Design of an Autonomous Surveillance Robot using Simultaneous Localization And Mapping

Dr. Hameem Shanavas
Associate professor,
Department of Electronics and communication
MVJCE, Bangalore
hameem.shanavas@mvjce.edu.in

Muhammad Hameem Safwat Hussain, Syed Aatif Ahmed
Students, Department of Electronics and Communication
MVJCE, Bangalore
syedaatifahmed@gmail.com, safwatchamp@gmail.com

Abstract— **In this paper, the design as well as complete implementation of a robot which can be autonomously controlled for surveillance. It can be seamlessly integrated into an existing security system already present. The robot's inherent ability allows it to map the interiors of an unexplored building and steer autonomously using its self-ruling and pilot feature. It uses a 2D LIDAR to map its environment in real-time and HD camera records suspicious activity. It also features an in-built display with touch based commands and voice recognition that enables people to interact with the robot during any situation.**

Index Terms—Robot Operating System (ROS), Autonomous Mobile Robot (AMR), Simultaneous Localization and Mapping (SLAM), Light Detection and Ranging (LIDAR).

I. INTRODUCTION

Building and industrial areas are commonly targeted by thiefs and intruders putting the lives of humans or valuables at risk. Recently, robots have found to be useful to provide surveillance and prevent such unauthorized access from taking place. With the advent of mobile robots capable of solving complex tasks and achieving real time decision making capability,autonomous robot have become a norm.

Autonomous Mobile Robot is a robot that can navigate by itself through an unknown environment. The AMR can sense its environment,create a model of it and localize itself within this environment. This enables the AMR to further create a navigation plan and dynamically execute this plan using a planning and path finding algorithm. This is known as Simultaneous Localization and Mapping. In a majority of SLAM based algorithms, some differentiable features in the map are noted as landmarks. When a robot moves across the map,the distances from these landmarks is taken as the feature extraction values for the robot to localize itself.

In this study, an Autonomous Surveillance Robot which can map and indoor environment is designed and implemented.

The Electronic design procedure has been explained in detail and functionality like video stream using a webcam, video recording and speech recognition is implemented.

The software used is Robot Operating System working on top of Ubuntu distro. ROS is a framework to develop robotic applications. It consists of algorithms to build maps,navigate,interpret sensor data. It also supports hardware and software abstraction with an inherently distributed - split workload and is backed by a community of developers and researchers.

II. LITERATURE SURVEY

In [1], the paper introduces a particle filter algorithm and the design method for developing an intelligent patrolling robot based on the ROS architechture. Then, by implementing the navigation package of ROS, the indoor intelligent patrol robot is able to carry out autonomous navigation while avoiding obstacles in real time. By using the ROS platform,the development and working of the robot platform was realized very quickly by achieving the route planning and dynamic decision making capabilities.

The mapping process we have chosen to implement SLAM is Gmapping. In [2], the paper discusses implementation of Gmapping on ARM based embedded system. The mapping process running on the system can create a 2D map of occupancy from the laser and the pose data that is read by the mobile robot. The experimental results that were carried out show the accuracy of the mapping process.

In [3], a fruit mapping mobile robot on a simulated agriculture land was simulated on gazebo simulator to increase the efficiency and quality of the crop monitoring and their treatment. The paper proposed a means to generate a map in simulated area through SLAM and detection of the fruits using a visual sensor.

Once the mapping process is performed, global path planning for the mobile robot is required for autonomous operation. The Adaptive Monte Carlo Localization algorithm is used for this purpose. In [4], for the locomotion of the robot, a micro-controller has to be used which gives the motion commands to the robots motors.

[5] describes an autonomous Mapping System which is mounted on a platform consisting of and IMU and wheel

978-1-5386-7523-6/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDI3C.2018.00022

encoders. The robot is controlled autonomously using ROS and can autonomously map the test site. Based on the above papers, an autonomous surveillance robot for Simultaneous Localization and Mapping using ROS is proposed, designed and implemented.

III. HARDWARE DESIGN PROCEDURE

Figure 1 contains a block diagram of the system used in developing the project.

