Design of an Autonomous Mobile Robot Based on ROS

Murat Köseoğlu[1], Orkan Murat Çelik[2], Ömer Pektaş[1]

[1]Dept. of Electrical-Electronics Engineering, Inonu University, Malatya, Turkey
[2]Havelsan Hava Elektronik Sanayi A.Ş., Ankara, Turkey
murat.koseoglu@inonu.edu.tr, omcelik@havelsan.com.tr, omerpektas003@hotmail.com

***Abstract*— In this paper, design of an autonomous mobile robot (AMR) adapted for robot operating system (ROS) is presented by considering both the hardware architecture and electronic communication protocols. Initially the purpose of the robot has been determined, and then the required components to construct the robot have been obtained. After mounting the electronic and mechanic hardware components on the platform, the required interconnections, data exchange system and software have been installed. Then the required tests have been run on the platform and some solutions methods have been proposed for the problems encountered during the tests. The results obtained and experience gained in the design of the platform have been quite satisfactory and will be a guiding light for future works.**

***Index Terms*—Autonomous mobile robot (AMR), robot operating system (ROS), embedded, light detection and ranging (LIDAR), simultaneous localization and mapping (SLAM).**

I. Introduction

The robots are generally used to serve or help people in several fields, from daily life works to industrial applications. The robot science has developed significantly due to the demands of people and industry recently. One of the main areas of the robot science is the mobile robots. The mobile robots have capable to navigate in an environment and interact with it via the sensors and the actuators. The mobile robots can be classified as autonomous mobile robots (AMR) and autonomous guided vehicles (AGV). The distinction between AMR and AGV is attributed to the autonomy mechanism of these robots. An AGV is based on a physical guidance and navigates in a pre-defined environment on a pre-defined path. AGVs have been frequently used in industry, but it leaves its role in industry to AMR recently. AGVs are suitable for repetitive tasks such as line follower robots, and they are commonly designed and produced for specific tasks. The task which will be performed by AGVs must be planned elaborately, and all the details must be defined to AGV by the programmer, since it cannot make a decision and doesn't have a decision mechanism based on an artificial intelligence. AGVs work according to preset systems and processes, which can make a rapid change difficultly. A business model that tends to react to trends or that is otherwise agile may not be the best fit for AGVs [1]. As mentioned, the main disadvantage of the AGVs is the inexecution of dynamically changing tasks.

An AMR is capable to navigate in an unpredictable environment. AMRs can sense the parameters of the environment and create a model of the environment and locate itself in this model. This behavior enables AMR to make a navigation plan and optimize this plan by a special planning algorithm. Shortly there is no predefined navigation plan for an AMR. Also, an AMR can create a map of environment using sensor data and localize itself in the map at the same time. This is known as simultaneous localization and mapping (SLAM). SLAM enables to create a quite sensitive navigation plan which can be revised and improved dynamically by the programmer.

AMRs have been widely used in industry and different fields due to their advanced features in recent years. AMRs have more complex hardware and software design in comparison with AGVs, since AMRs are equipped with more sensors and actuators. So, they have more complicated control systems which work in a coordinated manner in accordance with the data stream concurrently. This kind of operation is possible if and only if a smooth hardware compatible with default software is provided. As mentioned, it is effortful to design an AMR which operates properly.

In this study, an AMR, which can create the map of an indoor environment, has been designed and implemented. The design procedure and the integration of the hardware have been explained comprehensively. The major difficulties and solution methods in the implementation process have been mentioned. Some basic knowledge on AMR design has been tried to convey to the authors.

II. Design Procedure

The hardware side a mobile robot consists of both electronic and mechanical components that connected to each other and works compatible. Thus, both electronic and mechanical processes and procedures must be considered together in the design process.

978-1-5386-1880-6/17/$31.00 ©2017 IEEE

A. The Mechanical Design Procedure

The mechanical design procedure consists of all the mechanical parts of a mobile robot that can navigate in a physical environment.

The environment can be defined in two ways: i) indoor and ii) outdoor. The type of environment is the major factor which determines the fundamental working parameters and characteristics. If the robot is designed for an outdoor environment, the mechanics of robot must overcome the environmental stress which results in more difficult conditions and more complicated mechanical design in comparison with the indoor robots. This is the main problem in design for a mobile robot that can navigate in outdoor environment. If the robot is designed for an indoor environment, the design process is simpler in comparison with an outdoor robot design.

