Advanced Version Of Collision Detection Unit

Aavishkar 2017

Aditya Singh (B16085) *, Amrendra Singh (B16010) †, Anant Mishra (B16011)‡ , Chirag Vashist (B16094) §, Nikhil T R (B16066)¶, Sujetth Rangannath (B16036)‖

{ * B16085, † B16010, ‡ B16011, § B16094, ¶ B16066, ‖ B16036} @students.iitmandi.ac.in

***Abstract*—The project AVOCODU (Advanced Version Of COllision Detection Unit) aims to bring some important high-end safety features to low end cars at significantly low costs. Features implemented in AVOCODU include collision detection and warning while driving, blind spot detection, prevention of drunk driving, gear change assist and communication to family and friends in case of any mishappening.**

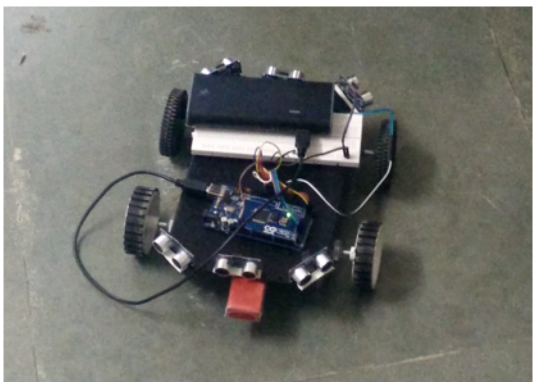

Fig. 1. Avocodu

I. Introduction to Avocodu

These days it is very essential to learn driving. But the busy roads, even in the morning, make it difficult for beginners to learn. They often end up hitting something with their vehicles. It's very difficult for them at beginning to decide on which gear should the car be driven. This gear problem is even faced by experienced drivers if they shift from a plain lying area to some hilly area or vice versa.

In order to solve these problems and assist drivers, this product is useful. Here is a brief introduction of what the product can do.

In case, the car is about to hit into an obstacle in the front and the driver doesn't apply the brakes in time, this gadget will slow the car down and when the collision is imminent, it will stop the vehicle. Similarly, during lane change, presence of a blind spot prevents the car from turning and push it forward instead. An alcohol sensor used in the car will allow it to start only if the driver blows into it and alcohol levels are found to be less than the legally permissible. The same features are highlighted in more detail in the trailing sections.

The following sections contain detailed description about

II. Gear Suggestion

Talking of introducing features to make drivers comfortable, one feature primarily targeted at new driving learners or the people new to the terrain e.g., a person from plains moving to hills or from hills to plains.

This can act as a way to suggest gears to the driver. If the driver has already put a gear, then they can verify if they used the correct gear. And if they are completely clueless, then they can use the suggested gear.

On simple research, we observed that change of gear depends on the inclination of the path, desired or possible speed of the vehicle, the torque of motors, the surface of the road and a few other factors [1].

The two primary factors found to be involved are speed and inclination. Due to budget constraints, we used motors which could not provide us with the speed, hence we kept suggestions based only on the inclination of the vehicle. As a thumb rule, lower inclination implies higher gear and vice versa.

To get the inclination of the vehicle we use the InvenSense MPU-6050 6-axis inertia measurement unit.

A. The InvenSense MPU-6050

The InvenSense MPU-6050 sensor contains a MEMS accelerometer and a MEMS gyro in a single chip. It is very accurate, as it contains 16-bits analog to digital conversion hardware for each channel. Therefore it captures the x, y, and z channel at the same time. The sensor uses the I2C-bus to interface with the Arduino. The MPU-6050 is not expensive, especially given the fact that it combines both an accelerometer and a gyro. The sensor has a "Digital Motion Processor" (DMP), also called a "Digital Motion Processing Unit". This DMP can be programmed with firmware and is able to do complex calculations with the sensor values.
This device communicates with the Arduino board using a I2C bus. This acts as a slave device to the microprocessor. This device can be connected to more device in which case the MPU-6050 will act as master device for it.[2]

III. The Buzzer Alert System

A major chunk of accidents take place due to distracted driving and fatigue. In both the scenarios, the driver's alertness is reduced.

"Distracted driving" is driving while doing something else that reduces attention towards the road. Activities like eating, talking on the phone, listening to music, texting or even using the GPS built in the car can distract the driver and endanger the driver and others. In 2013, 424,000 people were injured in motor vehicle crashes involving a distracted driver, an almost 10% increase since 2011.[3]

Countries have made rules to reduce use of phone and texting but their effectiveness has been very low.

