Environmental Monitoring Robotic System

Dr. Asad Yousuf, Savannah State University

Asad Yousuf is the Coordinator and Professor of Electronics Engineering Technology at Savannah State University

Mr. William Lehman, Bill's Robotic Solutions

William Lehman is President of Bill's Robotic Solutions which he started in July of 2013. He has had over twenty years of experience in software and hardware development. He has worked on numerous projects in digital communication systems, robotics, and aerospace applications. Mr. Lehman received his Bachelor of Science degree in Electrical Engineering in 1979 from Catholic University of America.

Dr. Mir M. Hayder, Savannah State University

Dr. Hayder is an Assistant Professor in the Department of Engineering Technology at Savannah State University, GA. He received PhD in Mechanical Engineering from McGill University, Canada. His research interest lies in the areas of engineering education, fluid-structure interaction, flow-induced vibrations, syngas and blended fuel combustion, and flow and structural simulations.
Environmental Monitoring Robotic System

Abstract

Robots are being developed and utilized as a fundamental data collection tool for environmental monitoring to meet the standards established by the Occupational Safety and Health Administration (OSHA). Monitoring of factors such as temperature, humidity, CO levels and etc. is a major concern to businesses, educational institutions, and health organizations because it can significantly impact the health, comfort, well-being and productivity of the individuals. Sensor data collected can provide the necessary Indoor Air Quality information to the designers and the maintenance crew to take the corrective measures to meet the standards of the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE). Furthermore, the environmental monitoring robot with automated machines and sensors can take place of humans to collect data in dangerous environments. To address the issues of environmental factors and to meet the requirement of Senior/Capstone design project course, students worked in a team to design, fabricate and test a mobile Robotic System to monitor and map data from the environmental sensors in a well-defined trajectory in an academic building. The Environmental Monitoring Robotic System consists of Inertial Measurement Unit (IMU) supported by the encoders on the Robot base to determine the location of the Robot and a Kinect® Microsoft Inc. camera to detect objects and obstacles in the trajectory. The proposed systems also consist of the Robot Operating System (ROS) and sensor section for monitoring the environment. The ROS section is used for mapping and localization of the mobile robot. The navigation stack in ROS is used for autonomous operation of the robot to move through a predetermined path to map data from the environmental sensors. The sensor section includes the humidity sensor, temperature sensor and a sensor for detecting the presence of different gases like CO, methane, propane and LPG. This paper discusses the design, fabrication, programming and implementation of a robotic platform to monitor the environmental changes. This paper also introduces an approach which enables students to work in teams to develop hardware/software system.

Introduction

Indoor Air Quality (IAQ) plays a significant role in our health and well-being. The Center for Disease Control (CDC) estimates the majority of Americans spends 90% of their time indoors. IAQ is in the top five Environmental Protection Agency’s (EPA) most urgent environmental risks to public health. IAQ problems are often associated with Heating Ventilation and Air-Conditioning (HVAC) system maintenance. If the building has improper temperature and relative humidity mold can grow. Volatile Organic Compounds (VOC) can be off gassed from building materials and use of cleaning and disinfecting compounds. Overcrowding can lead to excessive CO2 levels and are also a indicator of improper ventilation that can lead to other IAQ problems. Presence of outdoor air pollutants, radon and CO from cars in attached garages can present problems to IAQ. Human activity can also change the IAQ over time and space, for example with cooking, eating and cleaning or using printers [1, 2].
Software and Hardware Design of Mobile Robot

The design of the a Environmental robot to detect IAQ is a daunting problem due to the large number of separate sources of organic and inorganic compounds that present problems for human health. Developing a mobile robotic platform that can be controlled both manually and autonomously is also a significant challenge. We dealt with the first issue of sensors for the robot to include only the bare basics, of temperature and relative humidity. We also added a gas board that was used to detect CO levels in the building. We decided to use the Open Source Robotic Operating System (ROS) to develop our mobile robot. ROS provided most of the software for a mobile robot. ROS provided software to map the building, navigate the building using the map both manually and autonomously. Once the robot collected Environmental data, the data could be analyzed over time and with respect to position on the map.

