Final Report

RBE 2002 D01

Authors:

Teresa Saddler **Luke Trujillo** **Benjamin Ward**

Submitted On: May 1, 2019

Abstract

An autonomous robotic system was designed and constructed to locate and extinguish a fire in a apartment simulacrum, constructed of plywood, at various heights. The robot was required to respond completely autonomously - i.e. to receive no external input or information about field or robot status - and to report fire location to the user.

An air vortex cannon was selected as method of extinguishing the fire due to its ability to put out a fire on either floor without angle actuation. A drivetrain with two-wheel-drive and two ball castors was chosen for controllability, since it neatly centered the virtual turning center of the robot in the exact center of the robot, allowing the robot to turn on a dime. For both mechanisms, transmissions were selected to maximize power delivered to the output and ensure controllable linear and angular speeds.

Total power draw was restricted based upon physical battery limitations, and total operation time was limited to five minutes.

Overall robot design and construction proceeded taking into consideration physical constraints, material and component cost, and cost of machining time. Material and machining costs were minimized via use of simple 2D and 3D CNC fabrication methods (laser cutting and 3d printing).

Programming proceeded by taking into account the overall task requirements, utilizing a flow diagram to organize common tasks into repeatable code blocks. Various sensors, including encoders, an inertial measurement unit (IMU), and several rangefinders were used to supplement and integrate knowledge about the robot and the robot's environment into the system, enabling controllability. A* pathfinding and other algorithms enabled procedural decisionmaking.

Successful robot completion did not occur in the time period of the project, due to a number of systems and components breaking or behaving unexpectedly and other unforeseen circumstances, however many aspects of the robot were completed and demonstrated including pathfinding. Subsystems were successfully completed but not completely integrated.

Table of Contents

List of Figures

List of Tables

1 Introduction

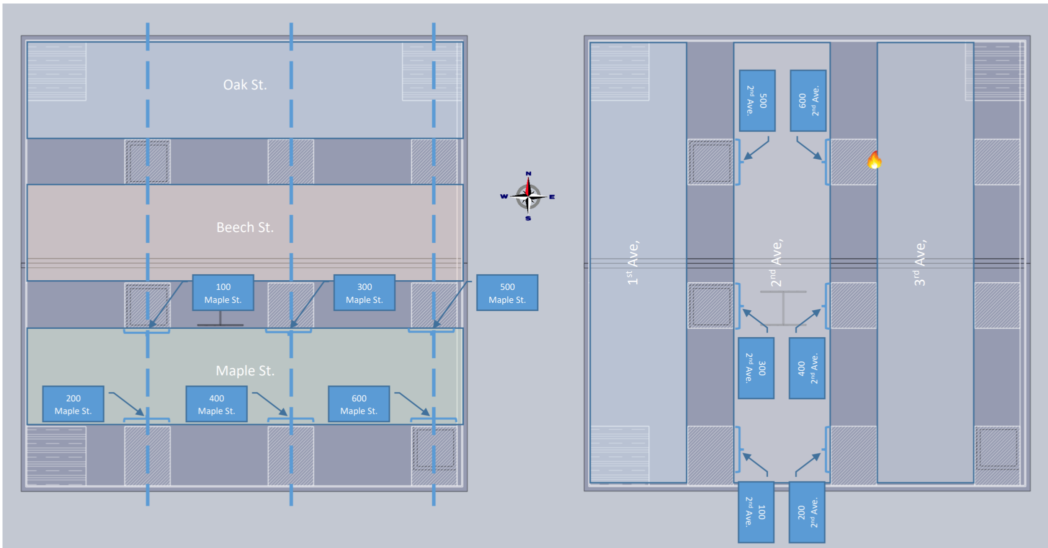

Figure 1: The full challenge field with possible starting locations marked α, β and γ and possible building locations marked A-I

In this project we were required to design and create an autonomous robotic system which efficiently navigate to, identify, and extinguish a fire, in the form of a lighter in a "window" of a model apartment building. Three apartment buildings were distributed randomly in any of the nine possible locations on the field, indicated by the letters A-I in figure 1, separated by streets, and apartment building each had two windows on each of its four faces, one above the driving surface of the field, and one above the driving surface of the field. There would only ever be one fire placed and it was guaranteed to be in a location not in the line of sight of the robot at the beginning of the demonstration period, and in a location physically reachable by the robot. The robot was permitted to start in any one of the three fire stations, indicated by α, β and γ in figure 1, but this location was chosen before placement of the apartment buildings and the fire. Additionally, a roadblock may be placed in one of the streets (but not in an intersection) to prevent robot traversal through that point of the field. The field was enclosed by a border, except for locations where there was a "cliff". In front of any cliff on the edge of the field, a line of white tape would be placed.