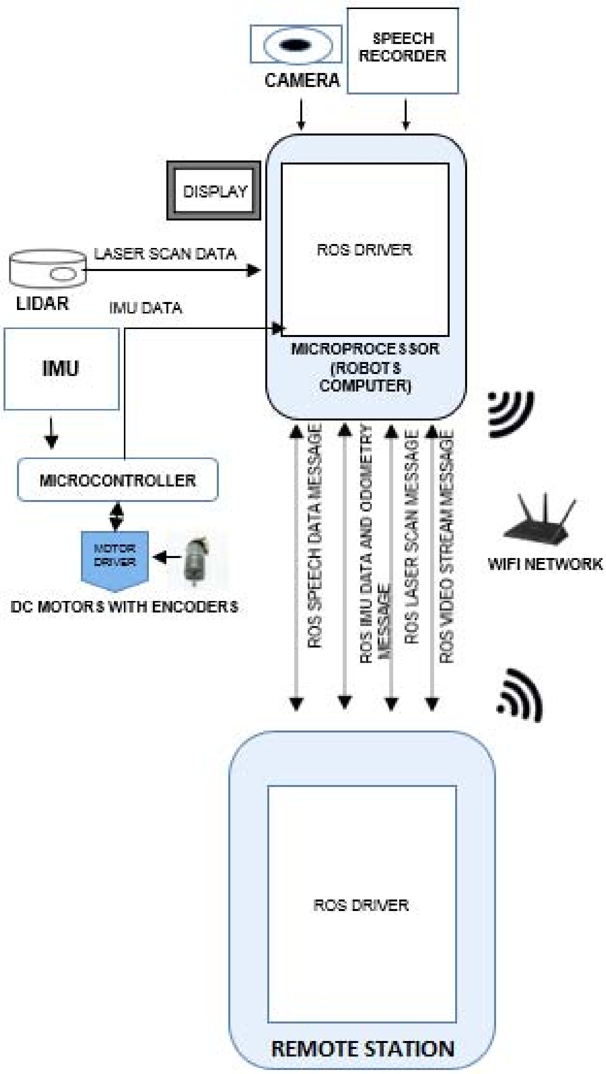

Fig. 1. Block Diagram of System

The main components of the system can be broadly classified into two groups, namely, Hardware and Software Components.

The main hardware components used in this project are:

i. Microprocessor

The microprocessor used in this project is the BeagleBone Black[6]. It is a lost cost development platform, commonly used in hobbyist projects. It has a 1GHz ARM Cortex A8 processor, helping it to process information at higher data rates. Its compatible with Ubuntu software, which can loaded into the microSD card. This acts as the RAM for the BeagleBone.

ii. Micro-controller

The Arduino Pro Mini is used as the micro-controller to control the motion of the motors. To do so, it takes feedback signals from the encoded motors and provides appropriate PWM signals back to the motor.

iii. IMU

The LSM9DS1 is a versatile, motion-sensing system-in-a-chip[7]. It contains a 3-axis accelerometer, 3-axis gyroscope and 3-axis magnetometer. Due to this, it allows 9 degrees of freedom in a single IC. It supports both I2C and SPI communication. It gives readings of acceleration, angular rotation and magnetic force in all the three axes, namely x, y and z. Using this data, it can measure acceleration, angular velocity and heading easily.

iv. LIDAR

The LIDAR is the acronym of Light Detection and Ranging. It is a surveying method used to measure distance from an object to the LIDAR[8]. It emits light in the form of pulsed laser and calculates the time taken by the reflected radiation to reach the LIDAR to calculate the distance. The LIDAR used is this project is the RPLIDAR A2[9].

The software components are:
i. ROS

ROS is Robot Operating System, the software framework used to write the robot's software[10]. It helps us in coding all the hardware components and interfacing them with each other easily.

ii. RVIZ

ROS Visualization (RVIZ) is the visualizer used along with ROS for displaying sensor data[11]. It can be used to build 2D and 3D models of robots. It helps is easier representation and analysis of sensor data.

IV. SYSTEM IMPLEMENTATION

First, the hardware components required are procured. Two DC motors with integrated quadrature encoders, having a counts per revolution of 120240, is used. Wheels of diameter 10cm are fitted to the motors. Therefore the minimum distance tick that can be detected by the encoder is 0.000083167cm. The RPLIDAR A2 is a 2D LIDAR with 360 degree scanning angle and a scanning frequency range of 10Hz. It has a rotation speed of 600 RPM, which can generate 4000 samples of distance data per second. A display unit of 15 inches is used to view the current status of the robot actions during its locomotion. The robot consists of a multi-layer platform as its chassis. The chassis is fabricated using acrylic, as its both sturdy and lightweight. The dimensions are decided and then, the acrylic sheets are laser cut to the required dimensions. The motors are attached to either side of the base layer's back portion using mounting clamps. A couple of castor wheels are fixed on the front end to balance the chassis. The second level of the chassis consists of a display unit mounted on it, along with a webcam. The webcam used is 5MP camera with built in microphone. The third level has the LIDAR mounted on top of it. The individual levels are separated by vertical metallic frames.

