Adaptive Cruise Control Using Simulink Image Processing

Miguel Asido
Electronics, Communicationa and Computer Engineering
Ateneo de Manila University
Quezon City, Philippines
miguel.asido@obf.ateneo.edu

Thomas Vincent Tarroza
Electronics, Communicationa and Computer Engineering
Ateneo de Manila University
Quezon City, Philippines
thomas.tarroza@obf.ateneo.edu

***Abstract*—Traffic congestion is a prevalent problem that is very much present in most major cities. Unfortunately, more roads do not mean less traffic. The fundamental problem of traffic is human drivers which are slow and inefficient at the task. One of the solution to the problem of traffic is to give the control to computer and sensor systems. Adaptive Cruise Control is the system that maintains safe driving speed and inter-vehicle distance between the follower vehicle and the lead vehicle. The study aims to develop a Unidirectional Adaptive Cruise Control system simulated through a mobot using Raspberry Pi.MATLAB and Simulink was used to develop the Tail Light Tracking Block Diagram that was deployed to the Raspberry Pi 3 Model B with a Camera Module. Using the HSV colorspace produced the best result because it can isolate the color red from the frame, leading to a cleaner thresholded frame. With the HSV method, a latency of 0.096 seconds was observed which is far faster than human reaction time. An Arduino Uno was used to interface the analog input of the IR Distance Sensor, PWM Input of the Encoders, and the PWM output to the Motor Drivers. The PID code in the Arduino still needs to be tuned to obtain a better response.**

***Index Terms*—Traffic Congestion, Adaptive Cruise Control, lead vehicle, HSV colorspace, latency, Raspberry Pi, MATLAB, Simulink, Tail Light Tracking, PID, Arduino**

I. Introduction

A. Background of the Study

With the rise of global use of private vehicles, one of the problems that arises is the slow flow of traffic. A larger traffic volume means more vehicles on the road leading to more frequent traffic jams. Traffic has become a major problem for cities around the globe causing delays and damages to economies of the world. Such is the case with the China National Expressway 110 and Beijing-Tibet Expressway Traffic Jam which lasted eleven days. Perishable products such as fruits and vegetables rotted on the road with some drivers being trapped for five days. [1] Traffic in the Philippines is also apparent and is much of a problem as well with Metro Manila having an average of 66 minutes stuck in traffic per day. The traffic situation in the Philippines' National Capital Region (NCR) was ranked the third worst traffic in Southeast Asia according to a study. [2]

Unfortunately, Adding more lanes to roads and adding more roads altogether does not solve the problem of traffic. If anything, it can sometimes worsen the traffic condition of an area. The Department of Transportation of California admitted that more roads lead to more traffic. In a study [3], it was concluded that an increase in roadway capacity where traffic congestion was present will likely induce an increase in vehicle road use because drivers will likely shift route to the newly expanded or built road. Thus, the problem of traffic returns with a larger vehicle volume.

The fundamental problem of traffic is human drivers. Although motor vehicles were made for human drivers, the latter are not the most efficient for the task. Human drivers are slower at recognizing signs and changes in traffic and are slower still in reacting to these compared to computer systems which can make recognize and make decision at near light speed. Human drivers, also, has a harder time communicating with other human drivers in other vehicles on the road unlike computers that can communicate with each other almost instantly as soon as both enter the range of each other. With slow reaction time and poor communication with other drivers, humans cause an instability and delays in traffic flow. Drivers take time to react to changes in the driving environment. Likewise, there is also a lag time before human drivers can execute changes to adjust to the changes in the driving environment. A brake in one driver will cause a brake in the driver behind and so on. This creates a "phantom traffic jam" that can extend for miles and last for hours even if there were no actual road hazard in the way. [7] An article [4] found that humans are the cause of traffic congestion. The bottlenecks such as road hazards are only triggers the instability of the collective driver traffic and are not the essential origin of traffic congestions.

The clear solution for the fundamental problem of traffic is to remove the vehicular control from the slow and uncommunicated driver to the fast and reliable computer control system and sensors. Autonomous Vehicles are already being road tested today and is an inevitable future. With full Automated Vehicle penetration to mass transportation, there would be significantly less conflict between conservative and aggressive drivers negotiating the merging and intersecting right-of-way. The automation of acceleration and intersection of vehicles on the road will improve traffic flow efficiency. [5]

B. Statement of the Problem

Traffic is major problem for major cities causing damages to economies. The problem of traffic seems to keep worsening with traffic jams spanning for days and weeks. Expanding road capacity by road widening and building more roads offer no long term solution to the problem of traffic and it sometimes adds to the problem it was trying to solve. An increase in road capacity is not a viable solution to the problem of traffic. The fundamental cause of traffic are human drivers. They cause instability to the dynamics and flow of traffic because of the lag that comes with their reaction time and adaptation time. The solution, therefore, to the problem of traffic is to remove huma drivers and replace them with automated systems. A solution is to develop a system that would minimize the reaction time and adaptation time to almost none.

C. Scope and Limitation

Previous studies have focused on using radar and lidar technologies as a means for depth perception for Automated Cruise Control. These studies implemented Cruise Control to specific vehicles either by rigged throttle and brake pedals or by manipulating the specific electronic control unit of the model vehicle.

The researcher's take on the traffic problem caused by slow reaction time is to propose a system for an adaptive cruise control algorithm for stagnant to low speed traffic conditions such as stop lights or in a traffic jam. The idea is for a follower vehicle to track a lead vehicle and maintain a constant distance between them. This method simulates a train like method where the vehicles behind a lead vehicle will move forward all at the same time like there's a chain connecting the vehicles. In a stoplight setting, this ideally enables more vehicles to pass in the time interval when the green light is on. As opposed to the typical setting where humans accelerate at different rates when the green light turns on.

This study is only limited to the implementation of the system to a Mobot equipped with a Raspberry Pi with Camera Module and an Arduino. Further studies such as upscaling and implementation to actual vehicles is outside the Scope of this study.

D. Objectives of the Study

The study aims to develop a Unidirectional Adaptive Cruise Control system simulated through a mobot using Raspberry Pi. The following are the expected output at the end of the study:

- To develop a Tail Light Detection/Tracking Algorithm in MATLAB/Simulink using a Raspberry Pi camera.
- To develop a Distance Controller using data from IR distance sensors and image processing.
- To develop an Electronic Speed Controller of the DC motor.
- To implement MATLAB/Simulink image processing algorithm, Arduino distance and speed controller, to form a system with Raspberry Pi and Arduino serving as a platform to follow a lead vehicle and maintain the desired distance.

II. Review of Related Literature

An article was published in the New journal of Physics which shows that even with no road hazards, human drivers' inconsistencies cause phantom traffic jams. The experiment consisted of twenty-two vehicles driving around a 230 meter circular road. All vehicles were equally spaced around the circumference of the road at the start of the experiment as can be seen in Fig.1a. The drivers were tasked to keep their distance as evenly as they can while driving. Three minutes later, a phantom traffic jam emerges that propagates backwards as can be seen in Fig.1b Inconsistencies in the the drivers caused one vehicle to be out of synchronization and causes a congestion leading to some vehicles to stop completely before accelerating again. The mechanisms by which traffic jams occur originate not by road hazards but by the instability of the human driver collective. Such hazards are only triggers for the development of traffic jams and are not the essential origin of traffic jams. [4]

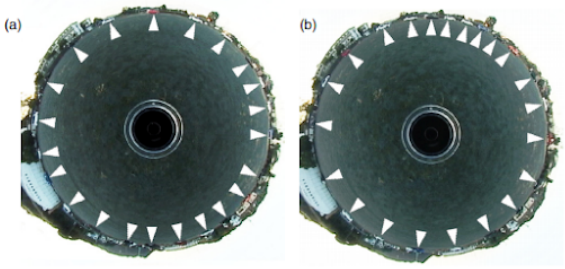

Fig. 1. (a) The snapshot at the initial stage. (b) The snapshot 3 min later where a jam has been formed.

