AutoBot

Project Manager
Jacob Kubisiak
Documentation Chief
Marissa Hintz
Financial Manager
Brandon Johnson

Project Engineers
Philip Wolschendorf
Connor Stone
Erik Romanski
Joe Venier

The goal of this project was to design, build, and program an autonomous vehicle which can successfully compete in the Intelligent Ground Vehicle Competition.

Introduction

The Autobot team of Michigan Technological University would like to introduce the location and obstacle aware robot, “Bishop”. Though there have been improvements, this is the third year that the mechanical design for Bishop will be used. The software has changed drastically from the last competition and the electrical design has been significantly improved upon.

The team was organized with one project manager, document chief, and a financial manager. The remaining members acted as project engineers. The role of the project manager assumed the responsibility of assuring that the project would be completed on time by allocating resources and keeping the team accountable. The document chief managed the documentation and the financial manager maintained the budget. Each member of the team functioned as an engineer by designing, building, and testing the robot. The team was also a multi-disciplined team made up of mechanical engineers, electrical engineers, and computer engineers. The divisions of the roles can be seen below in Table 1.

Position	Team Member	Major
Project Manager	Jacob Kubisiak	EE/CPE
Documentation Chief	Marissa Hintz	CPE
Financial Manager	Brandon Johnson	ME
Project Engineer	Philip Wolschendorf	EE/CPE
Project Engineer	Connor Stone	EE
Project Engineer	Joseph Venier	ME
Project Engineer	Erik Romanski	EE/CPE

Table 1: Member Information

The robot, Bishop, was constructed with the collaborative knowledge of many students throughout the last few years. The robot design team at Michigan Technological University is on a three year rotation, meaning that every three years the robot used for the IGVC competition is completely rebuilt mechanically and electrically. The software for the robot may stay, however with each new student group that works on the robot and with the drastic changes in software the design does not remain the same for long.

Mechanically, Bishop resembles a motorized tricycle with a mast. There are two wheels in the center of the robot and one in the front. The body of the robot houses in the motors, batteries, laptop, and the laser range finder. The mast extends from the center of the robot to allow the robot height to see its surroundings. The cameras are placed on the top of the mast and configured to limit blind spots. A GPS is also located on the mast to attain the current location of the robot. The e-stop button is placed in clear view in the center of the mast.

The computer that is located on the body of the robot controls the robot. All of the sensors on the robot are connected to the laptop via USB connection. The data from the sensors is read into the robot's code and analyzed to determine the shortest route to the given waypoint. The camera data is used to keep the robot in between the white lines on the course and to locate and navigate around the flags. Each component is read into one program and used to direct the robot.

Design Background

The majority of the material used for the construction of this autonomous robot is aluminum, with many of the parts being 8020 extrusion rails. Aluminum was used because of its high strength-to-weight ratio, relative inexpensiveness, and its ability to be easily machined. The 8020 extrusion rails provide a rugged structure, while allowing parts to be easily mounted or connected. The current design consists of a chassis containing two independent motors, with a peninsula in front containing electrical components and supporting an undriven caster wheel. The peninsula is constructed of lengths of right angle aluminum. Directly behind the peninsula are the battery powered motors, which power the two 29 inch driving wheels that are mounted directly to the shaft. Near the front of the robot is a smaller pivoting wheel, acting as a guide for the changing of direction. In the very rear of the robot is a dock for the connected laptop to securely rest on. As for the transportation of the payload, there is a hanging carriage below the peninsula where the payload will be carried during the competition.

The robot has a height of 55’’, width of 29’’, and length of 43’’. The central structure of the robot is the peninsula, which provides as housing for the electrical components, including the two batteries, with the laser range finder positioned at the front. An 8020 rail acts as a mast extruding upwards from the peninsula about 3 feet to give a higher perspective for the 3 cameras. The 3 cameras are mounted in a row with the outer two angled to the right and left to provide a wider field of view.

Design Improvement

The physical design described has been the general design for this robot for previous years, however mechanical improvements have been made to this design this year. At one time in the project, the mast was attached to the chassis which allowed it to tilt left or right a few degrees, compromising the accuracy of the camera, IMU, and GPS data. The mast was reattached with different fasteners and braced on the left and right with brackets that connect to the peninsula (Figure 2). These two measures eliminated this degree of freedom, and the mast can no longer move noticeably relative to the rest of the robot.

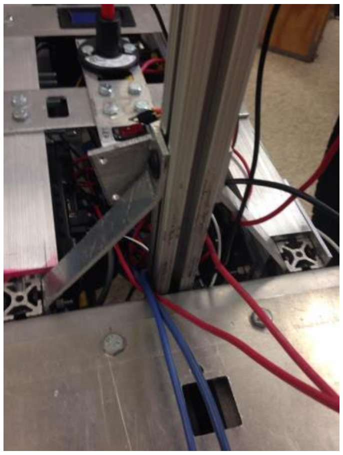


Figure 1: Mast

A problem for the robot this semester was that one of the wheels would gradually become loose while the robot operated for no obvious reason. After some inspection of the fasteners, it was concluded that the bolts holding the wheel to the motor were stripped. Part of what contributed to the stripping of these screws was that they had to be fed in at an angle because they were too long for the space provided for them. Bolts on both wheels were replaced with new, shorter bolts that still engage all the threading on the wheel. The clearance for the bolts was also reduced by bushings on the motor shafts that were left over from an old design but no longer served a purpose. Between removing these bushings and using shorter bolts, there is now enough space to feed the bolts in straight. After the bolts were replaced, the wheel did not come loose. The diameter of the heads of the bolts has to be reduced with a lathe for the bolts to fit, so extra bolts were modified this way and will be brought to the competition to serve as ready spares.

