Remote Controlled Carry-on and Checked Luggage Carrier for High Loads

Ali Qureshi, Mateo Restrepo, Victor Rodriguez, Neha Chawla, Sabri Tosunoglu

Department of Mechanical and Materials Engineering
Florida International University
Miami, Florida 33174

aqure010@fiu.edu, mrest031@fiu.edu, vrodr188@fiu.edu, nchaw002@fiu.edu, tosun@fiu.edu

ABSTRACT

In this paper, we explain the necessity for building a luggage carrier and the manufacturing steps taken to produce a working prototype. The scaled prototype that is developed in this work performs the primary functions of the actual proposed design.

Keywords

Remote controlled, luggage carrier, carry-on, checked baggage, 100 lb load capacity.

1. INTRODUCTION

1.1 Problem Statement

Nowadays travelling has become an important part of people's lives. People travel for leisure, vacations or for work and carry luggage during travelling. Passengers with physical limitations and senior citizens find it difficult to carry their luggage around at the airport or at hotels. They need to get special assistance just for this purpose and sometimes waiting for them is inconvenient. Similarly, it is difficult for people travelling alone with babies or toddlers to carry their luggage and simultaneously carry the baby. Since the airports are massive these days, walking between terminals becomes tiresome with the luggage. There are wheeled luggage carriers available in market these days but they need to be pulled across at all times. A system to diversify the carrying capacity will be able to assist customers with transporting checked-in luggage as well would be a great solution.

1.2 Motivation

In order to facilitate the ease of traveling and aid travelers, we designed a robot that will carry luggage bags without having to push them around. We created a platform that is remote controlled through a bluetooth connection to an android phone. The platform will carry most commercial luggage bags and will support heavy loads. The robot will have lights all around in order to illuminate the path if the photoresistor detects no light. It will have ultrasonic sensors all around in order to prevent any collision. If the robot detects something ahead of it, it will avoid collision either by stopping at a set distance or by rerouting. The platform will be available in two different sizes. A large size which will be used by customers before the TSA checkpoint for check-in baggage, and a small size used by customers after the TSA checkpoint for carry-on baggage.

This robot can also be used in many other applications. Every job that requires something to be carried around could use this robot. The platform will be large enough to carry most luggage sizes and could also be used to carry tool boxes. The possible benefits that the user will experience is ease of travel. Another added benefit is that it can carry anything that fits in the platform, and it can be used in the dark because the robot will light up the path.

Our robot can be used as a personal item and can and can also be used instead of renting out the carts that are currently present at airports. The airport luggage cart business has been on a steady decline. According to an article in the *Los Angeles Times*, at "Los Angeles International Airport, cart rentals once provided at least $2.75 million in annual revenue. Now, the airport is losing nearly $1 million a year under a deal that obligates it to provide free carts to foreign travelers." [1] By addressing this problem, we hope to turn those finances around by making robotic mechanisms that would significantly decrease the workload for luggage carrying.

1.3 Literature Survey

Research was conducted to determine the luggage carriers with integrated robotic system. We found quite a few robotic luggage in the market with robotic systems that use a variety of sensors and automation. One of the carry-on luggage bags was a user controlled motorized carry-on luggage bag on which the user can sit and control the motion using a throttle and steering handle bars. Figure 1 below shows the design of luggage.

Other carry-on luggage bags that were seen within the research were the Cowarobot and the Travelmate Robotics carry-on luggage bags. These bags can be seen in Fig. 2 and Fig. 3 respectively. These two carry-on luggage bags are very similar in design. They use a Bluetooth tracking device to track the user. The Cowarobot luggage bag tracks a bracelet worn by the user, while the Travelmate Robot bag syncs with the user's phone Bluetooth in order to track. They both use multiple sensors to avoid obstacles that arise when the luggage is following its user.

Figure 1. MODOBAG Motorized Carry-on Luggage Bag [3]

Table 1. Airline Carry-on Regulation Dimensions [2]

Airline	Dimension
United Airlines	9” x 14” x 22”
American Airlines	9” x 14” x 22”
Southwest Airlines	10” x 16” x 24”
Alaska Airlines	10” x 17” x 24”
Delta Airlines	9” x 14” x 22”
JetBlue Airlines	9” x 14” x 22”
Hawaiian Airlines	9” x 14” x 22”
Spirit Airlines	10” x 18” x 22”
Virgin America	10” x 16” x 24”
Allegiant Airlines	9” x 14” x 22”
Horizon Air	10” x 17” x 24”
Frontier Airlines	10” x 16” x 24”
Sun Country	11” x 16” x 24”

Table 1 shows that for most major airlines, the smallest dimensions used for carry-on luggage bag are 9” x 14” x 22”. It would be reasonable to take these dimensions as the standard because if the carry-on luggage bag meets these dimensions, the carry-on luggage can be taken on any airline.

