Senior Design
ENG EC 463

Memo

To: Professor Pisano

From: UPR Level Security: Juan Peralta, Eugenia Almandoz, Namir Fawaz, Brock Guzman, Brandon Webster

Team: 24

Date: 5/30/18

Subject: Final Deliverable Test Report

1.0 Test Objective and Significance

1.1 Remote Control

The significance of remote control is to essentially give remote users a way to take control of the robot if needed. It provides them a way to override autonomous navigation if they decide to navigate the robot to a specific location instead of it roaming around the substation. Since electrical substations pose unsafe conditions for people this is a useful function to have in order to check specific equipment or navigate to a desired area for urgent situations.

1.2 Gauge Detection/Gauge Readings

Gauge detection plays an important role in keeping the electrical substation as safe as possible and targeting problem equipment so that it can be dealt with accordingly before it has more serious effects. The purpose of this test is to show the full functionality of gauge detection and gauge readings working. This includes having our robot detect a gauge (clock in this case) through the raspberry pi camera while navigating. Then once the robot has detected the gauge it must move close enough to read the angle on the gauge. This will be used to take measurements on all the gauges around the substation because the robot will need a way in order to measure the status of all the equipment in the area.

1.3 Object Avoidance

Object avoidance is used mainly to complement navigation while the robot is on its way to its location. Its main purpose is to detect objects directly in front of it and calculate which way the robot must turn in order to avoid the object. The robot has an

ultrasonic sensor mounted on a servo at the front and based on the direction in which it encounters an object and the destination the robot has it will calculate the angle at which it should turn in order to avoid the object as well as still be on track to reach its destination.

1.4 Navigation:

Autonomous navigation is the most important aspect of our robot and serves as the basis to get all other aspects working as well. The significance is to get the robot from point A to point B while making certain decisions based on its current location. This is important because in order to surveill the substation our robot must be able to navigate itself throughout it while making smart decisions based on its target location and while maneuvering around objects that are in its path. For this test we wanted to show that based on a starting location and a given destination our robot was able to travel there while avoiding objects in its way.

2.0 Measurements and Data

2.1 Remote Control

Remote control works with our web application and will have its own page which has live stream video in order for the remote user to know where the robot is heading and in which way it should turn to go where it wants to go. Remote control works with 6 keys (W,A,S,D,Z,C) which correlate to straight forward, forward left, straight backwards, forward right, backwards left and backwards right. The escape key is also used to send the robot a command that will end remote control and return to autonomous navigation.

2.2 Gauge Detection/Gauge Readings

This is a built in opencv function and by adjusting the parameters we were able to get a reliable detection distance of 2 meters of a 10 inch clock, which we used since it had similar characteristics as the desired gauge.The GNSS system we used for localization was able to achieve distance of around 2 meters which worked for demo purposes, but on a substation environment we would have needed to be able to successfully implement the GNSS with RTK corrections, which would have provided accuracies up to 1cm. To be able to read the value of the gauge, reading the angle of the needle would need to be done. We accomplished this by using opencv function HoughLinesP(). Once the gauge was detected, we perform the transform on the ROI. Using the coordinates of the line output given by the function we are able to know the angle of the line. We were able to get reliable readings within 30 cm. From knowing the angle of the clock you would then need to have a function that transcribes these angles to actual values. For a clock you know each hour is 30 degrees from each other. Given more time we would have easily implemented this aspect so that the alert that is seen on the web application also gives the actual reading value of the “gauge”.

2.3 Object Avoidance

The servo moves 180 degrees and will take 12 measurements throughout this 180 degree turn. The robot essentially drives straight until it reaches an object in its path(an object that is within 0.5m in front of it). It will then take 13 measurements to detect the closest free space that will allow it to continue moving straight. We are essentially looking for the smallest angle from 90 degrees(which is straight ahead) to continue travelling and in turn get the robot to its destination in the smallest amount of time. Once it finds that direction it will continue driving in that direction but turn back to the position in which it was travelling before in order to get back on track to its destination.

2.4 Navigation

The GNSS-based navigation takes measurements from a U-Blox m8 GNSS Evaluation Kit connected to a Garmin antenna* and compares it to a known starting location of the robot (assuming it starts in the same spot each time), then corrects it with known distances, such as its known location subtracted from its measured location, the known location of MassDOT towers and the location received from the robot's ntrip connection to the MassDOT site. The robot makes the same corrections with known locations of gauges throughout the station. With these new values the robot makes a calculation to find the shortest path to the gauges, using its initial orientation (hard coded based on site specifications and translated to work with the MPU6050) to calculate how far in which direction the robot needs to turn to travel in the shortest path to the goal location. The robot is then signaled to turn slowly until the MPU6050 measures that it has turned a proper amount and the robot will proceed straight. If for any reason the MPU6050 measures anything different than the proper angle, the robot will receive signal to correct itself.

