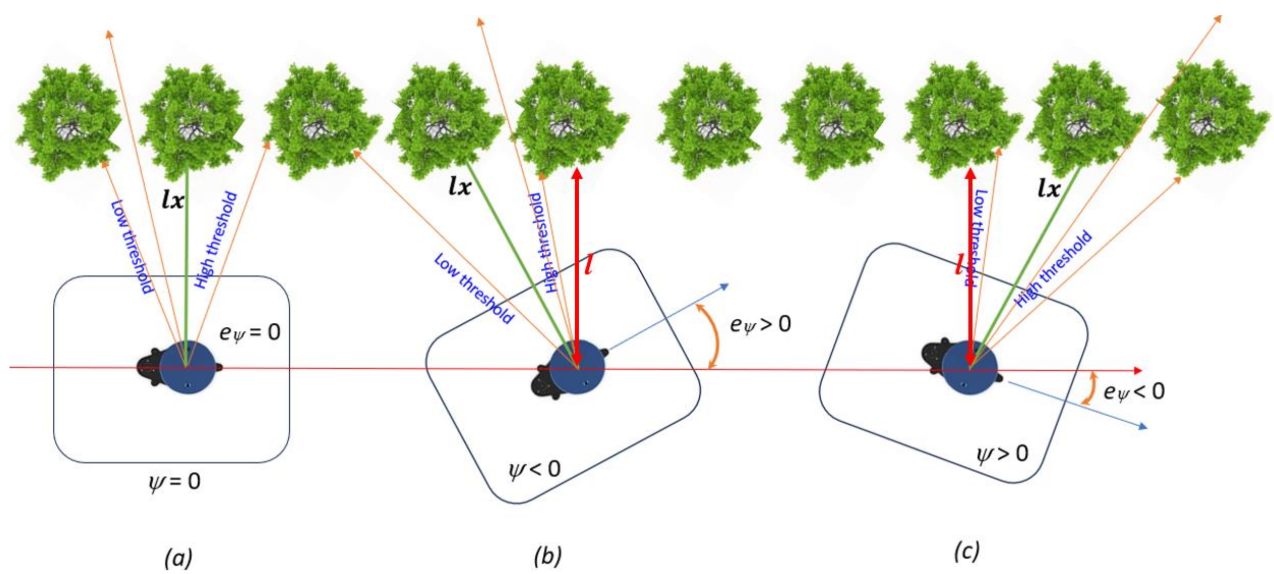


Figure 7. (a) Considerations measuring distance to a crop row, (b) approach to measuring distance to a wall.

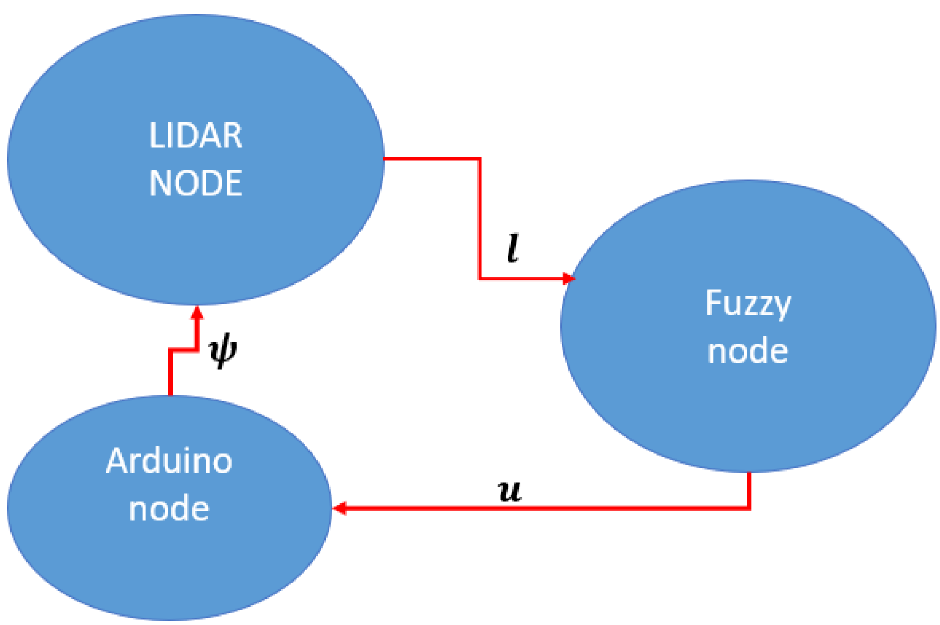


2.1 ROS undertaking

Let us use Figure 4 again to indicate that the mobile robot is controlled using ROS, which is installed on our mini-PC. As described in [8]. Due to this, the Arduino board in Figure 4 was configured as a ROS node (Arduino node), which is shown in Figure 8, and this node publishes the vehicle orientation. Figure 8 also illustrates a LIDAR node that is a subscriber to ψ topic. In addition to receiving information from LIDAR sensor, LIDAR node also calculates lateral distance based on vehicle orientation as discussed in previous paragraphs. Accordingly, LIDAR nodes publish a topic containing the lateral distance $\boldsymbol{l}$.

