Mobile Collaborative Robot with Obstacle Avoidance

Niculae F. Mihai, Angad S. Malhotra, Kartick R. Sivasubramanian, and Iulia N. Mihai

***Abstract*—This product has been designed to scan and inspect the flooring of a predefined space. Essentially, a camera at the tip of an arm mounted on a mobile base scans the environment as the robot moves. The robot's components can be programmed to halt all motion upon noticing any discrepancies in its scanning, and to then take photographs of said discrepancies before continuing along its trajectory. The applications of such a device can range anywhere from a simple workplace janitorial overview to industrial aircraft jet fuel tanks' safety inspections, wherein the robot can examine the tank for leaks, cracks, or corrosion.**

***Index Terms*—Mobile collaborative robot.**

I. INTRODUCTION

In today's world of robotics, fully autonomous robots have not yet been created, as even the advanced robots have not yet been upgraded to complete autonomy. As a potential solution, this mobile robot system is presented. This system stands out from other high-end robots due to its revolutionary features. Unlike other robots, which make use of 2-3 sensors, this system employs 8 ultrasonic sensors and a LiDAR module, all of which ensure 360° coverage and high-precision obstacle avoidance (Fig. 1). Furthermore, a camera is equipped to the system, serving to scan the environment, send a live video feed to a web server, and capture individual frames (images) when prompted by a program catered to the user's requirements. All the inputs are sent to a web server from which an intricately designed AI algorithm processes data and accurately commands direction. Despite receiving data from multiple sources, the AI processes instructions instantly. With 100% autonomy, this system opens up to several practical applications and it can also collaborate with other robots in order to branch its functionality out through a network of devices oriented towards a common goal .

For example, an aircraft's jet fuel tank is normally examined by human beings, and they are required to work within tight spaces and inspect the entire span of the tank, which can amount to strenuous work (Fig. 2).

This robot presents itself as a fitting solution to fuel tank inspection, as it can accurately detect any fault in the rivets of the tank or check for any traces of corrosion. Additionally, it can be combined with another robot so that it can be transported to different sections of the tank.

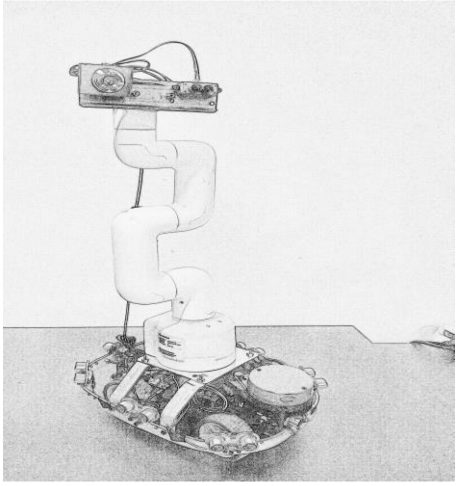

Fig. 1. Final product with erect arm

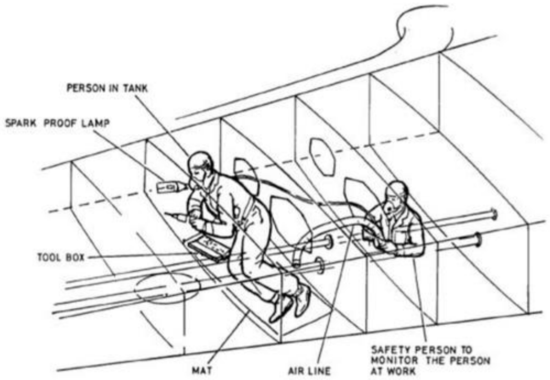

Fig. 2. Aircraft fuel tank inspection performed by crewmembers[1]

TABLE I: BILL OF MATERIALS

No.	Part Name	Quantity
1	Sun Founder Robot HATS	1
2	Sun Founder PCA9865	1
3	Sun Founder Motor Driver Module	1
4	SF-SR02	8
5	Sun Founder SF006C Servo	1
6	DC Gear Motor	2
7	Raspberry Pi 4	1
8	myCobot-Pi	1
9	SF-SR02 aluminum support	1
10	myCobot-Pi aluminum support	1
11	USB Camera	1
12	TF Mini LiDAR (ToF) Laser Range Sensor	1

II. MOBILITY

The product is built using a robotic car, controlled by means of a Raspberry Pi 4 running Python scripts, which

Manuscript received May 9, 2022; revised July 29, 2022.