Another upgrade has been the Laser Range Finder. The previous laser ranger finder, was replaced with SICK LMS (Laser Measurement System) 291. This new laser range finder was attached with a removable mount, which uses friction lock rails to allow the angle of the laser range finder to be adjusted.

Mounts were made for many other components. The previous mounts designed for the cameras were made of plastic and were becoming brittle after several semesters of use. The new clip design is split into two parts that are clamped onto the camera with bolts. This design does not require the plastic to bend as far as the old clips did, therefore lessoning the chance that the new clips will break in the future. The clips for the cameras were fabricated with a 3D printer which can be seen in Figure 3.

Safety was a top priority for the team as development was conducted. In light of this, terminal covers for the batteries were printed. This was done to avoid short circuited connection if contact to the aluminum body of the robot is made. Also, the physical E-Stop was mounted about half way up the mast (Figure 4). This allows for an easy and quick shut down of the robot if needed.

Figure 2: New Camera Clip

Figure 3: Physical Emergency Stop

To improve the reliability of the robot, some consideration went into weather proofing the robot. In case of inclement weather during the competition, two protective covers were constructed from vinyl sheets that can easily attach and peel free from Bishop using Velcro strips. These will prevent water from falling directly into the peninsula or onto the laptop. Additionally, a bulb was ordered to cover the blinking light board on top of the mast.

Design Process

The semester began by analyzing the previous electrical system which was already in place. There was a lack of reliable and understandable documentation, which made the initial analysis difficult. However, with the aid of a flow chart diagram showing the high-level function, a preliminary circuit analysis was accomplished. This was recorded by hand throughout the process by hand-sketching the circuit on paper (Figure 4). The hand-drawn circuit is accurate, but cumbersome and difficult to follow; it was quickly converted into a clearer and more useful circuit diagram using the Autocad software (Figure 5). The final circuit schematic shows the entire robot electrical system, displaying its connections and its components in great detail. This makes the entire project comprehensively documented, easy to understand, and easy to replicate.