Additionally, as shown in Figure 8, a Fuzzy node is a subscriber of l topic, allowing the required control action ($\boldsymbol{u}$) to be calculated based on this information. Likewise, the Arduino node was configured as a subscriber to u, and then the Arduino read the $\boldsymbol{u}$ word. It is necessary to convert the control action ($\boldsymbol{u}$) on the Arduino board into signals in order to modify the angular speed of DC motors in order to move the robot appropriately.

Figure 8. ROS nodes and topics



2.2. Fuzzy control system

In Figure 8, the Fuzzy node performs calculations according to a fuzzy logic algorithm [15], as part of our approach, we use an input that corresponds to the lateral distance topic as well as an output that corresponds to the control action. As part of the fuzzy logic control strategy, input fuzzy sets are defined, which were defined using the skfuzzy library's Python control module which we recommend to review in this link: *https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_control_system_advanced.html*.

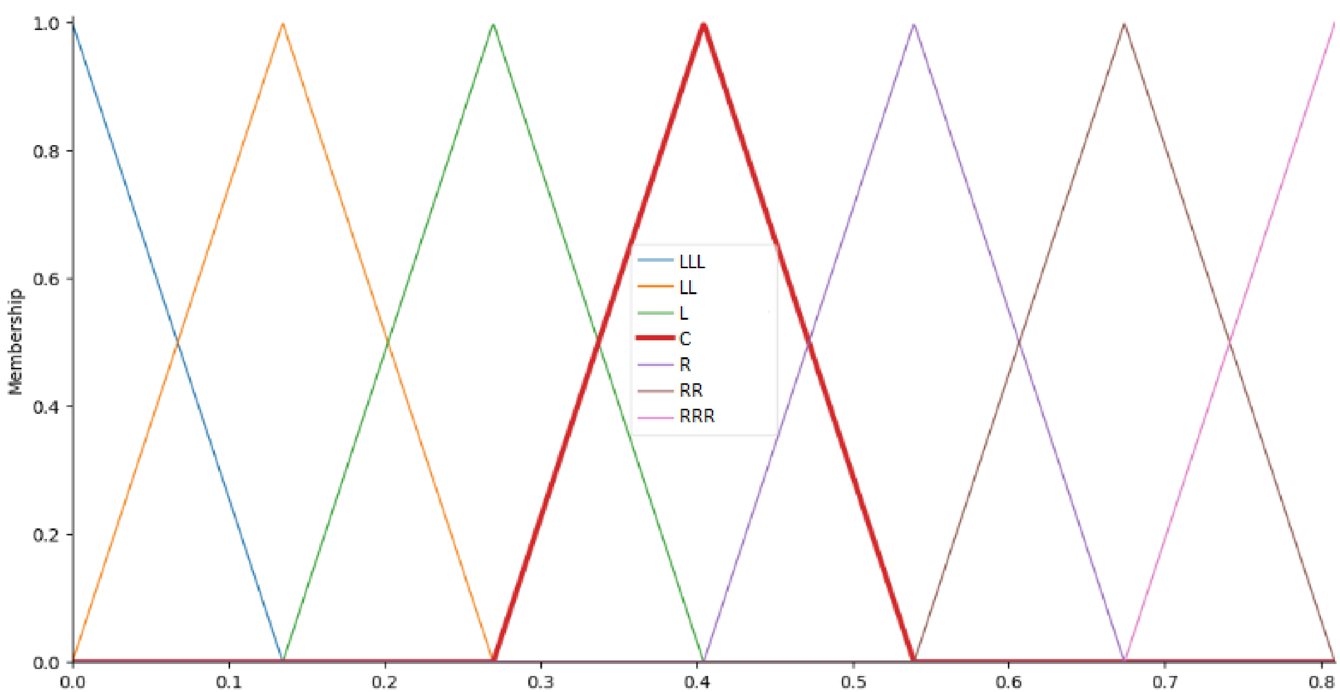
In order to describe fuzzy sets, let us include Figure 9, which illustrates that experimental crop rows were separated by 0.8 meters, and that the robot is positioned at the beginning of a crop path, facing forward, as indicated by the red arrow, allowing the robot to identify the left side row that it must follow.