A study done by Kesting and Treiber shows the relationship of Human Drivers to the Dynamics and Stability of Traffic Flow. Three characteristic time constants were tested to see their relationship with the dynamics and stability of Traffic Flow. Two of these characteristic time constants are Update Time and Reaction Time. The Update Time of Drivers is the representation of the finite attention to the traffic where only at certain time intervals does the driver look at the traffic situation and instantaneously modify their acceleration to match that of the current situation. The Reaction Time of Drivers represents the consequence of physiological aspect of sensing, perceiving, deciding, and performing an action. The Reaction time is the time it takes for the driver to recognize changes in the driving environment as well as react accordingly. A fleet of 100 vehicles was simulated on a single lane following a leader where it would brake and decelerate from 25m/s to 19m/s at time interval between 1000s and 1003s to simulate an instability in the traffic flow.

As can be seen in Fig. 3, with a reaction time of T' = 0.9s, the traffic flow becomes unstable and vehicles brake and accelerate more often unlike in Fig. 2 with zero reaction time T'=0s where the phantom traffic jam dissipates as it moves through the fleet of vehicles. The study concluded that the Reaction Time of driver is the cause of Traffic congestion as vehicles

stop then accelerate again in a phantom traffic jam. There were no road hazard and only a lead vehicle breaking which caused the traffic congestion. With zero reaction time, traffic flow is more stable and congestion easily dissipates.Automated Vehicles will make this possible as sensors and control systems minimize the time it takes for the Vehicle to adapt to changes in the traffic environment. [7]

Cruise Control is a system developed to assist drivers while driving along straight roads. Adaptive Cruise Control (ACC) is an extension of the Cruise Control system developed for congested traffic. ACC provides velocity and distance control. Low-speed ACC operates under traffic congestion to maintain distance from the lead vehicle, launching the vehicle when the lead vehicle moves, thus reducing reaction time. High-speed ACC operates on cruising speed when there are no lead vehicles to follow. It keeps the speed of the vehicle at a safe speed and decelerates when a slower vehicle ahead is detected to avoid collision.

A previous study done in the Asian Institute of Technology in Thailand developed an Adaptive Cruise Control for an intelligent Vehicle. A Mitsubishi Galant was used as the follower vehicle. Its Throttle Valve was modified to a drive-by-wire system where a servo motor controlled the throttle valve position to control the acceleration with a potentiometer attached to the gas pedal to measure its position. The Au-

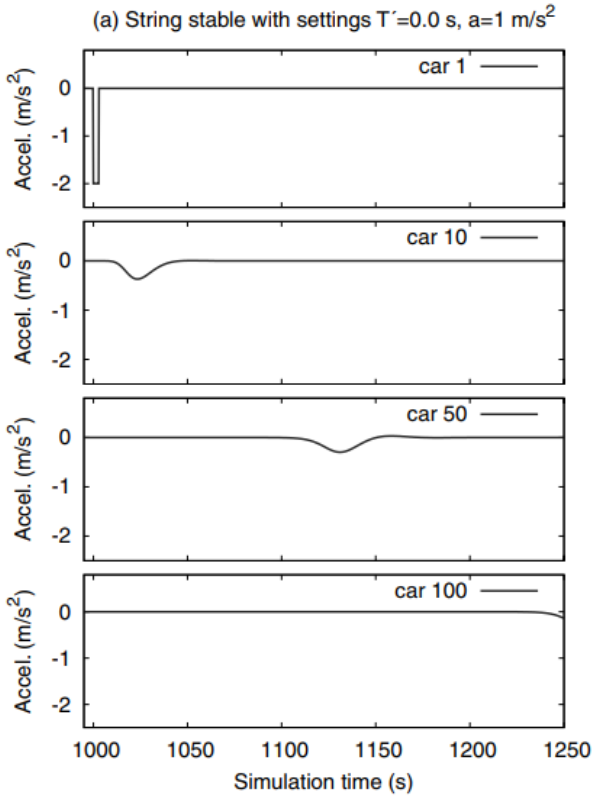

Fig. 2. Time series of the acceleration for selected platoon vehicles for zero reaction time T = 0 s

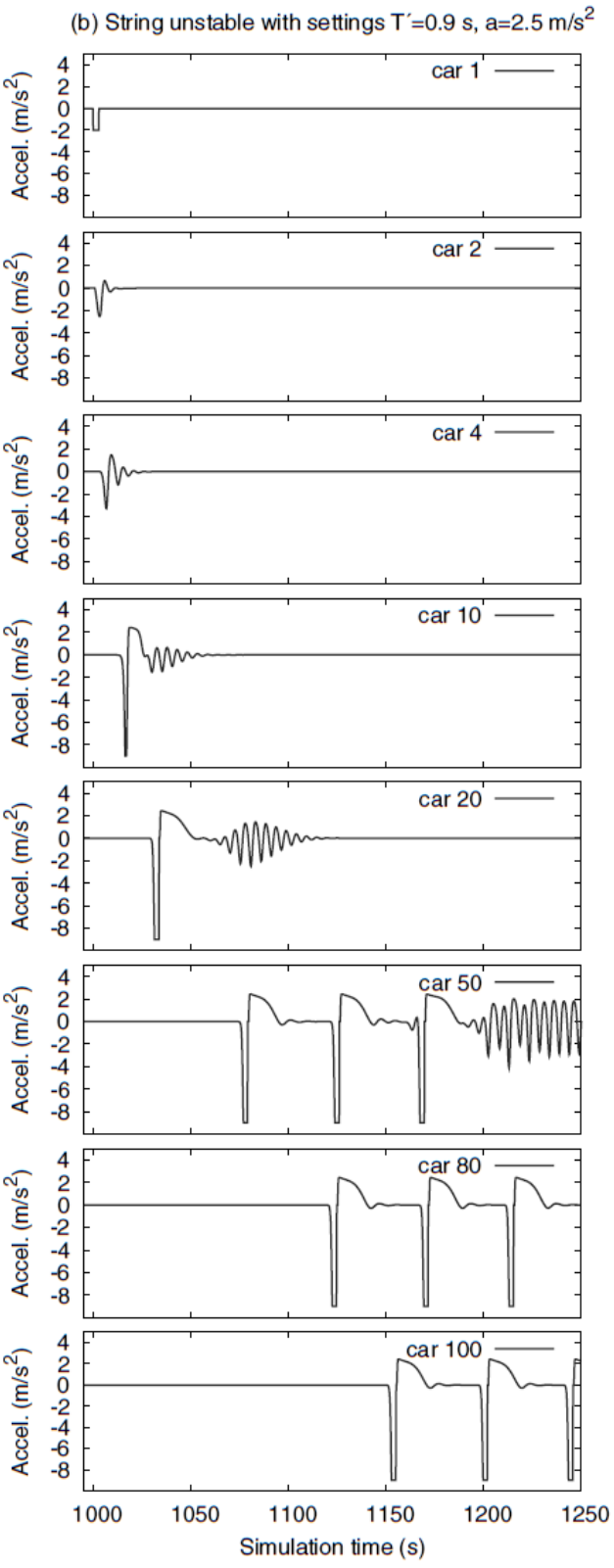

Fig. 3. Time series of the acceleration for selected platoon vehicles for non-zero reaction time T = 0.9 s

tomatic Braking Control System was made by controlling the brake pedal with a pulley and steel cable attachment controlled by a servo motor with an encoder to monitor its position. The distance sensor system utilized a laser range finder to detect the distance of the lead vehicle from follower vehicle.

