Executive summary

Initially our team set a goal of the project as to design a device that will help drivers identify and anticipate obstacles in their vehicle's blind spot. This device may prevent car accidents and reduce number of fatal collisions. It is expected that device will be able to control the user vehicle in order to prevent the user from colliding into obstacles in the blind spot.

Our device consists of three main modules. First, obstacle detection is implemented using input ultrasonic sensor SRF05 and microcontroller Arduino Uno. Tanzim is responsible for this part and I am assisting him for developing the algorithm, coding and testing the module. We already tested it for the case when both user car and obstacle are stationary, and results show that sensor gives correct distances to the obstacle. Next, we tested it for case when user car is stationary and obstacle is approaching the sensor. Here the goal was to develop the algorithm and write a code to correctly calculate a relative speed of obstacle with respect to user vehicle, and we accomplished that. Results show that calculated relative speed varies in range +-5% of the actual known speed of obstacle. Next task is to set whole module to the prototype (RC car) and test based on case of both user vehicle and obstacle moving. We expect that friction between wheel and ground will cause disturbances in sensor operations and give oscillatory noise in the result.

The second part is an output module. We need this module to show user a relative speed of approaching car, distance to obstacle car, and time remaining to change a lane safely. I am responsible for this module, and Tanzim assisted me in combining obstacle detection and output modules. After research on the operation of LCD s, I completed the code for displaying a data in LCD display. Also LED was programmed in a way it blinking according to distance to obstacle. Next step to consider is to improve the appearance of the data on LCD screen, especially relative speed of obstacle, which is changing its value too fast making it not fully detectable by human eyes at high rate of obstacle speed change.

In general, the big part of the project is accomplished, however the project is going not according to initial gannt chart plan, and delayed approximately for two weeks due to challenges that arise from implementing device on moving prototype. Next steps are to achieve correct results in testing moving user and obstacle, and implementing automation module on RC car, which the third module of the project.

Work Breakdown Structure:

In the following table the work breakdown structure is illustrated. Here the tasks that I am responsible for are bolded and tasks that I assist on are italized.

Task Number	Task	Tanzim	Hani	Valikhan
I. Initial Research				
1	• *Input Methods (Sensors)*	R		A
2	• **Output Methods (LED)**			R
3	• Processing Methods (Arduino)	A	R	
4	• Blind spot angles, sensor detection angles & ranges	R	A	
II. Obstacle Detection				
5	• *Program the Arduino to retrieve input from the sensors*	R	A	
6	• **Program the Arduino to set up output display**		A	R
7	• *Integrate the input and output components to provide user feedback according to the microcontroller's calculations*	R	A	A
8	• *Install the completed obstacle detection module to a RC car*	A	R	A
III. Vehicle Anticipation				
9	• **Design an algorithm to detect approaching vehicles and calculate their relative speed**	R		A
10	• Refine safe distances and time constants to make lane changes based on testing		R	
11	• Program the Arduino to provide feedback warning users of approaching obstacles	A		R
IV. Automation				
12	• Achieve full control of the RC vehicle through the Arduino	A	R	
13	• *Program the Arduino to lock the steering of the user vehicle to prevent unsafe lane changes*	A	R	A

Task 1: Research on Input Methods (Sensors)

- **Description**: The task consists of research on sensors which will have characteristics that best satisfy project technical requirements.
- **Category**: Old
- **Old Completion Date:** Sep. 10, 2011
- **Date completed:** Aug. 30, 2011

Responsibility : Valikhan Kuparov, Tanzim Mokammel

Status at start of reporting period: Started

Status at end of reporting period: Completed

- After researching on input sensors, several options are considered by design team. Based on different types and topologies the cost varies from $40 to $250

Actions

- My responsibility in this task was a research on laser sensors. Research on various models of laser sensors resulted in selection of following models:
  1. Simple laser range finder
  2. LR3 - Precision USB Laser Rangefinder

  I also did some research on how to improve a distance capability of our project, i.e. make it detectable for <3 cm. What I found is that almost all sensors require minimum of 4-5 cm to measure a distance. The following sensors show better distance capability:
  1. Ultrasonic Ranging Detector Mod HC-SR04 Distance Sensor
  2. Lego NXT Ultrasonic Sensor
  3. Eowave IR distance sensor

  Technical parameters of sensors above can be found in Appendix A.

