

Lawrence Technological University

Vehicle Name: **MASK**



Date Submitted: 5/15/2017

Team Captains:

Mark Kenney (mkenney@ltu.edu)

Adam Drotar (adrotar@ltu.edu)

Kristin Jordan (kjordan@ltu.edu)

Shuwei Xu (sxu3@ltu.edu)

Team Members:

Alec Wendt(awendt@ltu.edu)

Chris Suchezky(csuchezky@ltu.edu)

Ryan Moreno(rmoreno@ltu.edu)

Ryan Pizzirusso(rpizzirus@ltu.edu)

Ryan Woods (rwoods@ltu.edu)

Bryan Walther (bwalther@ltu.edu)

Lacy Pyrzynski(lpyrzynsk@ltu.edu)

Michael Dupuie II(mdupuie@ltu.edu)

Thomas Weeks(tweeks@ltu.edu)

Syed Athar(sathar@ltu.edu)

Zachary Carey (zackalak@gmail.com)

Ryan Pizzirusso(rpizzirus@ltu.edu)

Faculty Advisor:

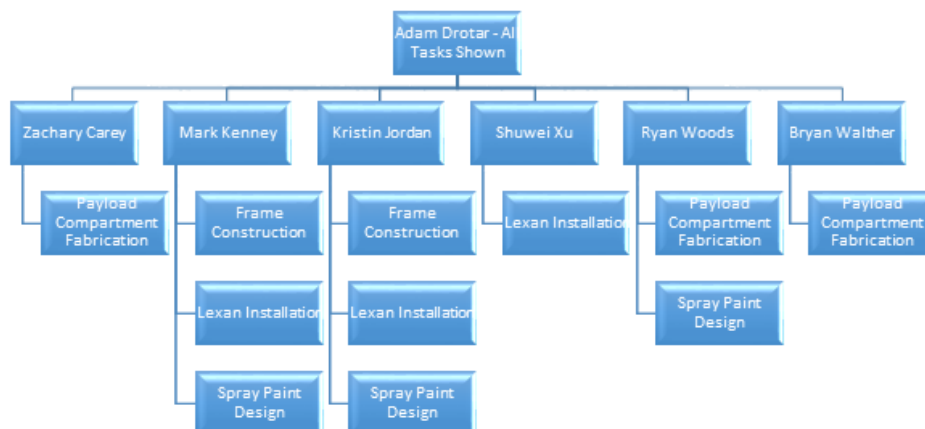
Dr. Giscard Kfoury(gkfoury@ltu.edu)

Introduction

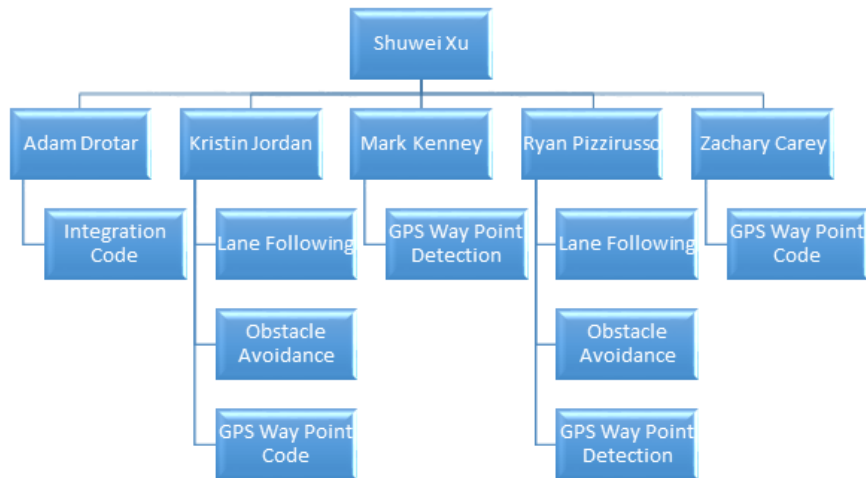
This past year the Intelligent Ground Vehicle Competition (IGVC) Team designed, built, and programmed a new robot to compete at the 2017 Intelligent Ground Vehicle Competition. The design for this year includes a rear differential drive, binocular stereo vision, and much more. The program uses both the lidar and views from the camera to follow lanes and avoid obstacles. Additionally, the code uses a compass that is attached to a Pololu A* board to determine movement for reaching specific GPS way points. The intent of this report is to provide the reader with an overview of the IGVC competition, Lawrence Tech’s team, our robot design, and the tasks we have completed for this project. Our vision for the IGVC team is to be a well established organization, composed of multifaceted members, that operates with a high quality design process that produces top ranking robots in the Intelligent Ground Vehicle Competition every year. The mission of Lawrence Tech’s IGVC Team is to design and build an autonomous robot to successfully traverse an outdoor obstacle course and navigate to GPS waypoints.

