An Autonomous Delivery Robot to Prevent the Spread of Coronavirus in Product Delivery System

Conference Paper · October 2020

DOI: 10.1109/UEMCON51285.2020.9298108

Murad Mehrab Abrar
The University of Arizona

Raian Islam
The University of Arizona

Abid Hasan Shanto
Ahsanullah University of Science & Tech

An Autonomous Delivery Robot to Prevent the Spread of Coronavirus in Product Delivery System

Murad Mehrab Abrar
Dept. of Electrical and Electronic Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
Email: mehrababrar@gmail.com

Raian Islam
Dept. of Electrical and Electronic Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
Email: raianislam16@gmail.com

Md. Abid Hasan Shanto
Dept. of Electrical and Electronic Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
Email: abidhasan.aust@gmail.com

***Abstract*— Due to the coronavirus situation around the world, safe and contactless home delivery services have become substantial concerns for the people while they are forced to stay at home. In this context, we have proposed a prototype robot that can be very helpful to reduce the risk of infectious disease transmission in the product delivery system during the extreme strain on healthcare and hygiene. The design and development of a cost effective autonomous mobile robot prototype have been presented that can deliver packages safely to a desired destination using Global Positioning System (GPS). The robot ensures a secure and human-contactless delivery by using a password protected container to carry the delivery package. The four wheel drive robot can successfully navigate to a preset location by receiving GPS coordinates from satellites and correcting its direction using a digital compass. After the robot arrives at its destination, it waits for the customer to unlock the container. The customer will have to use a password upon delivery to unlock the container and retrieve the ordered product. This password can be sent to the customer with the order confirmation message. After completing the delivery, the robot can autonomously return to its starting location. Heading angle accuracy test and trajectory completion accuracy test have been performed to ascertain the accuracy of the robot. Alongside an infection risk-free product delivery, our robot can be an effective technological solution of the last mile problem which will reduce the last mile delivery cost significantly.**

Keywords*— *Coronavirus, Autonomous, Robot, Delivery robot, GPS, Compass, Password, Last mile.

I. Introduction

The recent coronavirus outbreak has been labelled as a pandemic by the World Health Organization (WHO) and it is spreading rapidly around the globe [1]. To contain the spread of this highly infectious virus, governments around the world are taking necessary steps to impose home lockdown restrictions and to spread awareness against unprotected human contacts. Due to the enforced home lockdown measures and to avoid direct human contact, consumers are limiting physical shopping and turning to e-commerce for buying necessary essentials like food and grocery [2]; therefore, the strain on home delivery is increasing day by day. Robots and autonomous vehicles can help to ease the stress on the existing home delivery system while reducing the risk of virus transmission by mitigating direct human contacts [3]. The rapidly growing popularity of online market also raise the question of efficient product delivery to the customers. Studies have shown that the last mile delivery is the least efficient stage of the supply chain and comprises up to 28% of the total delivery cost [4]. A new approach to increase the last mile delivery efficiency introduces various autonomous delivery robots, among which the land-based autonomous ground vehicles have proven to be the most suitable technology to solve the last mile problem [5].

In this regard, we have developed a cost effective autonomous mobile robot prototype for the purpose of increasing the last mile delivery efficiency as well as ensuring a secure and contactless package delivery. An autonomous mobile robot is a self-driving vehicle that does not require any operation from operator to navigate the robot [6]. The movements and trajectory are predefined before the operation and the robot navigates accordingly. Among various navigation techniques, we have used the Global Positioning System (GPS) data for autonomous navigation of the robot and the destination is predefined as latitude and longitude points in the program of the robot. The main advantage of using GPS for navigation is that the data received from the GPS are independent of the previous readings; therefore it is easy to minimize errors [7]. A digital compass measures the heading angle of the robot and helps the robot to find the direction of the trajectory. The robot is equipped with a password protected container which protects the package against theft, damages and unprotected human contacts. This password can be sent to the customer by a text messages from the service company. Once the robot arrives at its delivery location, the only person who has the password will be able to unlock its delivery. It can carry up to 1KG of payload to desired destination. The proposed robot prototype has been designed for the accomplishment of four tasks-

1. Carry the package securely in a password protected container, 2. Navigate through the GPS waypoint and reach the destination successfully, 3. Open the container upon matching the password to deliver the package, 4. Return back to the home location.