The ACC has two operating modes. When the distance sensor detects that there are no hazards ahead, the ACC is in velocity control mode where the vehicle speed is kept at a safe speed. When the distance sensor detects that there is a hazard ahead, the ACC enters distance control mode where the vehicle is maintained at a safe distance from the hazard ahead. If the vehicle gets too close to the hazard or if the former is approaching the latter at a dangerous speed, the automatic braking system will decelerate the vehicle. If the distance between the vehicles become too great, the throttle valve system will accelerate the vehicle to maintain the inter-vehicle distance. The developed ACC was able to control the vehicle to cruise at the desired speed and efficiently maintain the distance between the vehicles. [8]

A study done by Ioannou and Chien developed an Automatic vehicle following using the Autonomous Intelligent Cruise Control system. First, the study used three human driver models to simulate the behaviors of actual human drivers on the road. The human driver in this case is the controller of the system as he is the one to detect distances, velocities, and accelerations of the vehicles and decides on changes accordingly.

The first model is the Linear Follow-the-Leader Model where a single lane of traffic with no passing assumes that each driver in the fleet reacts to the changes in the lead vehicle. The input is the difference in velocities of his vehicle to the one in front and the response is the acceleration command to the vehicle.

The second model is the Linear Optimal Control Model where the human driver is assumed to be mimicking a linear optimal controller based on a quadratic cost function. This model, however, does not consider the driver's reaction time in its computations.

The third model is the Look-Ahead Model where the human driver is assumed to observe the behavior of three vehicles ahead of him. The majority direction of acceleration of the three vehicles is computed then the driver adjusts accordingly.

To compare to the human driver models are the controller models for the automated vehicle. First is the Safety Distance Policy where the inter-vehicle distance stated by California is about a vehicle length for every 10 mph of velocity. Reaction time can be eliminated with automation and good sensors.

The second model is the Automatic Vehicle Following where the Automated Intelligent Cruise Control system is able to detect the inter-vehicle distance and relative velocity between itself and the lead vehicle in addition to its own vehicle velocity and acceleration. With these data, the model is assumed to correct its inter-vehicle distance and velocity relative to the lead vehicle according to the safety distance policy.

These models were then simulated across different number of vehicles and vehicle velocity as shown in Fig.4 and Fig.5 It can be seen that the Automated Cruise Control System was far more efficient in traffic flow across increasing number of vehicles. It can be also seen that the Automated Cruise Control System was more stable with a higher and more consistent flow rate compared to the human models which had fluctuations. [9]

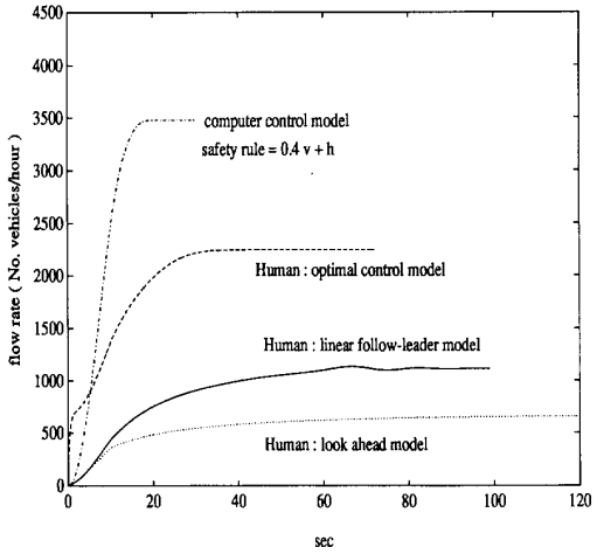

Fig. 4. Transient traffic flow rate (h = 4 m to 4.5 m).

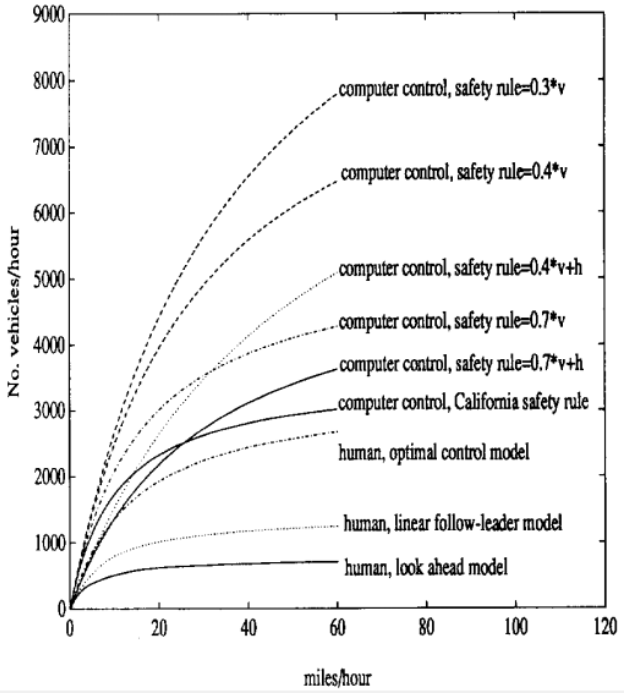

Fig. 5. Steady-state traffic flow rate versus steady-state speed (h = 4 m to 4.5 m).

Because of the way humans perceive the world, objects that are closer appear larger while objects that are farther appear smaller. Apparent size of images are measured not by linear length but by the rotation it covers in human eyes as well as camera lenses as can be seen in Fig.6. Closer objects cover more degrees in rotation in field of view compared to farther objects which cover less degrees in rotation in field of view. Thus, the apparent distance between objects increase when moving closer to the viewer and decreases when moving away from the viewer. [10]

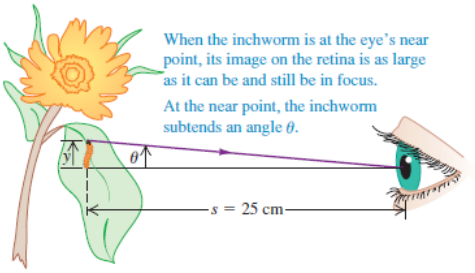

Fig. 6. Angular Coverage of Field of View Translates to Apparent Size of Object.

A study by Rohit and Patel focuses on detecting the tail lights of a vehicle based on the symmetry and its color characteristics. Their study focuses on night time driving conditions and have these four given guidelines about the characteristics of tail lights in general.

1) The tail light is the brightest object in the image
2) Close to each other in pairs
3) Symmetrical with same size and shape, and
4) Must be red in color.

Their study describes a system that detects the vehicles based on their tail lights and focuses on ranges up to 7 meters. Vehicles that do not follow the guidelines above were not included in the study.