The robot itself was required to start in a configuration in which it would fit into a 13 inch cube, but could expand during the demonstration to a maximum of 19 inches in any direction. The robot was required to include an IR camera, a range finding sensor and an IMU. The robot had to operate autonomously and its driving was restricted to the streets. When it located the fire, it had to report the fire's location by it's (x,y,z) coordinates, and preferably the street address of of the apartment building. The scheme for these street addresses can be seen in figure 2. After extinguishing the fire, the robot should indicate that the fire is extinguished and optionally navigate back to the starting location.

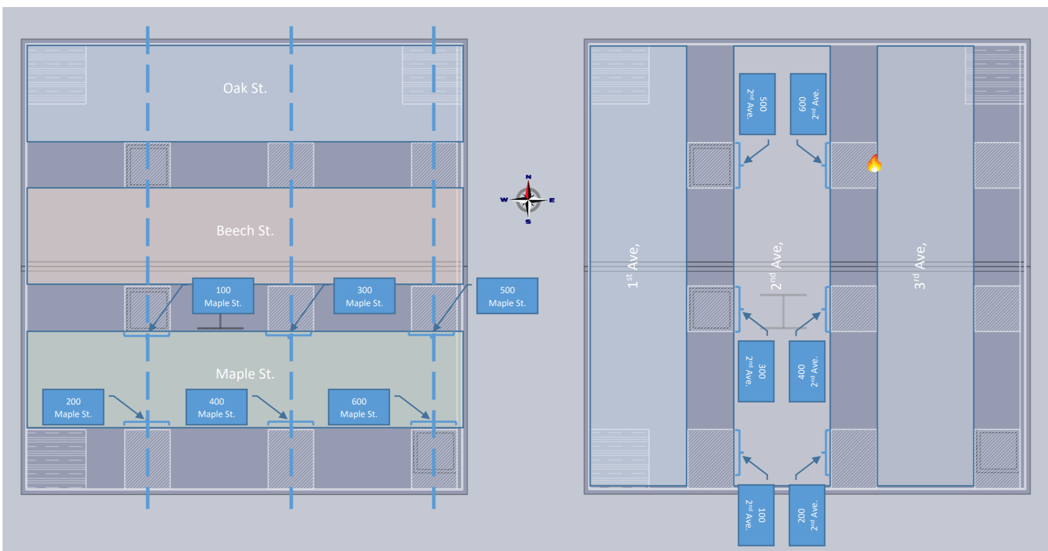

Figure 2: Naming and numbering scheme for avenues and streets

2 Methodology

In order to design our robot, we began by choosing a shooting mechanism. After an air vortex cannon was chosen as the method of putting out the fire, we proceeded to design the remainder under the additional constraints imposed by the air cannon. We created a CAD model of the entire robot design, starting with the chassis and drivetrain, calculating the ideal gear ratio to determine the appropriate gearing, the maximum speed at motor free speed at the chosen gear ratio and the current draw of the motors at slip.

Next, we designed a turret on top of the chassis, using a stepper motor to drive rotation of the turret plate which components were anchored to. We then added the fire extinguishing mechanism on top of the turret. To calculate the force required to draw back the air cannon pull, we used a load cell force sensor.

Based on these calculations, we created an appropriate transmission to allow the Pololu motor to operate the air vortex cannon, which utilized a rack and pinion including a section without teeth to draw back the pull to "cock" and then rapidly release. The optimal angle, as well as the maximum required distance of the air cannon was determined in order to ensure that no actuation of the air cannon's position needed to be implemented. Based on these measurements, the air cannon was positioned on the turret, along with the "cocking" mechanism.