Decisions

Once the list of alternative sensors is completed, we had to select one that will fully satisfy our project requirements. By analysis of pros and cons (Appendix B), we decided to use sensor based on sonar technology. Here ultrasonic sensor SRF05 is most optimal solution in terms of performance and cost for our project.

Testing & Verification, Final Results

- At this stage, testing was not performed. However in later stage when ultrasonic sensor is connected to microcontroller and microcontroller is programmed, testing of sensor was accomplished. But for now this is out of the scope of the Task1.

Task 2: Research on Output Methods (LED)

- **Category**: Old
- **Old Completion Date:** Sep 10, 2011
- **Date completed:** Sep 2, 2011

Responsibility : Valikhan Kuparov

Status at start of reporting period: Started

Status at end of reporting period: Completed

Actions

- I was responsible for research on output methods. These methods are based on different ways of human sensing (visual, auditory, physical). After analysis the following alternatives are proposed:
  1. graphical
  2. numerical
  3. simple-blinking light
  4. simple-beeping sound
  5. vibration of steering wheel

Decisions

- After analysis of alternative methods listed above, our team chose to incorporate several types into one output device. Proposed device consists of LCD display (numerical) and Blinking LED (visual).

Testing & Verification, Final Results

- Not available at this point.

Task 3: Program the Arduino to retrieve input from the sensors

- **Category**: Old
- **Old Completion Date:** Oct 20, 2011
- **Date completed:** Oct 12, 2011

Responsible : Tanzim Mokammel

Assisting: Valikhan Kuparov

Status at start of reporting period: Not started

Status at end of reporting period: Completed

Actions

- I assisted Tanzim in developing algorithm for a program of retrieving input data from sensor. Basic logic behind the program is that Arduino sends a set of pulse with delay after each pulse, and time of sending is stored. After pulse bounced from obstacle returns to sensor distance to obstacle is calculated by Arduino and current time is determined. Using physics law of motion, we are able to calculate relative speed of obstacle. Code is provided in Appendix C

Decisions

- On top level, there were no big conceptual problems to deal with. We clearly set a goal, and developed a code according to project requirements.

Testing & Verification, Final Results

- Set of tests includes placing different objects at known distance statically and comparing them with ones that is calculated by Arduino. All results from Arduino stay within +-5% error of measured values. Reading from the sensor of object placed at 19 cm away is shown in Appendix D.

Task 6: Program the Arduino to set up output display

- **Category**: Old
- **Old Completion Date:** Oct 20, 2011
- **Date completed:** Oct 15, 2011

Responsibility : Valikhan Kuparov

Status at start of reporting period: Not started

Status at end of reporting period: Completed

Actions

- I was responsible for programming Arduino to set up output display. Before writing a code, some research on operations of LCD displays is done. Piece of code I wrote is incorporated into Tanzim's code of retrieving data from input sensor. Part of the code is shown below:

```
lcd.clear();
// set the cursor to column 0, line 1
// (note: line 1 is the second row, since counting begins with 0):
```

```
  lcd.setCursor(0, 0);
  // print distance from obstacle

  lcd.print(current_distance);
  //lcd.print("");

  lcd.setCursor(0, 1);
  lcd.print(average_speed);
```

Decisions

- In LCD display we decided to show relative speed of obstacle and a distance to obstacle. Also time remaining to change a lane safely will be included in future. However due to delays in design, this is not possible now because we haven't finished implementing an algorithm for calculating a lane change safe time.

