SparkFun AVC Team

December 9, 2014

Dr. John Glover

University of Houston

4800 Calhoun Rd

Houston, Texas 77004

Dear Dr. Glover,

We are the SparkFun AVC senior design team at the University of Houston. The purpose of this letter is to inform you that the proposed target objective for fall 2014 has been met and the target objective for spring 2015, our vehicle competing in the SparkFun Autonomous Vehicle Competition, is on schedule. The team consists of Jorge Rivera, Alexander Bello, Gilberto Martinez, and Jorge Revuelta.

As you may already know, for the fall of 2014, the autonomous vehicle is currently in its initial design stages with the primary focus of achieving basic autonomous capabilities through the integration of various sensors. As of December 7, 2014, we have had extensive progress and have been able to reach our target objective. The team has been successful in acquiring data with all its sensors and has successfully translated all the information into servo commands in order to achieve the desired vehicle behavior. In this document you will find our full ADDIE components.

Thank you for your consideration in reviewing this report. If you have any questions or concerns, please contact us at jrivera23@uh.edu.

Sincerely,

SparkFun AVC Team

Your discussion of target objective, goal, objective, task, etc. is confusing because you seem to use these terms interchangeably. I want you to be very specific and clear about each of these things. They are not the same. See comments.

**SparkFun AVC**

Team Members:

Jorge Rivera

Alexander Bello

Gilberto Martinez

Jorge Revuelta

*Final Technical Report*

Sponsor:

Dr. John Glover

**Abstract**

The goal of the project is to place in one of the top three positions at the 2015 SparkFun Autonomous Vehicle Competition. The target objective for the Fall 2014 semester is to have an autonomous vehicle capable of performing basic navigation maneuvers with the aid of a graphical user interface using acquired data measurements, such as color, direction, and distance from its onboard electronics. Current results include a vehicle that is capable of navigating autonomously between GPS coordinates within 2 meters of desired waypoints. Servo control is performed by a pulse width modulated signal and image processing has given the capability of being able to detect objects at a rate of 10 frames per second to perform maneuvers based on the centroid information of different colors, and an ultrasonic sensor that provide distance measurements. Future work includes implementation of a navigation algorithm that accesses all the sensor data to perform various tasks as the vehicle navigates between GPS waypoints in a track mockup. Demonstration of sensor capabilities and maneuverability is demonstrated and operated through the use of a graphical user interface. Problems associated with the project include possible poor image processing due to fluctuations in lighting conditions, which we expect to address next semester. In addition there is the possibility of poor GPS navigation due to a weak GPS lock or interference from strong electromagnetic fields. Current material expenditures consist of $587.87, which includes temporary wireless communication devices, sensors, and the vehicle itself, while personnel labor expenditures reached $18,225.

**Background and Goal**

This document contains a detailed description of the deliverables completed during the fall 2014 semester including various sensor integrations and vehicle modifications. It also includes the engineering constraints, budget, and target objective of the project.

The goal of this project is to design and build a fully autonomous vehicle capable of placing in the top three spots in the SparkFun autonomous vehicle course in Niwot, Colorado on June 20, 2015. The target objective completed for the fall 2014 semester included building a vehicle that is capable of receiving GPS, magnetometer, ultrasonic, and image recognition data and using it to perform basic maneuvering tasks with the aid of a command interpreter graphical user interface. The overall target objective for the project is to have an autonomous vehicle capable of navigating a mockup course without a command interpreter, demonstrating that it is capable of prioritizing sensor information.

The SparkFun competition has four different vehicle classes. These include the micro, doping, non-traditional locomotion, and peloton classes. The micro class vehicles are smaller and must have a value less than $350. The doping class includes vehicles that weight more than 25 pounds and have a value more than $1000. The non-traditional locomotion includes vehicles similar to self-balancing pogo sticks, walker, and motorized hamster balls. We will be competing in the peloton class, which includes vehicles that do not fit into the other classes. The peloton class also tends to have the most vehicle entries. The competition course shown in figure 1 shows the various obstacles the autonomous vehicle will have to clear. There is a five-minute time limit that the vehicle has to complete the course. The team starts with 300 points and loses a point for every second the vehicle remains on the course. The course includes obstacles such as stanchions, barrels, ramps, hoops, and four additional vehicles. The team will earn bonus points for navigating through the barrels without making contact, successfully jumping the ramp, and navigating through the hoop. The following figure demonstrates SparkFun 2014 track and obstacle placement.