In this study, we have designed a mobile robot for a hard and smooth surfaced indoor environment. This case facilitates to use the wheels for locomotion. The basic mechanic parts of the designed robot consist of the chassis, two motors and four wheels. The chassis, which all of the mechanical and electrical parts are assembled on, is designed as a metallic circle plate which can work properly with differential drive system. We used a computer aided design (CAD) program for designing the chassis. and the chassis is capable to carry all of the load that weighted down. After the design process, the chassis was manufactured using a laser computer numerical control router (CNC). The motors have been chosen as geared DC motors which have quadrature encoder to get the rotation data from the motor shaft and transfer it to MCU. The motor type has been determined in accordance with the weight of the hardware and the desired speed which is compatible with the sensor resolution and data stream speed. This is an important factor in the navigation control of the robot. Besides two motor connected wheels, two free wheels (casters), which have been used to balance the robot, have been mounted in front and back side of the chassis.

B. Electronic Design Procedure

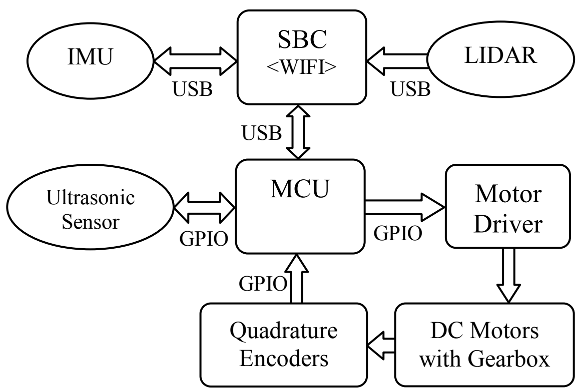
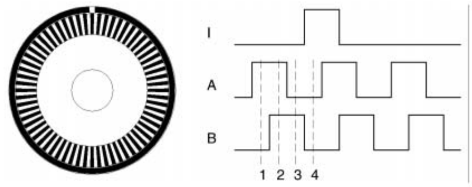


Fig. 1. Scheme of electronic design of the mobile robot

The electronic design of the robot is presented in Fig. 1. All of control and navigation algorithms run on electronic hardware. Electronic design of the robot is composed of three main parts which are power system, decision-control system and sensor systems. Power system is basically based on a battery and a power regulation system. We have used a Lithium-Polymer Battery (LiPo) and two switch mode power regulators due to the low dissipation and high efficiency. In order to protect the decision-control system from electromagnetic interference (EMI), the power system of decision-control system has been isolated.

The decision-control system is the main part of the electronic hardware. It consists of two different computers which work simultaneously and interdependently. One of the computers is a single board computer (SBC) which is the center of decision-control system. Different control systems can be run on SBC. The other computer is based on a 32-bit microcontroller (MCU) and designed to control peripherals and some sensors. A real-time operating system (RTOS) is embedded in MCU. A real-time system is characterized by its timing constraints in a sense that results or actions must be provided within a specific time window otherwise they are considered to be wrong, whatever their value is [2]. In this context, we have used an RTOS to execute real time tasks defined before. The MCU and SBC are connected to each other via a serial connection for communication. This communication is based on universal serial bus (USB). USB is used as virtual com port (VCP). The MCU sends the sensor data and the other parameters of the robot to the SBC. Simultaneously, the SBC gets the sensor data and uses these data to run robotic algorithms. Then the results of the algorithms are sent back to MCU to be executed in the navigation of the robot.

The last part of the hardware is the sensor system. The sensor systems are very important parts of mobile robots. The sensors which have used for our robot has been chosen by taking robot's mission into consideration. The robot has been designed for indoor navigation. So, we have chosen sensors to optimize this task. We have used a LIDAR, an ultrasonic range finder, quadrature encoders and an inertial measurement unit (IMU). LIDAR is a remote sensing technique that uses laser light in much the same way that sonar uses sound, or radar uses radio waves [3]. A LIDAR may also be built up by xenon or flash lamps, which are no laser sources [4]. It measures range at 360 degrees at same height. This gives a two dimension (2D) view of the environment of robot. The robot can create a 2D map of the environment by using LIDAR data and a SLAM algorithm. The ultrasonic range finder is a sensor based on ultrasonic sound signals. It works like lidar but it measures the range just in one direction. We use this sensor for the collision avoidance. The quadrature encoders are connected on the shafts of the right and left geared motors to get rotation data. As shown in Fig. 2, the quadrature encoders have two signal channel to get speed and rotation direction of motor which are A and B. The signals of channel A and B are square waves, and there is a 90 degrees' phase shift between them [5,6].



State	Ch A	Ch B
s_1	high	low
s_2	high	high
s_3	low	high
s_4	low	low

Fig. 2. A quadrature encoder produces pulses that are 90^0 out of phase [7].

IMU is an electronic device that measures and reports the velocity, orientation and gravitational forces applied to a craft, and it uses a combination of accelerometers and gyroscopes [8]. The IMU which have we have used 9 degree of freedom (9 DOF). We use data that taken from IMU to get more reliable odometry.

III. Implementation of AMR

After the design process, we assembled the mechanical and electronic parts on the chassis. The required programs were written for the proper operation of the robot by considering the used sensors and actuators. Each sensor was tested separately for different situations.