Organization

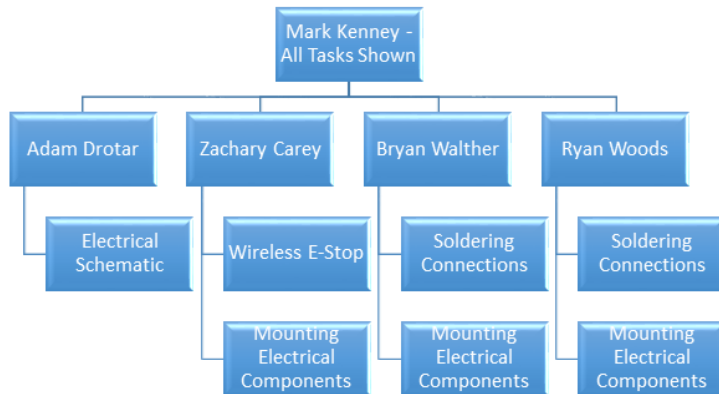
Mechanical Team



Programming Team



Electrical Team



Design Assumptions and Design Process

Simplicity

- A major driving principle for the control strategy and mechanical design was simplicity. The team aimed to create a design that would accomplish our necessary requirements in a fashion that would make the design easy to fabricate, control, and program.

Safety

- The robot must be safe in terms of controllability and operation. The electrical system must allow for a swift shutoff with both a manual and wireless switch (requirements detailed in ‘Design | Requirements | Safety’).

Performance

- We aim to design a robot that will fulfil the performance requirements and surpass the previous year’s team by qualifying in the competition and completing the first course.

Constraints

- Weather conditions may range from cloudy, rainy, and windy, to sunny and hot.

- The vehicle must be transportable to competition via a trailer.
- The vehicle should be transportable through a standard door frame.
- The control system cannot allow the robot to travel faster than the maximum speed or slower than the minimum speed.

Research

2015: Turtlebot (Made it through qualification)

- Used wheelchair hub motors for simple powertrain design
- Large size complicated transportation
- 3D printing for parts resulted in quick and cheap part builds

Bigfoot 2: Lawrence Technological University

- Completed all obstacle courses
- Used Emgu CV for program
- Performed simulation using Unity game engine

Key Design Considerations from Research

- Mechanical
 - Use hub motors to simplify powertrain design
 - Design simple suspension system to eliminate unnecessary complexity
 - Steer the robot using differential drive for simple design and control
 - Utilize 3D printer for quick and cheap parts
- Electrical
 - Segregate systems in relation to communications and voltage requirements for simple debugging and implementation
- Programming
 - Design program modules in classes for easier debugging.
 - Use Emgu CV for image processing

Innovative Concepts and Technology

For the controls strategy, the team investigated a rear differential drive system with a single front-caster, a tricycle drive system, and an ackermann steering system. Ultimately, a rear differential drive system was chosen as it provided a reliable controls platform that is simple to use, program, and design from a mechanical standpoint. Below is the additional research that the team performed, taking into account pros and cons of each system.

Description of Mechanical Design

Overview

The materials chosen for the robot include aluminum, lexan, PVC, and 3D-printed plastic. These materials were chosen over steel, wood, and other substitutes because they provide suitable performance-to-weight ratio.

Frame Structure

Our chassis is fabricated from aluminum square tubing and angle brackets which reduces weight when compared to a steel chassis. Our structure can withstand wear and tear that we may encounter on the course (running into a barrel/fence/etc. Or rainy weather conditions). Welding aluminum is more difficult when compared to steel, but this risk is mitigated through communication with skilled professionals at the fabrication laboratory. We use blue spray paint and LTU logo decal to improve aesthetics and sponsor logos are placed on side of robot base.