Figure 9. Experimental crop rows

Figure 10 illustrates input fuzzy sets where the central set (C) has its maximum value at 0.4 meters, L represents the set which is certainly close to the crop row at the left side of the robot, and LL and LLL represent the sets that represent situations where the robot is closer to the left crop row. The R, RR, and RRR sets represent situations in which the robot is far from the left-hand crop row. The maximum distance according to the depicted sets is 0.8 meters.

Figure 11 illustrates output fuzzy sets representing control actions. As the central set (Z) has its maximum membership at 0, the robot is in a position where corrective actions are not necessary while it is displacing forward. The sets TR and TRR represent stronger rightward turns, while TL and TLL represent stronger leftward turns. A negative value corresponds to an action on the left, whereas a positive value corresponds to an action on the right.

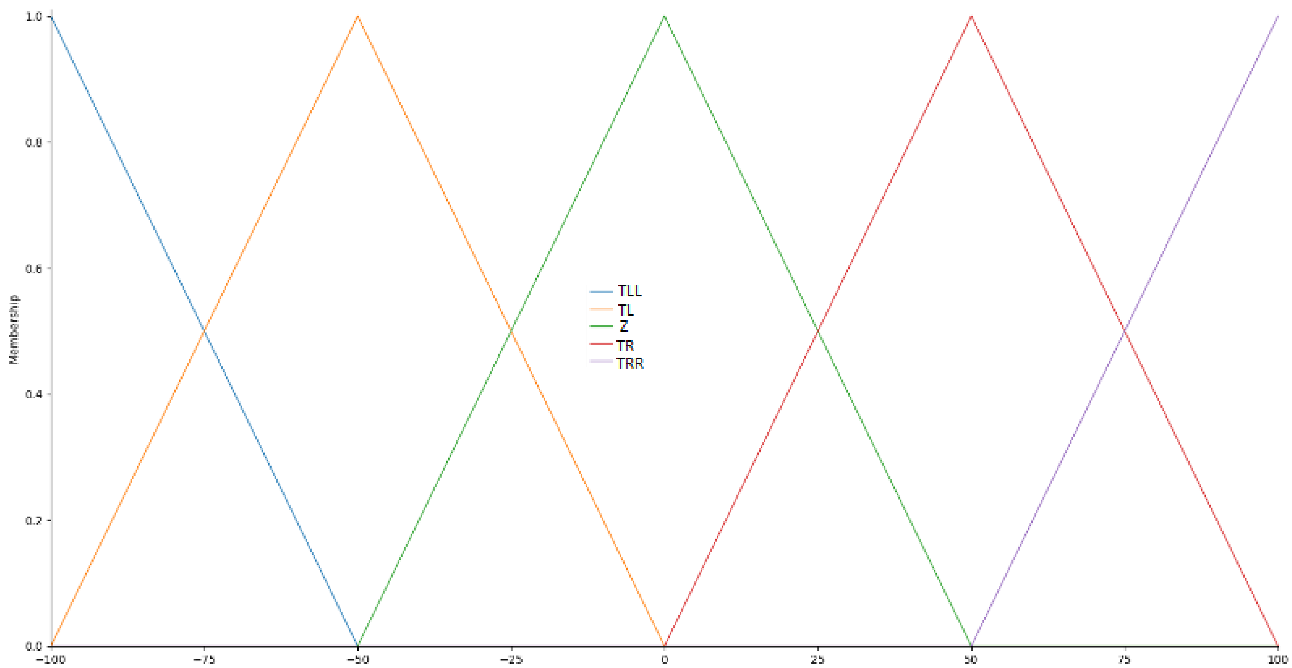
Figure 10. Input fuzzy sets.



It is possible to define fuzzy relations between input fuzzy sets and output fuzzy sets using the following rules:

rule1 = if ***l*** *is LLL, then* ***u*** *= TRR*
rule2 = if ***l*** *is LL, then* ***u*** *= TR*
rule3 = if ***l*** *is L, then* ***u*** *= TR*
rule4 = if ***l*** *is C, then* ***u*** *= Z*
rule5 = if ***l*** *is R, then* ***u*** *= TL*
rule6 = if ***l*** *is RR, then* ***u*** *= TL*
rule7 = if ***l*** *is RRR, then* ***u*** *= TLL*

Figure 11. Output fuzzy sets.