Then, based on the mechanical systems on our robot and the project requirements and constraints, we selected appropriate sensors necessary to complete the task. Using our selected sensors and the previous calculations for the power requirements of the drivetrain, we were then able to calculate both the current and power requirements at steady-state and peak conditions for the system. Current and power draw of electronic components at steady-state and peak conditions were determined using knowledge of use and published information/specifications of the electronic components.

All of the electrical components, including the sensors or their mounting solutions, were then included in the CAD, with mounting solutions in a configuration allowing neat wiring of the components.

Based on the mechanical capabilities of the robot design, and our sensor selections, as well as the project requirements, we designed a state machine for programming the robot tasks.

Complex code tasks included an original A* pathfinding implementation. This system is designed to recalculate the path when the roadblock is found using a distance sensor.

An ultrasonic rangefinder on the front face of the robot was used for building detection due to its superior range compared to the KoP infrared rangefinder. The infrared rangefinder was utilized for short-range building detection and alignment when extinguishing the fire.

Together, these sensors allow the robot to scan for buildings and detect an accurate distance, as they compensate for each other's ranging restrictions.

3 Analysis

We made the decision to prioritize core features (driving, electronics) with simple design, allowing for more time spent developing turret and fire extinguishing capability. Given initial air cannon testing, as well as the example of the previous term's robot, extinguishing functionality was deemed critical.

To determine the restrictions on the air cannon mechanism, we tested the air cannon at different distances and angles with different pull distances to ensure that our mechanism would be able to put out the fire. The results of our testing can be seen in the table below, which shows the distance the mouth of the air cannon was placed at the apartment building, and the angle of the mouth to the face of the apartment building when applicable, the distance the drawstring was pulled back, the success rate of the cannon in that configuration and the placement of the fire. This allowed us to determine that we needed to place the robot at least 5 inches from the apartment building structure, and pull the string back at least 5 inches from the back face of the air cannon in order to reliably extinguish the fire.

Table 1: Air Cannon Pull Distance Testing Results

Distance to Building (in)	Pull Distance (in)	Fire Put Out? (Y/N)	Fire Placement (High/Low)
5	6	Y	Low
4	6	Y	Low
6	6	Y	Low
9	6	N/Y (50%)	Low
7	6	N/N (0%)	High
5	6	Y	High
6	6	Y	High
6 (30 deg)	6	Y	High
6 (30 deg)	6	N/Y (50%)	High
6 (-30 deg)	6	Y	High
6 (-45 deg)	6	Y	High

Next, we used a load cell connected to the KoP SparkFun load cell amplifier to determine the force required to draw the air cannon cord back to different distances. The results of this experiment are shown in table below. From this, we were able to determine that for a suitable factor of safety, the transmission should be able to transmit 10lbs of force on the pull of the air vortex cannon. Using that information, we were able to design a transmission for this purpose.

Table 2: Air Cannon Force Requirement Testing

Pull Distance	Force Reading - Trial 1 (lb)	Force Reading - Trial 2 (lb)	Normalized Force 1	Normalized Force 2	Average Normalized Force
1	3.4	2	2.2	0.8	1.5
2	4.6	1.9	3.4	0.7	2.05
3	6.0	4.4	4.8	3.2	4
4	6.8	5.4	5.6	4.2	4.9
5	7.9	6.1	6.7	4.9	5.8
6	8.6	7.6	7.4	6.4	6.9
7	9.3	10.7	8.1	9.4	8.8

We chose a transmission by determining the minimum gear ratio and the maximum controllable speed.To find the minimum gear ratio, we used the formula:

$$e_{min_{cannon}} = \frac{\tau_S}{\tau_{cannon}} \eta \tag{1}$$

where τ_S is the stall torque of the motor and η is the efficiency. We found the minimum gear ratio $e_{min_{cannon}} = 1.133$. To find the gear ratio with the maximum controllable speed, we used the equation:

$$e_{ideal} = \frac{\omega_{ideal}}{\omega_{max\ power}} \tag{2}$$

where ω_{ideal}, the maximum controllable speed, is defined as 20 $\frac{degrees}{sec}$ and $\omega_{max\ power}$ is the angular speed of the motor at maximum power, which is defined as:

$$\omega_{max\ power} = \frac{\omega_{free}}{2} \tag{3}$$

where ω_{free} is the motor free speed. We found the gear ratio with the maximum controllable speed to be $e_{ideal} = 0.509$. We picked a gear ratio of about 1:3 in two 16:28 stages, which is between the minimum and controllable speeds and allows us to use standard 20DP gears which fit into our physical constraints with a double-stage gear reduction.