2. DESIGN CRITERIA

2.1 Limits of Weight and Power Calculations

We had in consideration the different applications that the luggage carrier may be designed for. In terms of weight, the carrier was designed to carry up to 100 pounds. For airports, the design will have two different versions, a small one personal up to 100 pounds and a scaled one up to 250 pounds considering the luggage requirements. The difference in load introduces a question for us about the torque required to carry the load.

Power required to move the vehicle in an inclined plane of 5 degrees:

$$Power = (F_{friction} + F_{grade} * Vel * \frac{1}{eff} \qquad (1)$$

$$F_{grade} = 9.81\left(\frac{m}{s^2}\right) * W_{carrier} * \sin(\alpha) \qquad (2)$$

$$F_{friction} = 9.81\left(\frac{m}{s^2}\right) * W_{carrier} * \ \mu_{min} \qquad (3)$$

$$Where: eff = 0.85, Vel = 10\left(\frac{km}{h}\right), \mu_{min} = 0.3$$

$$\& \quad W_{carrier} = W_{luggage} + W_{car}$$

Sample calculations:

For a maximum Luggage of 250 pounds and a carrier designed to weights itself 25 pounds the calculations are,

$$W_{carrier} = \frac{250 + 25}{2.22} = 123.87\ (kg)$$

$$F_{friction} = 9.81\left(\frac{m}{s^2}\right) * 123.87\ (kg) * \ 0.3 = 364.56\ N$$

$$F_{grade} = 9.81\left(\frac{m}{s^2}\right) * W_{carrier} * \sin(5) = 105.9\ N$$

$$Power = (364.56\ (N) + 105(N)) * 10\left(\frac{km}{h}\right) * \left(\frac{1000\left(\frac{m}{km}\right)}{3600\left(\frac{s}{h}\right)}\right) * \frac{1}{0.85} = 1534.51\ Watt$$

As previously explained the previous calculations are based on the assumptions that the vehicle will indoors with no significant air resistance force and that the vehicle will move always in a horizontal path. For cases in which these conditions differ the vehicle power consumption will need to be reanalyzed and recalculated. For the case of personal luggage carrier with 100 lb the power needed is 540 (W). This power is to be divided by the number of motors in use.

2.2 Size

The luggage carrier was designed to fit most standard airport doors and can be resizable upon request to fit future customers’ needs. The vehicle measures 22.69” wide and 24.41” in length, the principal measurements of the design can be seen in Figure 2. Even though the current system exceeds the standard dimensions for personal luggage carriers, the dimensioning was directed to luggage carrier assisters inside airports. So instead of having to look for a car, the passenger is greeted at the airport with one of this vehicle to assist him/her in translating the luggage up to the checking point. For personal luggage, we will design a smaller version to comply with airport standards. The current luggage design has a volumetric capacity of 1897 cubic inches.