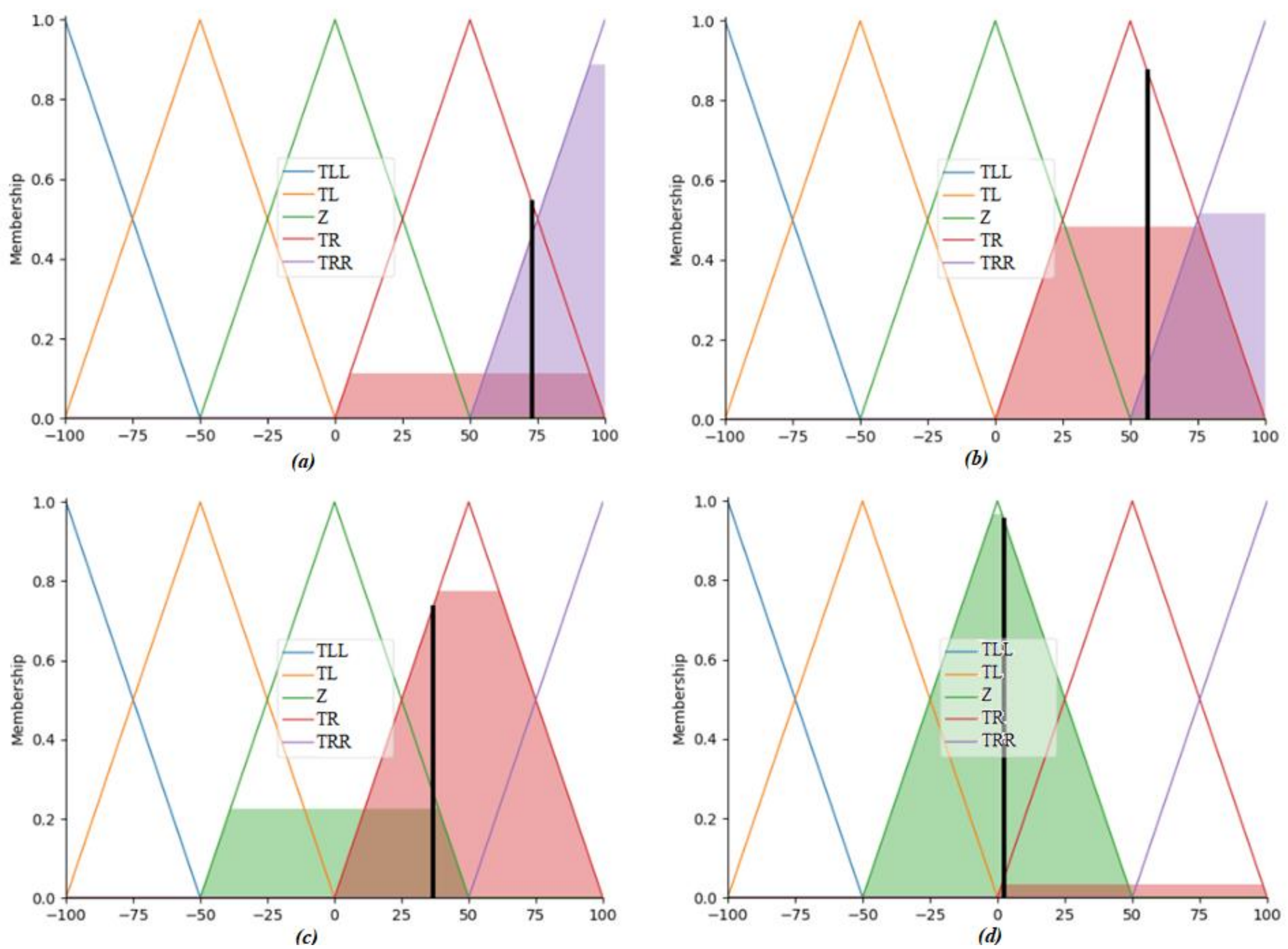


The Centroid method was chosen for the defuzzification process in order to determine the output value of the control action. In the phyton programming language, Centroid is part of the skfuzzy library's control systems module. Fig. 12 illustrates some examples of output values (u) when the robot is located close to the left based on the corresponding input values. If the vehicle is 0.15 meters close to the crop row at left in situation (a), the control action will be 72.87; if the robot is 0.2 meters close to its left-hand crop row, the control action will be 56.37; if the robot is 0.35 meters close to the crop row, the control action will be 36.71. As shown in figure 12(d), the control action is close to 0, since the vehicle is within 0.4 meters of the robot's left crop row.

Fig. 13 shows situations in which the mobile robot is 0.6, 0.6, 0.7 and 0.8 meters from the left side crop row, in which case the control actions are –33.63, -54.79, -82.82 and –83.24.