Their method starts with the input image frame, then filtering the image based on the tail light color. First, the Red Layer from the RGB frame was extracted and the grey converted image of the input frame was also gotten. Afterwards, these two frames were subtracted and unwanted noises were removed with the use of a Median Filter. Finally, the image was converted into binary using Threshold value. After processing the image, blob analysis was done and area of the bounding box was calculated and the detected lights were checked for symmetry, as seen in Fig. 7. [11]

There are different ways of representing color. Most of the time in computer vision, each pixel is represented as the array of mostly three values. Different ways of expressing these values are called Color Spaces. The Red-Green-Blue (RGB) Color Space is one of the simplest color space because it is the most intuitive. Because the human eye has cones that are most sensitive with Red, Green, and Blue colors of light, pixels are often represented as the combination of Red, Green, and Blue values. White is all Maximum values of RGB while Black

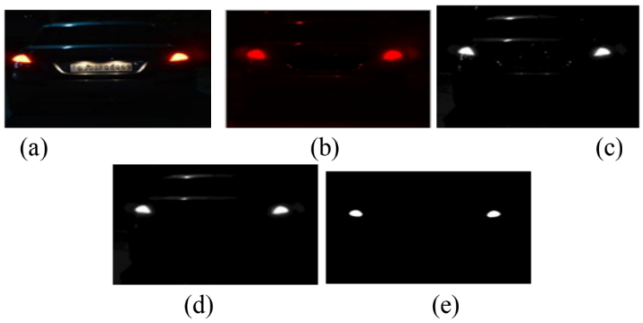

Fig. 7. (a) original image frame (b) Red color frame (c) gray converted image (d) Removed noise image (e) binary converted image frame.

is Minimum values. The Hue-Saturation-Value (HSV) Color Space is also widely used because of its efficiency of isolating colors. As seen in Fig. 8, the Hue corresponds to the color in the color wheel, the S stands for the Saturation where zero saturation is neutral gray, and V stands for Value or how Dark or Light the Color is. [16]

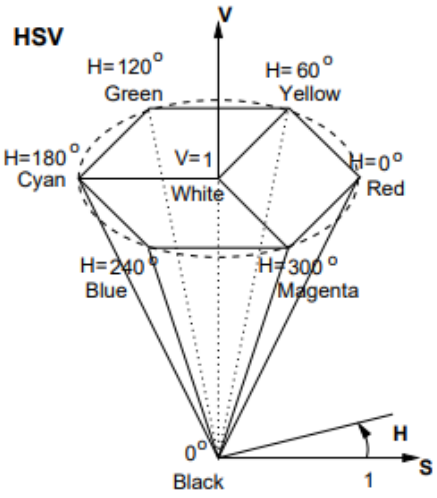

Fig. 8. HSV Color Space

A study was done by Rashmi et. al. where a program would detect traffic signs on the road. The proponents used the HSV colorspace to extract the threshold to isolate different colors of traffic signs. It was determined that red traffic signs had values of Hue greater than 240 degrees or lower than 10 degrees, Saturation of greater than 40, and Value of greater than 30. These values can also be used as thresholds to extract the red colors of the tail lights of vehicles on the road. [13]

Pulse Width Modulation (PWM) is a type of analog modulation of a carrier triangular wave by a modulator sine wave. PWM can be manipulated by the duty cycle of a square wave. The strength of the wave is changed whenever the length of the positive half cycle is controlled. The longer the "on-time," the more power the PWM wave can deliver.

As seen in Fig. 9, less power will be delivered with a lower duty cycle and when the on-time is shorter. On the other hand, more power will be delivered with a higher duty cycle and when the on-time is longer. PWM can be used as a more efficient way to control electric motors because it is a series of switching pulses with less power dissipation instead of a resistive control with high power dissipation and loss. It can also be used as a means of input where the duty cycle corresponds to the value of the input. One method to convert PWM to Analog value is by a Pulse In function where a counter is used to obtain the time of the positive edge and compare it with the period to obtain the duty cycle. [14]

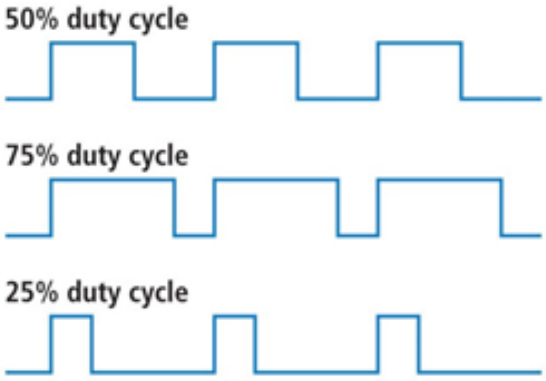

Fig. 9. Duty Cycle of a Pulse Width Modulated Signal

III. Methodology

A. Materials and Equipment

The Raspberry Pi 3 Model B is the third generation Raspberry Pi. This powerful credit-card sized single board computer can be used for many applications. The Raspberry Pi 3 Model B was used as the platform for the image processing. It contained the code for the Simulink blocks handling the Image Processing Algorithm. [17]

Fig. 10. Raspberry Pi 3 B

The Camera Module was used to take high-definition frames that was used as video feed for the Image Processing. The Raspberry Pi Camera will be used to record a live video in which the frames will be processed through Simulink. [18]

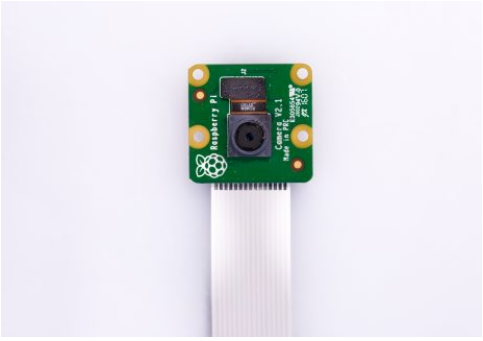

Fig. 11. Raspberry Pi Camera Module v2

The GP2Y0A21 Sharp distance sensor was used as a way to add obstacle avoidance, sensing to your robot or any other project. It has a detection range of 10 cm to 80 cm and an analog voltage indicating the distance. The IR sensor was used both as a distance detector and emergency stopper if the follower vehicle exceeds the distance threshold from the lead vehicle. [19]

Fig. 12. Sharp GP2Y0A21YK0F Analog Distance Sensor 10-80cm

The magnetic encoders output two digital pulses that are 90 degrees apart. The frequency of the digital pulse depends on the speed of the DC motor. The digital output is made useful by reading the frequency of the pulses, then converting to speed. This way, the encoder is used as a motor speed reader. The encoders was used to monitor the rotational speed of the motors. This was used as feedback to the system to adjust the relative speed of the vehicle. [20]

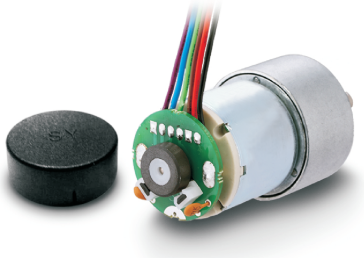

Fig. 13. Shayangye Two Channel Hall Effect Magnetic Encoder

The Mobot serves as the vehicle for the Assistive Cruise Control System.

Fig. 14. Mobot

The Arduino Uno Microprocessor board is a prototyping platform that is designed to interface with both digital, analog, and PWM input and output devices. The Arduino Uno was used to control the motor drivers with PWM as well as read the analog input from the IR Distance sensor, which the Raspberry Pi cannot since its GPIO pins are only digital, and the PWM inputs from the two encoders. [**?**]

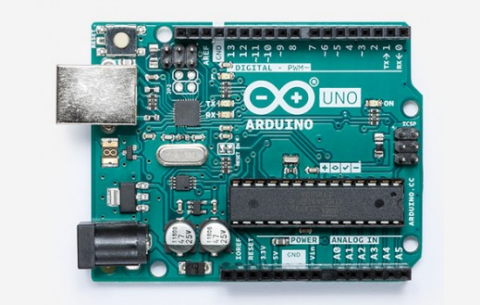

Fig. 15. Arduino Uno Board

B. Software

MATLAB + Image Processing Toolbox Image Processing Toolbox™ provides a comprehensive set of reference-standard algorithms and workflow apps for image processing, analysis, visualization, and algorithm development. [21]

Simulink is a block diagram environment for multi-domain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. [22]

C. Block Diagram

The Block Diagram for this study's system is shown in Fig. 16. The Raspberry Pi Camera feeds video input to the Image Processing of the system to determine the Estimated Distance from the Lead Vehicle. The Sharp IR Sensor is used for emergency purposes where the camera cannot see the obstruction. The Arduino is used to Control the PID as well as convert the different types of input as data that can be used. The encoders are used as feedback to make sure that the motors are running at the correct speed.