Testing & Verification, Final Results

- I verified the correctness of displayed speed value by comparing the numbers on LCD display with those calculated by Arduino (Appendix D)

Task 9: Design an algorithm to detect approaching vehicles and calculate their relative speed

- **Category**: Old
- **Old Completion Date:** Nov 15, 2011
- **New Completion Date:** Jan 29, 2012

Responsibility: Tanzim Mokammel

Assisting: Valikhan Kuparov

Status at start of reporting period: Not started

Status at end of reporting period: Delayed

Completed and tested for the case of a stationary sensor (stationary user vehicle), needs to be

verified when both user and obstacle vehicles are in motion

Actions

- Algorithm was designed to calculate relative speed of obstacle based on distance provided by sensor and internal timer of microcontroller. We store distances into array, and difference between consecutive distances divided by time it took to travel this delta distance gives us an instantaneous speed. By averaging it over particular number of times we get average relative

speed. This number was optimized through series of testing.

Decisions

- Selecting proper averaging factor and delay between consecutive pulses leads to better performance. Choosing averaging factor of 10 and delay time of 20 ms gives the best results, while other values result in speed oscillations on a plot.

Testing & Verification, Final Results

- We programmed Arduino in such a way that it gives a real time plotting of relative speed of obstacle. Here the flat portion of the plot corresponds to steady speed of approaching car. By comparing different plots, the averaging factor and delay time were chosen. Plot resulted from testing different averaging factors and delays are shown in appendix E.

Conclusion

At this moment a big part of the project is completed, including the retrieving data from sensor, programming microcontroller, displaying a relative speed of obstacle, and testing the module under static sensor and dynamic obstacle case. However project is delayed due to challenges with integration of three modules on RC car and achieving normal operation of the device. Approximate delay is two weeks. Next step to accomplish is integration of module on RC car and testing the entire system under different test cases. Once we are able to get correct performance of the device, we will integrate the automation module on the RC, which Hani is working on now.

Appendix A – Results of research on laser sensors

1. Simple laser range finder
I found the following rangefinder very cheap to implement. All we need is a simple laser and regular web-cam. The source code is also provided:
http://www.codeproject.com/KB/cs/range_finder.aspx

Pros:
a. low price.
b. easy to implement
c. source code is available

Cons:
a. only good at short distances (less than 2 m)
b. possibility of low accuracy
c. RC-car is going to be bulky if we use several lasers and webcams.

2. LR3 - Precision USB Laser Rangefinder

The second option is using existing device and make it possible to retrieve data
http://porcupineelectronics.com/Laserbotics.html

Range of detection: 0.1 to 30 meters
Accuracy: ±3mm
Interface: USB
Power Consumption: < 200mA (USB bus powered)
Measurement Rate: ~10Hz

Cost: ~100$, Fluke 411D laser measurement tool (amazon.com)
+ 149$ Laser Rangefinder Interface board

Pros:
a. more advanced technology
b. hardware is available
c. retrieving data (through USB cable to PC)

Cons:
a. costs totally ~250$ (not including software)
b. only demo software available. Further we must find a way to develop our own code.
c. RC-car is going to be bulky

Appendix B – Assessment of input sensor alternatives

Component	Possible solution	Type	Advantages	Disadvantages
Input - Obstacle Detection	SRF05 Ultrasonic Rangefinder		Good range (~ 3m) and accuracy (± 2% error), relatively cheap ($40)	Complex programming required; relatively expensive; prone to interference from miscellaneous obstacles
	LR3 - Precision USB Laser Rangefinder		Very high accuracy (+- 3mm) and range (30m)	Expensive ($150); requires external tools, very sensitive to light conditions
	Sharp IR Rangefinder R48-IR12		Very cheap ($15); immune to ambient light interference; low power consumption	Narrow beam width; prone to other IR sensor interference, low range (10 cm - 80 cm)
	Camera - image detection		Accurate; images can be reproduced for the driver	Complex image processing algorithms required; expensive