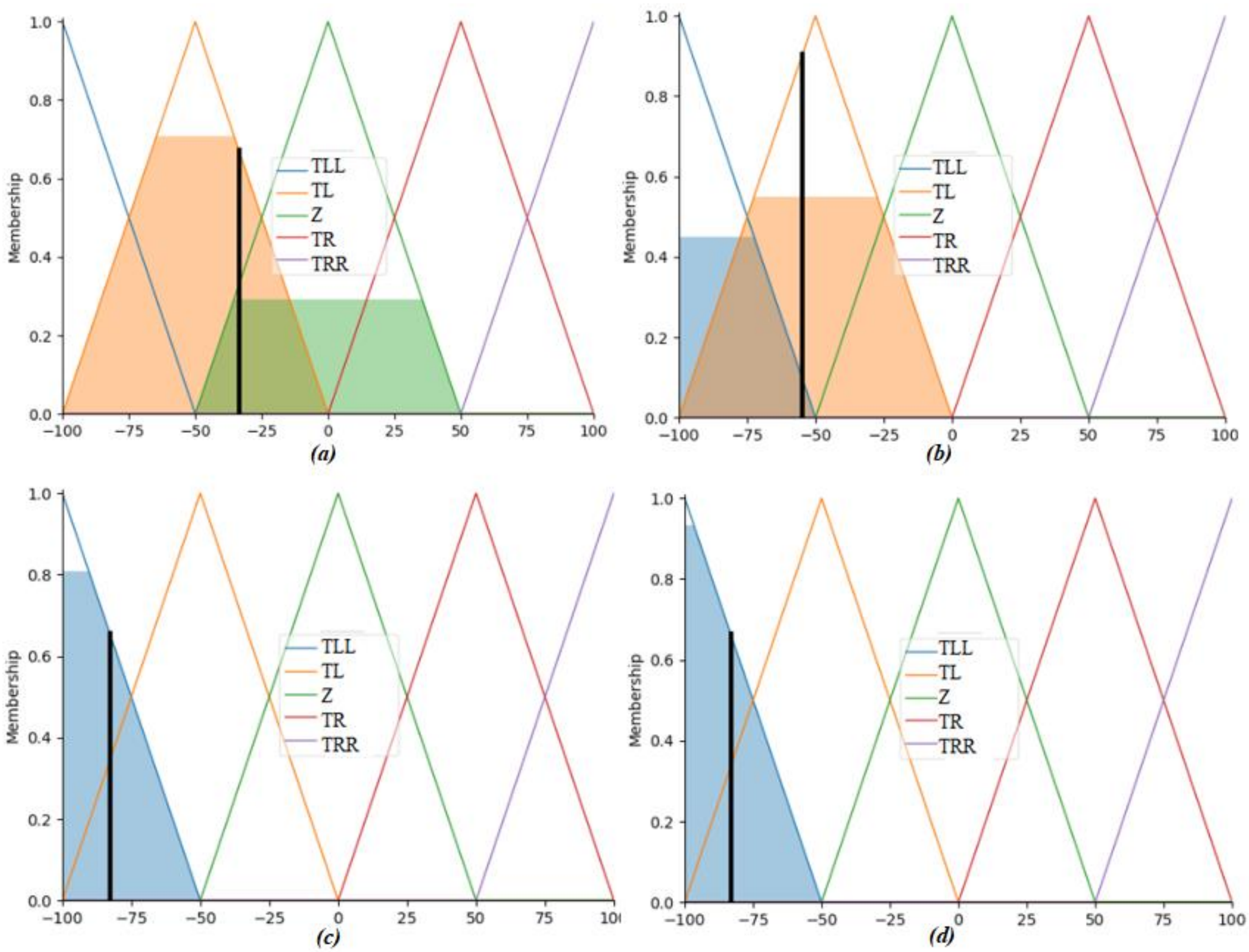
The defuzzification process leads to the determination of control actions values which according to diagram in Figure 8 corresponds to a message published on a ROS topic named ***u*** which the Arduino node is a subscriber to. Accordingly, u is a number that is received by the Arduino board that is programmed to generate the appropriate signals to control the two DC motors.

Figure 12. Defuzzification for situations when mobile robot is close to its left crop row.



Since we are using PWM signals to control the DC motors in our approach, Figure 14 illustrates how the Arduino is also used to control the two DC motors in accordance with the control action word received from the ROS fuzzy controller node. In this case, the control action $\boldsymbol{u}$ is converted into suitable PWM signals that are applied to the H-bridges that utilize the energy from the 12 Vdc battery to power the two DC motors to move the vehicle forward by merely reducing the angular speed of the right or left motor.