Using the measurements obtained from the graphical linkage synthesis, we then designed the gearbox in CAD. To attach the gearbox to the air cannon, we utilized a rack and pinion system including a partial gear (i.e. a gear with a smooth zone that has no teeth), where the rack attaches to the cannon elastic. The gearbox was designed such that the rack aligned with the center of the air cannon for optimum drawback and release performance.

To ensure that our transmission would endure the challenge, we calculated the shear forces on the gear teeth and the factor of safety associated with our chosen material. Specifically, we looked at the last gear in the transmission, a 60-tooth 32dp gear which acted as the mechanism pinion, because it would be under the most torque. To calculate the forces on these teeth, we used the equation:

$$F_{teeth} = \frac{\tau_{cannon}}{r_{pitch}} \tag{4}$$

where r_{pitch}, the pitch radius, is defined as:

$$r_{pitch} = \frac{t}{2P} \tag{5}$$

where t is the number of teeth on the gear, and P is the diametral pitch of the gear. Based on the forces calculated, the sheer stress on the gear teeth had a factor of safety of almost 10x.

$$T_{max} = \frac{\tau_{3dp}}{FoS_{desired}} = 0.757\ ksi \tag{6}$$

$$b_{req} = \frac{F_{teeth} * FoS_{desired}}{t_p * \tau_{3dp}} = 0.256 \tag{7}$$

$$FoS_{actual} = \frac{b_{actual} * FoS_{desired}}{b_{req}} = 9.773 \tag{8}$$

Maximum current draw was calculated based on torque applied to the motor at the position of maximum torque on the crank (when it is extended parallel to the field), using the equation:

$$I_{cannon_{max}} = \frac{T_{motor_{load}}}{T_{motor_{stall}}}(I_{stall} - I_{free}) + I_{free} \tag{9}$$

We found the maximum current draw, $I_{cannon_{max}}$, to be:

$$I_{cannon_{max}} = 1.586\ A$$

We continued by imposing constraints on our drivetrain. In order to navigate through the simulated city, we were able to constrain the width of the robot to 12 inches, which would allow for some error when driving straight. The robot was restricted to the shape of an octagon to allow for more leeway when turning. Due to the enormous height (relative to the 13 inch sizing cube) of the air cannon, drivetrain height was limited to 1.5". To accomplish this, DC motors were embedded into the top and bottom plates of the vertical parallel plate design, as was the stepper motor.

With these constraints, we then were able to design our drivetrain and transmission. We chose to have two wheel drive dropped slightly below one ball castor in the back-center of the robot and another in the front-center, allowing the robot to always have 3 points of contact with the field, to avoid overdefining the drivetrain contact plane. While the wheels used the same plastic O ring as the BaseBot, the wheels were driven by spur gears for packaging reasons as well as to avoid issues related to the BaseBot bevel gearboxes.

To choose a transmission, we determined the minimum gear ratio which we calculated the minimum gear ratio using the formula:

$$e_{min_{drive}} = \frac{T_S}{T_{f_{max}}}\eta \tag{10}$$

where T_S is the motor stall torque, $T_{f_{max}}$ is the maximum torque resisting friction, and η is the efficiency.

We determined a maximum "ideal" gear ratio assuming a controllable speed of 10 $\frac{in}{sec}$, at maximum mechanical power:

$$T_{max_{power}} = \frac{T_S}{2} = 85\ ozf \cdot in \tag{11}$$

$$\omega_{max_{power}} = \frac{\omega_{free}}{2} = 100\ rpm \tag{12}$$

$$\omega_{ideal} = \frac{10\frac{in}{sec}}{d_{wheel}} = 183.364\ \frac{deg}{sec} \tag{13}$$

$$e_{ideal} = \frac{\omega_{ideal}}{\omega_{max_{power}}} = 0.306 \tag{14}$$

where ω_{free} is the free speed of the motor and d_{wheel} is the wheel diameter in inches.