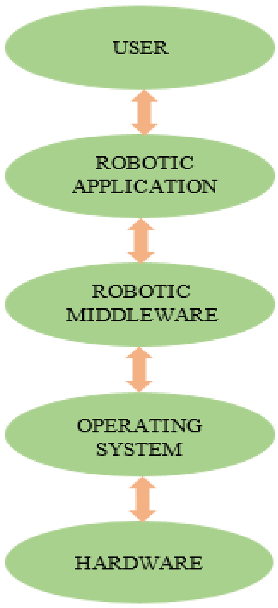


Fig. 3. Software and hardware interaction

As mentioned, two geared motors and differential drive systems were used for the navigation. After the sensor tests were completed, the motors and the chassis were tested for the navigation at different indoor environments. Some problems were encountered during these tests. For instance, the heading of the robot couldn't be rotated, since the friction on the caster wheels was bigger than the other wheels. So, we changed the caster wheels and mounted the new casters wheels by considering the friction. After the completion of these tests, the electronic parts were installed on the chassis, and the sensors and actuators were connected to the electronic parts and the chassis. Then, the required algorithms and software were established. As mentioned above, an SBC was used as the hardware for the decision and control system. The SBC is capable to execute tasks just as a personal computer. A linux distribution runs on SBC as operating system (OS). A robotic middleware ROS installed on OS was used to control the system. Bakken et al. defined middleware as follows: "a class of software technologies designed to help manage the complexity and heterogeneity inherent in distributed systems. It is defined as a layer of software above the operating system but below the application program that provides a common programming abstraction across a distributed system [9]. ROS is a "thin, message-based, peer-to-peer" [10], robotics middleware designed for mobile manipulators. The performed robotic applications were designed as message based applications. All of these applications are connected to each other with ROS. Thus, the user can see and dynamically control all the tasks running on the robot easily.

The designed robot has the capability of mapping of an indoor environment by using SLAM algorithms, and it can locate itself on a map created before. The robot can navigate in an area, which has a predefined map, by using data obtained from the sensors and the odometer. Odometry is the most widely used method for determining the momentary position of a mobile robot. In most practical applications odometry provides easily accessible real-time positioning information between periodic absolute position measurements [11].

After the integration of the parts and the software implementation were completed, a problem owing to an error in odometry calculation was encountered about the navigation algorithm, since the odometry calculation was just based on the encoder ticks. In other words, all the odometric calculations were made according to the data of position change in motor shaft. But in a special case, when the robot couldn't succeed in the obstacle avoidance in any time of moving process, it was seen that the rotation data obtained from the wheels were transferred to MCU even if there was no clear displacement, since the encoders had continued to send the ticks depending on the rotation of the wheels. In order to overcome this problem and get more reliable odometry calculation, an Extended Kalman Filter (EKF) was used to fuse the data obtained from the quadrature encoder and IMU sensor [12]. The Kalman filter, also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time, containing random noise and other inaccuracies, and produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone [13]. The Kalman filter has numerous applications in technology such as guidance, navigation and control of vehicles. In this method, the measured data were fused, so the desired estimations could be performed by using the EKF [14].

IV. RESULTS AND DISCUSSION

After the completion of the design and the implementation processes, some tasks have been charged to the mobile robot to test its capability of performing the ordered missions. For this purpose, the robot has been located in an indoor environment which consists of a hall and a room. At first, the task of mapping the environment is charged to the robot. The robot has navigated in the environment and created the 2D map of the environment as shown in Fig. 4.

The obtained 2D map can be used for navigation planning by following the optimal paths even if the robot is faced with several obstacles in the environment. In the mapping process, the software side of the robot has been tested. A joystick application, which provides the communication between the drive system and joystick, has been written for ROS and the differential drive system has been tested by using this application. During the test process, the robot has followed all the given directions successfully.

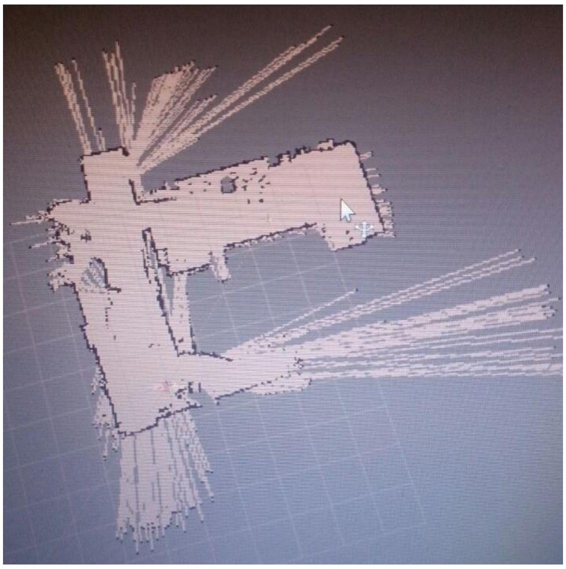


Fig. 4. The 2D map created by the mobile robot.