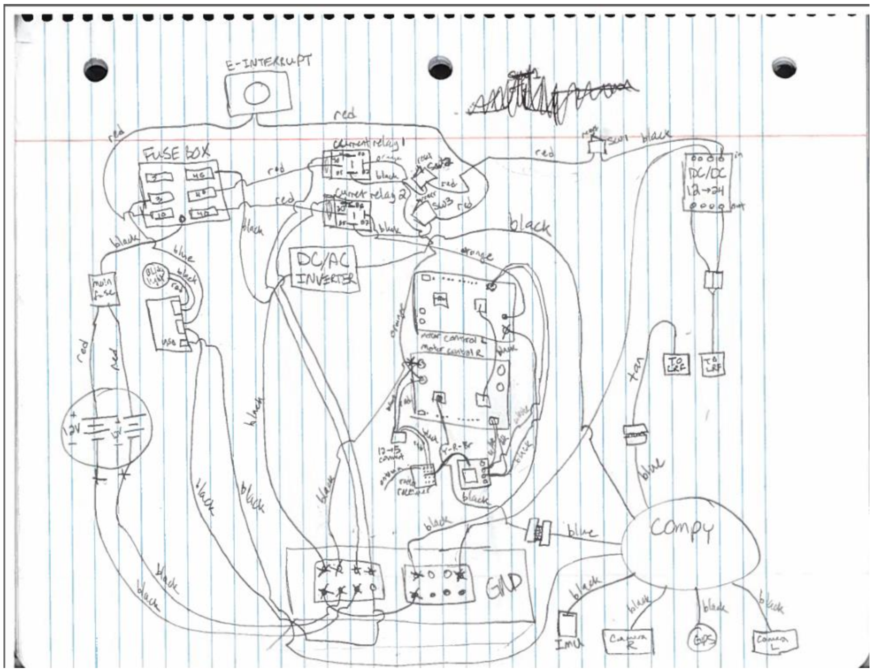


Figure 4: Preliminary Circuit Sketch

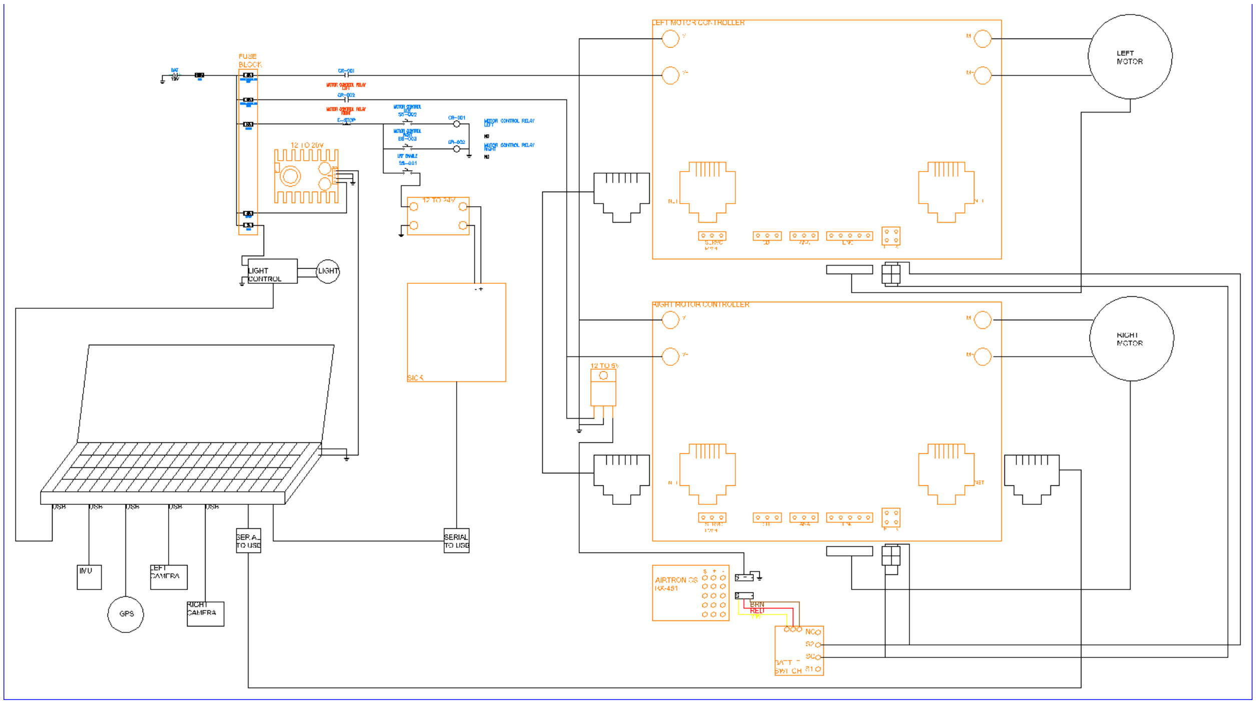


Figure 5: Final Circuit Schematic

Design Background

The robot’s electrical system runs from a 12V bus, provided by two large lead-acid batteries in parallel. The system has a common ground, a ground plate at the center of the chassis cavity which is electrically isolated from the chassis itself. The system is protected from overcurrent in case of fault or other error by a double-fused system. First, there is a large 125A fuse directly in series with the volt bus; second, the individual components are fed from a fuse box, reducing total current to each component to 40A, 10A or 3A, depending on their needs and tolerances.

Current is routed to the motor controllers through switches and then through current relays. Additional protections in the form of the emergency stops are also in place, as required by the IGVC rules. The “big red button”, or physical e-stop, directly interrupts current flow to the motor controllers when depressed, stopping the robot from moving. The wireless E-stop, when triggered, sends an interrupt signal to the motor controller, switching off the motors.

The voltage from the 12V bus is also scaled for the various needs of the components. At various points in the electrical system, the 12V bus is converted to 20Vdc, 24Vdc, and 5Vdc, for the computer, the laser range finder, and the radio receiver respectively. The electrical components such as the cameras and the GPS, for example, are connected directly to the computer through USB cables.

As with all engineering projects, safety was placed as the first and foremost consideration. The safety standards provided in the IGVC guidelines were closely and carefully followed in all aspects. The E-stop button is 1.5” in diameter and is bright red with a bright yellow casing, and is placed on the mast of the robot at a height of 3’4”, in plain view, unimpeded and easy to reach and activate in the event of an emergency. Current to the motor controllers is routed directly through the E-stop button from the multi-fuse box, meaning its execution will physically remove power to the motors, stopping the vehicle. The receiver for the wireless E-stop is placed on the front of the mast, with its antenna extending towards the front of the robot. When activated by the remote control, the radio receiver signals to the relay to open the normally shorted connection in the motor controllers, automatically shutting down all motion and stopping the vehicle. The safety light is constructed of a grouping of high-intensity, bright blue LEDs, which are lit solid while power is being supplied to the vehicle, and switch to flashing on and off only when the vehicle is placed in autonomous mode.

Additional care was taken in the vehicle’s construction to prevent any hazards from existing. The highest voltage value in the system is only 20Vdc--under ordinary conditions, nowhere near high enough to cause any danger to human life. However, precautions were still taken to prevent currents from reaching dangerous levels, for the protection of both human beings and equipment. The system is double-fused, meaning that the main fuse rated to 125A is connected in series between the voltage bus and the rest of the robot, while secondary fuses rated to 40A, 10A or 3A are connected by the multi-fuse box between the main fuse and the various components. Thus, in the event of a short taking place somewhere on the machine, if any fuse were to fail, the second-level fuse would remain as a backup to protect the system. Finally, the main power switch serves as a safety consideration as well as serving its functional use. Having the power supply physically disconnected from the circuit when the robot is not in operation adds an extra level of security during periods of storage or maintenance.

Design Improvement

Once the preliminary analysis was complete, the electrical hardware was repaired. It was necessary to replace almost all of the wiring in the robot system. Previous iterations had produced a great deal of excess wiring which made the circuitry difficult to follow and debug, caused loss in the system, and created a hazard in operation. Loose wires hanging in the chassis may become tangled around the wheel or axel, then be pulled loose and creating both danger to team members and damage to the robot. To amend these issues the physical location of several components was rearranged to decrease overall complexity of the system. This was followed by a complete replacement of the wiring itself. The wires were replaced to use only the minimum length for a secure connection without tension in the wire.

Next, the connections were upgraded. Previously the ends of loose wires were down. This was updated so that the wires were outfitted with appropriate connectors such as crimping loops to create a robust and secure connection without risk of a broken or shorted circuit. The connection from the power source to the radio receiver was also replaced, obtaining a snugly-fitting servo terminal. This replaced previous loosely fitting terminal pins. Since the batteries for the system were reaching the end of their lifetime, they were also replaced with new batteries of the same type.

Once this process was accomplished, several more improvements were made. A main power control switch was added. The component selected was a robust keyed switch, meaning the switch could be opened and the key removed entirely, preventing accidental powering to the robot. The switch was introduced in series with the main 12V voltage bus, between the main source fuse and the multi-fuse box. Thus, de-powering the switch removes power to the entire machine. A voltmeter was added in parallel between the input terminal to the multi-fuse box and the ground plate, powered from the same. Thus, when the robot is powered, the voltmeter displays the voltage of the 12V bus, while still being protected from overcurrent by the main source fuse.