Fig. 1 shows the model of our robot. The proposed robot consists of an Arduino Mega 2560 microcontroller board, a power supply unit, an Ublox Neo-M8N GPS with Compass module, an L298N motor driver board, a 4×3 keypad module,

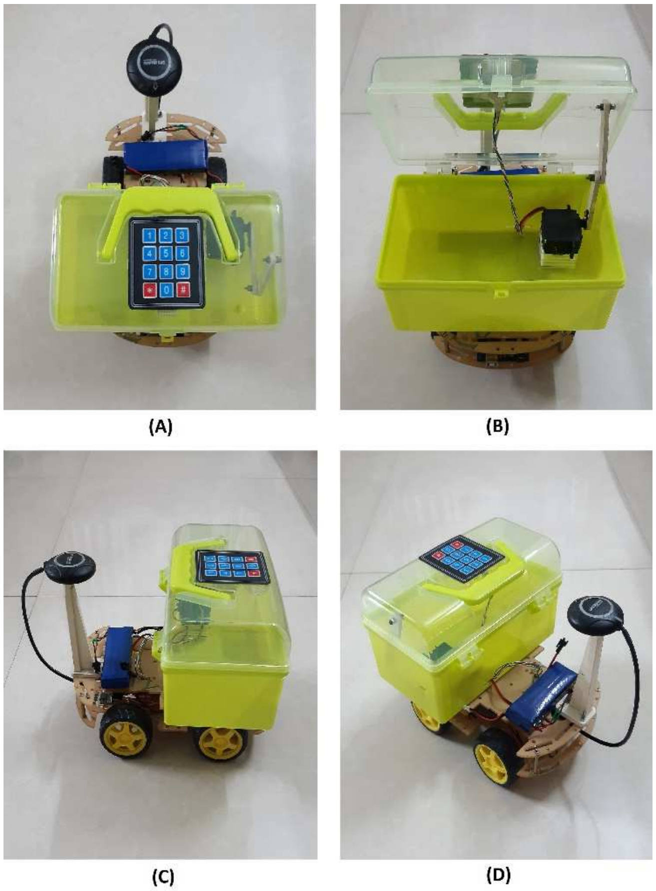

Fig. 1. Autonomous package delivery robot prototype. (A) Top view, (B) Container lid open function, (C) Profile view, (D) Rear view.

an HC-05 Bluetooth module, a Tower Pro MG996R servo motor and a four wheel drive robotic chassis. By choosing the cheap and commercially available components, we have minimized the cost of the robot.

II. System Hardware Design

A. Electronic Components

1. *Arduino Mega 2560 microcontroller:* The main processing unit of the robot is the Arduino Mega 2560 microcontroller board which processes all the input data and generates the PWM (Pulse-Width Modulation) signals. It is a microcontroller board based on the ATmega2560 microcontroller. It has 54 digital input/output pins (14 pins can provide PWM output), 16 analog inputs, 4 UARTs (Universal Asynchronous Receiver/Transmitter), a 16 MHz crystal oscillator, a USB port, a power jack and a reset button [8]. The operating voltage of the board is 5V.

2. *Power Supply:* We have used a 7.4V 2200 mAh (milliamp-hour) lithium polymer battery to power the whole robot. The battery provides sufficient power to run the robot for 30-45 minutes.

