Weekly Report - Tuesday, April 9, 2019

Auburn University IEEE SoutheastCon 2019 Hardware Competition Team

Current Major Development Tasks

Task Name	Category	% Compl.	Progress Updates
ROS Localization	Software	99%	AMCL fully working. Along with all its inputs(LIDAR, ODOM, Encoders, tf). Slight tuning may be needed to increase accuracy.
Integrate Encoders in ROS	Software	100%	After much difficulty, this has been achieved.
Fabricate, 3D Print, Assemble, and Wire New Robot	Electrical	99%	Everything that is mounted with the exception of a newly added protoboard board.
Differential Drive Control Algorithms	Software	100%	The control of the drive motors has been integrated in ROS and the PID loops have been tuned.
Driving in circles Algorithm	Software	75%	Algorithm exists, but needs to be tested. (Happening Tuesday morning)
ROS integration with 3D vision and Decision Making Intelligence	Software	100%	We can now return x and y coordinates of the objects in meters with respect to the robot. These values are used to tell the robot to move to that new location to collect the object. We also did some tuning to reduce lens distortion and improve the color thresholds.

Senior Design Team Members Time Management

Member Name	Task Name	~ Hours Spent (past week)

All Members	Team Meetings	3
Matthew	General team planning and support	12
Matthew	Mechanical updates	5
Matthew	Electrical wiring improvements	2
Matthew	Visual Algorithm Improvements	4
Nia	Servo and Motor Control	5
Joe	Navigation Stack	2
Josh	Navigation Stack, Electrical Improvements, Serial Comm Improvements	30

Tasks to be Accomplished Before Competition

Task Name	Category	Assignee
Calibrate to odometry with lidar localization	Software	Alex and Noah
Fuse AMCL with odometry	Software	Alex and Noah
Navigation stack (move in circles) and set servo positions	Software	Josh, Nia, and Joe
ROS integration with 3D vision and Decision Making Intelligence	Software	William and Matthew

Achievements, Obstacles, and Risks

We are in the final stretch of preparing for the competition. Unfortunately, due to the complexity of ROS and localization, we were later than we would have liked in terms of completing the integration of all of the aspects of software. Below is a summary of the major issues that we encountered and the solutions that we have come up with in order to solve them.

- The LIDAR has a cone of vision rather than a finite point. This means that if the LIDAR is mounted just below the top of the field it will see above the field. We had to mechanically lower the sensor so there is no way for the LIDAR to see above the field. In order to accomplish this, we had to redesign the mount and lower the electronics plate. Testing has proved our solution works as needed.

- The encoders were originally connected to the Raspberry Pi. Due to the nature of encoders, they must constantly be polled so that ticks can be converted to angular velocity. After developing a program to accomplish this, in practice, it used an entire core of the Pi's quad-core processor. We had a Teensy microcontroller laying around the lab. We programmed the Teensy to use its interrupt digital I/O pins to keep up with the encoder ticks and update the Pi through a serial port. We then integrated this data with ROS. Based on testing, the encoder data seems to be quite helpful in reducing drift from the LIDAR data.
- We integrated the serial program we wrote earlier in the semester with ROS. We found a differential drive ROS library and adapted it to control our motors. We tuned the PID algorithm to match the performance of our motors and limit the acceleration and deceleration to not flip the robot.