Housing and Weatherproofing

Sensor Box is fabricated out of clear lexan which allows visibility of sensors, is weather resistant, easily accessible, and easily modifiable if a cooling fan is necessary. Exterior sensor mounts for cameras will be mounted using 3D-printed plastic material which allows for custom-casing. We can print multiple cases in the event that there is an issue with breaks since it is low cost. Benefits include ease of design change and fabrication can be completed quickly with minimal skills required to run machinery. PVC mast will serve as a mount to the cameras and lidar which are adjustable in height and width, robust under loading conditions, and weather resistant. Lidar will be mounted on the front of the robot on the base with a wide range view to be able to detect obstacles at various heights.

Structure Design

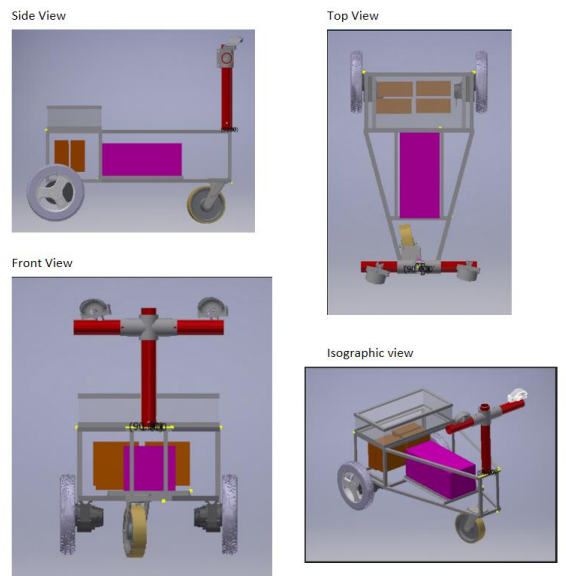


Figure 1 - CAD Design - Modeled using Autodesk Inventor

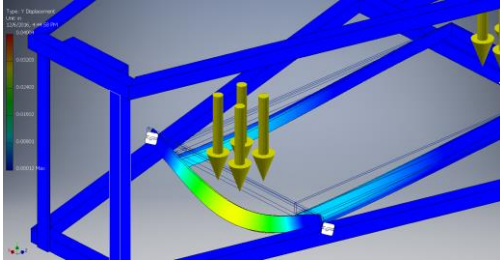


Figure 2 - FEA stress analysis of 20lb payload across aluminum chassis.

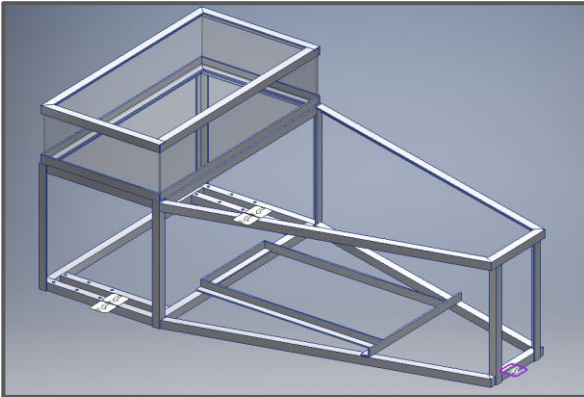


Figure 3 - Frame constraints for payload analysis.

Suspension

The team chose to mount the wheelchair motors to the chassis directly instead of using a mounting bracket or flexible suspension. This design decision was made in order to ensure that a welded bracket would not be a point of failure as it was in previous years.

Description of Electrical Design

Overview

A test robot was created to run simulations of electrical components. The components on the test platform are a direct relation to the Mask robot. Electrical components include a 12[v] battery, DC brushed motors, ESC, Laptop, Arduino UNO, GPS, IMU/Gyro, LIDAR/Ultrasonic, Relay, RGB LED, Switch, and a Camera.

On the MASK robot, the camera and lidar in front of robot, controllers, ESC, and IMU are in a sensor box, batteries in the base, and, motors are in the back.