3. *Ublox Neo-M8N GPS with Compass module:* The Ublox Neo-M8N GPS module incorporates the HMC5883L digital compass. The HMC5883L digital compass is highly sensitive to magnetic interference and any kind of interference can cause inaccurate results. The Ublox latest 8 series combined GPS-compass module provides a convenient method of mounting the compass away from sources of interference that may be present in the confines of a system. The Neo-M8N GPS receiver provides a high accuracy for position tracking under a good receiving condition [9]. Thus we get accurate latitude, longitude, altitude, time, and heading angle data from a single device.

4. *Motor driver:* An L298N motor driver board runs the 4 DC geared motors used in the wheels of our robot. The motor driver is directly powered by the 7.4V 2200mAh lithium polymer battery. It has a regulated 5V output pin which we have used to power the microcontroller. The motor driver provides 2 Amp peak output current per channel [10] which is sufficient to carry the load of the 4 geared motors.

5. *Keypad:* We have used a 4×3 membrane keypad module to input the password. The keypad has 12 keys, these are: number 0-9, '*' and '#'.

6. *Servo motor:* A Tower Pro MG996R servo motor is used to open and close the lid of the container. It provides a stall torque of 9.4kg/cm at an operating voltage of 4.8V.

7. *Bluetooth module:* An HC-05 Bluetooth module is used to monitor the current latitude, current longitude, target latitude, target longitude, current heading angle and target heading angle data of the robot. The Bluetooth module can transfer these data to android device via Bluetooth when the device is within the Bluetooth signal range.

8. *Buzzer:* We have used a buzzer (piezo speaker) to indicate a wrong password by generating a beeping tone.

B. Constructional Structure

1. *Chassis:* A "4WD (4 wheel drive) Smart Robot Car Kit" has been used in the robot which consists of two 3 mm thick acrylic boards for top and bottom plates, four 12V 180 RPM DC geared motors, four 65 mm wheels, M3 screws and nuts and copper columns [11].

2. *Stand for GPS module:* We have made a 12 cm long plastic stand for mounting the Ublox GPS and compass module with the robot. The stand has been made using 8 mm thick plastic board and it is mounted with the robot by M3 nuts and screws. This stand keeps the GPS and compass module away from any magnetic interference from the robot.

3. *Plastic container:* A plastic container (dimension = 22 cm × 12 cm × 13 cm) has been used for containing the delivery products securely. The container has been attached to the chassis with hot-melt adhesive. The servo motor is placed inside the container and a crank mechanism is attached with the servo arm and the lid of the container to control the lid.

III. Software Development

We have used Arduino Integrated Development Environment (Arduino IDE) to program the robot. The program is written in C++ programming language.

A. Extracting GPS data

At first we begin by extracting the GPS data from the Neo-M8N GPS module. We are surrounded by countless satellites which help a GPS module to find its location. A GPS module triangulates its position by measuring its distances from different satellites. It knows the positions of each satellite it is locked with and draws an imaginary circle around each one with radiuses equal to the distances of the satellites from the module. The common intersecting point of all the locked satellite-circles is the location of the GPS module. Therefore, the more satellites the GPS module is locked with, the more accurate the results are. We connected the GPS module Transmitter and Receiver pins with Serial Port 3 of the Arduino Mega microcontroller and set the baud rate to 9600. When a GPS module is powered, it immediately starts spitting out the "National Marine Electronics Association" sentences or NMEA sentences to the serial port [12]. NMEA sentence is a standard data format that is supported by all GPS manufacturers. NMEA gives users the ability to mix and match hardware and software, therefore, helps users to receive useful information directly from the satellites. We read the NMEA data strings using the Arduino Microcontroller and parse them into useful data. Under the NMEA-0183 standard, all characters used are printable ASCII test. Each sentence starts with a "$" symbol, a two letter "talker ID", a three letter "sentence ID", some data fields separated by commas, and a carriage return/line feed [13]. We get several types of NMEA sentences such as $GPGLL, $GPRMC, $GPVTG, $GPGGA etc. The $GPGGA NMEA string gives us the two dimensional latitude and longitude location, so we only need to extract the $GPGGA string or "GGA" standard. A complete $GPGGA sentence looks like this-

*$GPGGA,123429,4805.036,N,01129.000,E,1,11,0.9,530.4,M, 46.9,M,,*47*

Table I. explains this NMEA sentence. We need to extract the latitude and longitude data from the GGA standard NMEA sentences, which we did using the TinyGPS++.h library [14]. The TinyGPS++.h library converts the NMEA strings to decimal point latitude and longitude values.