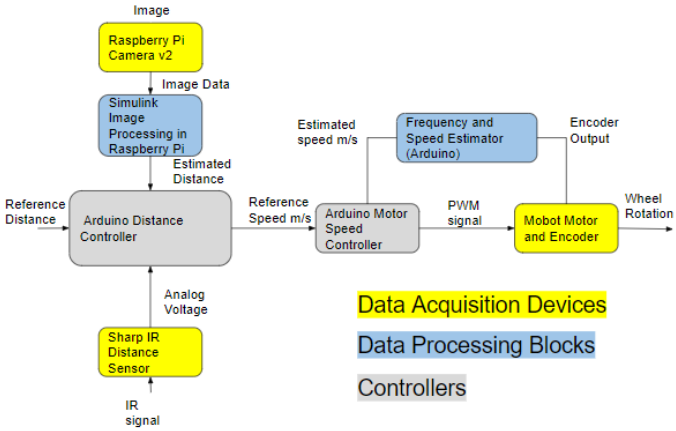

Fig. 16. Block Diagram

D. Procedure

The following Image Processing blocks and techniques were used in simulink to develop the tail light detector and tracker. This is limited to non-red vehicles with the conventional two tail light setup found in most private vehicles.

1) The first step to detecting red in an image was binarization. Using the RGB values, and subtracting the red frame from the greyscale converted image (or intensity), a certain threshold was set to eliminate non-red components in an image. The result was a binary image where only red components appeared white, and the rest appeared black. Upon testing, it was found that this method was not very reliable as non-red images were still picked up even with the threshold properly tuned.

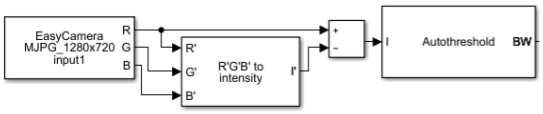

Fig. 17. Block Diagram Subtraction of Red Layer from RGB Intensity for Binarization

The method was revised. Instead of RGB values, the image signals were sent through a color space conversion block. The other color space available for processing in simulink was Y'CbCr. The Cr signal was directly accessed and a certain threshold was set to eliminate non-red components in an image. This method resulted to a more reliable red detection and binarization.
Another method was done using HSV color space.

2) The next step was to filter out noises from the image by using a median filter block. The binarized image is fed through the median filter block and the output was fed to the blob analysis block.
3) The binarized image is fed through the Blob Analysis Block which computes and outputs an area, centroid, and bounding box for each blobs in the binarized image. This block is used for calculating the statistics of regions of

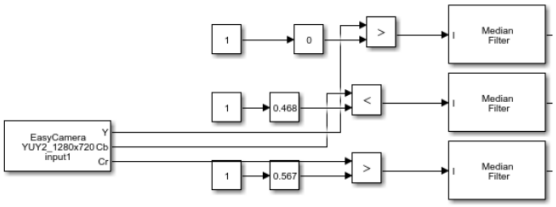

Fig. 18. Block Diagram Thresholding of YCbCr Color Signals.

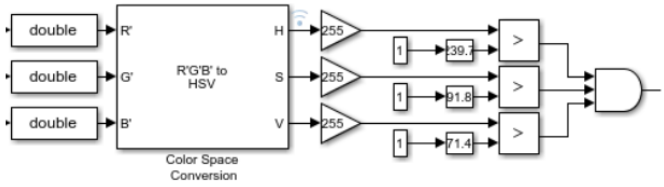

Fig. 19. Block Diagram Thresholding of HSV Color Signals.

Fig. 20. Median Filter Block

high value in a binary image. This block takes in input a matrix of the boolean values of a binary image where regions of HIGH value is white and those that are LOW is black. This block outputs several statistics of the white regions called "blobs" in the image. For a video feed, each frame is treated as an image. The threshold was set to a certain value such that only the tail lights are recognized and the other blobs that were detected are rejected.

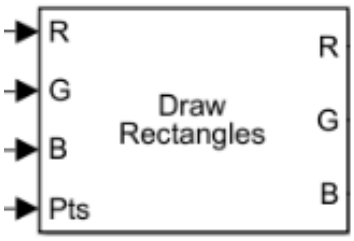

Fig. 21. Draw Rectangles Block

4) Finally, the output data 'area' was output for the pixel count algorithm, the centroid was fed through the draw "x" block which puts an "x" mark at the given coordinates. The output of the bounding box was sent to the draw rectangle block which bounds the blobs in a rectangle.

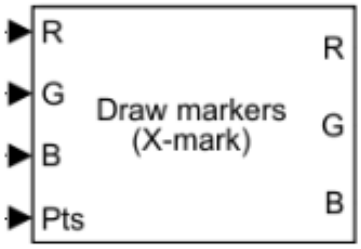

Fig. 22. Draw X-mark Block

5) A matlab function block was made which accesses the array of the centroids. Given that there are only two blobs, the horizontal difference between the two centroids were determined by accessing the x components of the centroids of the blobs. If the blobs are not strictly two, the output was considered an error.
6) Finally, A Write To File block was used to write the data into a .mat file, which is a time series of the horizontal differences of the centroids, which was used for the acceleration algorithm. Another Write to File block was used to store the difference between the current centroid length and previous centroid length. From this, an estimated relative rate of change is obtained. The .mat files were accessed and the values were copied to a spreadsheet for processing.

The image processing algorithm represented by Fig. 23, although captures the red parts of an image very well, and eliminates other colors, still is unreliable when unwanted red colors are present in the frame. To remedy this, a matlab function block was developed to distinguish which of the detected blobs are tail lights. First, it checks if the blobs are within bounds. It proceeds to compare each candidate from one another and checks for symmetry. If the candidates are symmetric and within bounds, then the x-components of the candidates are subtracted to obtain the distance.

Fig. 23. MATLAB Function

A simple algorithm that estimates the rate of change of the number of pixels was made from the blob analysis block in simulink. The previous number of red pixel was subtracted to the current number or pixels. This gave the rate of change of pixels per frame.

The same rate of change algorithm was done for the line that connects the centroids of each tail light blob seein in Fig. **??**. Similarly, an algorithm was developed to determine the rate of change of the output voltage read by the IR sensor.

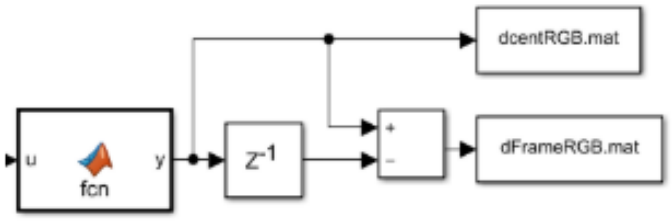

Fig. 24. Rate of Change

Unfortunately, the program above is too processing heavy for the Raspberry Pi to handle. A more streamlined version of the program is explained as follows.

