|| Volume 7 || Issue 5 || May 2022 || ISSN (Online) 2456-0774
INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH
AND ENGINEERING TRENDS

SELF NAVIGATING AUTONOMOUS ROBOT WITH 2D – MAPPING

[1]Mr. Kishore Vennela, [2]M Nitish Kumar, [3]M Bhavishya, [4]N Rohith

[1]kishore.vennela@bvrit.ac.in, [2]nitishgupta1007@gmail.com, [3]madarapubhavishya@gmail.com, [4]narlarohith.123@gmail.com

Department of Electronics and Communication Engineering, B V Raju Institute of Technology, Narsapur, Telangana, India

-- ***---

Abstract: To design a path finding robot with moderate velocity and to map the unknown environment using ARDUINO. Ultrasonic sensors (also known as transceivers when they both send and receive, but more generally called transducers) work on a principle similar to radar or sonar which evaluates attributes of a target by interpreting the echoes from radio or sound waves respectively. Ultrasonic sensors generate high-frequency sound waves and evaluate the echo which is received back by the sensor. Sensors calculate the time interval between sending the signal and receiving the echo to determine the distance to an object. The controlling device of the whole system is an Arduino. The Arduino reads the ultrasonic sensors data regarding the obstacle detected. The robot moves in the same direction until it reaches another turn. To perform this intelligent navigation, the Arduino is loaded with an efficient program written using Embedded 'C' language.

KEYWORDS: *ESP32, Ultrasonic Sensor, Servo Motor, L293 Motor Driver*

---***---

I INTRODUCTION

Robots are made to go and do what humans either cannot or do not want to do. SLAM describes the process of building a map of an unknown environment and computing at the same time the robot's position with the constructed map. Both steps depend on each other. A good map is necessary to compute the robot's position and on the other hand, just an accurate position estimate yields a correct map [1]. Because navigation, localization, path planning, and mapping are an essential steps in a mobile robot, many of the approaches used are simple heuristics successively searching for face connected cells or configuration with or without the minimization of a criterion. Wall following theory for a mobile robot is a technique that makes mobile the robot navigate besides the wall without shocking it and also avoid hitting any obstacles during navigation near the wall. Wall following is useful for SLAM projects inan indoor environment because the mobile robot can move in most areas of the plan without a supervisor by using Wall following technique. The problem of mobile robot navigation has been historically faced by decomposing it in three sub-problems: environment mapping, localization, and trajectory planning. Those three sub-problems have developed into broad research areas over decades, tackled separately and in a deterministic way, ignoring the uncertainty in robot sensing and motion. In the early 1990s, probabilistic robotics turned the classical approach to navigation upside down, by developing algorithms that were able to explicitly take into account the intrinsic uncertainty in robot sensors [2]. Probabilistic robotics focuses on representing uncertainty and information by probability distributions and not on the basis of a single guess. With the rise of those methods, the problem of Simultaneous Localization And Mapping (SLAM), also known as Concurrent Mapping and Localization (CML) arose: the mapping of sensor readings with respect to a global frame of reference depends on the robot's location in that frame and on the uncertainty in this robot's pose due to the cumulative odometry error that affects the process of building the map [3]. Visual odometry allows for enhanced navigational accuracy in robots or vehicles using any type of locomotion**.** Combining visual information from cameras or inertial sensors with wheel motion information allows us to overcome the drift and provides much more accurate localization. However, errors in the built map and the robot's pose are correlated. Thus, these two problems should be tackled together. SLAM techniques vary depending on whether indoor or outdoor robots are used. Outdoor environments are more challenging as they do not have specific limits and the criteria for choosing the next navigation point or termination can be completely different. Besides, indoor and outdoor classic SLAM systems differ from each other in aspects such as the individual requirements, sensors provided and morphology of the robots, Markovian, Kalman filter-based, and particle filter-based are the most common techniques used to solve the SLAM problem. Although SLAM in static environments can be considered solved, dynamic environments are still challenging [4]. SLAM can be considered to be a mapping process in which robot localization is uncertain. However, the planning task is put aside during the map-building process. Although strategies like coastal navigation can be used, while building the map the robot is usually guided by a human by means of teleportation. In this way, the wheels' drift can be minimized, ensuring as well the full coverage of the environment. Once the map is available, stochastic planning techniques, e.g., Partially Observable Markov Decision Process (POMDPs), are used for navigation. The advent of SLAM techniques motivated huge advances and opened new possibilities for robot development, but there are still considerable challenges in performance when adding environment dynamism or increasing dimensionality [5]. Moreover, tele operating the robot for mapping is usually a highly time-consuming task, especially in large areas or when the movement of the robot is limited. In other cases, it is difficult or even impossible for the robot to be guided due to insufficient connectivity or dangerous conditions, such as in rescue operations of natural disasters.