TABLE I. GGA STANDARD NMEA SENTENCE

NMEA sentence sections	***NMEA sentence meaning***
GGA	Global Positioning System Fix Data
123429	Fix taken at 12:34:29 Universal Time Coordinated (UTC)
4805.036,N	Latitude: 48° 05.036' North
01129.000,E	Longitude: 11° 29.000' East
1	Fix quality
11	Number of satellites the GPS is locked with
0.9	Horizontal dilution of position
530.4,M	Altitude: 530.4 meters above mean sea level
46.9,M	Height of geoid (mean sea level) above WGS84 ellipsoid
(empty field)	time in seconds since last DGPS update
(empty field)	DGPS station ID number
*47	the checksum data, always begins with *

From the latitude and longitude data, we calculate the target angle at which the robot needs to move.

B. Reading Digital Compass

The digital compass is used to read the current heading angle of the robot. We know the target angle from the GPS module, but the robot needs to know which direction it is pointed at, so the digital compass plays an important role to direct the robot towards the target path. The digital compass communicates with the Arduino Microcontroller using I2C serial communication. We have used HMC5883L.h library to read the compass [15]. These digital compass modules are not accurate and they need to be calibrated before using in any project. The HMC5883L.h library has a build in calibration example sketch which we used to read the X and Y axes offset values. The main program takes into account these offset values and calculates the current heading angle.

C. Keypad Interfacing

The delivery product is kept safe in a container which remains locked with a four digit password. When the robot reaches to its destination, it stops and waits for the customer to enter the four digit password that has been sent to the customer's cell phone along with the order confirmation. This four digit password would be preset by the vendor in the program. This preset password is saved in a string called "password". The customer's input password is stored in another string called "input_password". When the customer inputs the password and presses the "#" key, the "input_password" string is matched with the "password" string. If the input password matches, the servo motor turns from 0 degree to 90 degree and opens the lid of the container. If the input password does not match, the buzzer beeps three times and the customer needs to retype the input password. The "*" key clears the input password string if the customer mistakenly types a wrong key. We have used the Keypad.h library [16] to interface the 4×3 keypad and the Servo.h library to control the servo motor.

IV. WORKING PRINCIPLE

The program of the robot starts with the initialization of the GPS and compass sensors and the Bluetooth module. The function of this sensor initialization is to regulate the data transmission speed to Arduino Microcontroller. The basic program flowchart of the robot is illustrated in Fig. 2. The working principle of the robot follows 11 steps. These steps are described below-

1. The GPS module searches for satellites and reads the current latitude and longitude. To calculate the latitude and longitude accurately, the GPS module must need to be locked with at least 4 satellites.

2. Then the robot calculates the target angle from the latitude and longitude data. The angles must be measured in radians. The algorithm to find the target angle is-

difference_Longitude = radians(target_Longitude – current_Longitude);
Latitude1 = radians(current_Latitude);
Latitude2 = radians(target_Latitude);