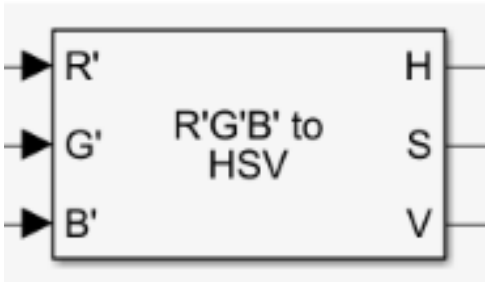

Fig. 25. Color Space Conversion Block

The block seen in Fig. 25 converts the input from the Video File or Raspberry Pi Camera from RGB colorspace to HSV colorspace.

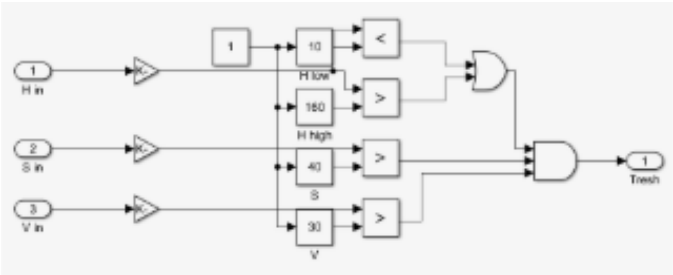

Fig. 26. Thresholding Subsystem

The subsystem seen in Fig. 26 binarizes the image. It will output a HIGH if the Hue is above 160 or below 10, and the Saturation is above 40, and the Value is above 30. This is based on the thresholds for red street signs. The output of this subsystem is the matrix for the pixels of the binarized image.

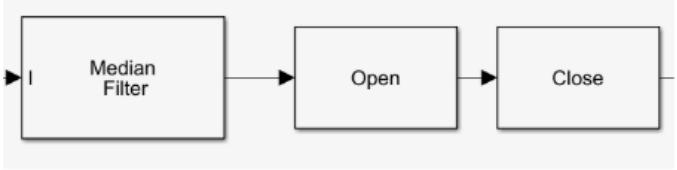

Fig. 27. Morphological Processes

As seen in Fig. 27, the following blocks cleans the binarized image so that noise is minimized. The median filter removes the specks and small holes in the blobs. The Morphological Open and Close are processes which greatly fixes binarized images. Morphological Open removes the boundaries of blobs then increases the boundaries which causes white specs to diminish and not comeback while larger blobs stay. Morphological Close removes holes in blobs by increasing the boundaries then decreasing them.

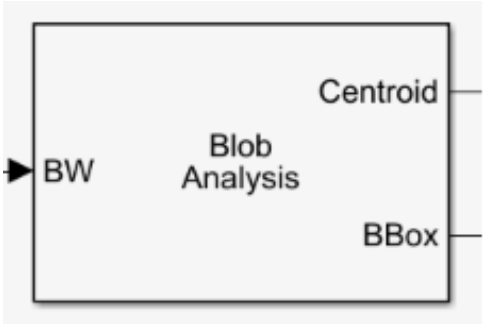

Fig. 28. Blob Analysis Block

The Blob Analysis block in Fig. 28 is a tool for analyzing objects in a binarized image. Some of its outputs are the Areas of the blob, the centroid of the blocks, and the dimensions and positions for bounding boxes of the blobs.

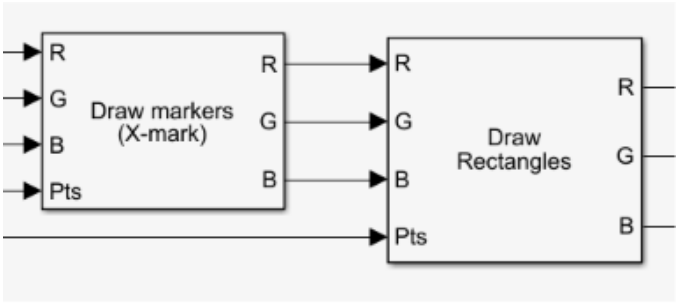

Fig. 29. Markers and Bounding Boxes

The drawing blobs in Fig. 29 help in showing the blob analysis to the user. THe Draw Markers highlights the centroids of the blobs while the Draw Rectangles Bounds the Blobs in a rectangle using the Bounding Box output of the blob analysis. A MATLAB function will then go through the centroids of the blobs and compute for the distance between the two horizontally aligned blobs that will most likely be the tail lights of the lead vehicle. The other noise blobs that were not eliminated by the Morphological Operations will be discarded.

Mapping the pixel distance between the centroids of the two tail lights to the actual distance was a simple process. The pixel distance was obtained by positioning the simulated tail lights at intervals of one inch from the camera. The pixel distance was recorded for every distance and the values were plotted in a graph.

Measuring the latency of the system was important because the point of the program was to be faster than human reaction time. With the first program, the Raspberry Pi would often crash and overheat with the shear processing power requirement of the blocks. For the times it did not crash, the system was beyond usable with the program taking up to several second to process each frame. The second program was much more streamlined and was viable for the Raspberry Pi. The Green LED would Light up everytime it detected at least one blob. A slow motion video capture was done where the camera was covered then quickly uncovered in front of the simulated tail lights at 480 fps. The number of frames was noted between lifting the cover and the LED turning on. The number of frames was divided by the frame rate to obtain the time between lifting the cover and the LED turning on which corresponds to the latency of the system. This test was done for when running the program externally on Simulink versus Deploying the program to the Raspberry Pi.

Using the pulseIn() function in Arduino, the time it takes for one positive count of the encoder. Since it takes 700 counts to rotate the motors shaft once, the time it takes to rotate the motor shaft once is obtained by multiplying the output of the pulseIn() function by 1400 (accounting for the negative half cycle of the encoder). The rotational speed of the motors is obtained by dividing one revolution by the computed time to rotate the motor shaft once.

As can be seen in Fig. 30a, the response of the Sharp IR sensor for Output Voltage vs Distance is hyperbolic and nonlinear. However, the response aforementioned for Output Voltage vs Inverse of the Distance is linear for around 10cm to 30cm, as seen in Fig. 30b. [19]

Calibrating the IR sensor involved mapping the output voltage to the inverse of the distance then getting the multiplicative inverse to obtain the actual distance reading in mm.

IV. Results and Discussion

A. Tail Light Detection Image Processing

Seen in Fig. 31 is a sample of a frame from a video in which the image will be processed through binarization, connected component analysis, and bounding. With the naked eye, it can be seen that the tail lights have a red glow. Also seen are unwanted red lights from other sources such as tail lights of cars that are not the subject, and from the stop light. Noise filtering and blob analysis will be used to attempt to ignore the noise and unwanted red color signals.

The first step was to access the colors signal of interest. In HSV shown in Fig. 32, the signal of interest was Hue, where it represented the color by values from 0 to 360 and shown in grayscale.

For the RGB method, the Red signal is subtracted from the RGB intensity signal to yield the image in Fig. 33. The figure shows a grayscale image where the red colors appear lighter. This grayscale image was then subjected to thresholding.

The third method was to access the Cr color signal from the YCbCr color space. The image in Fig.34 shows the output of the Cr color signal which represents red colors in very bright white. Cr is also called the redness of an image. This was also subjected to thresholding afterwards.