3.0 Equipment and Setup

3.1 Remote Control

For the remote control aspect of our robot we will be controlling it via the keyboard keys and we will track which direction it wants to go based on the key that has been pressed. When the key is pressed down, a begin command is sent to the robot using an Ajax request pointed at the server running on the Raspberry Pi. When the key is released, a stop command is sent using the same method. The raspberry pi will be serially connected with the arduino and will send the string over to the arduino in order to control the direction of the robot accordingly. Each key will correlate to a certain wheel direction making it travel as desired.

3.2 Gauge Detection/Gauge Reading

Once the robot reaches the GPS destination gauge detection is activated. Since the gauges that we would be looking for would be circular, and other circular object would not be near the desired gauge, we used the Hough circle transform for gauge detection. An actual gauge would be smaller than that size which means the reliable distance would fall. For differentiating between quadrants given the angle, you can easily just look at the location of the line coordinates given by the function and know which quadrant you are in, which means you can differentiate between for example 12 o'clock and 6 o'clock.

3.3 Object Avoidance

For the object avoidance technology we used a dynamixel ax-12a in order to mount an ultrasonic sensor for object detection. The servo is triggered when the ultrasonic sensor detects an object 0.5 m away from it at which point the servo begins to scan from left to right to take all the measurements between that range. To demonstrate this we used a cardboard box placed in the path of our robot and and had it avoid it and then readjust to get back on track to its destination. The robot stops momentarily while it takes all required measurements as well as calculates the angle it needs in order to avoid the object accurately and effectively.

3.4 Navigation

We demonstrated this by putting the robot down on an initial location and placed both a gauge which will stand as the final destination as well as an object in its path to this location in order to demonstrate object avoidance. The robot will calibrate for about a minute before beginning its trajectory. The arduino will handle the movements of the robot based on given parameters from the raspberry pi. The arduino controls the motor as well as the wheel servo which is how the robot knows how it should turn in accordance to the angles. For the test the robot will be set to start at the same location and will have a destination of 9 m east in order to demonstrate functionality.

4.0 Conclusion

4.1 Remote Control

What we tested for remote control for the demo is what we will be providing our customers with. This includes the remote control page on the web app as well as included functionality with for remote control using the 6 keys on their keyboard. Originally, we wanted to implement remote control with a joystick connected to the computer, this would make it much easier for the user to control the robot and navigate it more precisely to exactly where it wants given the precision of accuracy using x and y coordinates however, we were not able to add that functionality with the time we had. Our solution still is able to accurately drive the robot around a designated area and will work well in a substation setting.

4.2 Gauge detection/Gauge readings

For now we have our gauge detection working with just a regular analog clock. We also would like to modify this to work with concentric circles instead of just regular circles in order to avoid many false positives when testing our robot outdoors.This is a proof of concept to show our client how gauge readings can be taken with an autonomous robot however, in order for it to work in the substation we would have to get a model of their gauges and what they look like which we were not able to obtain. The website will show an alert when the gauge is detected as well as display the angle of the needle on it. This functionality in a substation setting would help the remote users know the

measurements of the equipment to understand the status of each and hopefully help them detect faulty equipment before it becomes a hazard to anyone or anything.

4.3 Object avoidance

For now, object avoidance is only working with one ultrasonic sensor placed at the front of the robot. There are points where only having one sensor does not allow us to detect all objects that are going to affect the turn of the robot. For example, if the robot found itself in a corner where it detected an object at every point between our 180 degree range then it would not be able to accurately navigate itself out of that. Therefore, we may look into adding a couple more sensors angled towards the wheel of the robot in order to make sure there are no other objects affecting the robot's route or turn. Along with that, adding sensors to the back end of our robot would allow us to implement the same algorithm we had for the front of the robot to the back in order to be able to back out of corners.

4.4 Navigation

We have the coordinates of the two gauges preset at the beginning so the robot can know where to navigate to. As the robot navigates it will use GPS longitude and latitude values to compare it to the desired location to know whether it has reached the gauge or not. Eventually this will be updated to having all the locations of the equipments we need to measure and the robot will then navigate to each one and measure the gauges accordingly.

Ideally with more time we would have liked to extend this into the functionality of our web app so our client could manually insert all the desired destinations(where the gauges were located in this case) and it would then loop through them to send the robot to each location. This would make adoption for different areas much easier, so that even if it was not a substation the robot would be able to know where to go based on the preset destinations. This is also useful since the locations of these gauges do not change so unless another gauge is added it would only need to be defined once.

Finally, with GPS we were only able to get so precise with our locations and therefore the robot had an error of about 2m which if in a substation would not be precise enough. Our solution provides our client with functionality to autonomously navigate through an area but given the range of error with GPS other solutions may have been able to yield more accurate results.