The sensitivity of speed control has been tested by trying different coefficients on PID controller. During the PID control process, all the data received from encoders has been collected in SBC, then the step response graphics based on these data has been drawn and the optimal PID coefficients given in Eq.1 has been determined by considering these graphics.

$$u(t) = K_p e(t) + K_i \int_0^t e(t)\, d\tau + K_d \frac{de(t)}{dt} \qquad (1)$$

In Eq. 1, K_p, K_i and K_d are all non-negative numbers which define the proportional, integral and derivative coefficients, respectively [15]. The obstacles were detected via the lidar, and the obstacle avoidance was provided by evaluating the measured data according to expression given as [16]

$$D_o = (\frac{V}{f_s} + x_j) \tan \Theta_L \qquad (2)$$

$$f_s = V/(\frac{D_o}{\tan \Theta_L} - x_j) \qquad (3)$$

where the largest obstacle D_o that can remain undetected is a function of robot velocity V, the tilt-angle of the laser scanner Θ_L, the scan rate f_s and the distance x_j required to separate the obstacle from the road.

V. CONCLUSION

In the study, it was seen that the sensors' sensitivity, sample rate and the related filters and algorithms have been prominent deterministic factors in the navigation and the mapping process. When the velocity of the robot has been increased, the mapping quality of the robot has decreased in a nonlinear manner due to the low sensor resolution and sensitivity as expected. The problems which can be encountered in a ROS based AMR during the design and implementation processes have been defined, and some solutions were submitted and realized. It is observed that the results of all the applied algorithms and submitted solution methods are reasonably, but it needs some improvements to minimize the obstacle avoidance and mapping errors.

REFERENCES

[1] R. C. Arkin and R. R. Murphy, "Autonomous Navigation in a Manufacturing environment", IEEE Transactions on Robotics and Automation, vol. 6, no. 4, pp. 444-454, August 1990.

[2] G. C.Buttazzo, Hard Real-Time Computing Systems, Kluwer Academic Publishers, 2002.

[3] S.F.A. Bender, P. J. Rodacy, R.L. Schmitt, P. J. Hargis, Jr., M.S. Johnson, J.R. Klarkowski, G. I. Magee, and G. L. Bender, "Sandia Report SAND2003-0184 Tracking Honey Bees Using LIDAR (Light Detection and Ranging) Technology", California, January 2003.

[4] M. D. Behera and P. S. Roy, "Lidar remote sensing for Forestry applications: The Indian context", Current Science, vol. 83, no. 11, pp. 1320-1328, December 2002.

[5] National Instruments, "Quadrature encoder fundamentals" http://www.ni.com/white-paper/4763/en/, 2008.

[6] National Instruments, "Encoder measurements: how-to guide" http://www.ni.com/white-paper/7109/en/, 2012.

[7] R. Siegwart, I. R. Nourbakhsh, and D Scaramuzza, Introduction to Autonomous Mobile Robots, Massachusetts Institute of Technology. The MIT Press, 2011.

[8] Paul D. Groves, Principless of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Artech House Second Edition, 2013.

[9] L.D. Styrcker, ECUMICT 2014: Proceedings of the European Conference on the Use of Modern Information and Communication Technologies, Gent, March 2014

[10] D. Bakken, "Middleware," in Encyclopedia of Distributed Computing , J. Urban and P. Dasgupta, Eds., Kluwer Academic, Dodrecht, The Netherlands, 2001.

[11] M. Quigley, K. Conley, B. Gerkey et al., "ROS: an open-source robot operating system", ICRA Workshop on Open Source Software, 2009.

[12] J.Z. Sasiadek and Q. Wang, "Sensor fusion based on fuzzy Kalman filtering for autonomous robot vehicle", Proceedings of the 1999 IEEE International Conference on Robotics & Automation Detroit, Michigan May 1999, pp. 2970-2975.

[13] C. Hajiyev, H.E. Soken, and S.Y. Vural, State Estimation and Control for Low-cost Unmanned Aerial Vehicles, Ch.4, p.51, Springer, 2015.

[14] S. Hutchinson, G.D. Hager, and P.I. Corke, "A tutorial on visual servo control", IEEE Transactions on Robotics and Automation, vol 12, no 5, October 1996.

[15] B.C. Kuo and F. Golnaraghi, Automatic Control Systems, John Wiley High Education, 2009, pp.528-532.

[16] J. C. Andersen, "Mobile robot navigation", PhD thesis, Tehnical University of Denmark, isbn.87-91184-64-9, 2006.