![Figure 1. Hardware Overview](image)

Figure 1 gives an overview of the system. The Base Station laptop is used to monitor and in manual mode control the robot. The robot consists of a small form factor laptop mounted on the robot base. Both laptops are running Ubuntu Linux 16.04. The ROS version running on both laptops is Kinetic. A Kinect 3D camera from Microsoft is fixed to the top of the robot. Communication between the Base Station laptop and the laptop on the robot is thorough a wireless access point. Since ROS is a distributed framework, the software for the robot’s operation is running on the Base Station laptop and the laptop mounted on the robot base.

A ROS system consists of a Master/Parameter Server and other ROS programs called nodes. The nodes communicate with the Master, Parameter Server and other ROS nodes though the network. There is only one Master and Parameter Server for a ROS system. The computer running the master is identified with a linux environment linux variable – ROS_MASTER_URI. ROS_MASTER_URI must be defined on each computer used in the ROS system. In our ROS system the Master ran on the robot laptop, in our case with a URI of [http://10.0.0.48:11311](http://10.0.0.48:11311). Port 11311 is used by default for communication with the Master. The other environment linux variable ROS_IP is set for the computer we are running the ROS node on. In our case that was 10.0.0.47 for the Base Station. SSH was used to run ROS software on the robots laptop.
(10.0.0.48) using connections via ssh robot@10.0.0.48. It was difficult at best to type commands on the robot laptop due to smaller keyboard and access to the laptop when mounted on the robot base. We accessed the laptop on the robot using only ssh.

The laptop mounted on robot is a 16 bit Atom processor, while the Base Station was a 32bit AMD V20 processor. How do we balance the load on each laptop? The answer to the question is dependent upon the hardware interfaces for the laptop mounted on the robot and the robot’s Mode of Operation.

**Modes of Operation**

The robot has five Modes of Operation:

1. Manual Control – Map Building
2. Off-Line Map Processing
3. Manual Control and Data Collection
4. Autonomous Navigation and Data Collection
5. Off-Line Data Analysis

The RVIZ robot simulator, robot state publisher and Navigation node were all run on the Base Station Laptop. The Master/Parameter server, freenect stack for the Kinect 3D Microsoft camera, robot base node and depthimage_to_laserscan node were run on the robot mounted laptop.

**Assignments**

There were ten students in the Senior Capstone section. We strongly encourage group work for our students to encourage good communication skills. The students were offered a variety of projects to work on, or submit a problem they wanted to work on. All the students wanted to work on the ESSUR robot.

Using ROS for the ESSUR had the advantage in that it has all the programs aka ROS Nodes to support all the Modes of Operation for the ESSUR to function. However three custom ROS packages needed to be developed along with embedded software running and the Arudnio micro-controllers, hardware and mechanics. We defined five groups for the students:

1. System Test
2. Robot Base
3. Environmental Sensors
4. Inertial Navigation Unit (IMU)
5. Part Procurement

With only students involved in the first semester of the project multiple students had more than one group assignment. Students provided us there primary areas of interest such as mechanics, hardware, software, test or ROS nodes.

We assigned students to balance the work load for each student on the ESSUR project. The
Professor and a mentor from industry who had ROS experience provided the management of the ESSUR team. The Professor evaluated and graded status reports, and other work products and deliverables from the students. All the students knew C++ and had micro-controller experience. Students for the most part had Linux experience. The mentor from industry provided technical support and scaffolding to students especially for ROS as directed by the Professor.

Weekly meetings provided time for training on how ROS systems function. Halfway through the semester we stopped weekly meeting which gave way to one-on-one training, or assisting small groups of students. Weekly status reports were sent each week by students with a simple format for accomplishments, next week’s goal, problems and assistance needed. Status reports were reviewed and discussed at the weekly meetings. Any deliverables with sections required for that week were submitted along with status report.

The mentor was asked by the Professor to define the serial interface format and commands between student groups for the Robot Base micro-controller and ROS node for the Robot Base and Environmental micro-controller and Environmental ROS node. The students had the option of modifying the suggested interfaces before accepting them. The mentor also provided scaffolding on the afore-mentioned ROS nodes interface to the ROS system to ease the learning curve of students for ROS.