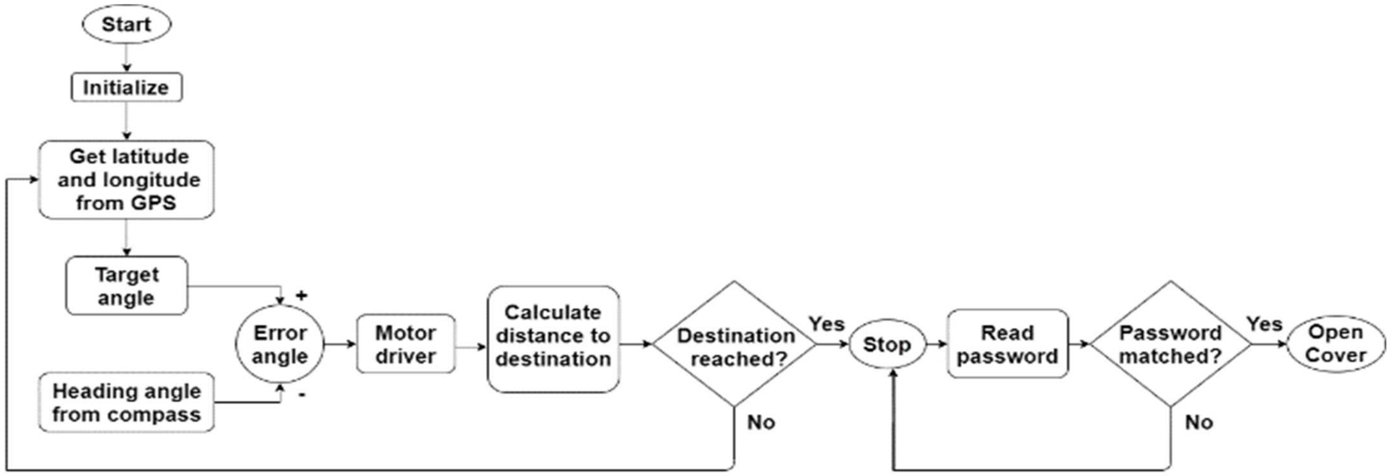

Fig. 2. The basic program flowchart of the proposed autonomous robot for package delivery

x = sin(difference_Longitude) × cos(Latitude2);
y = cos(Latitude1) × sin(Latitude2) – sin(Latitude1) × cos(Latitude2) × cos(difference_Longitude);
target_angle = atan2(x, y) × (180/3.14159);
if (target angle < 0) target angle += 360;

3. The current heading angle is received from the compass module and the error angle is measured from the difference between the current heading angle and the target angle.

4. By knowing the error angle, we know whether the robot needs to turn left or right. The 4 motors get pulses according to the error angle from the microcontroller. The L298N motor driver runs the motors to align the robot with the target angle.

5. The distance to the target destination is calculated by using this algorithm –
delta = radians(current_Longitude – target_Longitude);
sd_Longitude = sin(delta);
cd_Longitude = cos(delta);
Lat1 = radians(current_Latitude);
Lat2 = radians(target_Latitude);
sLat1 = sin(Lat1);
cLat1 = cos(Lat1);
sLat2 = sin(Lat2);
cLat2 = cos(Lat2);
delta = (cLat1 × sLat2) – (sLat1 × cLat2 × cd_Longitude);
delta = sq(delta);
delta += sq(cLat2 × sd_Longitude);
delta = sqrt(delta);
denom = (sLat1 × sLat2) + (cLat1 × cLat2 × cd_longitude);
delta = atan2(delta, denom);
distance_To_Destination = delta × 6372795;

6. The robot checks if it has reached the threshold value from its destination (in our program, we set the threshold value to 3 meters). If not, the robot calculates the target angle, current heading angle and distance to destination again and the loop continues till the robot reaches to its destination.

7. When the robot reaches its destination, it stops and waits for the customer to type the input password to receive the delivery.

8. Key "*" is used to start/re-start the input password. Key "#" is used to terminate the input password. If neither of these two keys are typed, the key is appended to the customer's input password string.

9. If key "#" is pressed, the robot compares the customer's input password string with the password to determine if the input password is correct, then clears the user's input password string. If the input password is incorrect, the buzzed beeps three times.

10. If the password matches, the servo motor gets PWM signal and the lid of the container opens.

11. After a 10 second delay, the lid of the container closes and the robot returns to its home location from where it started.