Figure 1. SparkFun autonomous vehicle course with barrels, hoops, stanchions, and ramp.

The goal of the project is to design a vehicle that is capable of placing in the top three positions in its competing class in the 2015 SparkFun Autonomous Vehicle Competition. This also includes bringing recognition to the University of Houston and the Electrical and Computer Engineering department by participating in a sponsored competition that promotes the winners of the competition.

**Problem**

The problem associated with our project is zero visibility has been raised in the SparkFun competition on behalf of the University of Houston through a senior design team. This team will be the first to compete and represent the University of Houston Electrical and Computer Engineering department in the competition.

**Need**

A fully autonomous vehicle capable of navigating the SparkFun course using GPS, image processing, and ultrasonic sensors. The deliverables for the fall 2014 semester includes the implementation of an array of sensors that we believe will be essential to successfully completing the course. As aforementioned, the target objective for the spring 2015 consists of using these sensors to navigate a mockup course. This semester is the first step toward prioritizing sensor information and developing a navigation algorithm to reach our final goal.

**Significance**

The University of Houston will gain recognition and continue to attract prospective engineering students. The team will also be able to observe the aerial competition and gather intellectual resources for Dr. Glover and future senior design groups.

**User Analysis**

The immediate users of the device will be the SparkFun AVC senior design team and the level of expertise consists of basic knowledge of microprocessor programming. For the fall 2014, the interfacing method consists of a graphical user interface to initiate navigation commands, while the end project in the spring 2015 will consists of a single push to start button.

**Overview Diagram**

In the following figure we introduce the overview diagram of our project. It captures the essence of what we have designed the vehicle to be able perform, which we believe to be one of the best approaches for successfully completing the track within three minutes. This is while avoiding collisions and with the aim of completing all the tasks along the way.

Figure 2. Project overview diagram.

In Fig. 2, we indicate that the vehicle will be given autonomous capabilities through the Raspberry Pi and Arduino Mega central controllers. The Raspberry Pi is responsible for performing image processing to detect colored blobs of the four obstacles and transmitting control commands to the Arduino. The Arduino is responsible for processing ultrasonic distance measurements to avoid near collisions, GPS/Directional positioning for waypoint navigation, and generation of servo pulse width modulated (PWM) control signals for steering.

**Target Objective and Goal Analysis**

The following figure, Fig. 3, provides the goal analysis for the fall 2014 semester.

Figure 3. Goal Analysis

The target objective consists of using the interfaced sensors with the processors and with the aid of graphical user interface to perform basic maneuvering tasks. Optimization of the sensors and development of a navigation algorithm for the vehicle is left for the spring semester. There were five objectives that were worked in parallel. The five objectives needed to be able to accomplish the target objective included generating control signals for the servos to execute forward, reverse, and steering control. In addition to interfacing a GPS sensor, infrared reflectance (IR) sensors for line detection, ultrasonic sensors for distance measurements, and a camera for object/color detection. For the GPS sensor the vehicle is able to move within three meters accuracy of a set waypoint set by the user. The IR sensor proved to be inefficient for our purposed and therefore will not be used for line following due to the low height tolerance of its recommended optimal operating conditions. However, if line following is deemed advantageous in the spring 2015, line following will be implemented using image processing with the Raspberry Pi camera module. Object and color detection was implemented and the vehicle was capable of moving towards and away from colored objects. The sonar sensor was interfaced with the Arduino and was capable of making distance measurements. Next semester the ultrasonic sensor will be implemented for vehicle avoidance as five vehicles run the course at the same time.

**Engineering Specifications and Constraints**

The vehicle has a constraints set by the competition and the class that the vehicle will be competing in. The vehicle must be under 25 lbs and under $1000. The project itself can cost more than $1000 but there must be less than $1000 worth of electronics on the actual vehicle during the competition. The vehicle itself cannot contain any wireless communications excluding GPS. Additionally, the vehicle must be tested in outdoor environments because the competition is outdoors.