Three rubrics, shown below, were used to judge Preliminary Design report and presentation as well as Final Report and presentation.

**ISL0 1: Written Communication**
- Organization of Project Report 30%
- Circuit Diagram, Software, Mechanical Drawings, Test Scripts 20%
- Results and Analysis 50%

**ISOL 2: Critical Thinking**
- Concept 10%
- Design 50%
- Hardware and Software Design 40%

**ISOL 3: Information Literacy**
- Determine the information needed 20%
- Access the needed information 30%
- Use the information effectively 50%

**System Tests**

Testing is a major task in complex systems. Although many of the ROS nodes have already been well tested, combining them into a ROS system for a custom robot is not trivial. ROS nodes have several communication methods using sockets to communicate with one another over the Internet. Topics are a one node to many node communication methods. Every Topic has a unique name which is associated with a message format. Nodes can be the source to the ROS system for multiple Topics and the same node can also receive many ROS Topics.
Every ROS system has a Master Program that acts as a Name Server between nodes. When executing ROS nodes it was difficult in some cases to set the node parameters to the correct Topic name. Many of the ROS nodes needed by the mobile robot also had dozens of parameters some of where default values were not always acceptable. Most of the solution was extracted using material on the subject of ROS mobile robots and Turtlebot ROS launch files [3, 4, 5].

There was a System Test developed for each mode of operation for the robot. Making students formally write out and document the tests proved invaluable to the project. A number of students in the project were surprised to learn that there were engineers in industry whose job was test.

We did not insist on formal written tests for the Environmental embedded software or the Robot Base embedded software. We asked students to demonstrate to us each command in the interface specification they agreed too, by manually sending commands through the serial interface. The Arduino micro-controller has a serial monitor interface which was used to test the interface.

The IMU as it turned out already had a ROS node written for it on the micro-controller using the ROS Serial node [6]. ROS Serial node provides ROS node interface for the Arduino micro-controller as well as other micro-controllers such as the MBED micro-controller [7].

**Robot Base**

We used the VEX Robotics parts to build our custom Robot Base. The curriculum for VEX contains Units 7-9 give a brief introduction to designing a Robot Base’s mechanical design [8]. Since our students are Electronic Engineering Design (EET) students, we had them focused on the DC motors and encoders used in the design.

The criteria for the Robot Base design required the Base to hold a laptop, mount a 3D Kinect Microsoft camera, a 5 ft boom to place three temperature sensors, an area to hold the electronics for Environmental sensors and encoders on the wheels. The Robot Base also was required to have a bumper switch on both the front and back of the robot. The final design of the Robot Base used two VEX PWM controllers and DC motors and encoders [9]. The mechanical design of the Robot Base was a differential drive with four wheels, with the motors/encoders driving the rear wheels and the front wheels were omni directional.

We used an Arduino Uno as the micro-controller for the Robot Base which meant we only had two interrupt digital ports to work with. The VEX encoder had two interrupt lines to detect both speed of the axle and direction. We needed two VEX encoders one for each of the rear wheels. Using only one of the two interrupt lines from the VEX encoder meant we could only detect speed of the axle and not direction. The robot was assumed to move as directed with the PWM signal sent to the motor to move forward or backward. This assumption is good provided the robot is not slipping due to external forces on the robot.

A ROS node running in the PC mounted on the Robot Base also needed to be developed to interface the Arduino Uno to the rest of the ROS system for the mobile robot. The ROS Base node translated Command Velocity topic (cmd_vel) to a serial command to set each of the two DC motors to the corresponding velocities. The Base node must also translate the encoder values
connected to each DC motor to determine the position and actual velocity of the robot using dead reckoning.

In the final design and build of the Robot Base the laptop was mounted on the top layer of the rectangular Base and the electronics were placed in the middle layer. The boom was attached to a corner of the Base for stability.