From the resulting images in Fig. 33, Fig. 32 and Fig.34, the next step was to binarize. A certain threshold value was

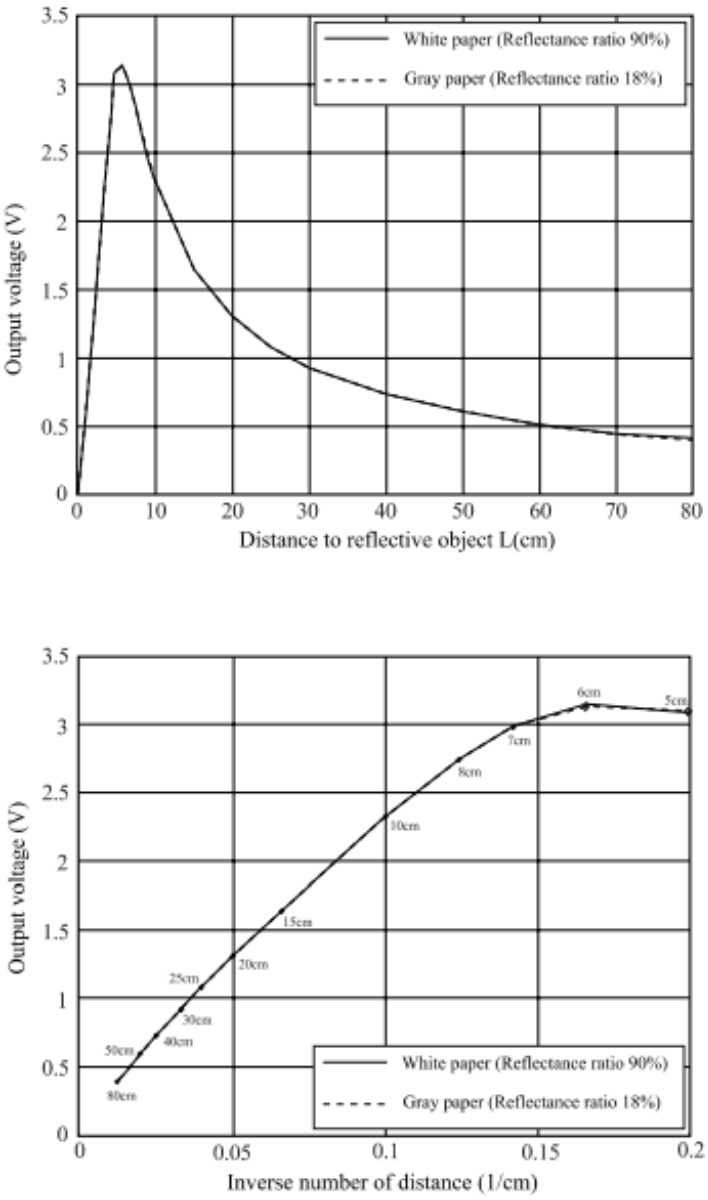

Fig. 30. Response of Sharp IR Sensor According to the Datasheet

Fig. 31. Full Color Image

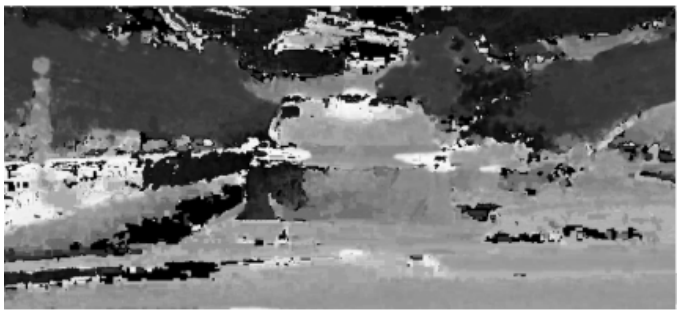

Fig. 32. H Signal from HSV

Fig. 33. R minus RGB Intensity

Fig. 34. Signal Image from Cr

Fig. 35. Binarized Result from HSV Signal

Fig. 36. Binarized Result from RGB Intensity Subtraction

Fig. 37. Binarized Result form Cr Signal

Fig. 38. Binarized Result from HSV AND RGB

tested for each case and the best value was found which eliminated the most noise and retained the most red signals in the image. Seen in Fig. 36, most of the red light in the image were retained but with very high noise. Comparing this to the YCrCb method in Fig. 37, it can be seen that the YCrCb was better at removing unwanted red noise. The reflection of the red light on the hood is not shown anymore in the binarized image, and there are minimal signals from the red lights in other sources. In the RGB binarization, most of the red signals like the stoplight and other tail lights are evidently seen. Seen in Fig. 35, most of the red lights were detected but the binarized image was very noise. Although noisy, it completely blocked out orange and yellow, which was detected a lot in RGB and YCbCr method.

After binarization, Rectangular bounding is done on the blobs seen in the binarized image. The Blob Analysis block in simulink outputs the coordinates of the bounding box and the centroids of each blob. These values were sent to the Draw Rectangle and Draw X blocks. Seen in Fig. 39, the RGB method was able to bound a box on the wanted region, which was the tail lights of the lead vehicle, without bounding other objects around.

Seen in Fig. 40, the RGB method had error eventually as the frame captured the other tail lights as its pixels get bigger. The bounding rectangle transferred to the other vehicle because the tail light was still recognized. The number of bounding boxes was limited to two. Only two blobs will be bounded. In this scenario, there are four blobs seen by the block, but it bounded the left vehicle tail light because the bounding starts from left to right. It does not matter how many detected blobs there are, the bounding box starts bounding from left to right,regardless of the limit of bounding boxes set. For this

Fig. 39. Output Image with Bounding Boxes

same initial test seen in Fig. 41 , the YCrCb did not show any shifts in bounding.

Fig. 40. Blobbing Error in RGB Method

Fig. 41. Blob Retained for YCrCb Method

The initial test show that RGB alone was a good red tracker but was bad at filtering our yellow and orange colors. HSV was a very good red tracker but had trouble eliminating noise from light colors. YCbCr was not very good with dark red areas and miss those pixels out. It was also prone to orange and yellow detection. The method that stood out was the use of AND for RGB and HSV. Since both RGB and HSV capture different shades of red very well, HSV rejected yellow and orange, while RGB did not, AND gate enabled the algorithm to eliminate both orange, yellow, and light colored noise. Only the desired red colors were retained and thresholded.

B. Streamlined Tail Light Tracking

The streamlined HSV version of the program produces a relatively clean output. A video file of a lead vehicle in traffic was fed to the program as seen in Fig. 42.

As seen in Fig. 43, the program successfully isolates the red pixels in the frame with a considerable amount of noise.

The binarized frame was cleaned in Fig. 44 with the white specks eliminated and the holes minimized.

The output in Fig. 45 shows that the taillights of the lead vehicle were successfully bounded and the centroids marked. Although the tail lights of other vehicles were also bounded, the MATLAB function only computed for the distance between the two horizontally aligned blobs that were the tail lights of the lead vehicle.

Fig. 42. Input Frame

Fig. 43. Thresholded Frame

Fig. 44. Cleaned Frame

Fig. 45. Bounded Output Frame

C. Distance Mapping for Image Processing

As can be seen in Fig. 46, the actual distance between the Raspberry Pi camera and the simulated tail light is inversely proportional to the distance in pixel between the centroid of the two tail lights. As the camera gets farther from the tail lights, the distance between the tail lights decreases. Conversely, as the camera gets closer to the tail lights, the distance between the tail lights increase. This pixel distance between the tail lights is fed to the Arduino as a PWM signal with maximum distance at maximum duty cycle (255) and minimum distance at minimum duty cycle (0).