Lack of sleep leads to fatigue which can reduce driver's alertness. Sleep deprivation increases the risk of crash related accidents. The less the driver had sleep, the greater is the risk. There is no test to determine sleepiness as there is for intoxication, i.e. like a "Breathalyzer". A study by researchers in Australia showed that being awake for 18 hours produced an impairment equal to a blood alcohol concentration (BAC) of .05, and .10 after 24 hours(.08 is considered legally drunk).[4]

A buzzer alert system can alert the driver of incoming vehicles, objects, and give sufficient time to the driver to react and prevent accidents. In this project, piezo-electric buzzer was used to alert for incoming objects. The buzzer gave different sounds depending on the distance of the object using the input given by the ultrasonic sensors.

IV. Alcohol Detection

Consuming alcohol in even small quantities causes decline in visual and mental functions. Higher dosage can cause reduced coordination, deteriorated muscle movement and poor information processing capabilities. Thus, intoxicated drivers are prone to be involved in accidents.

The "Alcohol Detection Unit" aims to reduce the number of road accidents caused by drivers under influence. There are primary two components in this unit : the MQ3 Alcohol Gas Sensor and SIM800C GSM Unit[Sec. V] The alcohol sensor can be placed on the dashboard or at the either side of the driver's seat so that the alcohol levels of the driver can be monitored easily.

When the driver tries to start the car and if his alcohol levels are higher than the legal levels, the car will not start. If the driver persistently tries to start the car, the buzzer will go off, attracting the attention of the nearby people. Additionally, a message will be sent to the relatives and the guardians of the driver via the GSM Unit, informing them of the driver's state.

If the drivers get drunk while driving, the alcohol sensor will slow down the car gradually, so that the driver can park it at the roadside safely. [5]

A. MQ3 Alcohol Gas Sensor

The MQ3 alcohol gas sensor gives analog value corresponding to the alcohol content of the air. The readings are generated within 2 seconds and pretty accurate. Thus MQ3 sensor is ideal for usage as a breath-analyzer. MQ3 uses a layer of Tin Dioxide (SnO_2) to detect the prescence of alcohol. In clean air, the conductance of SnO_2 is low, but as alcohol comes in contact with SnO_2, it converts into the respective acid, which has higher conductivity than the clean air. Thus MQ3 sensor gives its reading based on the changing conductivity levels of the SnO_2 layer. The sensitivity of the unit can be changed using a potentiometer.

Apart of alcohol, the MQ3 sensor also has a small response for benzene and gasoline. The MQ3 sensor also requires preheating time of approximately 20 minutes before usage in order to get stable values. This is because the above reaction requires heat.

There are lot of MQ3 unit types and each has its own load resistance. Thus individual units need to tested and calibrated for the precise trigger point. Through experimentation, the reading of 280 of the MQ3 analog was finalized as the trigger point. Overall, MQ3 gives accurate and reliable readings quickly and is relatively cheap. Thus it was considered appropriate for this project. [6]

V. GSM Communication System

GSM stands for Global System for Mobile Communication and is the most widely used system for mobile communication. GSM operates at various frequency channels, but the GSM Unit SIM800 operates at 900MHz or 1800Mhz.

Using the SIM800 unit, we can get the all functionalities of a mobile phone including sending and receiving SMS and phone calls. The SIM800 unit requires a 2G SIM for its functioning, which needs to properly calibrated and registered. GSM Units use a set of commands called the AT commands, for setup, control and transmission. Here's the summary of a few AT commands that we used to setup this unit.

The AT command that we deployed to alert the relatives/guardians of the driver was AT+CMGS.[7]

$AT + CMGS =< number >< CR >< message >< CTRL - Z >$

VI. Algorithms

All the below algorithms are implemented based on the readings of the 6 sonars. 4 of these sonars are placed on the four corners and the remaining 2 are placed on the front and the back.

A. Collision Detection Assist

A huge problem that new drivers face is that they fail to judge the nearest neighbour distance for the vehicles around them are (mostly because of the fact that the steering wheel is placed on one side).