II. REAL-WORLD AUTONOMOUS NAVIGATION

To achieve this feat, AMRs must be able to localize within their far-reaching boundaries, as well as plan pathways through congested areas that may look very different every time the AMR passes through. In this environment, especially working in and among persons, obstacle detection and avoidance are of paramount importance. Fortunately, the same sensors that give the AMR the capability of localizing itself are able to detect and report the occurrence and location of obstacles that the AMR must negotiate. The recent changes in power and cost-effectiveness of these sensors, typically lidar, make AMRs feasible and have ushered in a wave of entrants into what is a rapidly growing product segment. Competition and the shrinking electronics have made a typical lidar used in mobile applications go from more than $10,000 and the size of a basketball a decade ago, to comparative models today at sub-$2,000 cost and as small as a baseball. Higher scan rates and more angular resolution provide ever more data feeding hungry algorithms used for modern localization algorithms. SLAM can use many sensors as input, with lidar being the most common and readily available at present. It often uses several different types of sensor inputs, with algorithms tailored to the powers and limits of the various sensor types. Each sensor has associated with it an algorithm for determining motion or position from that input. For example, odometry, or wheel revolutions, is used with the mathematical kinematics of the mobile robot base, to calculate movement based on wheel rotation. Sonar or ultrasonic sensors can estimate position based on standoff distances to local infrastructure. And lidar range scans can be used in several ways to estimate mobile robot motion or position. As an iterative inference problem, SLAM starts with a known condition, being the location and pose of the AMR, a modeled prediction of a future condition, being the location and pose estimate based on current speed and heading, and sensor data from multiple sources with estimates of error and noise quantities. SLAM uses statistical techniques, including Kalman or particle filters, to approximate a solution to the robot's location and pose iteratively.

III.PROPOSED SYSTEM

The below figure (1) shows the block diagram of the proposed system. In this servo motor, 12V battery, Ultrasonic sensor, DC Motor, Motor driver circuit, cloud database and path planning are used. ESP32 is a feature-rich MCU with integrated Wi-Fi and Bluetooth connectivity for a wide range of applications. Here, with help of ultrasonic sensors, it receives the data where the obstacle is there. When the program starts running, the servo motor starts to rotate from 0 degrees to 180 degrees, one degree at a time. If there is any obstacle detected then, it will decide whether to go right or left. Motor drivers act as an interface between the motors and the control circuits. For the path plotting purpose, we use "FIREBASE" as a cloud database. The protocol for the data transmission used is "UART". The Path planning can be seen on PC will the help of data stored in FIREBASE.

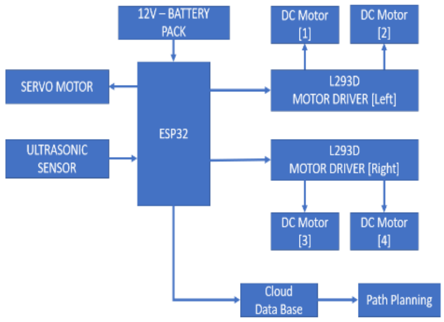

Fig. 1: BLOCK DIAGRAM OF PROPOSED SYSTEM

ESP – 32

It is completely unfair to compare ESP32 with Arduino; both are advantageous and functional on their own. In terms of power and features, obviously, the dual-cored microprocessor-powered ESP32 will surely take down the microcontroller-powered Arduino UNO. The ESP32 has built-in Bluetooth and Wi-Fi with a good number of GPIO pins and communication protocols for a very cheap price. The Arduino might look a bit handicapped when competing with ESP32 but it has a large number of shields in the market that can be readily used, also advanced Arduino boards like Yun have good processing power as well.

