THE MOROCCAN MONSTER

Faculty Advisor Statement:

I, Dr. Jennifer Wang of the Department of Mechanical Engineering at The College of New Jersey, certify that the design and development on the Moroccan Monster by the individuals on the design team is significant and is either for-credit or equivalent to what might be awarded credit in senior design course.

5/15/2016

Introduction:

The College of New Jersey is happy to present the senior team consisting of 5 peoples with different major. Each of them was valuable for creating the *Moroccan Monster* that will compete in the Intelligent Ground Vehicle Competition 2016. The *Moroccan Monster* followed every conditions discussed in the rules. The car has one laser sensor or LiDAR used from the 2007 team, one camera, Raspberry Pi as the micro-computer and Sabertooth. The 2016 car is very different from the previous car from 2014 from every different aspect of it (hardware and software).

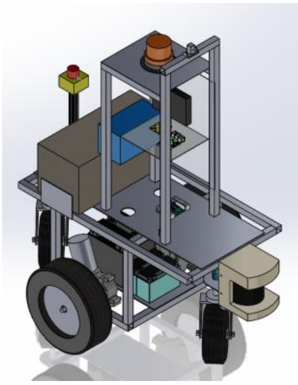

Figure 1: The final design of the car

I. Team Management

1. Team Organization

As discussed in the introduction, the 2016 team consist of 6 engineers from different major. Mehdi is the team leader/team manager. His job was to keep track of the project, guide his team members, communicate with the professors and staff, deal with the paper work and help his team members if they need it. Charlie Gordon is the assistant manager and mechanical engineer in this project. He dealt with the design and construction of the frame. Additionally, he dealt with the drive train without forgetting that he was helping and assisting the team leader for some decisions. Nicholas Cuzzola is the mechanical engineer in this team. He dealt with the robotics of the car such as camera mounting and the housing of the electronics. He also dealt with the mechanical analysis of the car. Eric Mauro is the electrical engineer. He dealt with the physical wiring and the wiring diagram. He also dealt with the programming of the sensors and the motor movement and path planning. Joseph Martinico is the computer engineer. He dealt with the image processing and mapping. Finally, Justin Cheng is the mechanical engineer in this project. He was more like a help for our team, especially to our computer engineer.

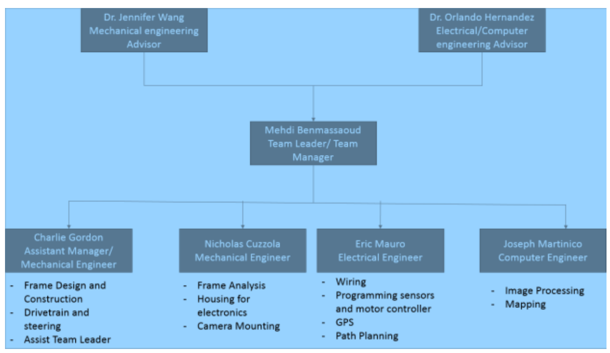

Figure 2: The structure of the team involved in the team

2. Time and Money Management

The team went through five different step in this project: Conceptual phase, planning phase, prototype phase, implementation phase and testing phase. The team spent about three months in the conceptual design to understand the project. In the planning design, the team spent about four months to plan to construct the car. Then 2 months for the prototype, 1 month for implementation and one month for testing.

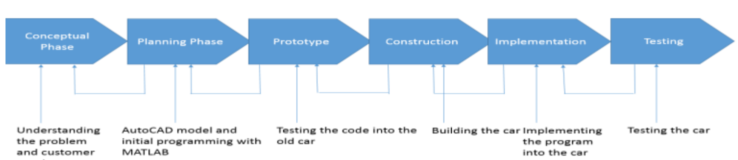

Figure 3: The Gantt chart of the project

Regarding the budget, the team had to be careful with managing the money especially that the dean of the school of engineering give $100.00 per each team. For the *Moroccan Monster* case, the team had an initial budget of $500.00. Even though, we had an opportunity to ask the dean for more money. The team needed to recycle most of the components such as the motors, LiDAR, motor controller and many other component as shown in the budget below.