The aim of the collision detection assist is to alert the driver is case a collision is imminent. The algorithm primarily takes two things into consideration:

1) How far are the other vehicles from the driver's vehicle (Distance)
2) How fast is the driver approaching the surrounding vehicles (Relative velocity)

$$Relative\,Velocity\ =\ \frac{Current\ Distance - Previous\ Distance}{Time\ gap\ between\ the\ readings}$$

Note : The relative velocity is measured by subtracting the current and previous distances measured by the sonar and then dividing the result by the amount of time it took the loop to execute

This algorithm accounts for 5 major cases:

1) There are no surrounding vehicles. In this state, nothing happens. This could be the case when the driver is on a highway or is driving between two cities, etc.
2) There is atleast one vehicle in the vicinity of the driver but the driver isn't approaching the vehicle at an alarming speed. In this case also, no alerts are shown.
3) There is atleast one vehicle in the vicinity of the driver and the driver is approaching it at a very fast pace. (i.e The speed of the driver's car relative to the other vehicle is very high). In this case, a buzzer starts beeping. This is to notify the driver to reduce his/her speed.
4) The driver's vehicle is really close to a surrounding vehicle and there still exists a negative relative velocity between both the vehicles. In this case, the buzzer starts beeping (this time, the beeping pattern and the frequency of the beep is different). This is to notify the user to immediately stop the car and maintain a safe distance from the other vehicle.
5) The driver's vehicle is really close to a surrounding vehicle but the relative velocity is almost zero. In this case, nothing happens. This could be the case when the driver is driving in traffic.

B. Parking Assist

The parking assist is mostly similar to the collision detection assist. There is a parking switch on the remote, which when pressed will call a function to change the distance and relative velocity thresholds. This is done because the minimum allowable distances are obviously different in case of normal driving and in case of parking. Apart from this, the remaining part of the algorithm is mostly same as Art. VI-A.

C. Lane Change Assist

A lot of road accidents take place simply due to the negligence of the driver when he/she is trying to change lanes but doesn't notice a vehicle travelling on that lane at a pretty high speed and still tries to turn. The aim of the lane change assist is to try and prevent a collision in such cases by intimating the driver about this beforehand.

The lane change assist mainly takes into consideration the 4 sensors on the corners of the car. In order to understand the working of the algorithm, consider the following situation:
The driver is trying to move towards the left lane but there is a car diagonally opposite to ours on the left side. Please consider the sonar at the bottom left corner. Now there are two cases:

1) The distance isn't decreasing between both the cars. In this case, it is safe to turn (If there is a safe distance between the two cars).
2) The distance is decreasing : This means that the other car is already in a position where it is impossible for the car to turn. In this case, the buzzer starts beeping. If the driver doesn't respond quickly enough, the lane change assist will automatically apply mild brakes (given there is no vehicle directly behind ours) and prevent the collision.

TABLE I
MAPPED FORWARD VALUES

Y Axis	forward
4 to 505	100 to 0
525 to 1020	0 to -100

TABLE II
MAPPED TURN VALUES

X Axis	turn
4 to 505	-100 to 0
525 to 1020	0 to 100

VII. THE REMOTE CONTROL OF THE DEMO MODEL CAR

A. Bluetooth

Bluetooth Remote Control: The car used for demonstration purpose is controlled using a bluetooth remote control requiring one extra arduino (mega or uno) board (other than one already placed onto the car chassis), two bluetooth modules (HC-05 used for this project) and connecting wires. Pairing only occurs between two bluetooth modules if and only if one of them is master and the other is slave. AT commands are used for controlling the Master/Slave behaviour of HC-05. By default, all HC-05 modules are in Slave configuration. Two Bluetooth HC-05 modules are used such that one is connected with an arduino (mega) board further connected to a Joystick (acting as remote-control) in Master configuration (AT+ROLE=1) and the other one is connected to the arduino board of the car in the default Slave (AT+ROLE=0) configuration (there's absolutely no issue doing it other way round).

B. The Joystick Based Control

The axes vary from 0 to 1023 where the centre rest position for the joystick is around (512, 512). The Y axis was chosen to control the forward and backward movement of the car, while the x axis controlled turning of the car.

According to the orientation of the joystick that was chosen, the 0 on the Y axis represented a full throttle towards front, and 1023 the exact opposite. . The 0 on the X axis represented left and 1023 towards right. The value of the Y axis was stored in forward and the x axis was stored in turn.

It was decided to leave a little margin of error for the rest position. On the Y axis, 4-505 was selected as forward, and 525-1020 was selected as backward. Doing this gave us various levels of speed. For X axis 0-505 was selected as left and 525-1020 as right.

To disable rotation at one place, any region on the joystick that had a higher magnitude for turn that forward, was modified for interpretation. Wherever the magnitude of forward was lesser than the magnitude of turn, its magnitude was raised to the magnitude of turn, while maintaining its original sign.