Another improvement made was replacing the laptop power supply. Previous designs had used the laptop's commercially provided 20V AC/DC rectifying power supply. However, since commercial inverters run on standard household power, an inverter was required to convert 12Vdc from the batteries to 120V at 60Hz AC. This resulted in two conversions, 12Vdc to 120Vac to 20Vdc. This double conversion was a significant source of loss in the system, and therefore a significant power drain on the batteries. Furthermore, the hardware itself was large, bulky, and difficult to secure. The system was replaced by a simple 12-20Vdc boost converter, which had much less loss and was easily installed underneath the laptop docking station.

Testing Results

No significant testing was required for the electrical systems. The electrical functionality was observed while testing was performed while testing the robot’s autonomous navigation systems, and no problems or errors in hardware were detected.

Design Description

Much of the functionality of this project comes from the code, because of this many decisions for the robot were based off of different aspects of coding. Using different components including GPS, Laser Range Finder, and Cameras data was gathered and analyzed throughout the code to determine the shortest path to a given waypoint. Once the path was determined, the robot moved in the direction given in the path.

Coding Features

Before any code could be written, the platform, operating system, and programming language had to be decided upon. A couple constraints that were considered were the flexibility, reliability, speed, and availability of prewritten software.

The first decision to be made was the type of computing platform. There were two main options available: a distributed system, or a centralized computing system. A distributed system would include many different coded components that each managed one aspect of the project while a centralized system would have only one component that would manage the entire project's software. Our team decided upon the centralized system as it is much easier to modify and update. The distributed system may introduce more non-software related bugs, or malfunctions. In essence, the code for the robot is included in one laptop that controls each aspect.

Next, the operating system was chosen. One of the greatest debates for PC users often is whether a distribution of Linux or Windows should be selected. Linux offers more flexibility when creating a software platform and it is often much more reliable than Windows. In light of this, Linux has been chosen instead of Windows. However, there are many different distributions of Linux. In the past the Autobot team has had issues with the reliability of the selected distribution of Linux, so when choosing the operating system for this project, a very reliable and well tested and supported system was picked: Ubuntu 12.04 Long Term Service Package.

There are many different programming languages to choose from when developing. Each language has different strengths and weaknesses. The C programming language is a procedural program that is designed for efficient execution. As our robot will be competition to finish a course quickly, an efficient language was preferred. C, however, is older and does not include many of the features that would make the coding for the robot easier. One of the defining factors for our selection of a programming language is the availability of libraries. C++ was the language that was decided upon as it contained two libraries that our code is very heavily based upon: the Mobile Robot Programming Toolkit (MRPT) and OpenCV.

To increase the ease of development, flexibility of the code, and the ease of testing the robot, the code was designed to be modular. In a coding sense, this means that the code is divided into smaller segments that don't depend on each other or depend minimally on each other. This allows testing and development of the code to occur on one part of the robot even if another part of the robot is not functioning. For instance, the cameras can be tested independent of the rest of the robot's systems. This allows them to be easily added and removed as needed.

As the final goal of this project is to enter the robot into a competition, speed was a very important consideration. There were some components in the code that slowed down the completion of the calculations. To amend this, some of the calculations are run concurrently by using threading. Threading is launching different calculations at one time and having them run simultaneously. Although this does complicate the programming, it greatly increases the speed of the robot.

GPS

Part of the competition is to design a robot that can travel autonomously from one location to another. In order to do that, the robot must know where it is. A GPS unit was used to find the robot's current location. The GPS is constantly pulling data in a separate thread from the rest of the code. This allows the robot to have an updated current location as soon as it is available.

Although GPS are very useful in this project, they GPS that was used was not perfectly precise. Below in Figure 6 the precision of the GPS is shown. This is a display of data taken over 5 minutes. The inner circle represents the area that 50% of the data falls within and the outer circle the area that 90% of the data falls within. The standard deviation and covalence were found and this information is used for outlier detection; to avoid any coordinate that is significantly different the other data.