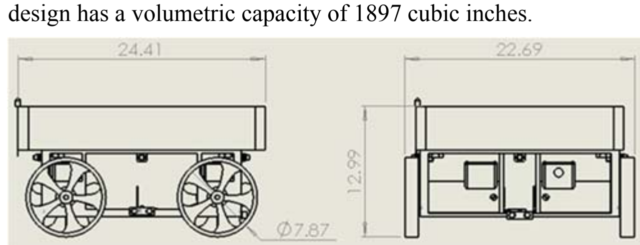

Figure 2. Smaller Platform Conceptual Design Size

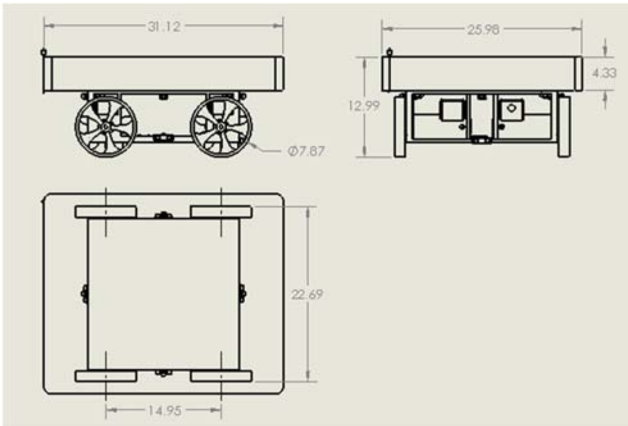

Figure 3. Larger Platform Conceptual Design Size

The current design will be able to fit even regular house door which are normally less wide than commercial doors. This design will also be useful to handymen and workers that need to carry several tools and want to work more efficiently.

2.3 Type of Case

We went for a basic type of box case. The box type of case will avoid objects from falling and the client will be able to organize his/her belongings using the case as a container. We will incorporate for future design modifications compartments to the case in order to assist better the intended market.

2.4 Possible Markets

As explained before the carrier can be a useful tool for the following markets:

- Airports
- Warehouses
- Supermarkets
- Moving Companies
- Care

2.5 Bluetooth and Override Control

We will control the carrier by Bluetooth. The Arduino microcontroller will have a Bluetooth sensor that will locate the other end of the Bluetooth connection (the user) and will send the location information to the controller. The controller will map its trajectory to the desired destination having in consideration the nearby obstacles. The carrier will have 4 ultrasonic sensors in the 2 plane directions.

We will also connect the microcontroller to a controller android application with an override function to avoid collisions in case the carrier may lose its connection with the Bluetooth device or the trajectory towards the location is full of obstacles and the installed sensors are not capable to input the necessary information to get out of the jam. The override instructions will also assist the design team in the testing and tuning of the robot.

3. FINAL CONCEPTUAL DESIGN

The luggage carrier proposed will have either a small platform, shown in Fig. 4, or a large platform, shown in Fig 5, that will carry most commercial luggage. The large platform will be rented out before checkpoint and are mainly for suitcases. While the smaller ones will be rented out passed checkpoint and are meant for carry-on luggage bags. This way the platforms will not be crossing through security, and doesn't create any safety liabilities. The small platform has a carrying space that measures 24" by 20". While, the large one has a carrying space of 30" by 25". The height of both the holding platform is 3.94". The luggage carrier has four 7.87" diameter wheels which have 5 spokes each. The robot has four ultrasonic sensors, one on each side in order to detect any obstacle. Figures 4 and 5 show the 3D model of the luggage carrier that we made using SolidWorks.

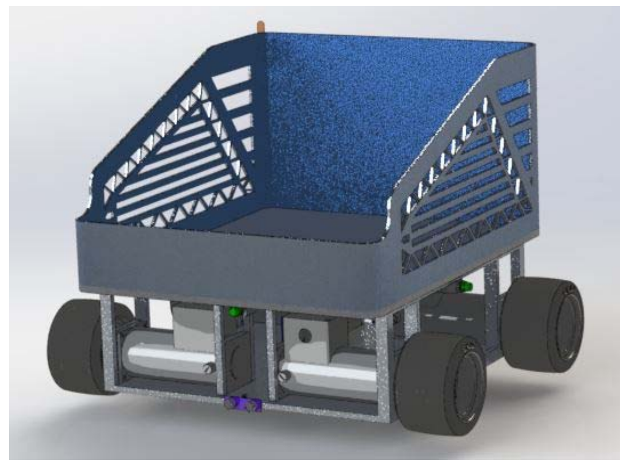

Figure 4. Smaller Platform Overall View of the Final Design

Figure 2. Larger Platform Overall View of the Final Design

Figure 6 shows a top down view of the layout of the components. The LED's that will be used to illuminate the ground are shown by the color green. The ultrasonic sensors are shown by the color purple and can be seen on each side of the robot along with the LED's. The microcontroller which is an Arduino UNO is placed in the center and is shown by the color purple. The photo-resistor sensor is shown by the orange color and is located at the top of the platform.