In order to communicate the three values: parking state, forward, and turn with the remote effectively and quickly, a method was developed. It involved to send a 10-character string to the car in each packet of data sent to the car.

The first character was made to represent the parking state. It was either 0 or 1. The next character represented the sign of the value of forward. 1 for negative and 0 for positive. The next 3 characters represented the absolute value of forward. 0100 for 100 and 1100 for -100. The next character represented the sign of the value of turn. 1 for negative and 0 for positive. The next 3 characters represented the absolute value of turn. 0100 for 100 and 1100 for -100. The last character was chosen to be '\n'. This actually represents newline. This was chosen to act as a trigger to the decoder to begin interpreting the string. The interpreting of this string was done at the car using a simple algorithm. First of all, the input string was converted to a 9-digit number. The data type unsigned long on the Arduino was a good selection for this. Using modulo operator, a reminder for a particular operand can be found, the numbers are decoded. In this way, the car is able to interpret 3 values: parking state, forward, and turn.

The function Steer(): The function Steer accepts 2 values, namely, forward and turn, both in the range of 100 to 100. Since PWM pins were used to control the speed of the motors on left and right, it was identified that selecting 2 variables left and right (whose values would be calculated using forward and turn) would be appropriate. The value of left was calculated as the sum of forward and turn. The value of right was calculated as the difference of forward and turn. This logic came up by trial and error. To ease calculation, the 2 values left and right were constrained to be in the range -100 to 100. Later on, their absolute values were scaled from 0 to 255. This value was then fed into the respective PWM pin for the motor, where 0 meant rest, and 255 meant rotation at full speed.

Associated with each motor are 2 Digital pins that help in controlling direction of rotation of the motor. When one of the pins is high, and the other is low, it will rotate in one direction. When the polarities are reversed, the direction of rotation of the motor is also reversed.

VIII. Conclusion and Future Possibilities

At the end of the project, the following features in the model were working: [The model was controlled with a joystick]

1) The buzzer beeped based on relative velocity and distance triggered collision detection assist, lane change assist and parking assist units.
2) Brakes were automatically applied if collision is imminent.
3) The alcohol detection unit detected alcohol above a particular threshold and sent a SMS to the preset phone number.
4) The gear suggestion system displayed the appropriate gear on the seven segment dispaly. For the model, we assumed the speed to be linearly proportional to motor input voltage because it was not possible to use a tachometer due to budget limitations.

We achieve nearly 100% of our expectations. The responsiveness of the unit was exemplary. The algorithm also handled basic and most common exceptions or corner cases that arise in practical situations. Also, if this is made a marketable product, it would be very easy to play out appropriate pre-recorded audio messages using a speaker instead a simple beep. This audio playback was not required in this simple model.

TABLE III
Equipment Price List

Motors x 4	760	HC-05 x 2	618
Motor drivers x 2	380	Wheels x 4	100
Chassis	220	Seven segment x 2	20
Wires	360	Dual axis joystick	149
MPU6050	229	SIM800C GSM module	620
MQ-3 Sensor	219	Arduino Mega2560 x 2	1498
Buzzer	90	Sonars x 8	872
Breadboard	70	Li-Ion battery	600

Total budget : Rs. 6805.
The budget spent on the essential modules(collison detection, parking assist, lane change assist, gear suggest and alcohol sensing) was Rs. 3699 while the rest of the amount was for building a model to demonstrate the working of the system.

Driving is something that every one of us mostly does some stage or the other. Self driving cars available in the market are not available in India, and the ones avilable in developed countries are not affordable by anyone but the very rich. It will probably take a decade for them to be available to commoners. Since with driving there naturally comes the risk of accidents we think that a product like this, which does not aim to completely take over the driving task but merely assists the driver by adding an extra layer of safety would be a significantly useful product.

References

[1] Gear vs Torque, http://www.iosrjournals.org/iosr-jmce/papers/vol11-issue2/Version-7/G011273541.pdf

[2] MPU6050 arduino.cc http://playground.arduino.cc/Main/MPU-6050

[3] National Center for Statistics and Analysis, Distracted Driving: 2013 Data

[4] , Driving Facts http://drowsydriving.org/about/facts-and-stats/

[5] AT Commands, https://www.cdc.gov/motorvehiclesafety/impaired_driving/impaired-drv_factsheet.html

[6] AT Commands, http://www.ijemr.net/DOC/AlcoholDetectionWithVehicleControlling(20-23).pdf

[7] AT Commands, http://m2msupport.net/m2msupport/atcops-plmn-selection/