Power Distribution System

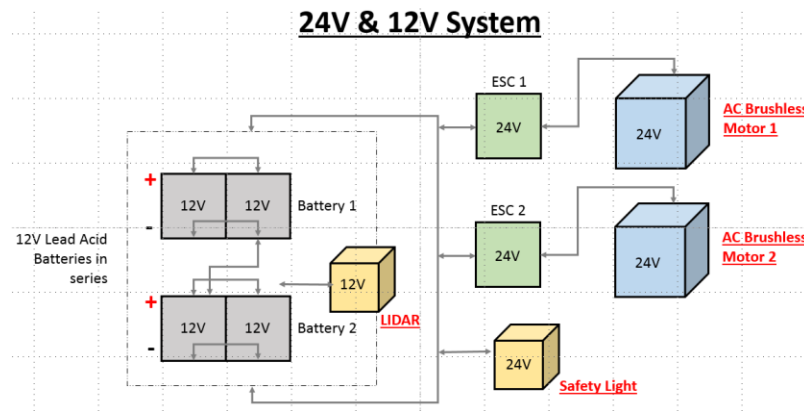


Figure 4 - 24V and 12V electrical schematic

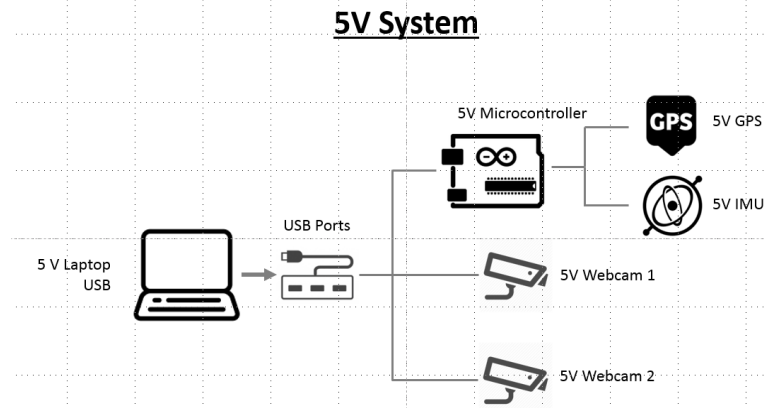


Figure 5 - 5V electrical schematic

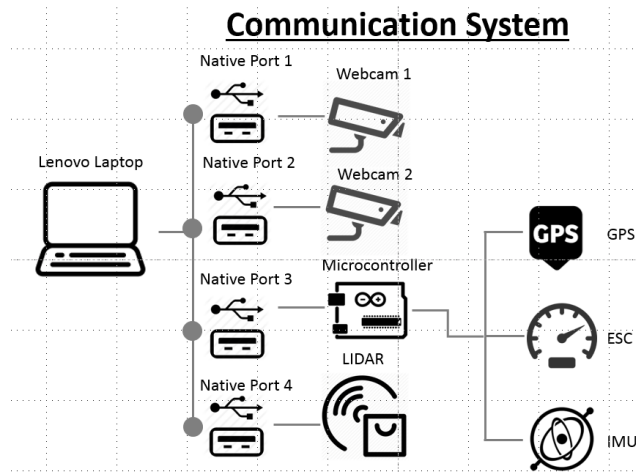


Figure 6 -Sensor communication network

Run Time Calculations

- Power supply capacity
 - 2x (18 [Ahr]) Batteries = 36 [Ahr]
- Current draw

- Motors: 21[A] max
- Lidar: 1[A]
- Computer: 2.25[A]
- Etc Components (microcontroller, sensors): <1[A]
- Total Current Draw: ~25[A]
- Continuous Average Run Time: $36[\text{Ahr}] / 25[\text{A}] = 1[\text{Hr}] 26[\text{min}]$

Electronics Suite Description

Our robot has a GPS which is capable of 66 Channels (22 for tracking, 66 for searching), 1[Hs]-10 [Hz] updates (1[s] - 0.1 [s]), and 1690 [f/s] maximum velocity tracking. We have cameras which support full HD 1080p. Our lidar has a 30[m], 270* scanning range, with 25[msec/scan]. Our actuators are brushless wheelchair motors which allows the robot to operate between 1-5 [mi/hr]. To control the motors, we have an ESC (Electronic Speed Controller) which has a rating of 90[A] max draw. We are using a Pololu A* which has 32[kb] flash memory (very small, but sufficient), 14 digital I/O pins (6 PWM outputs), and 6 analog inputs. The computer is battery operated and has an i7 Processor 1.80 [GHz]. For power, we are using lead acid batteries which are 12[v] with 18[Ahr]. These are connected 2 - series (to enable 24[v]) and 2 - parallel (double [Ahr] for longer run time).