**Environmental Sensors**

The Environmental Sensors system consisted of three temperature sensors, a relative humidity sensor and a Carbon Monoxide (CO) sensor. The Arduino Uno board was chosen for the microcontroller. As with the Robot Base Arduino Uno board communication with the robot mounted laptop was through serial to USB connector built into the Arduino Uno board.

The first temperature sensor was mounted at the lower end of the boom. The second temperature sensor was mounted at seat level. The third temperature sensor was mounted at the upper end of the 5ft boom. The code for the relative humidity sensor (HS1101) was implemented as a state machine with the following states:

0) Wait for humidity reading request  
1) Charge RC circuit (sensor is a variable capacitor)  
2) Wait for RC circuit to discharge (interrupt occurs)  
3) Calculate relative humidity  
4) Reset for next reading

The capacitance of the sensor changes due to changes in humidity. The resistor is 330 ohms, the sensor is a variable capacitor, so the time of decay of the RC circuit determines the relative humidity. The RC circuit is charged by setting a high on a interrupt digital line and interrupt occurs when the line goes low [10].

The gas detection board also is implemented as a state machine, to cycle the heater to the gas sensor on and off before taking a reading. The gas board only indicates if the level of the gas is beyond an acceptable threshold. We planned to use the gas board to initially detect Carbon Monoxide levels (CO). The gas detection board parts did not arrive on time so we were only able to take temperature and relative humidity readings [11].

A ROS Environmental node had to be developed and run on the PC mounted on the Robot Base to read the sensor values from Environmental Arduino Uno through the serial port. The sensor data was combined by the node with position data and time from the Odometry (odom) topic and written to an Environmental data file. The Environmental data file was analyzed off line with tools such as Microsoft Office Excel.

**Part Procurement**

For previous Senior Capstone projects, the students in each group were responsible for determining parts and ordering them. Due to the high cost of the robotics project we used a grant
that was available to students to obtain parts for student projects. We assigned one student to maintain the parts list and do the work submitting the parts to be filled through the grant. The student assigned did their job, but delays with the grant meant the IMU and encoder were never filled. Missing the encoders and IMU meant we could only partially test the Robot Base.

We worked around the Robot Base problem by having the students test the mechanics of the base only using the DC motors. The students from the Robot Base group were assigned to write the System Test plans since they had more time due to an incomplete Robot Base. The Robot Base group tested the Robot Base ROS node on the PC with simulated data from the embedded Robot Base Arduinio.

![Image](image1.png)

**Figure 2. Robot Base using iRobot Create 1**

We had an iRobot Create 1 Robot Base laying around which we used for the ESSUR, see Figure 2. We had to use the Create 1 ROS driver to control the Base. We modified the Create 1 Base to make it rectangular for our tests.

In Figure 2 the black Kinect 3D Microsoft camera is mounted on the top of the Robot Base. The white front of the iRobot Create 1 can be seen at the bottom of Figure 2. The laptop was mounted just above the iRobot Create 1. The Robot Base was designed so the laptop could ride the robot while open, to monitor the system. The boom which is normally attached to the back left corner of the robot is not shown in Figure 2. The boom has the three temperature sensors mounted that connect to the Arduinio Uno's analog ports. The humidity sensor and CO sensor are mounted on a breadboard behind the Kinect 3D Microsoft camera. Batteries for the Kinect 3D Microsoft camera, humidity sensor, CO sensor and Arduinio Uno are located on the top of the robot mounted behind the camera. The Kinect 3D Microsoft camera required 12V DC which was provided by two AA and one nine volt battery.
Conclusion

Usually for Senior Capstone projects we have teams of two or three students. The students on Senior Capstone projects in the past were responsible for buying any parts not available in the Engineering lab. The ESSUR project required more expensive parts and had ten students assigned. There were problems getting parts in a timely manner for the project. We used a iRobot Create 1 as a base to test much of the ROS system for a mobile base. We also tested the custom Robot Base mechanics under manual control and the ROS Base node through simulation. The iRobot Create 1 was not located at the University and was controled and tested by students using PuTTY and in linux with Secure Shell (ssh).

The software and hardware developed by the ESSUR team is being used as a starting point for other robotic projects by Senior Capstone student projects.

References