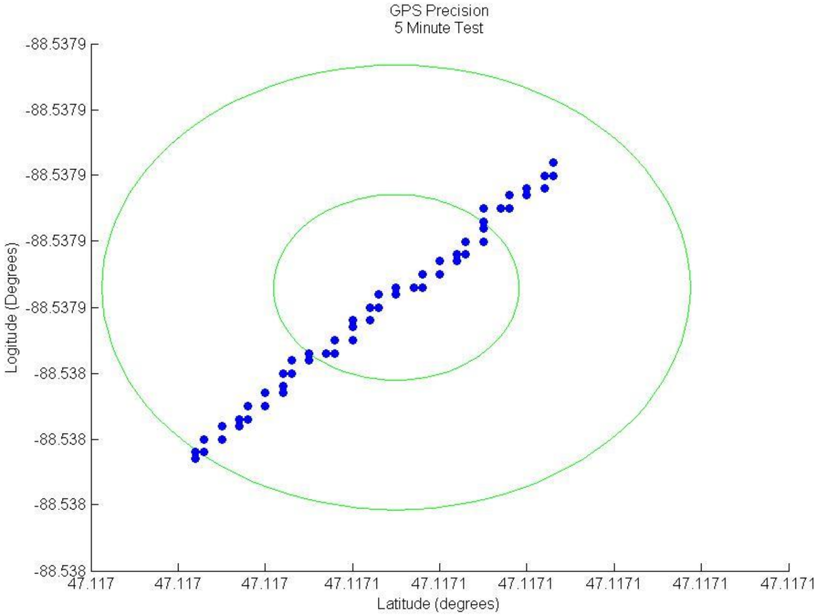


Figure 6: The precision of the GPS

Laser Range Finder

The laser range finder is utilized during the competition to discover obstacles and how far they are from the robot. Since the course will be littered with obstacles, the ability to navigate around them is paramount, especially since striking obstacles during the course of the challenge adds penalties to the final result. As the obstacles most likely to strike the robot are in front of the robot, the laser range finder is situated at the front of the peninsula, facing forward. In order to find obstacles, the laser range finder sends out a laser pulse and determines the range (max 65 meters) to an object by the time before the laser returns. The algorithm utilized determines a probability of an obstacle's presence, from 0-100%, where 0% means there is no obstacle present. When the algorithm finds these probabilities, it marks them on a generated map of the surroundings. The map uses the varying probabilities of obstacles, with a 0% chance of an obstacle's presence marked as white up to black for a 100% probability. An example of the variable probability is shown in Figure 8, a map of the DOW courtyard which was created in a test run. The map shows a large white area in the middle, which was the obstacle-free path the robot took on the run, while the black squares inside that white area represent obstacles placed in the courtyard. An interesting thing to note is that at the top-right, top-left, and bottom of the map, there are areas that slowly go from 0% probability (white) to 50% probability (darker gray), with a few shades of darkening gray in between, showing that the laser range finder's range and loss of accuracy farther away.

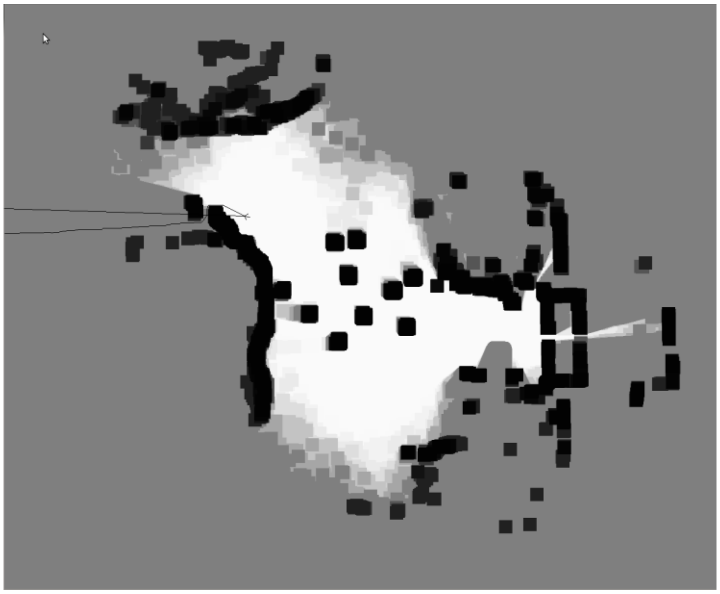


Figure 8: Map of DOW Courtyard

Cameras

The cameras allow the robot to accomplish the lane following objectives for the competition. This entails staying within the lane defined by white lines drawn on the grass. Additionally, the cameras are used to traverse a series of colored flags. The robot should travel to the left of red flags and to the right of blue flags. In order to obtain distance information, the 3 cameras are orientated in a downward manner. Imaging processing is handled using the Open Computer Vision (OpenCV) set of tools. These are a free to use set of imaging processing tools. Functions such as the Hough Line detection, inverse perspective transform, threshold, and Canny filters are handled through this library.

Each camera is calibrated using a precision map. This precision map is used to obtain the transformation matrix to perform an Inverse Perspective Transform on the images received from the camera. This transform generates a top-down view of the camera image. This viewpoint simplifies the integration with the rest of the mapping algorithms as well as providing an equal distance projection.

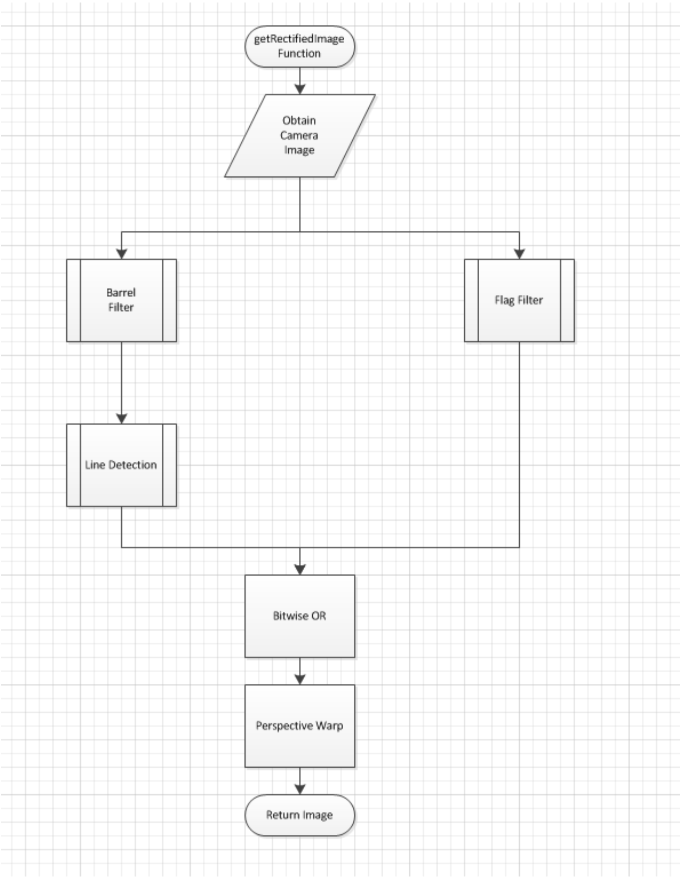


Figure 9: A high level overview of camera functionality

The camera applies various filtering algorithms to the images in order to detect the flags and the lines. The flag detection uses the unmodified camera image, splits the image into its Red, Green, and Blue (RGB) components. For red flags, it subtracts the Green and Blue layers from the red. This yields an image consisting only of red objects. A blur filter is applied at this point to remove some noise that may have remained. Since orange construction barrels may appear to be similar to a flag, shape recognition is run on the image. This detects polygons located within the image. If the polygon’s area is too small or too big, it is ignored. At this point, the polygon is checked for the number of vertices and edge length. If these parameters indicate it is a square, the algorithm determines that the polygon is a flag. Flags are handled by creating obstacles located to the side of the flag. This will cause the robot to avoid these points when it is navigating.