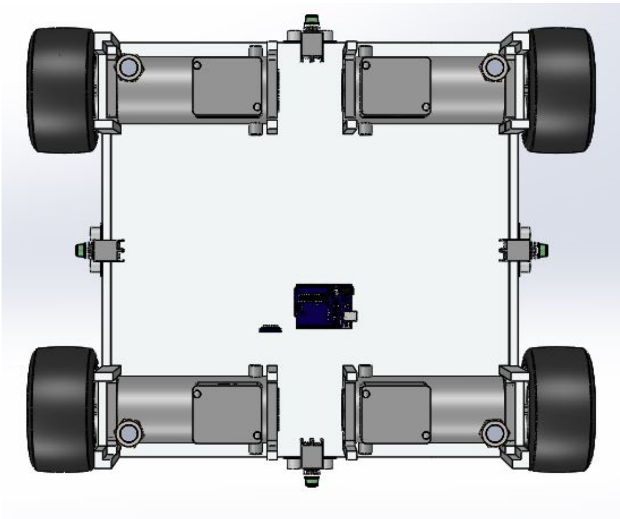

Figure 6. Top Down View of Components

To further show the configuration of the motors and the positioning of the ultrasonic sensors with respect to the microcontroller a cut view is shown in Figure 7. This figure shows that the ultrasonic sensors are located on every side of the robot, and that the microcontroller is in the center of the robot.

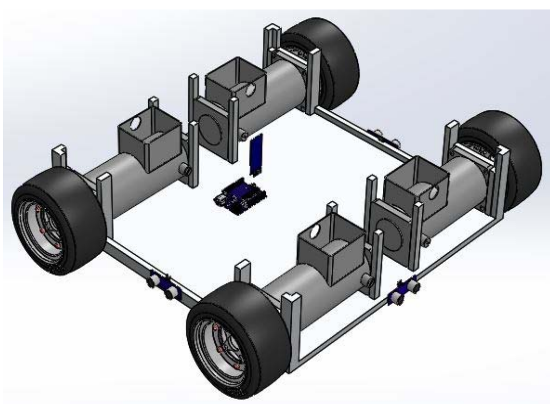

Figure 7. 3D View of the Internal Components

The luggage bags can be placed in the carrier in two ways as shown in the Figures 8 and 9. They show the representation using the larger platform, but the concept of placement is the same for the smaller platform.

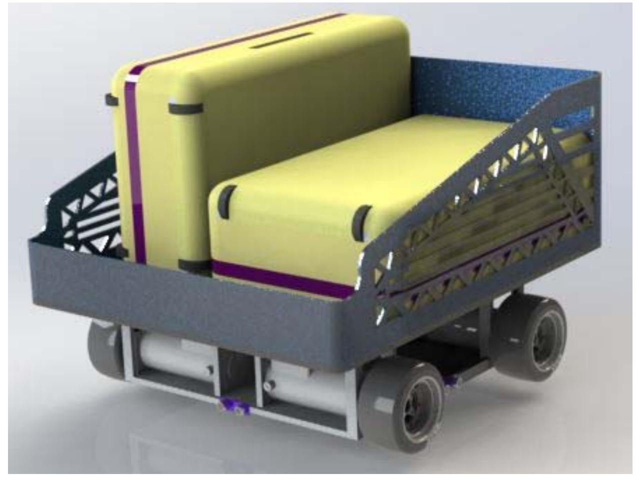

Figure 8. Carrier Loading Style 1

Figure 9. Carrier Loading Style 2

4. MODELING

4.1 Smaller Platform Simulations

Simulations were conducted on the smaller luggage carrier to simulate the weight of carry-on luggage bags. A 100lb distributed load was placed on the platform. The maximum stress occurring the platform 121 psi. Figure 10 shows the outcome of the simulation.

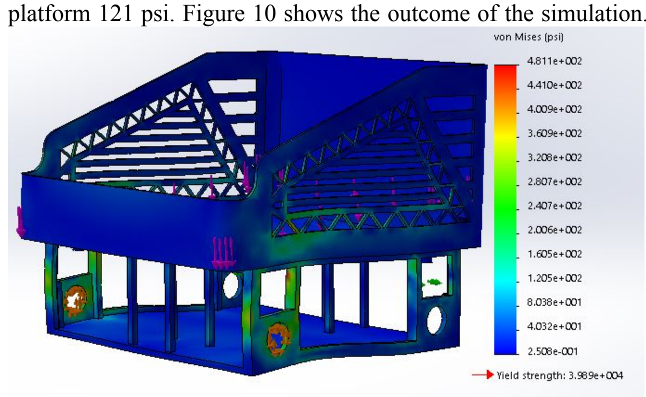

Figure 10. Maximum Stress on the Smaller Platform

Figure 11 shows the maximum displacement occurring on the smaller platform. It can be clearly seen that the maximum displacement occurs in the center of the platform. The maximum displacement occurring is 0.0002963 inch, which is insignificant.