V. Tests and Results

GPS modules work the best at outside in fields or in open areas. Trees or buildings around the testing location can block the satellite signals and can cause the accuracy of the GPS module to decrease. The ideal testing location for our prototype robot should be an open field where the ground is obstacle free. But during the tests, these factors came into account and added some errors in the results.

In our work we have conducted two types of experiments to determine the performance of our robot: A heading angle accuracy experiment and a trajectory completion accuracy experiment. In the heading angle accuracy experiment the actual heading angle and the current heading angle are measured and the accuracy is found from the deviation between the two data. D. J. Paul et al. showed an accuracy of the heading angle in a GPS guided object transporter robot where the performance of the robot is determined only by the direction of movement [17]. However, the data from GPS modules can fluctuate sometimes, causing some error in the robot operation. While conducting the test runs we have found that, when the

TABLE II. DATA FROM HEADING ANGLE ACCURACY EXPERIMENT

Actual heading angle	*Current heading angle*	*Error*	*Accuracy*
109.66°	107.75°	1.74%	98.26%
190.23°	186.54°	1.94%	98.06%
39.90°	41.88°	4.96%	95.04%
325.73°	330.55°	1.48%	98.52%
265.11°	259.95°	1.95%	98.05%
70.38°	67.47°	4.13%	95.87%
211.48°	216.15°	2.21%	97.79%
21.92°	23.09°	5.34%	94.66%
306.57°	299.71°	2.24%	97.76%
145.41°	148.50°	2.13%	97.87%

robot reaches approximately around the destination point and the GPS data fluctuation occurs, the robot moves out from the trajectory and misses the destination, causing a trajectory completion failure. This matter has been taken into account in our test runs and we conducted this trajectory completion accuracy experiment to evaluate the trajectory completion performance of our robot. In this experiment, the number of successful trajectory completion attempts determine the performance of the robot. We calculated the trajectory completion accuracy rate using this formula-

$$Trajectory\ accuracy = \frac{number\ of\ successful\ attempts}{the\ total\ number\ of\ experiments} \times 100\%$$

The two accuracy experiments are discussed below.

A. Heading Angle Accuracy Experiment

In the heading angle accuracy experiment, we measured the actual heading angles of the robot and the current heading angles after the motor drive. The robot sends these data via Bluetooth. We measured 10 samples and the heading angle accuracy ranged from 94.66% to 98.52%, resulting in an average accuracy of 97.19%. Fig. 3 illustrates the heading angle accuracy of our robot. The data from the heading angle accuracy experiment are given in Table. II.

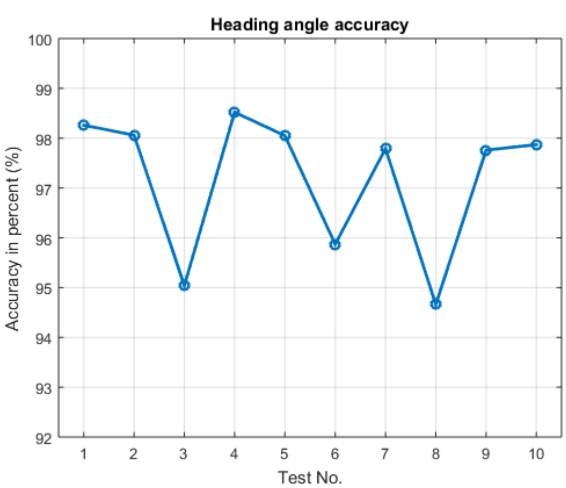

Fig. 3. Heading angle accuracy rate.