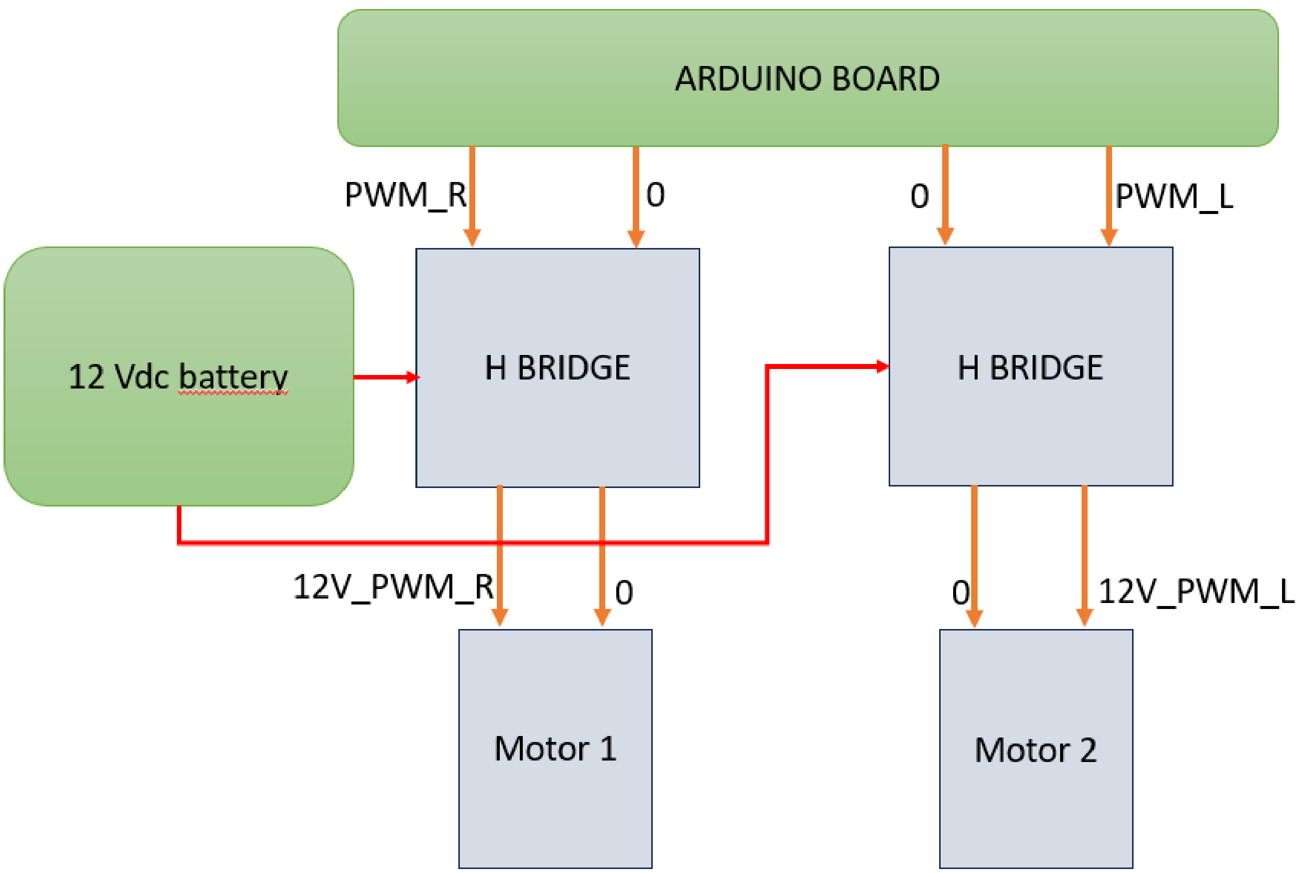
Figure 13. The defuzzification of situations where the mobile robot is considered to be far away from its left crop row



2.3 Reached results

In the course of the experiments, promising results were obtained. To describe the results shown in Figure 15, where three experimental results were included, let us refer to Figure 9. It was observed that the vehicle was capable of going over 0.4 meters when it was collocated at the beginning of a crop row, but approximately certain distances far to the left crop row, in this case 0.57, 0.4, and 0.28 meters, when necessary in an acceptable time, based on our goals, when it was required to do so.

Figure 14. Block diagram about Arduino and DC motors connections



As we noted in this first report, the number of reflected beams is significantly lower when the crop row is completed than when the robot is following it, so this condition was used to compel the robot to search for the next crop row by a curvature, which finishes when the reflected LIDAR beams return to normal quantities, at which point the robot continues searching for the next crop row. Figure 16 shows the results of a test in which the robot was required to move through three paths of crop growth. According to the show, the vehicle enters the next crop path with a non-suitable orientation, which is corrected together with the distance to the crop over a considerable period of time.