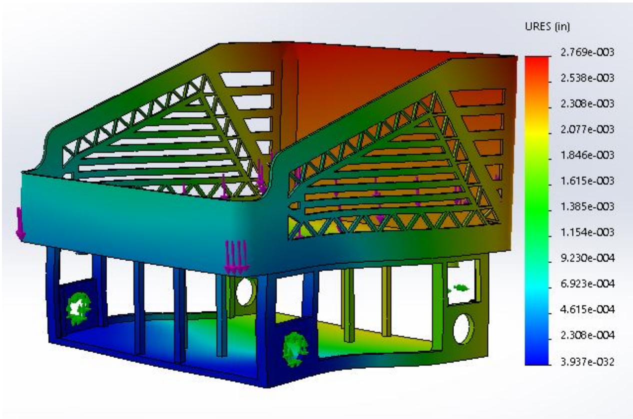

Figure 11. Smaller Platform Displacement Analysis

Figure 12 shows the resultant factor of safety from placing the load. The minimum factor of safety was 3.246.

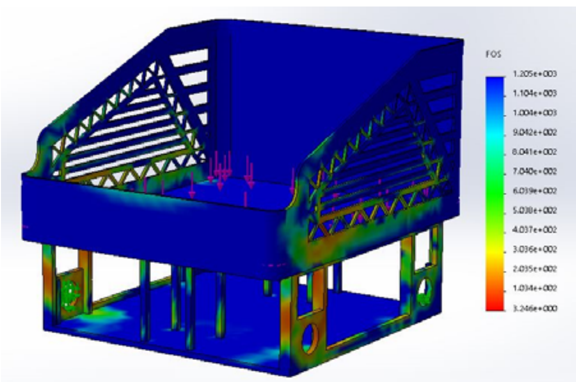

Figure 12. Smaller Platform Factor of Safety

4.2 Larger Platform Simulations

The larger platform was also simulated with 100lb loading. The maximum stress occurring on the platform according to the simulation is 120.7 psi as shown in the Figure 13.

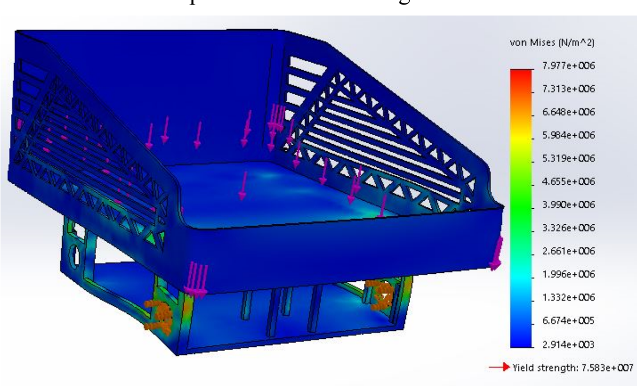

Figure 13. Larger Platform Stress Analysis

The maximum deflection occurring on the platform is 0.0002189 in. Similar to the smaller platform, the larger platform's largest deflection occurs towards the center as shown in Figure 14.

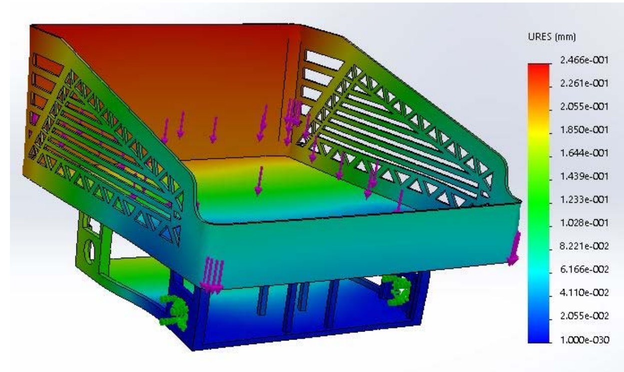

Figure 14. Larger Platform Deflection Analysis

The minimum factor of safety occurring on the platform is 3.156 as shown in the Figure 15.