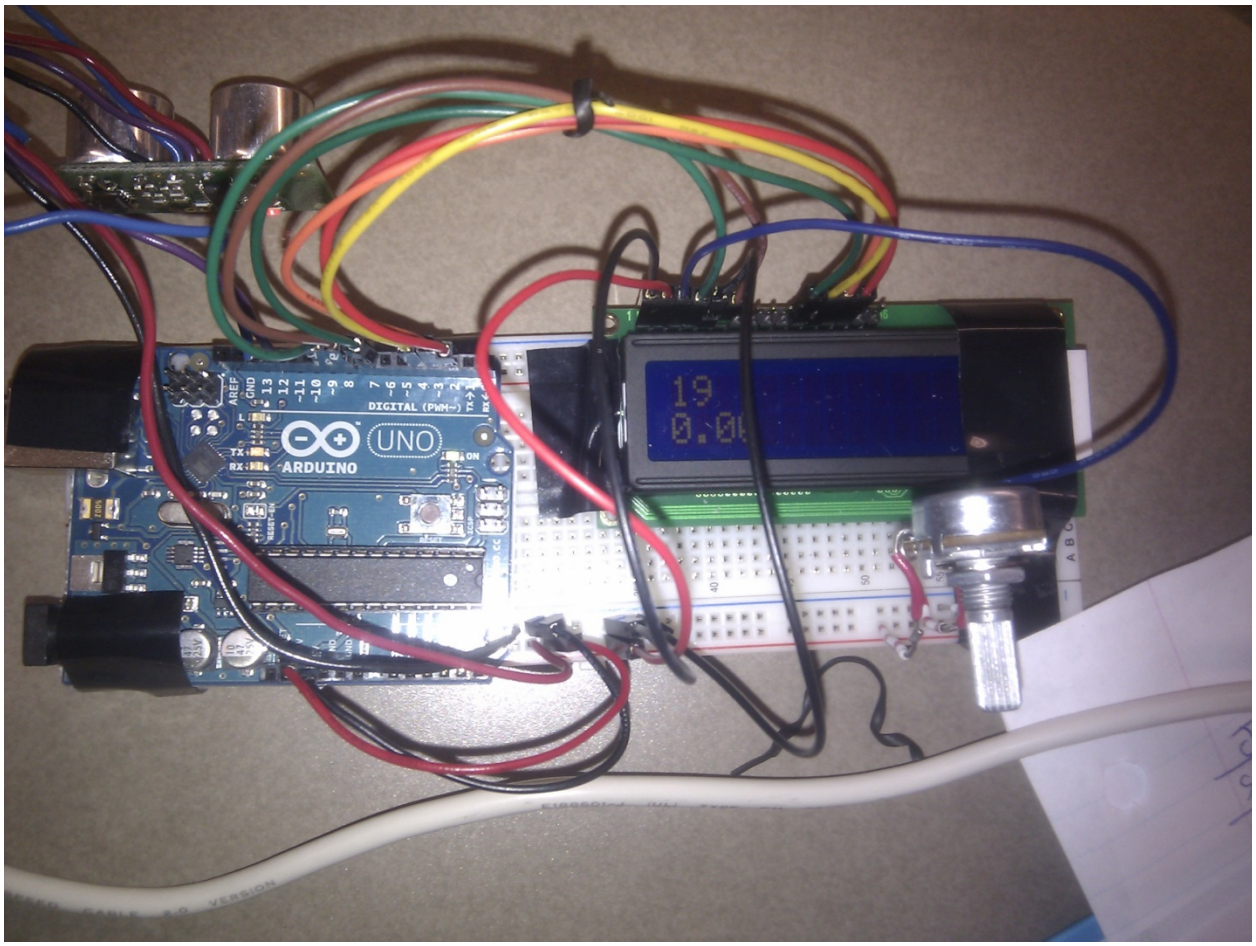
Appendix C - Code developed for retrieving data from input sensors

```
#include <LiquidCrystal.h>

const int NUM_READINGS =3;//number of readings for distance array (needs to be at least 2 for the speed calculations to work;
const int NUM_SPEED_READINGS =5;//number of readings for speed array
const int TIME_DELAY=20;//delay time after the entire process
const int TRIGGER_DELAY=10; //how long to wait for echo (don't change this from 10)
const int MAX_DELTA_D=40; //maximum allowed distance delta b/w two intervals to steady speed calculations

int total=0;
int a_index=0;
int current_index=0;
int average_distance=0;
int current_distance=0;
int distance [NUM_READINGS];

//for speed calculations
unsigned long startTime=0 ;              // start time for stop watch
unsigned long elapsedTime=0;             // elapsed time for stop watch
int old_distance=0;
int delta_d=0;
float inst_speed=0.0;
float average_speed=0.0;
```

```
float total_speed=0.0;
float speeds [NUM_SPEED_READINGS];
int s_index=0;

float time_left=0.0;    //time left until distance is zero

//hardware variables setup
int pin_echo=2;  //srf05 echo pin (dig 2)
int pin_trigger=3 ;  //srf05 trigger pin (dig 3)
int led_trigger=13; // (digital 13)
unsigned long pulseTime=0;  //stores pulse in microseconds
//lcd definition
LiquidCrystal lcd(12, 11, 7, 6, 5, 4);

//setup
void setup () {
   pinMode (pin_trigger, OUTPUT);
   pinMode (pin_echo, INPUT);

   //LED setup
   pinMode (led_trigger, OUTPUT); //sets dig. pin as output

   lcd.begin(16, 2); //sets up the lcd

   for (int cur_reading = 0; cur_reading < NUM_READINGS ; cur_reading ++) {
    distance [cur_reading] = 0;
  }
   //initialize speed array
   for (int cur_reading = 0; cur_reading < NUM_SPEED_READINGS ; cur_reading ++) {
    speeds [cur_reading] = 0.0;
  }
  Serial.begin(9600);
}
//external function to avoid rounding errors:
// printFloat prints out the float 'value' rounded to 'places' places after the decimal point
void printFloat(float value, int places) {
  // this is used to cast digits
  int digit;
  float tens = 0.1;

..... rest of the code is omitted
```