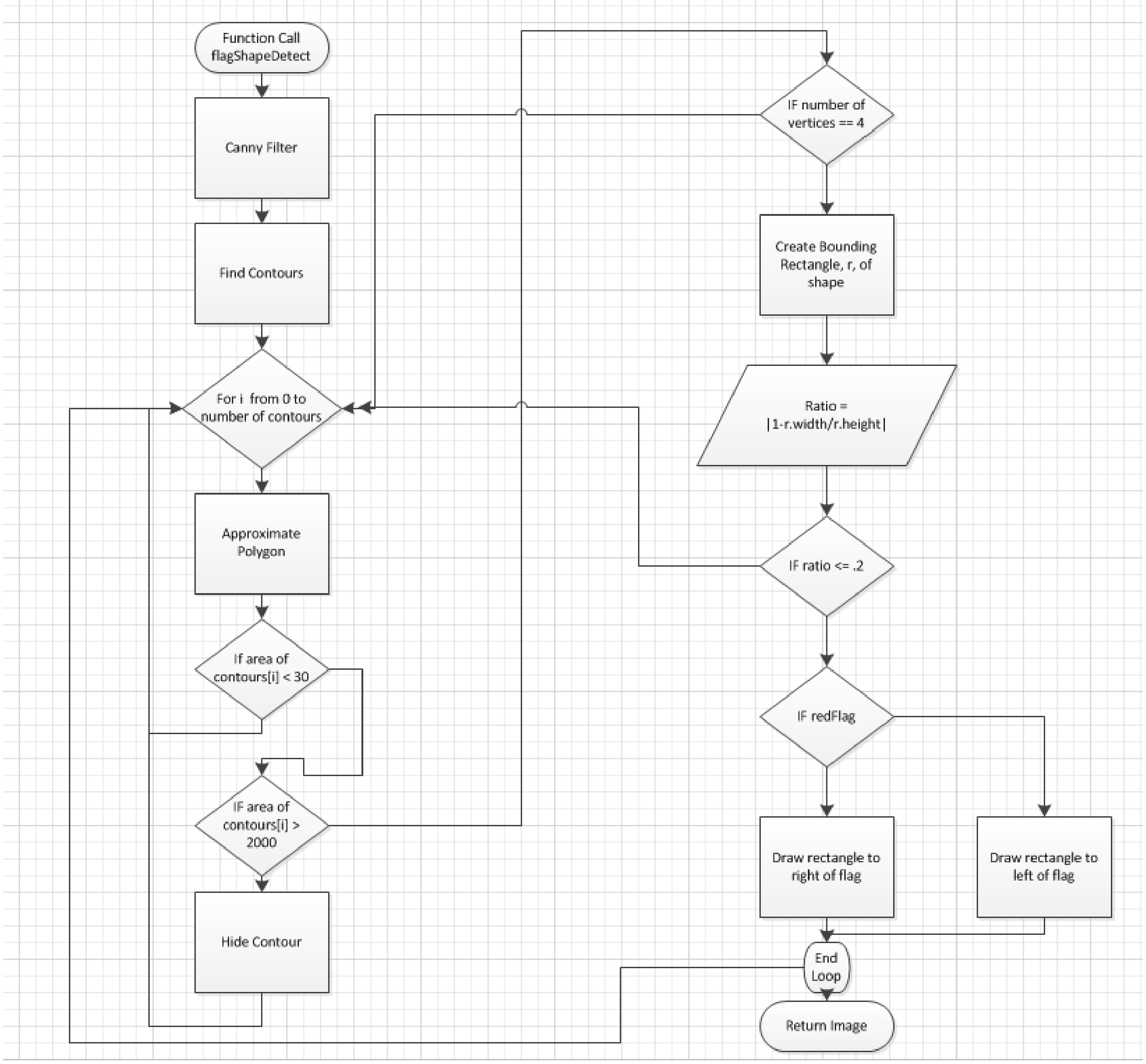


Figure 10: The flowchart representation of the flag shape detection function

Line detection uses has been updated from the previous method of simply detecting and avoiding the color white. The updated detection uses Hough Transforms to detect lines in the image. This is done using a three step process. The image is passed through a low pass filter to remove noise. A threshold filter is also applied at this point. This processed image was the extent of the previous image processing for line detection. The updated method uses this filtered image and determines the "skeleton" of the line. The skeleton is a single pixel wide version of the line obtained through repeated erosions and dilations. Ultimately, this skeleton representation of the lines in the image is passed through a probabilistic Hough Line Detection function. This function attempts to find lines in the image while also considering that gaps may exist in the skeleton line. These lines that are found are returned to be used in the camera mapping.