We selected a 20:60 ratio (1:3) transmission in order to balance speed and controllability, with our focus being on controllability since speed was not an essential component of the

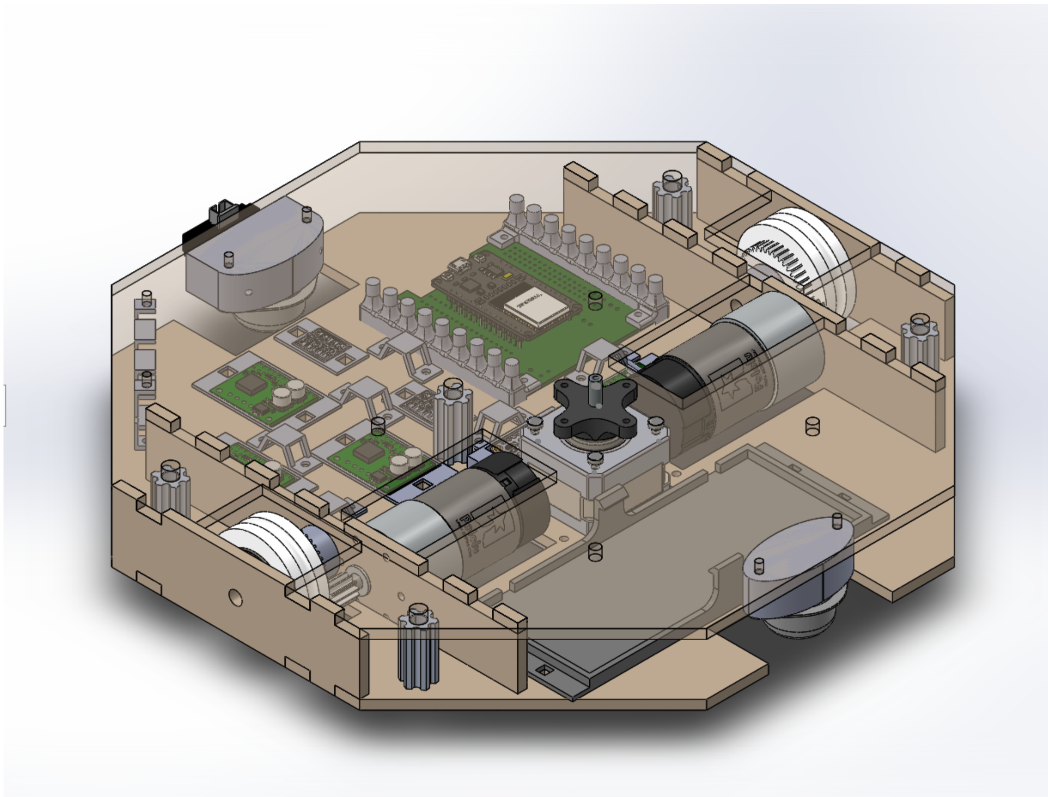

Figure 3: Robot CAD Assembly (Chassis)

challenge. This ratio is between the controllable ratio and the minimum ratio, and errs on the side of the controllable ratio.

Using the weight of the robot as calculated in our SolidWorks model, the transmission, the robot design and the motor specifications, we were then able to calculate the maximum speed of the robot and the power requirement of the robot at both slip and stall. To calculate the max speed, we converted motor free (angular) speed to linear speed:

$$v =_{free} \frac{d_{wheel}}{2} \omega_{free} e_{drivetrain} = 10.908 \ \frac{in}{sec} \tag{15}$$

$$v_{max} = v_{free}\eta = 10.363 \ \frac{in}{sec} \tag{16}$$

where $e_{drivetrain}$ is the overall drivetrain gear ratio, and η is the efficiency.

Power requirements were determined using the calculated gear ratio. Since the maximum torque due to friction is less than the stall torque, the robot is guaranteed to slip instead of stall.

$$T_{output_{stall}} = \frac{T_S}{e.drivetrain} = 31.875\ in \cdot lbf \tag{17}$$

If the two motors on the drivetrain were to stall, the current drawn would be:

$$I_{stall} = 2I_S = 10\ A \tag{18}$$

where I_S is the stall current of one motor.