Appendix D – Module showing correct distance

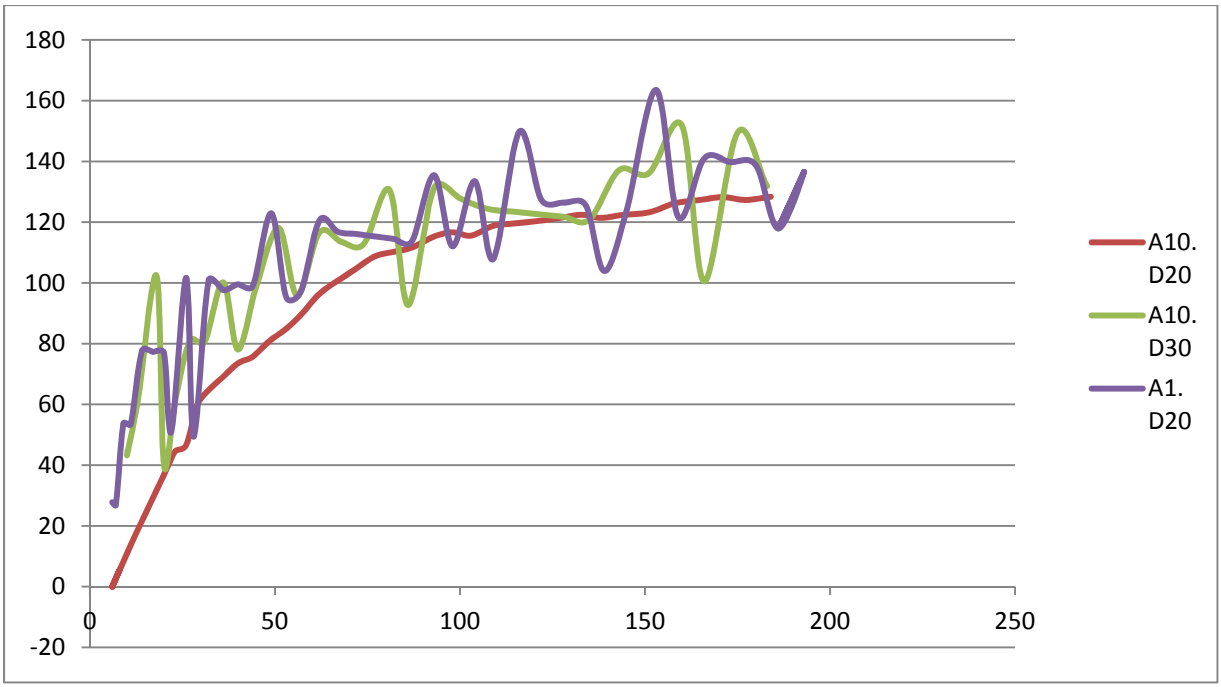
Object was placed 19 cm away from sensor. Programmed LCD display shows correct value.