The engineering standard used in this project is IEEE 802.15.4 which handles the wireless communications between the Arduino and computer. According to FCC regulations the transmitter must be less than 1 W. The transmitter currently used is 60 mW and is well within FCC regulations. Additionally, this transmitter will not be used in the final vehicle as it is only for debugging purposes.

**Objectives/Milestones**

**i. Distance Measurements**

The planned method to avoid vehicle-vehicle collision is through the implementation of several ultrasonic sensors. Currently, a single sonar sensor has been integrated to the vehicle. The sonar sensor used in this project is a Maxbotix LV-MaxSonar-EZ1 sonar range finder. The sensors operates with a voltage from 2.5 V to 5 V with a current draw by the sensors is 2 mA. The sensor returns the distance as a signal to the analog pin of the Arduino that is directly proportional to the distance measured in inches. The resolution for the distance measurements is 1 inch, with a maximum measurement of 255 inches. A constraint that should be taken into account is that these sensors cannot detect objects closer than 6 inches and they are not equipped to be used in wet conditions.

**ii. Line Following**

Initial line following experimentation began with the implementation of an infrared reflectance sensor to detect a white line over a gray tone background. The sensor used was the QTR-8RC Reflectance Sensor Array. The operating voltage for the sensor is 3.3 V to 5 V and the supply current draw is 100 mA. The IR sensor can detect black from white by using the reflectance detected and sending a digital value corresponding to the time it takes for the capacitor to discharge. Operating constraints of the sensor included a maximum height from the line of 9 mm and required use of indoor settings. It was concluded that the IR sensors could not be used for line following due to the height constraint and the outdoor conditions.

**iii. Image Processing**

Image processing is performed on a Raspberry Pi operating at a processing speed of 700 MHz using a Raspberry Pi CSI camera board module. The current setup consists of using the camera as a live video feed that is used to capture frames that are then analyzed using a series of algorithms to extract the desired information. The algorithms used for image processing consists of the Python libraries from the open source Open Computer Vision (OpenCV) project. As OpenCV is designed to operate natively with video for Linux (V4L2) USB cameras, the Raspberry Pi had to undergo several modifications such that the CSI camera module could operate with the libraries. This included the installation of a custom user video for Linux (UV4L) such that the Raspberry Pi recognizes the camera as a USB device. Once the camera had been successfully interfaced, the resolution could be modified to achieve a desired frame rate. Initial testing of the Raspberry Pi Camera provided the following results as seen in the following table.

Table 1 Frames per second based on resolution

|  |  |  |  |
| --- | --- | --- | --- |
| Resolution | Frames | Time [s] | FPS |
| 640 px x 480 px | 500 | 49.296 | 10.143 |
| 320 px x 240 px | 500 | 16.907 | 29.574 |

Table 1 demonstrates initial frame rates obtained at two different resolutions. However, these results only demonstrate the frame-capturing rate from a live video feed obtained from the Raspberry Pi camera without performing image processing, which in turn lowers the frame rate considerably. At a resolution of 640 pixels by 480 pixels, the camera operates at about six frames per second. Lowering the resolution to 320 pixels by 240 pixels resulted in a processing rate of ten frames per second. These frame-rates were obtained from the GUI interface developed for the fall 2014 semester that provides diagnostic information from the interfaced devices. Image processing information is transmitted through a dedicated communication line between the Raspberry Pi and Arduino with a UART interface.

As aforementioned, the algorithms used for image processing come from the OpenCV Python libraries. As the objective for color and object detection is to focus on the largest object on the cameras field of view. This is done because the closest object is the highest priority as opposed to an obstacle that is small and in turn far away. Captured frames are in the Blue Green Red (BGR) color space. These frames are then converted to Hue Saturation Value (HSV) color space arrays for the image processing algorithms. In HSV, the colors are primarily identified within the hue parameter, while in BGR it depends on the mixture of the three primary colors. This first step in recognizing an object of interest is to convert from the BGR to HSV colors space, which is done using one of the OpenCV algorithms. The second step is to threshold the image by providing a range of color space values for the color of interest. The third step consists of finding the contours of the pixel groups identified by the threshold. The contours are then analyzed to extract pixel area and centroid information of the largest collection of pixels. This has been achieved for red, blue, green and yellow color ranges. However, lighting plays an important role in detection, which can result in either good or poor object tracking. Good object tracking consists of centroid that remains steady as an object is moved. Poor tracking occurs because of hue color shifting by the lighting environment or through the fluctuations of brightness levels. This results in a centroid that jerks around sporadically or the color being ignored by the image processing algorithms. Lighting implications can be seen through a provided object video debugger provided by the image processing code running on the Raspberry Pi that can be initiated and terminated with a GUI command.