Input/output

There are a total 39 digital Pins on the ESP32 out of which 34 can be used as GPIO and the remaining are input-only pins. The device supports 18-channels for 12-bit ADC and 2-channel for 8-bit DAC. It also has 16 channels for PWM signal generation and 10 GPIO pins support capacitive touch features. The ESP32 has a multiplexing feature, this enables the programmer to configure any GPIO pin for PWM or other serial communication through the program. The ESP32 supports 3 SPI interfaces, 3 UART interfaces, 2 I2C interfaces, 2 I2S interfaces, and also supports CAN protocol.

ULTRASONIC SENSOR

The HC-SR04 Ultrasonic (US) sensor is a 4-pin module, whose pin names are Vcc, Trigger, Echo, and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which form the Ultrasonic transmitter and Receiver. The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module.

SERVO MOTOR

There are lots of servo motors available in the market and each one has its own specialty and applications. The following two paragraphs will help you identify the right type of servo motor for your project/system. Most of the hobby Servo motors operate from 4.8V to 6.5V, the higher the voltage higher the torque we can achieve, but most commonly they are operated at +5V. Almost all hobby servo motors can rotate only from 0° to 180° due to their gear arrangement so make sure your project can live with the half-circle if not, you can prefer a 0° to 360° motor or modify the motor to make a full circle. The gears in the motors are easily subjected to wear and tear, so if your application requires stronger and long-running motors you can go with metal gears or just stick with normal plastic gear.

L293D MOTOR DRIVER

The L293D is a popular 16-Pin Motor Driver IC. As the name suggests it is mainly used to drive motors. A single L293D IC is capable of running two DC motors at the same time; also the direction of these two motors can be controlled independently. So if you have motors thathave operating voltage less than 36V and operating current less than 600mA, which are to be controlled by digital circuits like Op-Amp, 555 timers, digital gates, or even Microcontrollers like Arduino, PIC, ARM, etc.This IC will be the right choice for you.

IV.RESULTS

The below figure (2) shows the hardware setup of the proposed system. In this servo motor, 12V battery, Ultrasonic sensor, DC Motor, Motor driver circuit, cloud database, and path planning are used.