The current drawn at slip is maximized when the resistive torque due to friction is maximum.

$$T_{load_{motor}} = T_{friction_{max}} e_{drivetrain} = 1.872\ in \cdot lbf \tag{19}$$

$$I_{load_{motor}} = \frac{T_{load_{motor}}}{T_S}(I_S - I_{free}) + I_{free} = 1.128\ A \tag{20}$$

$$I_{slip} = 2 * I_{load_{motor}} = 2.257\ A \tag{21}$$

Overall power requirements were based on several factors: peak drivetrain current draw, peak air cannon cock current draw, and steady-state current for electronics components.

$$\begin{aligned} P_{max} = V_{ref}(I_{slip} + I_{cannon_{max}} + I_{steadystate}) &= V_{ref}(2.257\ A + 1.586\ A + 155\ mA) \\ &= V_{ref}(4.008\ A) = 48.096\ W \end{aligned} \tag{22}$$

where V_{ref} is 12V, aka 8x 1.5V (max) 2Ah rechargeable batteries.

The steady-state current calculations are based on manufacturer specifications for idle current draw in each of the electronic components, given by:

$$I_{idle} = \sum_{i}^{n} I_i \tag{23}$$

where I_i is the idle current draw for the component i. In this case the components sum as outlined in Table 3.

Table 3: Current Draw of Electrical Components

Component	Current Draw
ESP32	25 mA
WiFi Reciever	100 mA
Encoders	30 mA

$$I_{idle} = 25\ mA + 100\ mA + 30\ mA I_{idle} = 155\ mA \tag{24}$$

Given an ideal 12V Battery pack with a capacity of 2Ah, the total energy the pack can provide is given by:

$$\begin{gathered} E = Pt = V_{ref}It \\ E = (12\ V)(2\ Ah)(\frac{3600\ s}{1\ h}) \\ E = 86.4\ kJ \end{gathered} \tag{25}$$

Now that we know the energy the pack can provide, we can divide by the power requirement to determine operation time. Steady state requirements:

$$\begin{gathered} t = \frac{E}{P} = \frac{E}{VI} \\ t = \frac{86.4\ kJ}{(12\ V)(0.155\ A)} * \frac{1\ hr}{3600\ s} \\ t = 12.9\ hr \end{gathered} \tag{26}$$

As we can see in the equation above, running at only steady-state, the battery will last for about 12 hours, 54 minutes.

Peak requirements:

$$\begin{gathered} t = \frac{E}{P} = \frac{E}{P_{max}} \\ t = \frac{86.4\ kJ}{48.096\ W} * \frac{1\ min}{60\ s} \\ t = 29.9\ min \end{gathered} \tag{27}$$

Even at a constant max current draw, the battery pack will still allow 29 minutes of operation, more than enough time to complete the challenge. This exceeds the goal of 10 minutes - the maximum time allocated to complete the entire list of tasks.

Our pathfinding and localization code took the form of a weighted directed graph (grid map), where streets are navigable edges and buildings are visitable nodes. Using a heuristic based on the total distance to the destination:

$$H = x^2_{destination_{total}} + y^2_{destination_{total}} \tag{28}$$

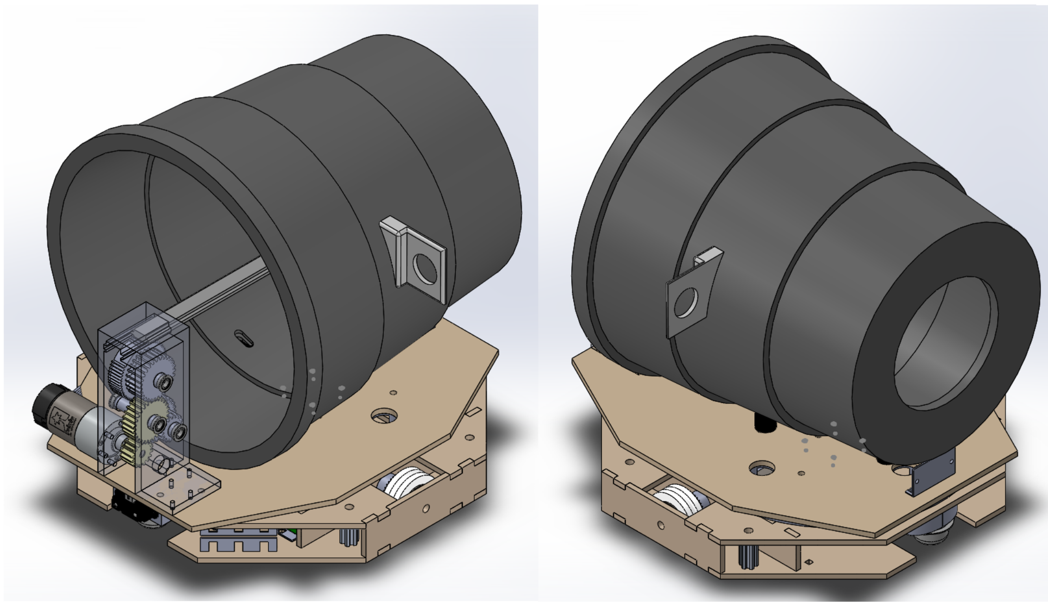

Figure 4: Robot CAD Assembly (Overall), 2 Views

And a movement cost roughly based on total time for completion in the form of linear distance to travel and the circumference from required rotation:

$$M = x + y + arc \tag{29}$$

Cost is multiplied by 2 if the node has been visited before, providing incentive to visit unchecked nodes without blocking prospective paths.

Overall code design included emphasis on object-oriented structure and non-blocking operation. This was completed through the implementation of feedback loops which would result in corrections in every loop. In order to create a method of executing and order of driving instructions, a queue was created through a linked list which instructed the robot to either drive forward, turn a certain amount of degrees, and check for the fire. We utilized a global coordinate system and used the GetIMU class to maintain absolute positioning, allowing us to visualize robot location in Bowler Studio.

Using a SolidWorks sketch for precise and automatic dimensioning, the IR camera was mounted in the top section of the robot as to see both prospective fire locations at a distance of 6". To reduce the distance required between the camera and flame, the camera was mounted such that the FoV was 33°(long axis) vertically. This allowed for an effective usable range of about 3 feet due to IR intensity from the flame and a large blockage in field of view by the air cannon.

A pair of rangefinders was used to determine distance as follows:
IR rangefinder (80cm @ 0.4V, 10cm @ 3.1V)

$$cm = mapFloat(Vcc * reading/4096, 3.1, 0.4, 10, 80) \tag{30}$$

Ultrasonic rangefinder (6" @ 0V to 155" @ Vcc)

$$inches = 512/Vcc * reading/4096 \tag{31}$$

4 Results & Discussion

We found during the testing and demonstrations of the different systems of our robot that our calculations and predictions were accurate. However, some mechanical design oversights required major changes to our robot even after the Wednesday demo deadline, and code did not progress at the rate required - integration was poor.

This experience demonstrated to each of us that the real world is "messy", that CAD cannot account for human error in design, and that programming requires an extended timeline which allows for very extended test and integration time

Concerning one of our primary tasks, utilizing IMU and encoder for navigation, we had extremely positive results. The robot was able to quickly and accurately traverse the field surface with straight paths and accurate turns, except where covered with tape - tape surfaces proved too slick to generate enough traction to turn the robot, since the robot was very light. This task proved much easier once our custom drivetrain was developed, as mechanical problems plagued the BaseBot drivetrain. PID tuning was relatively straightforward and was successful for controlling the drivetrain in both default and "Y-correction" modes when driving straight, and for the turning controller.

A* pathfinding proved successful but was only minimally integrated into overall robot code - robot was easily able to follow pre-set paths and generate adjustments to account for roadblocks, but this code was not integrated into functionality to properly search for the fire (i.e. if the fire was not found on the path, the robot would stop looking).

Turret functionality was somewhat successful, after several roadblocks. Conceptually, the turret was sound, but due to human error in CAD, bolt heads collided while turning. After adjustments (3d printed risers), the turret worked while stationary and successfully could be used to rotate the air cannon and scan with the IR camera. However, the turret was taped into a stationary position for testing since the stepper motor displayed extremely low holding torque. As the turret bearing made the system effectively frictionless, this meant the turret was uncontrollable while moving the robot.