V. ELECTRONIC DESIGN PROCEDURE

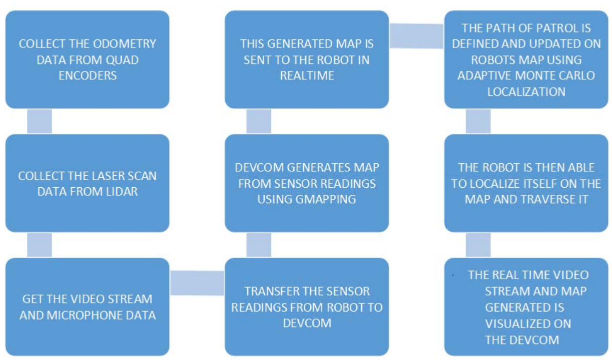

Fig. 2. Scheme of Electronic Design Procedure

Figure 2 above shows the electronics design procedure which mainly consists of 2 communicating systems. One is the robot's microprocessor, acting as the brain, and the other is a remote station system. The connection between the remote station and the robot is established over Wi-Fi through an intermediate wireless router.The encoder readings from the quadrature encoders are read by the microcontroller. The microcontroller sends this readings to the microprocessor of robot through USB interface. The IMU with 11 bit resolution, consisting of built in 3 axis accelerometer,gyroscope and magnetometer, is connected to the microcontroller, which is also communicating with the microprocessor. The 2D LIDAR is directly connected to the USB port of the microprocessor. The encoder readings and the LIDAR scanner readings are fused together to provide the odometry node for the robot. The odometry data, along with the IMU readings and the web camera video stream, is sent to the remote station. The remote station can stream the live video stream by subscribing to the 'image_raw data' stream on the RQT console. The map generated in real-time can be subscribed as a '/map' topic on the RVIZ console to visualize the LIDAR data-points on the map.

The BeagleBone Black, used as the robot's microprocessor, helps to process the sensor data. Ubuntu OS is installed onto the memory card of the BeagleBone Black. A swap memory of 2GB was allocated separately to handle the memory extensive ROS processes smoothly. A Wi-Fi module is connected to it and assigned a static IP address for ease of communication between the different nodes.

To implement SLAM, Gmapping algorithm is used, which is based on Rao-Blackwellized Particle Filter. The Rao Blackwellized particle filter combines the encoder ticks count and the LASER sensor readings to calculate the odometry and localize the robot during its movement.

The web camera used requires UVC camera driver to be installed to support it in Ubuntu. The parameters and properties of the webcam were fine tuned in its launch files for best performance and video clarity.

To provide speech recognition capability,the CMU pocket sphinx speech recognizer is used. The speech input is taken from the webcam's inbuilt microphone. It makes use of 'gstreamer' to split incoming voice signals into recognizable utterances and offers means to begin and end the recognition. The 'voice_cmd' script is used to control the robot using locomotion commands like forward and backward.

A GUI application is made, which consists of various locations in a given area, displayed by their names, on the touch based display unit. Each location is linked to the x,y coordinates in the map, which the robot creates. When the user touches a particular name, the corresponding coordinates are sent as a location goal for the robot to go to.

VI. RESULTS

First, all the hardware is tested individually and connected.

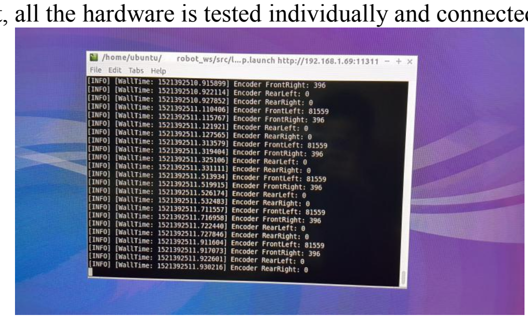

Fig. 3. Encoder Readings

Figure3 shows the encoder readings of the motors connected to the robot. The encoder readings increase for forward motion and decrease for reverse motion.

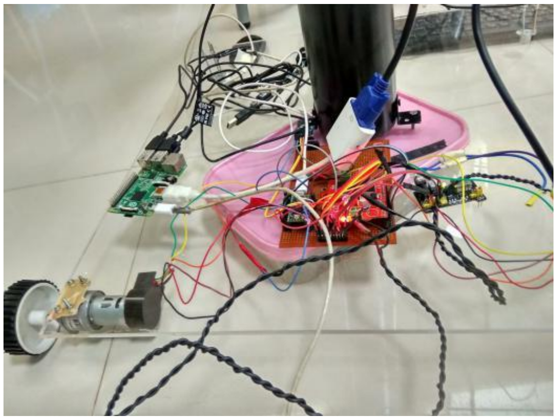

Fig. 4. Hardware Connections

The figure 4 above shows all the hardware components connected and being tested together.