Description	Supplier/Store	Quantity	Unit Price	Tax	Shipping	Total Price	Cost to Team
Mechanical:							
Front Pneumatic Caster Wheel with Suspension (210lb load capacity)	Amazon	2	$88.73	$0.00	$0.00	$177.46	$177.46
Multipurpose 6061 Aluminum rectangular tubing- unpolished: (72" x 1" x 1")	McMaster-Carr	10	$23.38			$233.80	$233.80
Electric DC Motors for driven wheels	NPC Hobby Robotics	2	$196.17			$392.34	$0.00
Blazer C48AW LED Strobe Light	Amazon	1	$17.99			$17.99	$17.99
Mechanical E-Stop- Baco	McMaster-Carr	1	$45.85			$45.85	$0.00
Sheet metal	McMaster-Carr	1	$219.92			$219.92	$0.00
Plexiglass	McMaster-Carr	1	$82.69			$82.69	$82.69
Computer/Electrical:						$0.00	
LMS 291-S05 Laser Measurement System		1	$3,411.00			$3,411.00	$0.00
Power Sonic PS-1270 F2, 12 Volt 7.0 Ah Rechargable Battery		1	$19.00			$19.00	$0.0
CS3 GP 12170, 12V 17Ah Rechargable Battery		1	$35.00			$35.00	$0.0
Sabertooth 2x25 (v2.00) Motor Driver		1	$125.00			$125.00	$0.00
Raspberry Pi 2 - B w/ Power Supply and Case	Amazon	1	$49.99			$49.99	$40.00
Adafruit Ultimate GPS Breakout- 66 channel w/ 10Hz updates- version 3		1	$39.89			$39.89	$39.89
Raspberry Pi 5MP Camera		1	$24.00			$24.00	$24.00
Flex Cable for Camera		1	$9.90			$9.90	$9.90
4-channel I2C-safe Bi-directional Logic Level Converter - BSS138	Adafruit	1	$3.95			$3.95	$0.00
Competition expense							
Hotel Room	Quality Inn website	2	$480.00			$960.00	$865.06
Registration	IGVC	1	$300.00			$300.00	$300.00
Total:							**$1,790.79**
					Total cost:	**$6,147.78**	
					Total spent:	**$1,790.79**	
					Saved:	**$4,356.99**	

Figure 4: The budget used for the project

II. Hardware design

1. The vehicle design

Among the constraints set by the rule, the student team, along with the suggestion of the team's advisors, set additional design constraints for the vehicle. The vehicle must fit, with ease, in a standard doorway of the engineering building in which the student team will be constructing and performing preliminary testing of the vehicle during the winter months. The standard doorway measures 32" wide between the two points with the least amount of clearance. The mechanical engineers came up with a simple design that requires two driven wheels and two caster tire with suspension to follow the zero turn radius. Additionally, the team has to find the way to put batteries, sensors, safety light and mechanical e-stop into the car. Therefore, after different thinking about alternative solution, it was decided to have frame having two levels with two brackets in front and back for supporting caster tire as shown in the figure 4. Each of the component of the frame has a role.

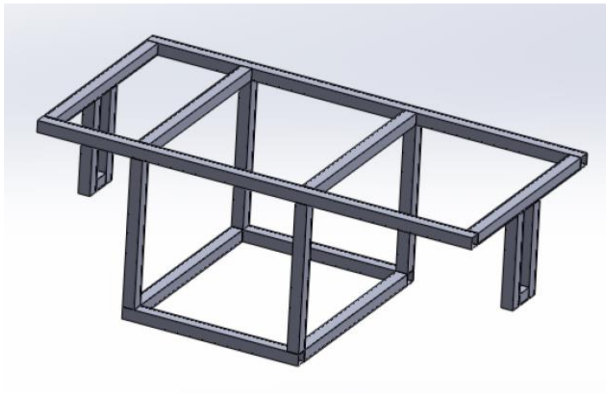

Figure 5: the frame design for the Moroccan Monster

a. Drive Train and suspension

The mechanical student team researched several methods of suspension. With a four-wheel design, the logical options for suspension are all wheels have suspension, only the two driven wheels have suspension, only the supporting casters have suspension, or no suspension for the vehicle. In past years, TCNJ autonomous vehicle teams have used all combinations of the methods mentioned above. An engineer from the team came across caster wheels used on gates that are sold with springs attached to them to provide suspension. These spring caster wheels are shown in Figure 5. In addition, suspension for the driven wheels was too complex for the scope of this project. It is why the team decide to provide suspension for only two caster wheels since the terrain is a grass field and the maximum incline experienced is 8.5°. This decision compromises the past TCNJ autonomous vehicle teams' suspension designs. Therefore, the team bolted the two caster tire to the front and back bracket in the frame and the two driving wheel in side.