Safety Devices

Our safety light operates at 24[v] and the wireless E-Stop is attached solely to the motor circuitry, so it will not disable the sensors or controllers. Per regulation it is hardware based (not controlled by software). The hardwire E-Stop is a push to stop, red in color with a minimum of one inch in diameter and is attached solely to the motor circuitry and also will not disable the sensors or controllers. It is easily identifiable and accessible, located in the center rear of vehicle 2-4 [ft] from the ground.

Description of Software Design

Overview

The microcontroller that we are using is the Pololu A*. We choose this controller board due to the specs that allow it to efficiently perform serial communication and motor control communication to our ESCs. As can be seen in figure 7, we are sending and receiving data from the computer, while simultaneously pulling information from the compass and cameras. After the computer process all information, our motor speed information is passed from the computer, to the controller, to the ESC.

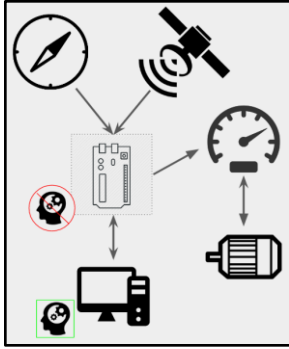


Figure 7 - Microcontroller communication design

Obstacle Detection and Avoidance

In order to detect obstacles and barrels on the lane, we are using the Hokuyo LIDAR processing library through the computer. Hokuyo LIDAR provides an open source C# library available to perform operations such as obstacle detection, 3D space scanning, and other distance detection processes/manipulation. We choose to use Hokuyo LIDAR because we inherited this LIDAR from last a previous team and this LIDAR perfectly fit in our competition environment.

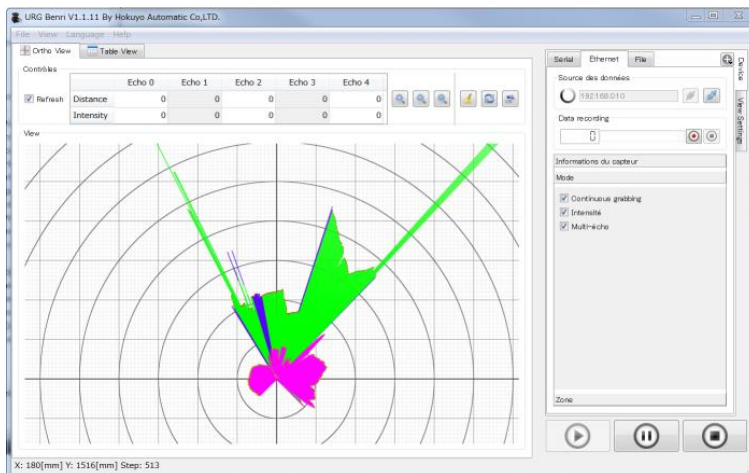


Figure 8 - Hokuyo LIDAR detect the objects around it.

Software Strategy and Path Planning

Computer vision programming with EmguCV:

In order to process the visual information that we are receiving from our dual HD cameras, we are using the EmguCV image processing library through the computer. EmguCV is an open source C# library available to perform operations such as filtering, image detection, color detection, and other visual processes/manipulation. We choose this interface due to its prevalence in the LTU Robotics curriculum. Furthermore, the color detection prowess is valuable to our core strategy.

Notably, we are not currently using stereo vision (dual camera processing often used to detect depth). Instead, we aim to gather data from each camera in order to determine which side of the robot is currently closest to a line and to more effectively stay within the lane. E.g. if the robot is on the left side of the lane and needs to avoid a barrel, we can use the right camera to pick up white line on the right side of the lane and continue. We feel that this is a strong preliminary strategy and may update/improve this our algorithms to integrate more advanced techniques closer to the competition.

