Senior Design
ENG EC 464

UPR

Memo

To: Professor Pisano

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

Team: 24

Date: 4/3/18

Subject: Final Deliverable Test Plan

1.0 Remote Control

1.1 Goal: The goal of remote control is to have the ability to have a remote user take control of the robot and be able to drive to where it is needed. This is important for the remote operator in the case that it wants to send the robot to check specific equipment or surveill a specific area, without them needing to be inside the substation. The operator will be able to simply control the robot using 6 keyboard keys and using the camera on the robot it will be able to know where the robot is and in which way it is headed.

1.2 Procedure: To demonstrate the remote control module of the robot we will interrupt the normal operation of the robot with commands from the AWS web page. As soon as the keys are pressed on the remote control page, a flag will be sent to the Arduino that will turn the remote control operations on that take the key commands from the web page

1.3 Result: The remote control will work with our web application and will have its own page where video will be streaming from the camera in order for the operator to know where the robot is heading and in which way it should turn to go where it wants. What we will be testing for remote control for the demo is what we will be providing our customers with. Originally, we wanted to implement remote control with a joystick connected to the computer, we may look into implementing this if it deems to be a simple task.

2.0 Gauge Detection/Gauge Reading

2.1 Goal: The goal of gauge detection is for the robot to be able to detect a gauge using the camera while the robot is navigating. Since we were not able to get pictures of the gauges in their substation we used a clock to represent the gauge. 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.

2.2 Procedure:
When the robot has reached destination, which means the raspberry pi signals the arduino, the robot will scan its camera. While scanning, the raspberry pi will perform Gauge Detection. Once detected the robot will center the gauge in the frame in order to get the turn direction it will need, and start travelling until it gets to within 30 cm of the gauge. Then the raspberry pi will perform Gauge Reading. After this, the robot will wait to get its new target location.

2.3 Result: For now we have our gauge detection working with just a regular analog clock. This will be a proof of concept for the gauges in the substation. 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.

3.0 Object Avoidance

3.1 Goal: The goal of object avoidance is for it to compliment navigation for when the robot encounters an object in its path. The goal is for the robot to have 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 in track to reach its destination.

3.2 Procedure: For the test we will show object avoidance by not only detecting the object when the robot encounters it but then accurately avoiding it with the right angle provided. Object avoidance works with the ultrasonic sensor mounted onto a servo which sweeps 180 degrees from directly left to directly right of the robot. The sensor takes 13 readings within this range when an object comes within 50 cm directly in front of it. Any reading that is more than 150 cm is what it considers as “free space” and then calculates in which direction the robot should turn to then avoid the given object. Once it has avoided the object it will get a new turn angle from the raspberry pi which will take it pack to a new path towards the target.

3.3 Result: 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. 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.

4.0 Navigation

4.1 Goal: The goal for navigation is to show that we have a way to calculate a bearing for the robot given its current gps location and a goal gps location. The raspberry pi gives the robot a turn angle and a direction(left or right) based on its current angle/position and its goal location. For the test, the raspberry pi will give a preset turn angle and direction when the robot initially starts and will also give it preset turn angles when it has avoided an object. The robot will then navigate in that direction until it has reached a specific equipment/ the gauge or clock in this test case.

4.2 Procedure: We will demonstrate this by putting the robot down on an initial location and will place 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.

4.3 Result: 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.