b. Top of the frame

The top of the frame was one of the most important part that all mechanical engineers have been taking seriously since January 2016. This part of the car was a place where the camera, the

GPS, safety light will be bolted on. In addition, the housing of electrical components supposed to be bolted on it. After testing the camera at different height and decide that 4 ft was perfect height for the camera and realize how dangerous the lights coming out from the safety light on the camera. One of our mechanical engineers came up with the idea as shown in figure 6 (a). The top has dimension of 10.5"width, 11.5"length and 27.5"height. The lower base handles the safety light and the camera will be in the front. The Top of the frame is bolted into a sheet that is in the top level pf the frame.

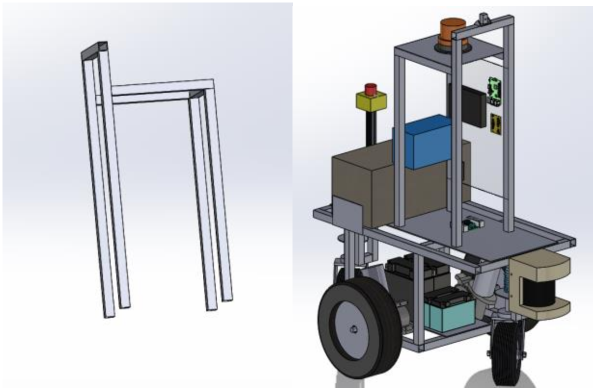

Figure 6: (a) The top of the frame (b) The design

2. The electrical design

a. Laser Range Finder (LiDAR)

The laser range finder is a device that detects obstacles on the course and is used as our main component for avoiding collisions. The Sick LMS 291 laser, one that had been obtained in one of NJAV's previous competition years, was used for this project. The device sends out laser beams and captures the reflections off of objects in a certain range. This is similar to the technology used in ultrasonic sensors, but it uses light instead of sound. The laser can detect object in 180° range.

Figure 8: Sick LMS-291 Laser Range Finder

The laser range finder communicates via an RS-232 serial connection. Hexadecimal bytes are sent to and from the LRF to get information. The information sent to the laser determines settings and data requests while the information received contains two-dimensional data received from the surroundings. This data can be plotted. To optimize the speed for each data collection cycle, we set the laser to measure distances every degree (181 points) and to only use values from a single scan instead of averaging several. It is bolted in the front bracket

b. Raspberry Pi

Most autonomous vehicles, especially those created for IGVC, contain a laptop computer to process all of the data and control the systems in the vehicle. However, with recent developments and affordability of Raspberry Pi microcontrollers, the Raspberry Pi 2B was used in this project as both a microcontroller and a replacement for a laptop. The Raspberry Pi 2B was released earlier last year, after the most recent IGVC, so it has not been used for the competition yet. The Raspberry Pi 2B has significantly more processing power compared to its previous iterations. The quad-core processor, $1GB$ of SDRAM, and additional USB ports set the Raspberry Pi 2B apart from the other models.