GPS Waypoint Calculations:

For the GPS way point detection code the latitude, longitude, and head angle are sent through serial communication to the computer, which then does the following calculations. These calculations are used to determine the distance between the robot's current location and the target waypoint as well as the adjustment angle needed for the robot to directly face the waypoint. The movement program is still in the process of being completed.

$$Distance = (\Delta Longitude^2 + \Delta Latitude^2)^{-2}$$

$$Angle = \tan^{-1}\left(\frac{Latitude}{Longitude}\right)$$

$$AngleToTurn = currentAngle - Angle$$

if $(180 > AngleToTurn > 0) \rightarrow$ turn right
 if $(0 > AngleToTurn > -180) \rightarrow$ turn left

Figure 9 - GPS Waypoint Calculations

Map Generation

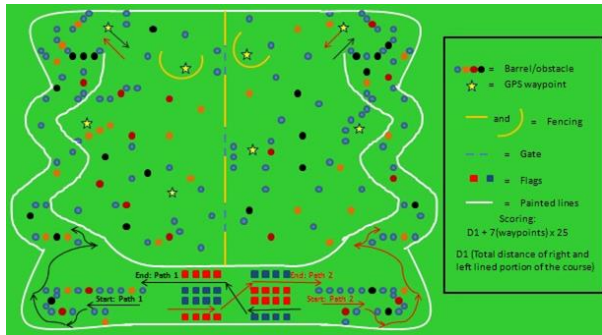


Figure 9 - Sample Course Map

Goal Selection and Path Generation

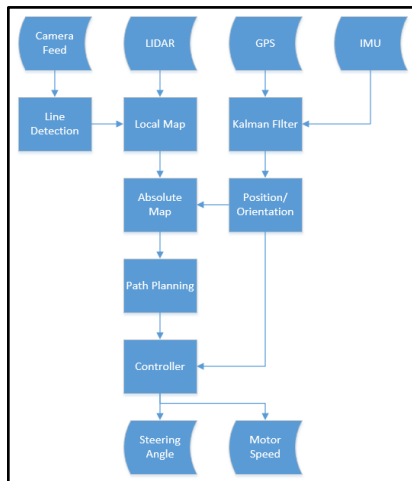


Figure 11 - Logic Flowchart

- Microcontroller starts receiving data from sensors and send data to computer
- Computer receives data from microcontroller and camera
- Computer calculates data and finds shortest path
- Computer gives command to microcontroller
- Microcontroller sends moving instructions to motors
- Battery provides power for motors to move

Additional Creative Concepts

- Create a complete map and path planning
 1. Open CV - gather image data - Lane detection: To mark the lane as white and everything else as green
 2. Or Emgu CV (for windows use)
 3. Remote control for manual controlling
 4. Lidar data: To detect obstacle or traffic cones
 5. GPS/IMU data: To guide the vehicle to follow the lane
 6. Ultrasonic sensors: To detect close obstacle

Software Architecture

- Programmed in C# using Windows Presentation Format for the Graphical User Interface
- Use MVC structure (Model-View-Control)

Figure 12 demonstrates how our software architecture relies on all sensor data in order to make a turning decision. Redundancy is used throughout the system and during decision making in order to ensure that we are making the optimal choice in direction at any given moment.

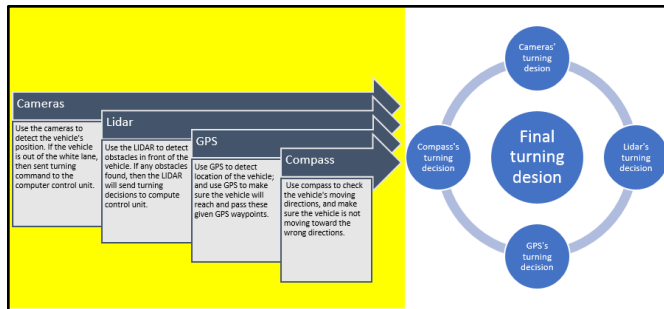


Figure 12 - Software architecture takes into account all sensor data in order to promote redundancy and make a final turning decision.