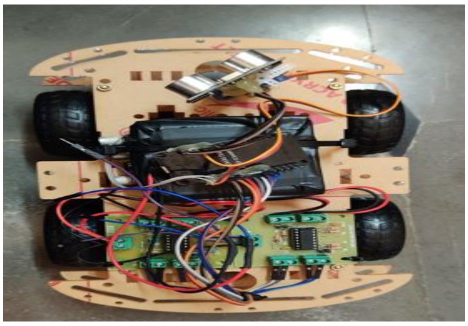

Fig. 2: FRONT VIEW OF PROPOSED SYSTEM

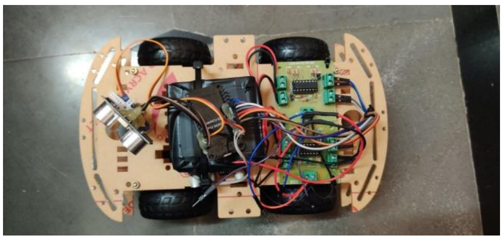

Fig. 3: TOP VIEW OF PROPOSED SYSTEM

Fig. 4: OUTPUT -1

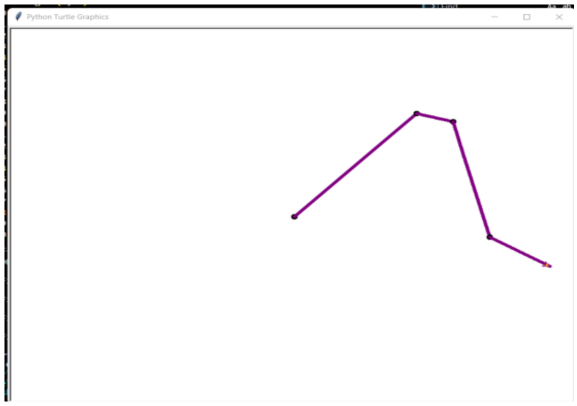

Fig. 5: OUTPUT -2

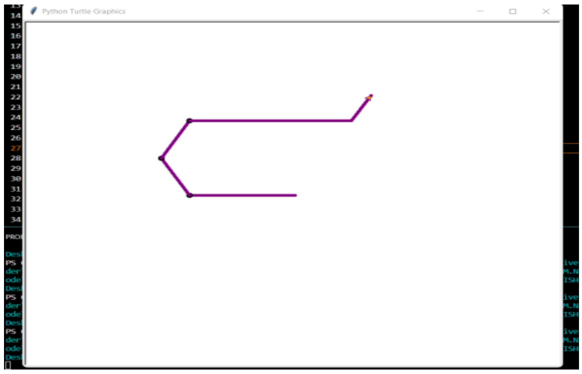

Fig. 6: OUTPUT - 3

V.CONCLUSION

The following points summarize the main conclusions of navigation and mapping with path plan result derived from Experiments work: Wall following technique make the mobile robot move without a supervisor and also make the mobile robot navigate in an unknown environment without the need the map or plan. Increasing speed to high value makes mobile robots in danger of crashing and affects the stability of path plan, and also decreasing the speed to low value makes the time consumed in high value. Decreasing the value of the robot vision range of mobile robots, especially near range makes the mobile robot move close to the wall and that make it in danger of crashing. Increasing the value of robot vision of mobile robot, especially near range make mobile robot not able to move in the narrow corridor. The effect of varying sensor range or robot vision showed that it does not only affect the distance between the mobile robot, obstacles, and walls, but also it affects the shape of the path from start to end point increasing the sensor range may force the robot to take an entirely different path to the goal especially if it has to go through small openings with respect to the sensor range.

VI.REFERENCES

[1] Elmar A. R uckert, "Simultaneous localization and mapping for mobile robots with recent sensor technologies", Graz University of Technology Institute for Computer Graphics and Vision, Master Thesis December 2009.

[2] Ahmed Rahman,"A Fuzzy Logic Control for Autonomous Mobile Robot ", University of Baghdad, Al-Khawarizmi College for Engineer, Mechatronics Department, Msc thesis, 2009.

[3] Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: Part I. IEEE Robot. Autom. Mag. 2006, 13, 99–110.

[4] Howie Choset, KeijiNagatani, " Topological Simultaneous Localization and Mapping (SLAM) ", IEEE Transactions on robotics and automation, 2- APRIL- 2001.

[5] Tom Duckett, " A Genetic Algorithm for Simultaneous Localization and Mapping", IEEE International Conference on Robotics & Automation, September- (14-19)-2003.

[6] Young-Ho Choi ,Tae-Kyeong Lee ,Se-Young Oh, " A line feature based SLAM with low grade range sensors using geometric constraints and active exploration for mobile robot", Journal Autonomous Robots, 1-January-2008.

[7] Jun Okamoto Jr., Vitor CampanholoGuizilini, " Online SLAM Using Clustered Landmarks with Omnidirectional Vision", Journal of the Brazilian Society of Mechanical Sciences and Engineering, Dec-2010.

[8] Ngai Ming Kwok, Quang Phuc Ha, Shoudong Huang , Gamini Dissanayake, and Gu Fang, " Mobile Robot Localization and Mapping using a Gaussian Sum Filter", International Journal of Control, Automation, and Systems, vol. 5, no. 3, pp. 251-268, June-2007.

[9]. Roy, N.; Burgard, W.; Fox, D.; Thrun, S. Coastal navigation-mobile robot navigation with uncertainty in dynamic environments. In Proceedings of the IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C), Detroit, MI, USA, 10–15 May 1999; Volume 1, pp. 35–40.

[10] Mu, B.; Giamou, M.; Paull, L.; Agha-mohammadi, A.a.; Leonard, J.; How, J. Information-based active SLAM via topological feature graphs. In Proceedings of the IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NA, USA, 12–14 December 2016; pp. 5583–5590