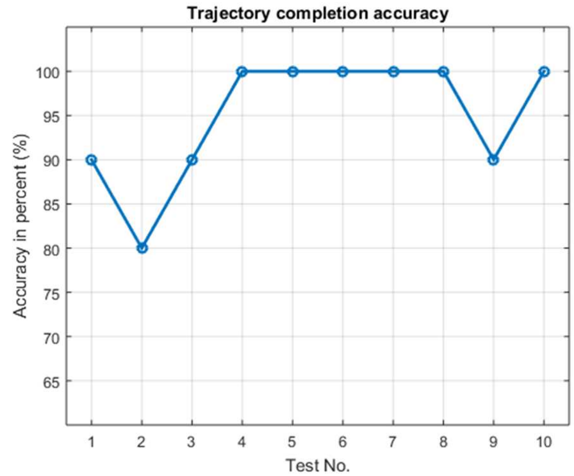

Fig. 4. Trajectory completion accuracy rate.

B. Trajectory Completion Accuracy Experiment

In the robot trajectory completion experiment, the robot was tested in different locations for 10 times. In each test set, we operated the robot 10 times and counted how many times the robot reached the threshold value from its destination successfully.

In the trajectory completion accuracy experiment, we also tested the password protected container each time we operated the robot. We changed the password before operating the robot and tested it after the robot completed its trajectory. The password protected container worked successfully in all the tests and we achieved a 100% accuracy rate. The trajectory completion accuracy rate varied from 80% to 100%, resulting in an average trajectory completion accuracy of 95%. Fig. 4 illustrates the trajectory completion accuracy of the robot. The data from the trajectory completion experiment are given in Table. III.

TABLE III. DATA FROM TRAJECTORY COMPLETION ACCURACY EXPERIMENT

Test No.	*No. of experiments*	*No. of times trajectory completed*	*Trajectory completion accuracy rate*	*Number of times password worked*	*Password accuracy rate*
1	10	9	90%	10	100%
2	10	8	80%	10	100%
3	10	9	90%	10	100%
4	10	10	100%	10	100%
5	10	10	100%	10	100%
6	10	10	100%	10	100%
7	10	10	100%	10	100%
8	10	10	100%	10	100%
9	10	9	90%	10	100%
10	10	10	100%	10	100%

The Neo M8N GPS module readings do not vary considerably within 3 meters around a location. So, we considered a 3 meters radius circle around our destination as the threshold value from our destination. When the robot reaches within this circle, it stops as it approximately reaches the destination. While returning back to its home location from where it started, the robot receives new GPS data, so it does not follow the same route always. The four wheel drive chassis of our robot gives it enough power to carry a 500 gm to 1KG load and helps the robot to track its path accurately.

One of the advantages of our robot is that, being a GPS guided delivery robot, it has no range limits as long as the GPS location is received. Starship food delivery robots are available in the market but they are only used for local delivery services within a 6 kilometers radius [18]. The final version of our prototype robot can be used for long distance product deliveries and can be free from any range limits.

VI. Discussion and Conclusion

We have developed an autonomous package delivery robot prototype which can deliver up to 1KG of packages or products to a certain GPS location safely in a password protected container without any human contact. The results show that our robot ensures the package protection and transportation, maintaining a 100% accurate password protected delivery with 97.19% heading angle accuracy and 95% successful attempts or trajectory completion accuracy. The perfect accuracy of the password protection indicates that the products can be delivered without being touched, damaged or stolen. This robot can be very useful to deliver food, grocery and daily useful products to our desired destination during the recent coronavirus pandemic, where unprotected human contact is considered lethal and outdoor movements are restricted. Also, during this extreme pressure on the healthcare system, delivering medical equipment to hospitals can be very risky. Our robot can play an important role to accomplish these tasks as well. In the same time, our robot can be an effective smart logistics system to solve important logistics problems in the supply chain, particularly minimizing the last mile delivery cost and time. The robot provides a simple user interface to the customers, which makes it smart and user friendly. Also, the lightweight construction of our robot makes it crash-safe, thus it can effectively contribute to the reduction of urban traffic, congestion and accidents.

VII. Future Work

In future this robot can be developed as a google map guided robot to enhance the accuracy. A Lidar sensor based mapping can be implemented for avoiding obstacles and recognizing the surrounding environment. To accommodate more packages and heavy load, the size of the robot can be increased and powerful motors can be attached. These updates will highly improve the performance of our robot and it will become a more reliable source to bring our daily needed products to our door.