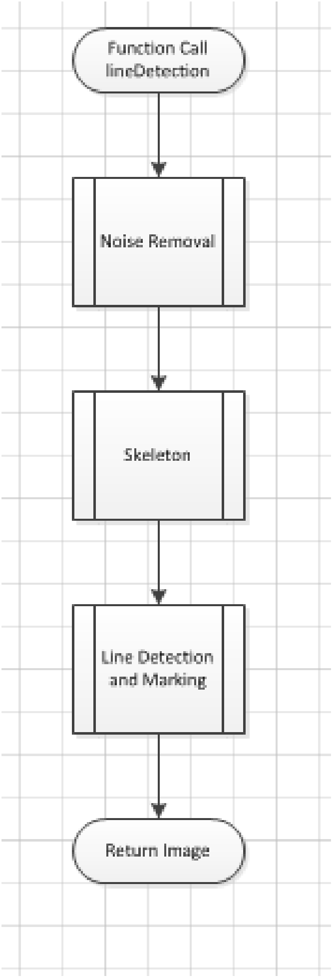


Figure 71: A high level overview of the line detection

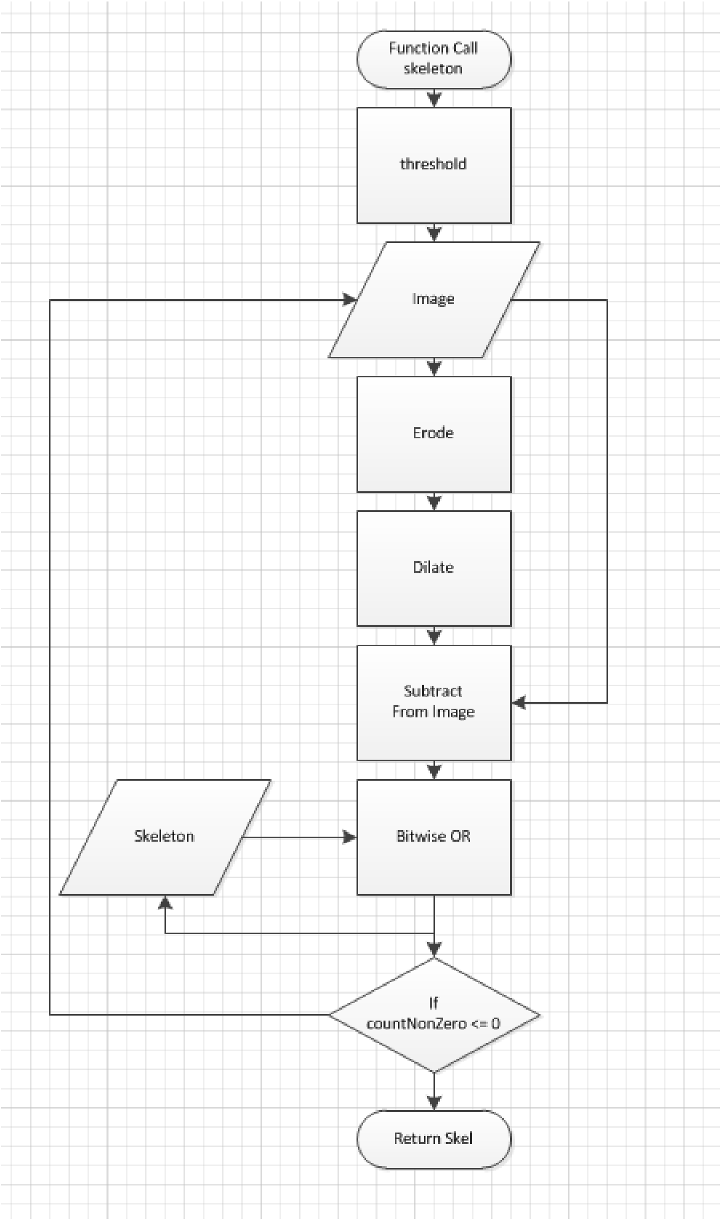


Figure 82: A representation of the skeleton function

The results of the flag detection and the line detection have a bitwise OR operation performed on them. This allows any obstacle that is seen by either method to be present in the map. Now that the filtering has been applied, the image should be binary in nature. This means that all obstacles are white and everything else is marked as black. This simplifies the mapping and navigation because probabilities will not have to be considered. The image is inverse perspective transformed at this point according to the predefined configuration inherent to that camera and its position.