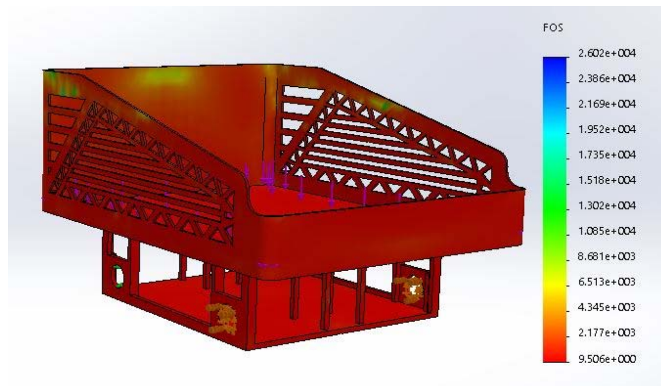

Figure 15. Larger Platform Factor of Safety

5. CONTROL

5.1 Theory and Language

We used C++ language for programming. The code uses a switch case statement for communicating with the user's phone. Within each case an if statement was used for determining whether or not the ultrasound detects an obstacle or not. The following code example shown in section 5.2 is a code developed using this theoretical principle to be used on the small scaled prototype constructed for this project. This code simply makes the platform stop and wait for user command whenever it detects an object.

5.2 Sample Code

The code starts opening the Bluetooth library communications and establishing a Bluetooth object as well as defining the variables that will be transmitted through Bluetooth.

```
#include <SoftwareSerial.h>
SoftwareSerial bt(0,1);
#define GO_LEFT     'a'
#define GO_RIGHT    'd'
#define GO_FORWARD  'w'
#define GO_BACK     's'
#define STOP        'x'
```

Variables for the motor driver are created, enA and enB will be the pins that will indicate the velocity of each of the motors, and from in1 to in4 will be the variables that indicate the direction of rotation of the DC motors, in the next code can be seen the assignation of the pins to these variables.

```
//motor one
int enA = 10;
int in1 = 9;
int in2 = 8;
// motor two
int enB = 11;
int in3 = 7;
int in4 = 6;
```

The Arduino UNO that We used for the project is limited to 13 output pins, because of that We use the same trigger pin for each of the ultrasound sensors, each of the sensor though will have an independent echo pin, which is the receiver for each of the cases.

```
#define trigPin     12 // define the trigger pins of your sensors
#define echoPin1 5  // sensor at the front left
#define echoPin2 4  // sensor at the front right
#define echoPin3 3  // sensor at the back
```

In the void setup code We stablish the output and input pins, and start the Bluetooth serial port.

```
void setup() {
  Serial.begin(9600); // begin serial communitication
  Serial.println("Motor test!");
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  bt.begin(9600);//start the bluetooth serial port - send and recieve at
9600 baud

  pinMode(trigPin, OUTPUT);// set the trig pin to output (Send sound
waves)
  pinMode(echoPin1, INPUT);// set the echo pin to input (recieve
sound waves)
  pinMode(echoPin2, INPUT);// set the echo pin to input (recieve
sound waves)
  pinMode(echoPin3, INPUT);// set the echo pin to input (recieve
sound waves)
}
```

Subroutines:

A series of subroutines were programed to stop, move the robot and check the distance on the ultrasound sensors. These subroutines control the rotation direction of the DC motors and the speed, which is an input variable for each of the subroutines to be able to control that velocity in the main code.

```
void allOff(int speed)  {  //stop the robot
  digitalWrite(in1, speed);
  digitalWrite(in2, speed);
  digitalWrite(in3, speed);
  digitalWrite(in4, speed);
  delay(5);
}

void goForward(int speed)  {
  Serial.println ("No obstacle detected. going forward");
  for (int i=0; i <= 2; i++){
    analogWrite(enA, speed);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    analogWrite(enB, speed);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    delay(5);      //this delay is the amount of time that is going to be
applying this operation
   }
}
  void goBackward(int speed)  {
  Serial.println ("No obstacle detected. going Backward");
  for (int i=0; i <= 2; i++){
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    analogWrite(enA, speed);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    analogWrite(enB, speed);
    delay(5);
   }
}
  void goRight(int speed)  {
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  analogWrite(enA, speed);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  analogWrite(enB, speed);
  delay(5);
}
  void goLeft(int speed) {
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  analogWrite(enA, speed);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  analogWrite(enB, speed);
  delay(5);
}
```