Figure 15. Plots about time response while the robot was moving between crops rows

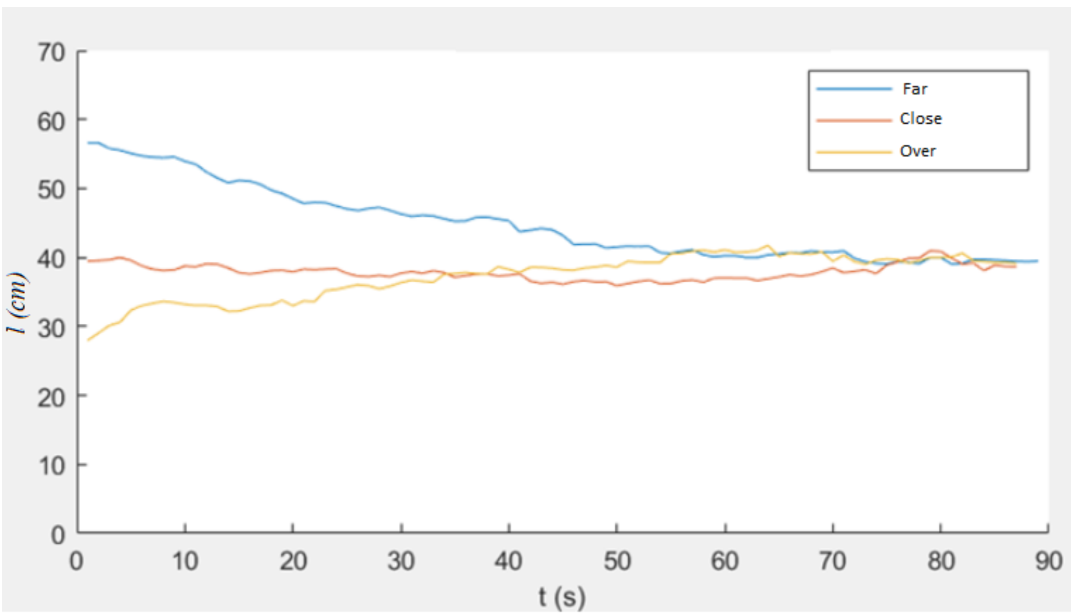
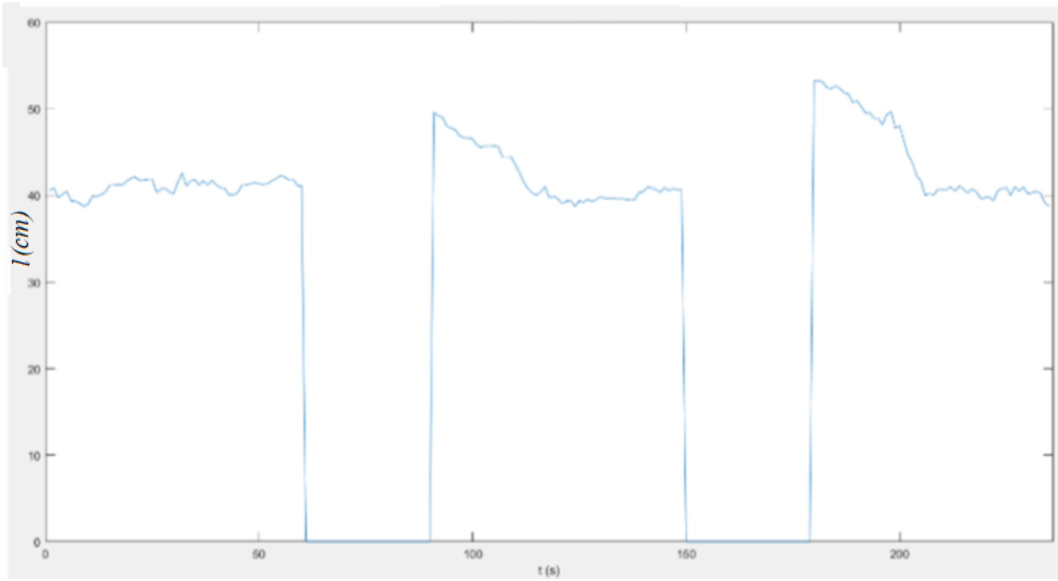


Figure 16. Plots about performance while the robot was moving between three crops rows



3. Conclusions and Future works

We used a fuzzy control system based on the experience of members of the authors' team in this approach. As a result of the execution of experiments, our team continues to be motivated to pursue an ongoing process aimed at reaching an undertaking involving the development of a mathematical model of the robot for simulation prior to real experimentation, thus preventing mistakes that could have been detected beforehand.

It is our intention to make use of the mathematical model mentioned in the previous paragraph in this project since we are also interested in experimenting with modern control systems for our platform in the future.

As the process for entering a next crop path is conducted in an open loop dynamic, it is not appropriated, and so this maneuver must be considered as a future work.