Appendix E – Calculations of speed for case of stationary sensor

Initial Tests:

Here the speed of the obstacle against time is illustrated as it moves away from the sensor. This graph was constructed based on data from Arduino serial monitor and using Excel program. The averaging factor (A), and delay time (D) are varied.



Y-axis: Speed of Obstacle in cm/s

X-axis: Sample Point

A=number of stored instantaneous speed averages;

D= delay between consecutive pulses;

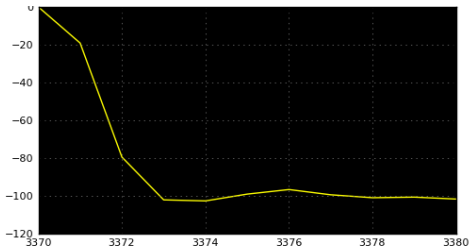
Based on initial data above, we get an estimation of the A, and D required to produce an acceptable output, which resulted as A=10, D=10.

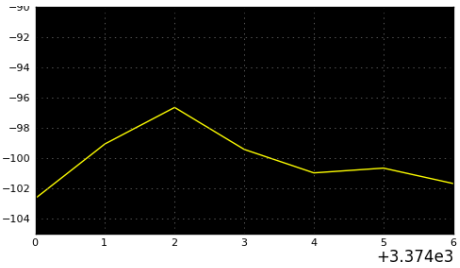
The following cases were tested:

A5, D15	A10, D15	A15, D15
A5, D20	A10, D20	A15, D20
A5, D25	A10, D25	A15, D25

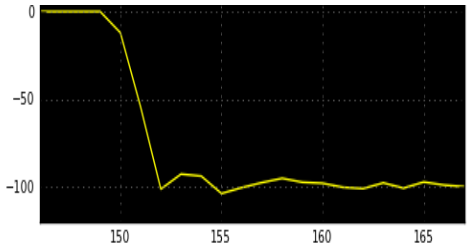
The following plots represent the speed of obstacle approaching the sensor. The right graph is zoomed in look at the flat portion of the left graph representing the constant average speed of the car. On the left graph, the slope shows how fast the sensor adjusts to the speed of the car. By adjusting averaging factor A and delay time D we can control the falling slope.