To check the ultrasound sensors we also created two subroutines, one to check the two sensors on the front and the other for the back sensor. Because the sensors are using the same trigger pin one sensor is activated first and then the other. The code returns a value of 1 if an object is detected with less than 25 cm of distance.

```
int checkfront(){
   int result;
   result = 0;
   long duration1, duration2, distance1, distance2; // start the scan
   //////////////////////////////////////
```

```
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2); // delays are required for a succesful sensor operation.
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10); //this delay is required as well!
    digitalWrite(trigPin, LOW);
    duration1 = pulseIn(echoPin1, HIGH);
    distance1 = (duration1/2) / 29.1;// convert the distance to centimeters.
    //////////////////////////////////////
    delay(10);
    //////////////////////////////////////
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2); // delays are required for a succesful sensor operation.
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10); //this delay is required as well!
    digitalWrite(trigPin, LOW);
    duration2 = pulseIn(echoPin2, HIGH);
    distance2 = (duration2/2) / 29.1;// convert the distance to centimeters.
    //////////////////////////////////////
      if (distance1 < 25 || distance2 < 25)/*if there's an obstacle 25 centimers, ahead, do the following: */ {
       result = 1;
      }
      else {
       result = 0;
       }
       return result;
 }
   int checkback(){
    char result;
    result = 0;
    long  duration3, distance3; // start the scan
    //////////////////////////////////////
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2); // delays are required for a succesful sensor operation.
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10); //this delay is required as well!
    digitalWrite(trigPin, LOW);
    duration3 = pulseIn(echoPin3, HIGH);
    distance3 = (duration3/2) / 29.1;// convert the distance to centimeters.
    //////////////////////////////////////
      if (distance3 < 25)/*if there's an obstacle 25 centimers, ahead, do the following: */ {
       result = 1;
      }
      else {
       result = 0;
       }
       return result;
  }
```

In the main code of the Arduino the first two lines check if there is Bluetooth connection with other device and stores in the variable 'data' the information sent by the android app from the smartphone, if there is communication then the code will run.

```
  If (bt.available()) {
  char data = (char)bt.read();
```

A switch selection command was inserted to indicate what action to take by the robot depending on the information sent by the phone, the switch will read the information stored in the variable data and will perform the operation indicated on the phone. For going backwards and forward the code will check first for objects closer than 25 cm, and if no object is close enough the action of forward or backward will take place. The distance of forward and backward motions are controlled by for loops, if the programmer wants to increase the distance in forward or backward motion, the iteration just has to increase the number of loops.

```
  switch(data) {
    case GO_BACK:
      crash = 0;
      iter = 0;
      while(crash == 0 && iter < 70){
       crash=checkback();
       goBackward(105);
       iter ++;
      }
      iter=0;
     allOff(0);
     break;
    case GO_FORWARD:
      crash = 0;
      iter = 0;
      while(crash == 0 && iter < 70){
       crash=checkfront();
       goForward(105);
       iter ++;
      }
      iter=0;
     allOff(0);
     break;
    case GO_LEFT:
     for (int i=0; i <= 50; i++){
     goLeft(150);
     delay(20);
     }
     allOff(0);
     break;
    case GO_RIGHT:
     for (int i=0; i <= 50; i++){
     goRight(150);
     delay(20);
     }
     allOff(0);
     break;
    case STOP:
     allOff(0);
     break;
   }
  }
  }
```

6. PROTOTYPE

A small scaled prototype was constructed to develop a running code for the system. The figure below shows the prototype that was created for this project.