The Raspberry Pi was programmed to interface with the laser range finder through USB, the camera through the dedicated CSI-2 port, the GPS through the TTL serial pins (RX and TX), and the motor controller board through the two PWM signal pins. It was programmed in Python and the image processing was done using OpenCV, an open source computer vision program. The Raspberry Pi 3, an even more powerful model, was released during the construction of our vehicle and is something to look into for next year's team. The Raspberry Pi 2B is bolted into plexiglass that is parallel to the sheet metal

c. Camera

The camera is one of the most crucial components in this project. It is used to capture images of the course to detect lines and obstacles. The LRF cannot detect the white lines and that indicate the boundaries and some other traps, so the camera is necessary. There were initially several factors that needed to be considered when choosing a camera, and something that was simple to integrate into the system while still being effective seemed like the best option. The Raspberry Pi camera is a small 5-megapixel camera that provides high sensitivity, low crosstalk, and low noise image capture. It connects to the Raspberry Pi with a CSI port which is specifically designed for interfacing with cameras. The port is dedicated solely to the camera, so it is capable of sending pixel data at high rates. For our purposes, the images need to have a small resolution to reduce image processing computational time but must also be clear enough to detect all of the possible traps and obstacles. The camera is bolted into the front top of the car

d. GPS

IGVC requires that our vehicle navigate to several different GPS waypoints throughout the course. We chose a GPS that would be accurate and easy to interface with the Raspberry Pi. The Adafruit Ultimate GPS is a 66 channel GPS with 10 Hz updates that communicates via TTL serial. The GPS was placed near the top-center of the vehicle. For our project, we used the "gpsd" program and libraries to process the GPS data. This allowed us to easily receive latitude, longitude, velocity, and other information through a simple Python script. The GPS is bolted into the front of the Top

e. Motors and Motor Controller

The motors and motor controller board were previously used by other NJAV teams and were available to use in this project. The two NPC-41250 motors are designed to run on $12\ V$. However, the microcontroller must interface with a motor controller to provide enough power. The Sabertooth 2x25 motor controller is capable of providing a continuous current of $25\ A$ to two motors and has several different operating modes. An RC-based pulse width modulation (PWM) design was used to change the speed of the motors. This is one of the most commonly used motor control methods. By changing the pulse width of each signal to different values between $1000\ and\ 2000\ microseconds$ (based on the controller board specifications), the speed and direction of each individual motor can be controlled. The car can then turn by creating a difference in the motor speeds. The two motors are welding into driving wheels and the micro-controller is bolted into the second level of sheet metal

f. Power System

All of the components run on either $12\ V$, $24\ V$, or a built-in voltage regulator. There is one 5V and one 3.3V regulator on the Raspberry Pi. There are several batteries available to us, including the CSB GP-12170 $12V\ 17Ah$ and PS-1270 12V 6Ah batteries shown below in Figure 12. The battery configuration is also shown above in Figure 13. The two 17Ah batteries are used for the 12V line to the motors. However, one is used as a back-up, and there is a three-way switch to change which battery is connected. The two 6Ah batteries are placed in series and only used to power the LiDAR. This separated configuration was designed for simplicity and safe operation. The batteries are bolted into the lower level of the car

Figure 12: (1) CSB GP-12170 12V 17Ah Battery and (2) PS-1270 12V 6 Ah Battery

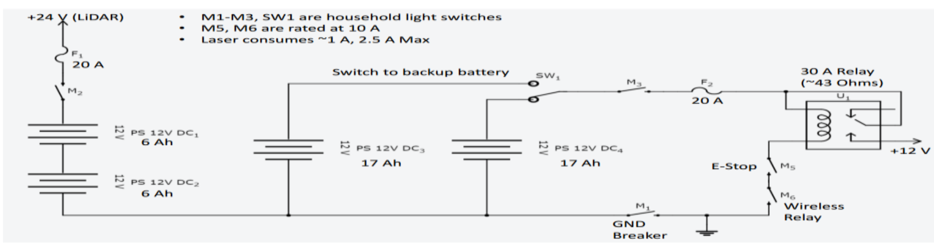

Figure 13: Relay and Power Circuit

III. Software Design

1. Path Planning

Our path-planning algorithm incorporates a sum-of-forces method for deciding which direction to move in. The desired waypoint creates an attractive force that pulls the vehicle towards it, and the obstacles create repelling forces that push the vehicle away. When these vectors are summed, the vehicle is given a single force vector representing the desired speed and direction.

Originally, we created a MATLAB program to simulate the path-planning algorithm by plotting a large matrix of randomly generated obstacles. While the program helped simulate what the actual vehicle would do, it was not perfect. Since it was based on matrix algebra, it was difficult to predict an accurate position for the vehicle during each iteration. The plotted paths were jagged and jumped from point to point, but in reality, the vehicle would be moving much more smoothly and linearly. The obstacles were also only represented by individual points which does not match the format of the LRF data.