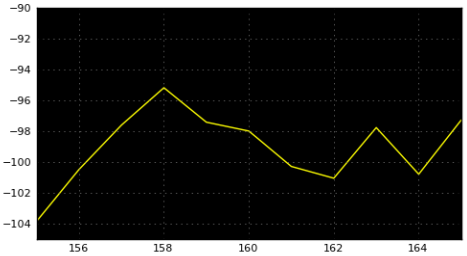
A5, D15



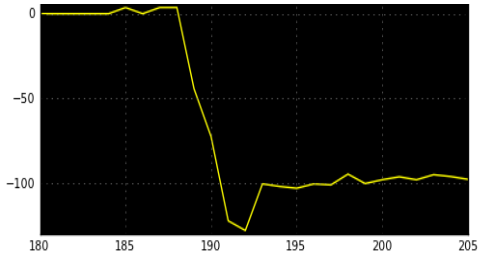


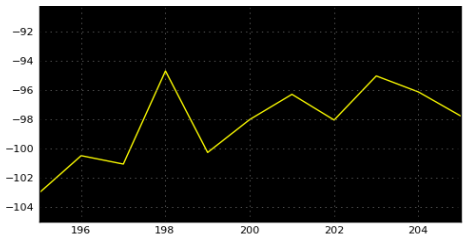
A5, D20



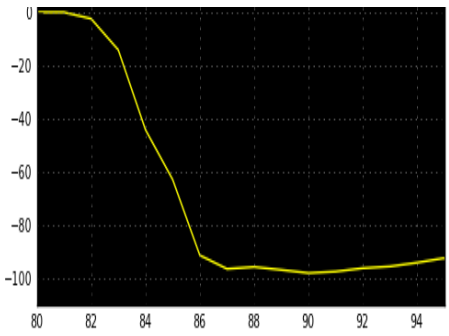


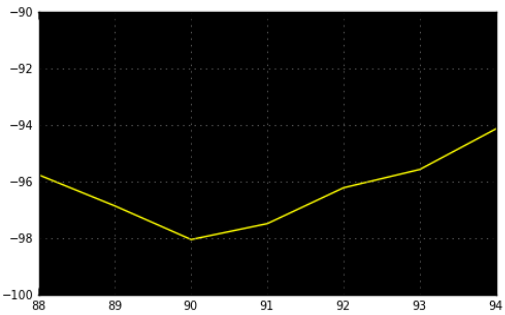
A5, D25



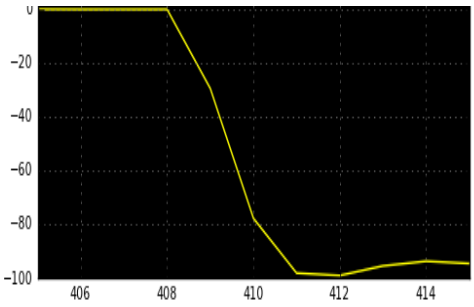


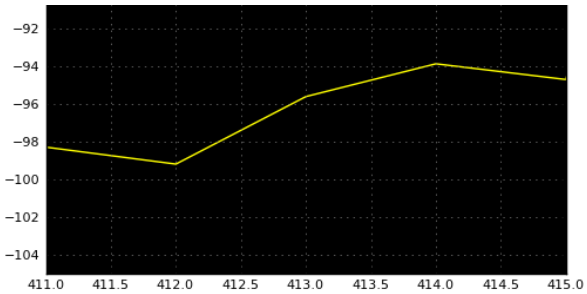
A10, D15



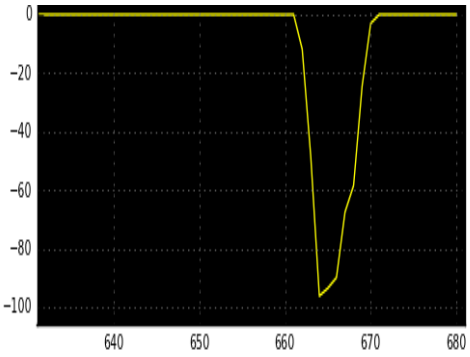


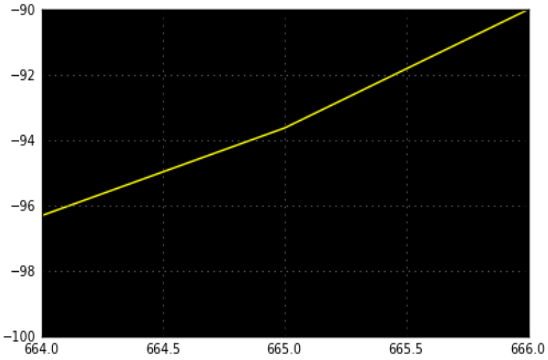
A10, D20



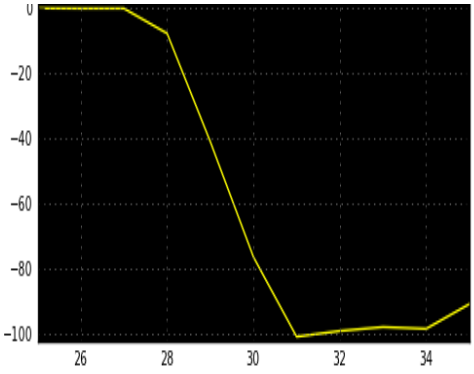


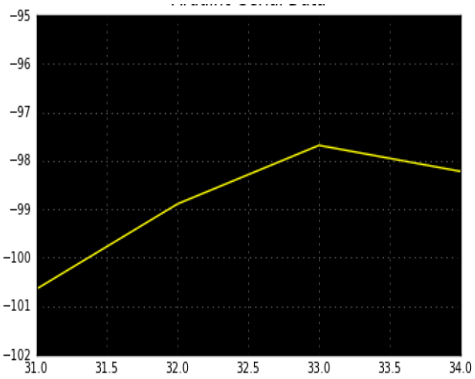
A10, D25



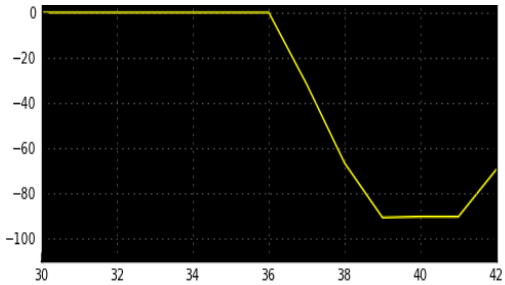