Movement away from an obstacle is achieved by transmitting the centroid and area information to the Arduino, which in turn controls the servos for steering. This capability has been tested at low speeds, successfully being able to avoid the objects of interest with the current ten frames per second processing speed in good lighting conditions. The primary implementation of avoidance is performed according to the location of the centroid with respect to the camera’s field of view. To avoid an obstacle, the vehicle is maneuvered using a proportional controller such that the centroid is moved either to the far left or far right. Once the object is out of sight, the avoidance algorithm is ended. Similar to moving away form an obstacle, the capability of moving towards blue or green has been tested at low speeds. To move towards the obstacle, the vehicle is maneuvered using a proportional controller such that the centroid is moved within a desired center threshold. The vehicle is then capable of navigating towards what it perceived to be the center of an object.

**iv. Vehicular Motion**

The vehicle is controlled through the Arduino. There are two motors on the vehicle, a steering servo and the main forward and reverse locomotion motor located at the rear of the vehicle. The steering servo is directly configured with the Arduino and is controlled through a pulse width modulated signal of ms ms. This controls the steering servo through its full range of motion. The rear main motor, named the vehicle throttle, is controlled through a pulse width modulated signal and an electronic speed control. A pulse width modulated signal is sent to the electronic speed control (ESC) and the ESC converts the signal to a high frequency signal to control the speed of the rear main motor. Using two pulse width modulated signals, the steering servo and vehicle throttle is controlled by the Arduino. This capability was tested by cycling the vehicle through different steering angles while the throttle was tested by throttling to different values.

**v. Waypoint Navigation**

Waypoint navigation is performed on the Arduino Mega, which navigates according to GPS, magnetometer, and gyroscope sensor values. The magnetometer and GPS were first interfaced with the Arduino. After interference issues arose with the magnetometer, the gyroscope was added to primarily rely on gyroscope readings for short-term navigation. The basic principle behind the waypoint navigation system is as follows. A cardinal heading towards a waypoint is computed using the current latitude and longitude coordinates from the GPS and the vehicle heading is adjusted towards the computed waypoint heading using the magnetometer and gyroscope.

The GPS Module is the LS20031, this sensor is capable of updating at 10 Hz and has an accuracy of about 2.5 – 3 meters. The GPS is interfaced through a UART Serial Port on the Arduino Mega. NEMA sentences are transmitted from the GPS and are interpreted by the Arduino to obtain latitude, longitude, speed, heading, and satellites in view. Currently the GPS updates at 5 Hz due to increased accuracy as opposed to 10 Hz.

The magnetometer is the HMC5883L, this sensor is capable of updating at up to 140 Hz and is interfaced through the I2C bus. Currently the magnetometer is updated at 75 Hz. The magnetometer measures the vector direction of the Earth’s magnetic field along the X, Y, and Z axes. A tangent trigonometric function is used to relate the X and Y axe to the vehicle heading. Power lines, surrounding metal objects, and vehicle electronics can all cause noise in the magnetometer reading. Early in the development of the navigation system, this posed a huge issue and caused sporadic steering errors in vehicle navigation. Nevertheless, the magnetometer is an essential component as it provides absolute heading, which must be used to navigate towards a waypoint.

The gyroscope is the ITG3200, this sensor is capable of updating at 140 Hz and is interfaced through the I2C bus. Currently, the gyroscope is updated at 75 Hz. The gyroscope measures angular velocity along the X, Y, and Z axe. Only the Z-axis information is used and if the angular velocity is multiplied by the change in time since the last update, an angular heading change can be computed. This angular heading change is added to the current vehicle heading which allows the vehicle heading to change according to the gyroscope angular velocity in the Z-axis.