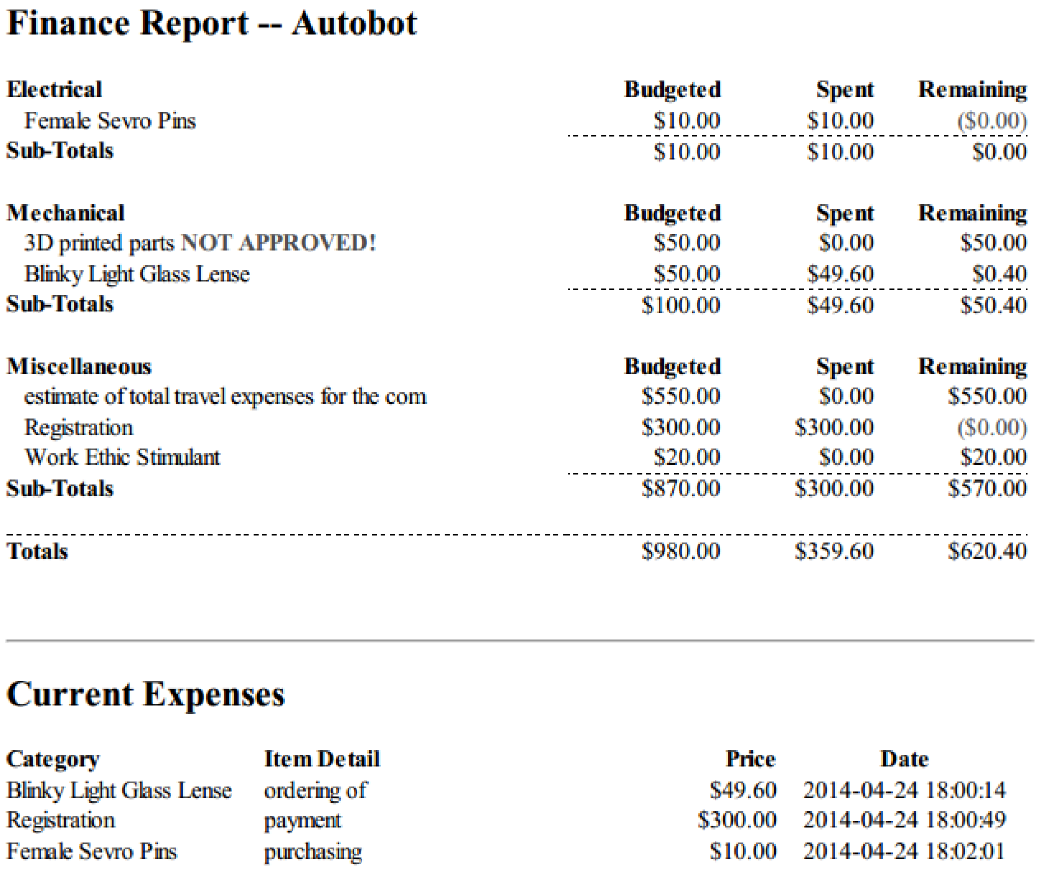
Navigation

The navigation of the robot is dependent on the data it receives from its various sensors. Fundamentally, the navigation is based on the GPS. The competition centers on the robot navigating to a pre-determined GPS waypoint. In order to avoid the obstacles, the path finding algorithm has to incorporate them. The map is a top-down view of the obstacles that the sensors have seen. These obstacles come from the camera data and the laser range finder data. The laser range finder map is used as the base map for these operations. This is due to the lack of variability in the data due to noise. This base map is used to obtain the robot's current position in relation to the origin of this map. This data is then used in the camera map building in order to keep track of positional changes in the map.

The camera map utilizes the top-down views previously generated and adds the images to the map. The exact location where the images are added it dependent on the configuration of the cameras. Since the data from the cameras can be "noisy" and false positives may arise, the camera map is designed to be cleared over time. This is accomplished by applying a scalar addition to the map images. Over time, the older obstacles will gradually fade away. This would allow the robot to recalculate a path if the false positive was due to the transient nature of the camera or the conditions around it (i.e. glare). However, obstacles that really exist would continue to be added to the map and would not fade away on the map.

The camera map is superimposed on the laser map allowing obstacles from both to exist on this map. By having a singular map, the navigation code is reduced in complexity. This is possible since lane following and flag navigation is implemented as simple obstacle avoidance. A shortest path algorithm is implemented in order to obtain a shortest path to the destination, the GPS waypoint. This path is continuously updated as the robot traverses the course and obtains new sensor data.

The total expenditure of the team was much lower than expected this semester. Many of the parts that we expected to use were not needed for the current design. In fact, the most expensive aspect of this project was the registration fee for the competition itself. Before can be seen the current finance report for this last semester.



Finance Report -- Autobot

Electrical	**Budgeted**	**Spent**	**Remaining**
Female Sevro Pins	$10.00	$10.00	($0.00)
Sub-Totals	$10.00	$10.00	$0.00

Mechanical	**Budgeted**	**Spent**	**Remaining**
3D printed parts NOT APPROVED!	$50.00	$0.00	$50.00
Blinky Light Glass Lense	$50.00	$49.60	$0.40
Sub-Totals	$100.00	$49.60	$50.40

Miscellaneous	**Budgeted**	**Spent**	**Remaining**
estimate of total travel expenses for the com	$550.00	$0.00	$550.00
Registration	$300.00	$300.00	($0.00)
Work Ethic Stimulant	$20.00	$0.00	$20.00
Sub-Totals	$870.00	$300.00	$570.00

Totals	$980.00	$359.60	$620.40

Current Expenses

Category	**Item Detail**	**Price**	**Date**
Blinky Light Glass Lense	ordering of	$49.60	2014-04-24 18:00:14
Registration	payment	$300.00	2014-04-24 18:00:49
Female Sevro Pins	purchasing	$10.00	2014-04-24 18:02:01

Figure 94: Finance Report

Conclusion/Summary

The robot named “Bishop” is, after much hard work this semester, considered competition ready. The robot can autonomously navigate to a GPS waypoint while avoiding obstacles and staying within painted lines. This is accomplished through a using a variety of components. The updates done mechanically increased the stability of the robot and its acquired data. Electrically, a boost converter was exchanged for the previous design. This increased the efficiency of the robot and lowered the power consumption. New batteries were also added. Many revisions were made to the code, including updates to the navigation, GPS, camera, and IOP interface code. These changes allowed the robot to calculate an efficient path to the waypoint.

Another goal for this semester was to improve the ability to test and monitor the robot. To accomplish this a volt meter was added in parallel with the batteries. This allows the team to view the voltage as the robot is running, enabling the team to shut the robot down if the voltage dips too low. The code was also updated to be modular to increase the ability to test the code without all of the components included.

The documentation was improved mechanically, electrically, and coding. Mechanically, the models for each of the components were updated to match the robot. At the beginning of the semester, there was minimal electrical documentation. This was improved upon by using both a hand drawn and a computer modeled schematic. The software code was also modeled using various flow charts.

The team has traveled through many design iterations and this semester has reached a polished design. Each aspect of the robot is working together to reach the final goal and to leave the robot in an open condition for future design work. While there is always room for improvement, Bishop has reached a point of completion that leaves the team comfortable with its ability to compete in the upcoming IGVC.