Then, a program to simulate the sum-of-forces algorithm for a single frame of data was created. This was the program that would ultimately be used for the vehicle as it could be run once during every iteration of the vehicle's overall code. Figure 14 shows how the laser data is used to decide which direction to go. There are 181 points (0 to 180 degrees) that the LiDAR receives. Within a certain threshold radius R, the LiDAR detects points at a distance of $r(\theta)$. Obstacles are given repelling forces while an initial attractive force is directed toward the waypoint. The equations for this program are:

$$F_x = \frac{\omega a y_x}{R} - \frac{1}{N+1}\sum_{\theta=0}^{N}\frac{1-r(\theta)}{R}\cos(\theta),\ F_y = \frac{\omega a y_x}{R} + \frac{1}{N+1}\sum_{\theta=0}^{N}\frac{1-r(\theta)}{R}\sin(\theta)$$

2. Image Processing

a. Color Spaces

In order to be able to detect and filter out certain colors in images, color representation in OpenCV must be understood. In general, when working with color, everything is based on a scale of red, green, and blue values. These colors are represented on a scale of $0 - 255$ with 255 meaning that there is full color. Different combinations of values for red, green, and blue will result in different colors. Another color space that can be utilized for color representation in OpenCV is the HSV color space. When dealing with HSV, colors are represented by hue, saturation, and value. A representation of the HSV color space can be seen in Figure 15. The HSV color space is one of the most commonly used cylindrical-coordinate representation of points in an RGB space. The hue value refers to the angle around the central vertical axis. The saturation and value are represented by the distance from the axis and the distance along the axis, respectively. HSV color space is commonly used for image processing because unlike the RGB color space, it separates the image intensity from the color information. This proves to be very useful when filtering colors because it will decrease the negative effects that may be caused by lighting changes and shadows.

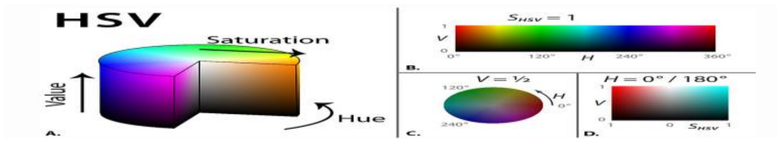

Figure 15: Representation of HSV color space

c. Line Detection

The algorithm used is Canny edge. Canny edge detection is a popular algorithm used for edge detection and it works by going through multiple stages. These different stages are noise reduction, finding intensity gradient, non-maximum suppression, and Hysteresis thresholding. By combining all of these stages into one function, OpenCV makes it easy to detect edges in an image. In order to test this function, it was applied to a sample image of the IGVC competition course.

d. Image Smoothing

In order to successfully use the Canny edge detection, the original input image needed to be tested through various filters in order to reduce unwanted noise and edges. Color detection will play a very important role in our image processing because we will need to detect the color white to see the painted path and potential potholes. Also, red and blue have to be detected so that flags placed throughout the course can be identified. In order to filter out the red and blue in an image, color filtering was done in OpenCV. The OpenCV in Range() function works by taking in a pixel range for an image and filtering out all of the pixels that do not fall within that range. After doing research it was found that it is much easier to detect colors other than white if you first convert the image from RGB to HSV. Once in the HSV color space, colors can be filtered out easily. In order to test this color filtering, an image with red, green, and blue was passed through a filter. As seen in Figure 17, the program was able to clearly separate the blue and red parts of the image. This is significant because on the course there will be red and blue flags placed that will need to be detected.