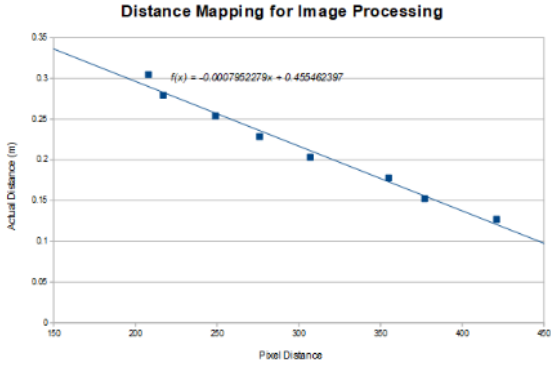

Fig. 46. Plot of Pixel Distance vs. Actual Distance

D. Latency

When external mode was used, the number of frames between lifting the cover and the LED turning on was 570 frames which corresponds to a latency of 1.1875 which is quite slow. When the program is deployed to hardware, the number of frames between lifting the cover and the LED turning on was 46 frames which corresponds to 0.046 seconds which is almost instantaneous and is far faster than human reaction time.

E. Motor Response from Encoder

As can be seen in Fig. 47, the responses of both of the motors to a PWM output from 0 to 255 is nonlinear. The rotational speed of the motors at becomes jittery and all over the place at around 200 RPM. Fortunately, the response of the two motors are similar to one another in the fact that their behaviour is similar across the PWM output.

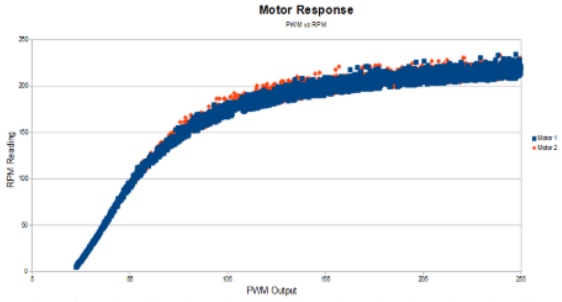

Fig. 47. Motor Response

F. IR Sensor Response

As seen in Fig. 48, the output voltage of the Sharp IR sensor is nonlinear. The response of the Sharp IR sensor is hyperbolic where the voltage output is proportional to the multiplicative inverse of the distance in cm. This behaviour is in line with the datasheet of the device. The sensor was properly calibrated by computing for the inverse of the mapped voltage. The output was accurate for distances from 10cm to about 30cm where voltage vs inverse of distance is linear according to the data sheet.

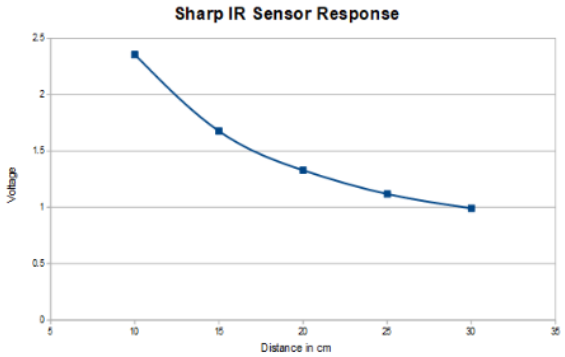

Fig. 48. Sharp IR Sensor Response

V. Conclusion

The initial test show that RGB is relatively unreliable compared to YCbCr method. The blobs were noisy with RGB method, while it was better tracked in YCbCr. The test was not very conclusive on which method is superior since a more controlled environment with no noise is needed to accurately see the response of the image processing algorithms. Both RGB and YCbCr method sees the color yellow, orange, and red as one. Thresholding becomes a challenge because of this problem. Using the HSV colorspace produced the best result because it can isolate the color red from the frame, leading to a cleaner thresholded frame. With the HSV method, a latency of 0.096 seconds was observed which is far faster than human reaction time. The PID code in the Arduino will still be tuned to obtain a better response from the motors.

REFERENCES

[1] D. Bates, "Great Crawl of China: Vendors cash in on 60-mile traffic jam that's lasted 11 days - with no end in sight", Daily Mail, 2010.

[2] V. Chin, M. Jaafar, J Moy, M. Phong, S. Wang, M. McDonnell, and I. Prawiradinata, "Unlocking Cities: The Impact of ridesharing in Southeast Asia and beyond," The Boston Consulting Group (BCG), November 2017.

[3] S. Handy, "Increasing Highway Capacity Unlikely to Relieve Traffic Congestion," National Center for Sustainable Transportation, October 2015.

[4] Y. Sugiyama, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S. Tadaki and S. Yukawa, "Traffic jams without bottlenecks—experimental evidence for the physical mechanism of the formation of a jam", New Journal of Physics, vol. 10, no. 3, p. 033001, 2008.

[5] J. Bierstedt, A. Gooze, C. Gray, J. Peterman, L. Raykin, and J. Walters, "Effects of Next-Generation Vehicles on Travel Demand and Highway Capacity," FP Think Working Group, January 2014.

[6] "Honda Sensing", Honda Automobiles, 2018. [Online]. Available: https://automobiles.honda.com/sensing. [Accessed: 10- Oct- 2018].

[7] Kesting and M. Treiber, "How Reaction Time, Update Time, and Adaptation Time Influence the Stability of Traffic Flow", Computer-Aided Civil and Infrastructure Engineering, vol. 23, no. 2, pp. 125-137, 2008.

[8] W. Pananurak, S. Thanok and M. Parnichkun, "Adaptive Cruise Control for an Intelligent Vehicle," In International Conference on Robotics and Biomimetics, 2008.

[9] Ioannou and C. Chien, "Autonomous intelligent cruise control", IEEE Transactions on Vehicular Technology, vol. 42, no. 4, pp. 657-672, 1993.

[10] H. Young, R. Freedman and A. Ford, University physics. Harlow: Addison-Wesley, 2011, p. 1146.

[11] B. M. Rohit and M. M. Patel, "Nighttime vehicle Tail light detection in low light video frames using MATLAB," International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 3, no. V, pp. 27–31, May 2015.

[12] "Median FIlter," 2-D median filtering - MATLAB. [Online]. Available: https://www.mathworks.com/help/images/ref/medfilt2.html. [Accessed: 10-Oct-2018].

[13] H. Rashmi, M. Shashidhar, and G Prashanth Kumar, "Automatic Tracking of Traffic Signes Based on HSV", International Journal of Engineering Research & Technology (IJRASET), vol. 3, Issue 5, May 2014.

[14] Pulse-width modulation. Massachusetts: Massachusetts Institute of Technology, 2000, p. 1.

[15] T. Tarroza, M. Bouziane, E. Beudaert, and H. Carpentier, "INCASE: Lowcost mobile robot pilot using integrated design:Rapid prototyping using MATLAB,Simulink and Stateflow."

[16] S. Kolkur, D. Kalbande, P. Shimpi, C. Bapat, and J. Jatakia, "Human skin detection using RGB, HSV and YCbCr color models", August 2017.

[17] "Raspberry Pi 3 Model B: Welcome to RS," RS Components International. [Online]. Available: http://www.rs-components.com/raspberrypi. [Accessed: 10-Oct-2018].

[18] "Camera Module V2," Raspberry Pi. [Online]. Available: https://www.raspberrypi.org/products/camera-module-v2/. [Accessed: 10-Oct-2018].

[19] "Distance Sensor (100-550cm) - Sharp IR GP2Y0A710K," Circuitrocks Philippines. [Online]. Available: https://circuit.rocks/distance-sensor-100-550-cm-sharp-ir-gp2y0a710k. [Accessed: 10-Oct-2018].

[20] "Magnetic Encoders" Data Sheet, Shayang Ye Industrial, [Accessed: 10-Oct-2018].

[21] "Image Processing Toolbox," MATLAB. [Online]. Available: https://www.mathworks.com/products/image.html. [Accessed: 10-Oct-2018].

[22] "SIMULINK," Simulink Product Description - MATLAB & Simulink. [Online]. Available: https://www.mathworks.com/help/Simulink/gs/product-description.html. [Accessed: 10-Oct-2018].