Based on the preliminary results of the experiments, we are confident that future work will achieve improved performance and results.

Acknowledgments

The authors acknowledge the academic and administrative communities at Universidad Distrital Francisco José de Caldas as they are always willing to support the development of innovative research strategies.

References

[1] H. Borrero Guerrero, A.E. Baquero Velasquez, J.F. Barrero, D.Z. Côco, J.C. Risardi, D.V. Magalhães, y M. Becker, "Orientation (yaw) fuzzy controller applied to a car-like mobile robot prototype," in 2014 IEEE 5th Colombian Workshop on Circuits and Systems (CWCAS), 2014, pp. 1-6. https://doi.org/10.1109/CWCAS.2014.6994603

[2] V.A.H. Higuti, H.B. Guerrero, A.E.B. Velasquez, R. Pinto, L.M. Tinelli, D.V. Magalhães, y D. Milori, "Low-cost embedded computer for mobile robot platform based on raspberry board," in ABCM International Congress of Mechanical Engineering (Cobem2015), Rio de Janeiro, Brazil, 2015.

[3] H.B. Guerrero, "Desenvolvimento de um sistema de controle em um robô móvel agrícola em escala reduzida para deslocamento entre fileiras de plantio," Ph.D. thesis, Escola de Engenharia de São Carlos, Universidade de Sao Paulo, 2016.

[4] H.B. Guerrero, "Desenvolvimento de um sistema de controle em um robô móvel agrícola em escala reduzida para deslocamento entre fileiras de plantio," Ph.D. thesis, Escola de Engenharia de São Carlos, Universidade de Sao Paulo, 2016.

[5] J. Ni, Y. Wang, H. Li, y H. Du, "Path tracking motion control method of tracked robot based on improved LQR control," in 2022 41st Chinese Control Conference (CCC), 2022. https://doi.org/10.23919/CCC55666.2022.9902113

[6] D. Ben Halima Abid, N.Y. Allagui, y N. Derbel, "Navigation and trajectory tracking of mobile robot based on kinematic PI controller," in 2017 18th International Conference on Sciences, 2017. https://doi.org/10.1109/STA.2017.8314966

[7] N.Y. Allagui, D.B. Abid, y N. Derbel, "Autonomous navigation of mobile robot with combined fractional order PI and fuzzy logic controllers," in 2019 16th International Multi-Conference on Systems, Signals Devices (SSD), 2019, pp. 78-83. https://doi.org/10.1109/SSD.2019.8893176

[8] J. Lentin, "Robot operating system for absolute beginners," Apress, Berkeley, CA, 2018.

[9] I. Nevludov, O. Sychova, O. Reznichenko, S. Novoselov, D. Mospan, y V. Mospan, "Control system for agricultural robot based on ROS," in 2021 IEEE International Conference on Modern Electrical and Energy Systems (MEES), 2021, pp. 1-6. https://doi.org/10.1109/MEES52427.2021.9598560

[10] R.K. Megalingam, D. Nagalla, K. Nigam, V. Gontu, y P.K. Allada, "PID-based locomotion of multi-terrain robot using ROS platform," in 2020 Fourth International Conference on Inventive Systems and Control (ICISC), 2020, pp. 751-755. https://doi.org/10.1109/ICISC47916.2020.9171152

[11] I. Alam Bhuiyan, "LiDAR Sensor for Autonomous Vehicle," 2017. https://doi.org/10.13140/RG.2.2.16982.34887/1

[12] Z. Lin, Y. Xiong, H. Dai, y X. Xia, "An experimental performance evaluation of the orientation accuracy of four nine-axis MEMS motion sensors," in 2017 5th International Conference on Enterprise Systems (ES), 2017, pp. 185-189. https://doi.org/10.1109/ES.2017.37

[13] B.G. Henry, Q.Y. David, C.M.J. Estivent, C.C.L. Arbey, C.R.Y. Alexis, y S.R. Andrés, "Lidar readings based mobile robot wall-following task using a reactive fuzzy control system - a low-cost experimental approach", 2020. https://hemeroteca.unad.edu.co/index.php/memorias/article/view/4201

[14] H.B. Guerrero, "Desenvolvimento de um sistema de controle em um robô móvel agrícola em escala reduzida para deslocamento entre fileiras de plantio," Ph.D. thesis, Escola de Engenharia de São Carlos, Universidade de Sao Paulo, 2016.

[15] S.N. Sivanandam, S. Sumathi, y S.N. Deepa, "Introduction to Fuzzy Logic using MATLAB," Springer-Verlag, Berlin, Germany, 2007. https://doi.org/10.1007/978-3-540-35781-0