References

[1] Coronavirus Confirmed As Pandemic by World Health Organization. [Online]. Available: https://www.bbc.com/news/world-51839944

[2] US Ecommerce Sales Rise 25% Since Beginning of March. [Online]. Available: https://www.digitalcommerce360.com/2020/04/01/us-ecommerce-sales-rise-25-since-beginning-of-march/

[3] V. Chamola, V. Hassija, V. Gupta and M. Guizani, "A Comprehensive Review of the COVID-19 Pandemic and the Role of IoT, Drones, AI, Blockchain, and 5G in Managing its Impact," in IEEE Access, vol. 8, pp. 90225-90265, 2020, doi: 10.1109/ACCESS.2020.2992341.

[4] L. Ranieri, S. Digiesi, B. Silvestri and M. Roccotelli, "A Review of Last Mile Logistics Innovations in an Externalities Cost Reduction Vision," Sustainability, MDPI, Open Access Journal, vol. 10(3), pp. 1-18, 2018.

[5] G. Prause, I. Boevsky, "Delivery Robots for Smart Rural Development," *Bulgarian Journal of Agricultural Economics and Management*, vol. 63, No. 4, pp. 57-65, 2018.

[6] M. H. A. Hamid, A. H. Adom, N. A. Rahim and M. H. F. Rahiman, "Navigation of mobile robot using Global Positioning System (GPS) and obstacle avoidance system with commanded loop daisy chaining application method," *2009 5th International Colloquium on Signal Processing & Its Applications*, Kuala Lumpur, 2009, pp. 176-181, doi: 10.1109/CSPA.2009.5069211.

[7] W. Rahiman and Z. Zainal, "An overview of development GPS navigation for autonomous car," *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, Melbourne, VIC, 2013, pp. 1112-1118, doi: 10.1109/ICIEA.2013.6566533.

[8] Arduino Mega Datasheet. [Online]. Available: https://docs.rs-online.com/9ab3/0900766b80e8ba22.pdf

[9] A. L. Boer, M. C. Luculescu, L. Cristea, S. C. Zamfira and I. BARBU, "Comparative Study Between Global Positioning Systems Used on Remotely Piloted Aircraft Systems," *Scientific Research and Education in the Air Force*, vol. 18, pp. 127-132, 2016, doi: 10.19062/2247-3173.2016.18.1.16.

[10] L298N Motor Driver Module. [Online]. Available: https://components101.com/modules/l293n-motor-driver-module

[11] 4WD Smart Robot Car Two Layer Chassis Kit. [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/Seeed%20Technology/110090263_Web.pdf

[12] P. McWhorter, "Build an Arduino GPS Tracker," 2014. [Online]. Available: https://toptechboy.com/lesson-22-build-an-arduino-gps-tracker/

[13] J. Parthasarathy, "Positioning and Navigation System Using GPS," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*, vol. 36, pp. 208-212, 2006.

[14] Tiny GPS Plus Library for Arduino. [Online]. Available: http://arduiniana.org/libraries/tinygpsplus/

[15] HMC5883L Triple Axis Digital Compass Arduino Library. [Online]. Available: https://github.com/jarzebski/Arduino-HMC5883L

[16] Arduino Keypad Library. [Online]. Available: https://www.arduinolibraries.info/libraries/keypad

[17] M. Abdullah-Al-Kaiser, D. J. Paul, A. I. Khan, C. Shahnaz and S. A. Fattah, "A cost effective GPS guided autonomous object transporter robot for disaster management and industrial automation," *TENCON 2017 - 2017 IEEE Region 10 Conference*, Penang, 2017, pp. 2637-2641, doi: 10.1109/TENCON.2017.8228307.

[18] Starship Self-driving Delivery Robot. [Online]. Available: https://www.starship.xyz/business