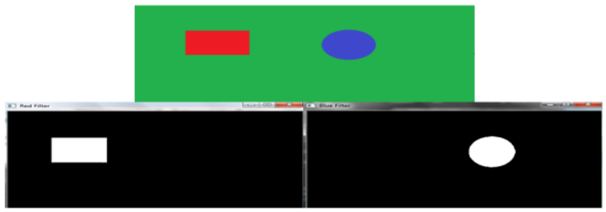

Figure 17: Blue and red color filter test

Being able to detect the color white is going to be a big part of the image processing algorithm because ideally the camera will be able to identify the white path lines and help keep the vehicle between the specified course lines. When detecting the color white in OpenCV, it is simpler to keep the image in the RGB color space. Since white is represented by a value of *255* for each red, green, and blue, a filter was applied to the input image to get rid of all pixels that are not white. This filter was applied to the original input image as well as the same image after it has been converted into grayscale. The results from this test can be seen in Figure 18.

Figure 18: White filter applied to original image (left) and grayscale image (right)

In both of the images that have been passed through the white filter, it can be seen that there is significant amount of noise, with the grayscale image having slightly more noise than the other.

This noise has to be reduced so that our path planning algorithm does not receive skewed input data.

A common method that is used for noise reduction in images is called a Gaussian Blur. Applying the Gaussian Blur to an image will result in a smoother image with reduced image noise as well as reduced detail. OpenCV provides a GaussianBlur() function that will perform this algorithm and give you a resulting smoother image. In order to test this, the smoothing function was added to our white color filter using the same input image. As expected, this resulted in a significant noise reduction that can be seen in Figure 19

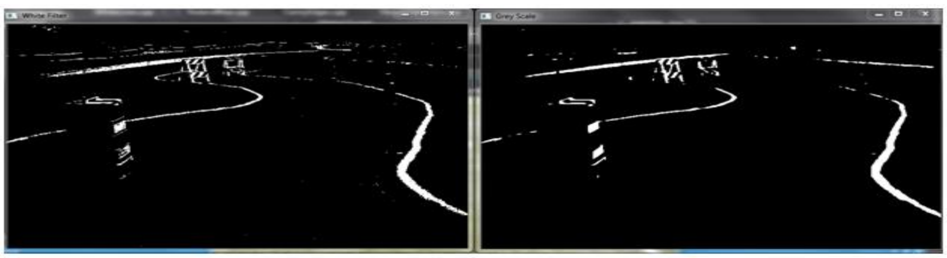

Figure 19: White filter test with Gaussian smooth

Since the Gaussian Blur proved to successfully remove most of the unwanted pixels from the image, the last thing to do was to implement our new smoothed image with the edge detection function. By doing this, the Canny () function will detect only edges that are formed from white lines or obstacles. The figure 20 shows the way used

Figure 20: Image processing pipeline

e. Image Processing for Navigation

The resulting data from the image processing will be used to help the car navigate between the lines by defining the pixel location. In order to do this, we needed to calibrate our camera using known distances. The camera was set on the car with a known location and then a picture was taken of the tile floor in The College of New Jersey school of engineering where each tile is 1’x1’. Using these known distances, we were able to get pixel locations that relate to real world locations. Once these locations were determined, we used the openCV function getPerspectiveTransform () to create a relationship between pixel locations and location relative to our car. In order to test if our image processing will work properly, we took an image and used all of the white pixel locations to be set as obstacles for the algorithm. The results from this test can be seen in Figure 22.

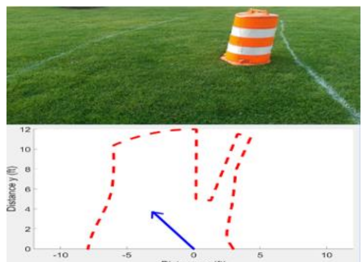

Figure 22: Camera Testing

IV. Innovation

This year team have been using many different ways to innovate. The first one is to use the Zero Turn Radius. This later helped the electrical and computer engineers not to worry about implementing the breaks into the software, additionally not to back up. Compare to the old team, the Moroccan Monster team decided to weld the frame to create one car instead of bolting as discussed before. The other innovative idea is to use the Raspberry Pi since the first time the TCNJ team got involved, they always use the Arduino. Now, by using the Raspberry Pi, we saved lots of place to give to the computer. Putting the safety light in lower level and the GPS underneath it helps the camera and GPS to be safe. Finally, the openCV has been for the first time by the Moroccan Monster team since the Moroccan Monster is the first team to use the Raspberry Pi instead of the Arduino.