Fig. 5. Completely assembled robot

Once the hardware connections are done, the software is tested.

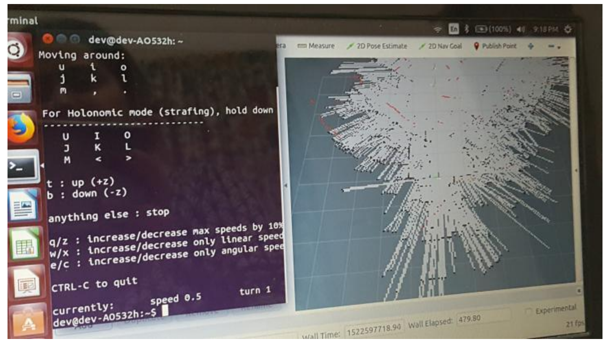

Fig. 6. Map building

Once the different soft-wares installed on the robot are tested, the robot is run around different territories to build the map. The map is built and dynamically saved on the robot's computer and live streamed to the development computer.

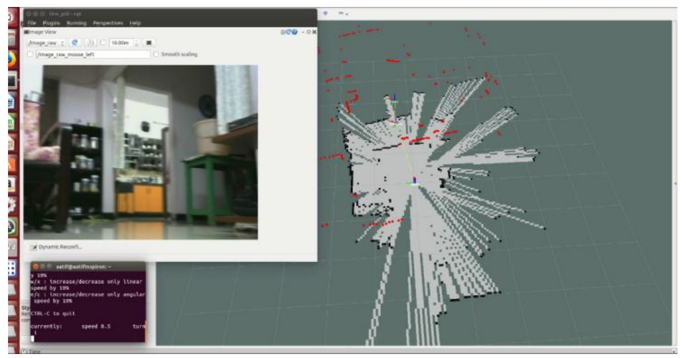

Fig. 7. Map and Live stream on Development Computer

Once the map is built and saved, the coordinates for navigation are given as inputs to the robot. The robot then starts its autonomous navigation, real time updating of the map and live streaming of the webcam video feed to the development computer.

VII. CONCLUSION

Once the complete setup is built, tested and checked for errors, it can be mounted on any kind of chassis, depending on the application being built for. Such systems can act as remote surveillance systems when the user is not present in the location of the robots. They can be used to monitor the security critical areas near borders 24/7 without any hassle and loss of precious soldier lives. Such systems can also be added to any prevailing security system to make it more robust and user friendly. It can be used in war zones and landmine detection to move in unknown territories.

VIII. REFERENCES

[1] Zhihua He,Lei Cheng,Wentel Zheng,Mengxia Sun,Qiuyue Yu,"Intelligent Patrol Robot based on ROS architechture",2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM).

[2] Qinjie Lin,Zhaowu Ke,Sheng Bi,Sirui Xu,Yuhong Liang,Fating Hong,Liqian Feng," Indoor mapping using gmapping on embedded system", 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO).

[3] Novian Habibie,Aditya Murda Nugraha,Ahmad Zaki Anshori,M.Anwar Ma'sum,Wisnu Jatmiko,"Fruit mapping mobile robot on simulated agricultural area in Gazebo simulator using simultaneous localization and mapping(SLAM)",Micro-NanoMechatronics and Human Science (MHS),2017 International Symposium.

[4] Zhengang Li,Yong Xiong,Lei Zhou,"ROS-Based Indoor Autonomous Exploration and Navigation Wheelchair",2017 10th International Symposium on Computational Intelligence And Design(ISCID).

[5] Matthew Young,Xiaoqi Chen,Chris Pretty,Stuart Ralston,Mathias Roehring,"Development of an autonomous robotic system for terrain mapping",2017 24th International Conference on Mechatronics and Machine Vision in Practice(M2VIP).

[6] https://beagleboard.org/black

[7] https://www.sparkfun.com/products/13284

[8] https://oceanservice.noaa.gov/facts/lidar.html

[9] https://www.robotshop.com/en/rplidar-a2-360-laser-scanner.html

[10] http://www.ros.org/about-ros/

[11] http://sdk.rethinkrobotics.com/wiki/Rviz