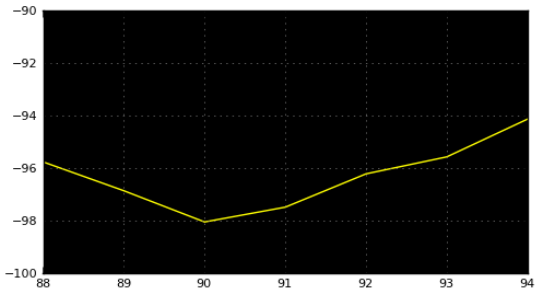
A15, D15



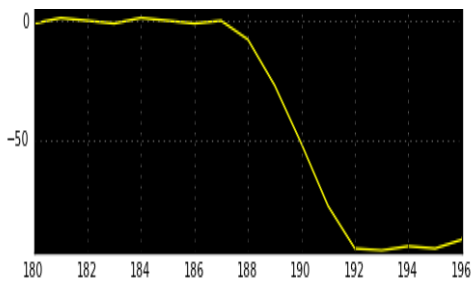


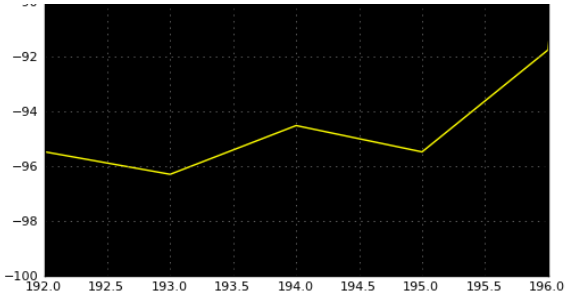
A15, D20





A15, D25





After analysis of the plots, our team has decided to use A10, D20 at this moment. It has less oscillatory output and a fast setup time (falling slope). However these parameters may change as we start testing a system on dynamic prototype. It is expected that friction between wheels and ground will introduce extra noise to sensor input and cause more oscillations on plots.