An onboard filter is implemented to perform a sensor fusion of the information from the magnetometer, gyroscope and GPS. This onboard filter currently produces a filtered vehicle yaw value, or vehicle heading. The filter combines short-term data such as angular change,

, (1)

where and are filter gains, is gyroscope angular change, and is the magnetometer angular change. In the current filter, and . This insures that the short-term heading relies mostly on the gyroscope readings. The filter also combines the various headings computed by the magnetometer and GPS, when in motion, by

, (2)

where is the magnetometer heading, is the GPS heading when the vehicle is in motion, and and are filter gains. All filter gains are adjustable through the Guantanamo GUI built for this project. Using this onboard filter, the vehicle heading found more reliably than if dependent on each sensor independently.

The waypoint navigation system is capable of navigating with 2 meters of a waypoint. Waypoint navigation is initiated through the Guantanamo GUI. Once the vehicle reaches a waypoint a “waypoint reached” is transmitted and the GUI can chose to transmit another waypoint for multiple waypoint navigation or allow the vehicle to return to an idle state.

**vi. Guantanamo GUI**

For the target objective of this project, the vehicle required a command interpreter to allow the user to send commands to the vehicle, which would cause the vehicle to perform different tasks. The tasks explicitly stated in the scope of this semester were, GPS waypoint navigation, avoid and going towards the colors, and control of the vehicle steering servo and throttle. This vehicle needs to be optimized and further developed next semester, the ability to have a powerful GUI to control and monitor the vehicle will allow the project to progress smoother by significantly easing debugging and vehicle monitoring tactics. A GUI was created for this project in Visual Studios, which monitors the vehicle and sensors, and allows commands to be sent to the vehicle. The GUI was codenamed Guantanamo GUI, named after the vehicle.

The vehicle and PC communicate through a 2.4GHz wireless communication device called XBees. This allows the vehicle to send and receive information through a UART port and allows the computer to send and receive information through its UART COM port. The GUI built for this project allows the user to connect to a UART port on the computer. The GUI then interprets the information on the UART port and presents the information on a magnitude of different windows.

The Guantanamo GUI currently shows data from the magnetometer, GPS, ultrasonic sensor, Raspberry Pi image processing, steering and throttle servo positions, and vehicle diagnostics such as uptime, sensor updating frequency, and vehicle state. The GUI can also send commands such as, navigate waypoints, avoid and go to colors through image processing, calibrate sensors, set steering and throttle, and configure onboard filter gains. Additionally, the GUI can plot special data from the Arduino called “Probe” data, which is currently set to compare gyroscope, magnetometer, and vehicle heading to better set filter gains.

**Budget**

 This semester the team has spent a total of $587.87 on the autonomous vehicle as seen in Table 2. This includes the cost of some devices that will not be in the final vehicle, such as the XBees and reflectance sensor arrays.

Table 2. Materials Budget


In addition the personnel labor cost was $18,225 as seen in Table 3.

Table 3. Labor Budget


Table 4 shows the total expenditures to date in white. The materials highlighted in yellow indicate the possible anticipated expenditures for the following semester. The Linux image-processing unit might be necessary if the current Raspberry Pi and Raspberry Pi camera module board proves to be underpowered. An alternative to this problem is to overclock the raspberry pi from 700 MHz to 1 GHz. The TIVA C or PixHawk are possible alternatives to the Arduino Mega that is running the vehicle at the moment. Some advantages of the Tiva C microcontroller is its capability of running a real time operating system. An advantage to using the PixHawk is that it includes all the necessary sensors to run the autonomous vehicle.

Table 4. Current expenditures for this semester including possible expenditures for the following semester.

**Conclusions**

Sensor and processor integration has been achieved through the use of the Arduino as the central controller of the system, with the Raspberry Pi operating as a slave image-processing sensor. This was achieved through the use of I2C and UART communication interfaces provided by the Arduino and Raspberry Pi. As mentioned in the report, the target objective of performing waypoint navigation and navigating to and away from objects was accomplished using the GUI. The GUI is constantly being expanded along with the continual optimization of the vehicle. GPS navigation has proven to be reliable when a good GPS lock is obtained resulting in navigation within 2 meters of desired waypoints. Image processing has yielded the desired 10 frames per second when processing individual colors, however, measures to increase frame rate is an ongoing objective. Basic object maneuverability has been achieved at low speeds as we still lack a fundamental navigation algorithm, which is scheduled to be implemented in following semester. The navigation algorithm is the final goal of the spring 2015 semester to successfully be able to navigate a mockup course using the currently implemented onboard sensors.