Concerning the IR camera, a successful scan procedure was developed to detect the fire. This functionality was developed but not well integrated/tested due to reported turret problems. The location of the fire, once found, was successfully reported via Bowler Studio.

An ultrasonic rangefinder on the front face of the robot was used for building detection due to its superior range compared to the KoP infrared rangefinder. The infrared rangefinder was utilized for short-range building detection and alignment when extinguishing the fire, but was not integrated. Both sensors worked adequately, but did require working around their dead zones.

The manufacturing process of our robot went smoothly. We created a vertical standoff-based design using Birch plywood and the popular-with-FIRST "churros". The churros are aluminum hex shafts that provide dual function as both standoffs and axles in our robot design. The churros are lightweight, strong, and simple to cut, which made them an excellent part for our build. We tapped the churros ourselves to allow for accurate placement of threads for assembly. This combination materials and methods of construction allowed us to assemble our chassis rapidly, with not much time spent waiting for 3D printed parts. In addition, the robot chassis was strong, lightweight, and flexible, giving us a robust and damage-resistant chassis to work with.

The vast majority of our bugs in software during development resulted in segmentation faults, for which the ESP32 does not provide any insightful information. This greatly impeded the rate of development on our robot.

Overall, the robot performed well in the demo compared to other teams, but due to large mechanical and code problems that were not resolved until too late, our performance fell back towards the benchmark as we were not able to integrate all of our systems together.

5 Conclusions

Overall, the project was unsuccessful at completing the given tasks, as we struggled to integrate our different working components and faced continued issues with failing components and electronics. While we made many major changes to the robot, the sequence of issues made it impossible to demonstrate the tasks being completed in sequence.

Given additional time, and continued access to the robot, the main improvements to be made on our robot are continuing work on the code and mechanics, including replacing the stepper motor driving the turret with a more suitable motor such that the turret would not freely rotate, so that all tasks can be completed in one run, and improvement of the air vortex cannon cocking mechanism to increase the reliability of the rack and prevent it from breaking in operation.

While the robot currently is ineffective at completing the project tasks when faced with the entire system at once, most of the individual components (pathfinding, air cannon, turret) work well when isolated, and continued work on integration would result in a successful robot. Use of sensors was successful, as driving was our most successful task and the chassis our most solid subsystem, utilizing the IMU and encoders.

Comments

Optional section, included as per format suggested by Professor Bertozzi (RBE 2001)

Large inconsistency was introduced when driving on the field due to the alternate field surface (tape) in some areas, which often caused the robot to lose traction since it is fairly light.

The term project was a somewhat good application of concepts learned in class, but more focus on analog filtering/dynamics portions would have been useful as good practical experience. The PDR and CDR should have been located earlier in the course, and more emphasis should have been placed on Lab 4's completion on an earlier timeframe (perhaps 1-2 weeks earlier).

For purposes of the project and class in general, a semi-comprehensive powerpoint or other documentation covering concepts expected for class (especially the types of math for mechanism analysis) would have been very useful.

We had several problems with the ESP32. One of the pins we were told to use for the H-Bridge did not work (we switched it to another pin successfully). There should be more documentation on the ESP32 timers and how they related to PWM pins, and which can be used.

We had to replace the Robot Interface Board several times - once due to a "known problem" where the 5V regulator dies closed and provides V_{bat} to all devices on the 5V rail. Soon after repairing the RIB for this problem, an undiagnosable (by Kevin or Alex Camilo) short occurred and we were forced to waste a lot of valuable time replacing the RIB twice, as the first one provided was not usable.

The example robot base was a poor reference as its wiring was messy (it may be good to use 3D printed use wire routing parts, such as we used on our robot). Wiring diagrams on GitHub were unavailable for some components.

Wire wrapping was an arbitrary and inefficient way to make connections. We sometimes had issues with loose wires even when a sufficient number of well-done wraps. The wires were very difficult to keep tidy, and required significant ahead-of-time design effort for best practices.

Appendices

Appendix 1: Contributions

Table 4: Team Member Contributions for Lab and Final Project

Student Name	Contribution to: (%)		
	Lab	Final Project	Final Report
Teresa Saddler	35	33	43
Luke Trujillo	30	33	2
Benjamin Ward	35	33	55