Description of Failure Modes, Failure Points, and Resolutions

Vehicle Failure Modes and Points, and Prevention Strategy

There are many risks to assess when managing a project of this size. It is important to note that there are safety risks.

- **Welding/Fabrication**
 - The welding process can be dangerous. As such, we made sure complete the necessary training in order to enter the fabrication lab, used the right safety equipment, and made sure to utilize our fab lab personnel resources for complex fabrication tasks.
- **Stability and Control**
 - It is possible for the robot to lose control and crash into a person. With this in mind, we made sure to follow the speed and e-stop requirements of the competition in our electrical and programming design.

Vehicle Safety Design Concepts

Simulations Employed

Simulations in Virtual Environment

Figure 13 shows our current EmguCV and Windows Forms interface. Through the use of ImageBoxes and Capture objects, we can display our camera data to the user and run multiple filters and manipulations on the data. This feedback provides us valuable insight into what the

camera sees, and allow us to tailor or algorithms to effectively use this data. The EmguCV Capture object allows us to gather and process image data in the form of pixels, each of which have a red, blue, and green component, or RGB value.

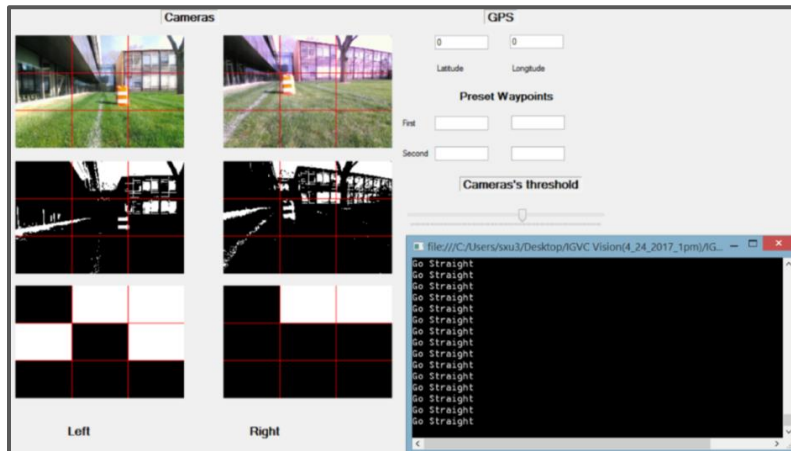


Figure 13 - EmguCV integration with WinForms interface and ImageBox/Capture objects

Theoretical Concepts in Simulations

When processing the image, we are following the following process:

1. Filter colors into black and white
2. Strip noise through threshold adjustment on white detection
3. Further sharpen image through noise reduction operations

Performance Testing to Date

Component, System, and Subsystem Testing

When testing our software design, we created a few metrics that can be seen in the following procedure:

- Test cases
 1. Motors test
 - a. Connect single motor to microcontroller;
 - b. Send three signals to motor to let motor at varying speeds (fast, normal, slow) [simulates turning].
 2. Serial communication test
 - a. Connect computer and microcontroller;
 - b. Sent char 'A' from microcontroller to computer;
 - c. Display 'A' on computer (verify proper communication)

3. Emgu CV Vision test
 - a. Test color detection in bright lighting conditions
 - i. Turn on lights in lab on floor with high reflection
 - b. Test color detection in dark lighting conditions
 - i. Create dim lighting condition in lab and test white detection on varying shaded backdrops
4. Integration test
 - a. Create an interface to display:
 - i. GPS data
 - ii. Camera data
 - iii. Lane detection
 - iv. Object detection
5. Lane detection:
 - a. Preliminary:
 - i. Run small-scale robot indoors on small test track
 1. White lines/black backdrop
 - b. Advanced:
 - i. Create white spray painted lines in grass per competition spec
 - ii. Run robot with on test track

Initial Performance Assessments

Vehicle Performance to Date

Currently, the robot is capable of being powered and controlled manually. Basic testing is done to do lane navigation, obstacle avoidance, and GPS navigation, however the integration of these modules is not complete. In the future, we plan on improving GPS waypoint following program, complete outdoor testing for individual components, and integrate vision system, control system, and other sensor data. Each sub-system is currently functioning independently.