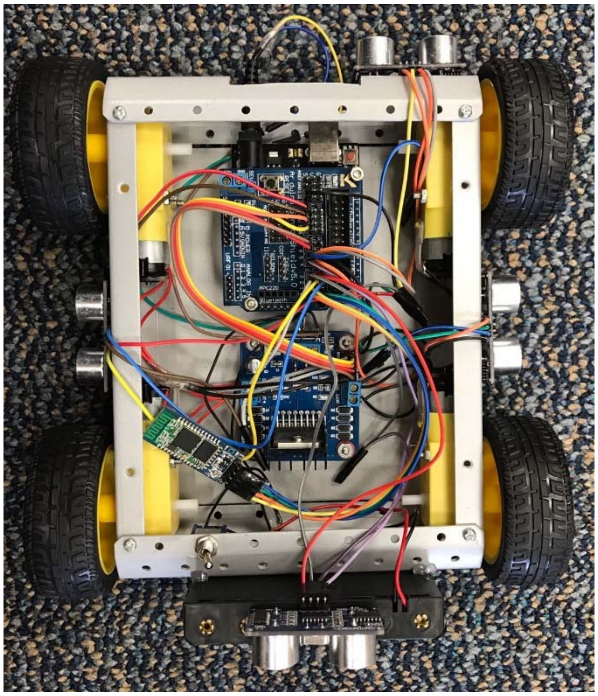

Figure 16. Prototype Model

The Arduino controller has a set of 14 digital pins which are used for control. Out of the 14 pins, pins 6 through 11are used for motor control. Pins 2 through 5 are used for retrieving the unique values of each respective ultrasonic sensor. Pin 12 is used as the trigger pin to allow the ultrasonic sensors to start detecting. Pins 0 and 1 are specific Rx and Tx respectively used to connect the bluetooth device to the Arduino in an interchange manner (i.e. Rx connects to Tx and Tx connects to the bluetooth Rx). The power for the prototype is obtained from a series of 6 1.5V batteries connected to a switch and connected to the power input of the Arduino respectively. The bluetooth controller, the ultrasonic sensors, and the motors are all connected to the power input.

7. EXPERIMENTS

An android phone was used to control the prototype. The prototype was made to move in all possible directions. In order to test if the override function will make the robot stop at a safe distance without crashing, we used a person and an object as a barrier.

8. RESULTS

The robot was very responsive and achieved a positive outcome. The luggage carrier prototype moved in the desired direction and stopped automatically when it detected something in its path thus avoiding a crash. Initial experimental results, although limited due to time constraints, were judged to be encouraging.

9. CONCLUSION

We successfully built the luggage carrier prototype which can be controlled by phone and is capable of moving in all directions and is also able to avoid crashes by detecting the obstacle in the path. For future design we are planning to include machine learning in the robot, and maybe transform it in a personal assistant equipped with voice recognition.

10. REFERENCES

[1] Martín, Hugo. "Idle luggage carts mean lost revenue for airports and cart renters." *Los Angeles Times*. Los Angeles Times, 01 July 2011. Web. 27 Feb. 2017.

[2] Tsatips. "Can I Bring It On a Plane? TSA Carry-on and Checked Bag Rules." Airport Security | TSA Travel Tips. N.p., 04 Jan. 2017. Web. 27 Feb. 2017.

[3] Hodgkins, Kelly. "Don't walk through the airport, ride on your motorized luggage with Modobag." *Digital Trends*. N.p., 21 July 2016. Web. 27 Feb. 2017.

[4] "My TSA." My TSA - Home Airport. N.p., n.d. Web. 27 Feb. 2017.

[5] Cowarobot. "COWAROBOT R1: The First and Only Robotic Suitcase." Indiegogo. N.p., 26 July 2016. Web. 27 Feb. 2017.

[6] "Travelmate Robotics." Company. N.p., n.d. Web. 27 Feb. 2017.

[7] "Elegoo UNO Project Super Starter Kit with Tutorial, 5V Relay, UNO R3, Power Supply Module, Servo Motor, 9V Battery with DC, Prototype Expansion Board, ect. for Arduino: Computers & Accessories." *Amazon.com: Elegoo UNO Project Super Starter Kit with Tutorial, 5V Relay, UNO R3, Power Supply Module, Servo Motor, 9V Battery with DC, Prototype Expansion Board, ect. for Arduino: Computers & Accessories*. N.p., n.d. Web. 27 Feb. 2017.

[8] "Buy Products Online from China Wholesalers at Aliexpress.com." *Buy Products Online from China Wholesalers at Aliexpress.com*. N.p., n.d. Web. 27 Feb. 2017.

[9] "Arduino UNO Rev3." Arduino Store USA. N.p., n.d. Web. 27 Feb. 2017.

[10] Jazar, Reza N., Vehicle Dynamics: Theory and application. Springer, 2017.

[11] "Design and Manufacturing Laboratory." Design and Manufacturing Laboratory. Web. 28 Feb. 2017.