N. F. Mihai is with the Technoaccord Inc, Laval, QC H7B1E6, Canada (e-mail: mniculae@ technoaccord.com). ,

A. S. Malhotra, K. R. Sivasubramanian, and I. N. Mihai are with Concordia University, Montreal, QC, H3G 1M8 Canada (e-mail: malhotra.angad@gmail.com, karthicksiva0320@gmail.com, iulia@iumirobo.com).

DOI: 10.7763/IJMO.2022.V12.811

execute directional instructions based on data received from the eight ultrasonic sensors surrounding the car (Fig. 3 and Fig. 4), as well as from a LiDAR module (Fig. 19) for full view.

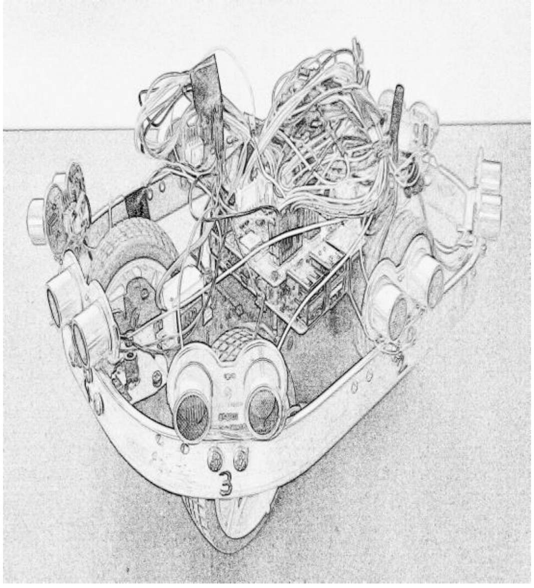

Fig. 3. Mobile base equipped with ultrasonic sensors, side view

The robotic car is composed of three modules, along with a servo and two DC gear motors. The modules are all Sun Founder modules, namely, Robot HATS, PCA9865, and the Motor Driver Module. The servo used is Sun Founder's SF006C, and the two DC motors: F130SA-11200-38V.

The servo is used to control the robot's direction by means of the front wheel axis, whereas the DC motors control the robot's linear direction via the rear wheels.

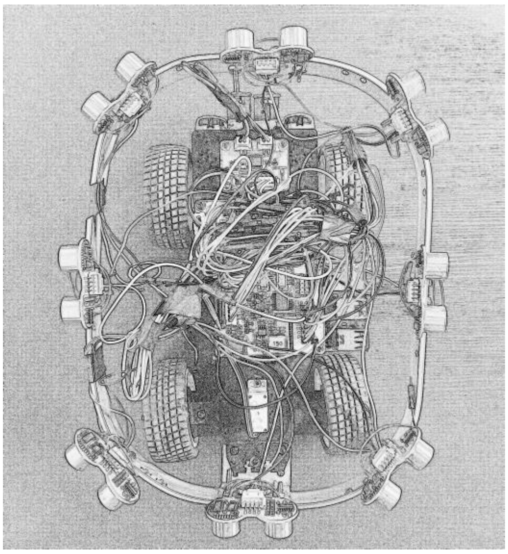

Fig. 4. Mobile base equipped with ultrasonic sensors, top view

The Robot HATS is directly connected to the Raspberry Pi via general purpose input-output (GPIO) pins. Receiving the data processed through Python scripts, it relays the data to the pulse-width modulation driver (PCA9865), which in turn transmits data directly to the servo. Additionally, the PWM sends data to the Motor Driver Module which, itself, controls the DC motors' speed and direction.

The ultrasonic sensors coupled to the Raspberry Pi send data to an artificially intelligent algorithm via Python scripts, which in turn call the IP address of a Flask-based web server linked to the AI. Once the algorithm processes the data, it returns it to the Raspberry Pi in the form of directional instructions for the motors.

III. ULTRASONIC SENSORS

The ultrasonic sensors used are of the SF-SR02 model by Sun Founder (Fig. 5), and eight of them are employed. The purpose of the ultrasonic sensors is to calculate the distances of nearby objects in order to avoid them accordingly.

The sensors are connected to the Raspberry Pi via the computer's GPIO ports. These sensors are unique, compared to conventional ultrasound sensors, as there are normally two distinct physical connections to be made: one to send ultrasound waves (trigger), and the other, to receive them after they bounce back from an object (echo). The SF-SR02 sensors, however, implement this function under a single connection, hence simplifying the wiring.

TABLE II: SPECIFICATIONS OF SF-SR02 SENSOR[2]

Working Voltage	DC5V
Working Current	16mA
Working Frequency	40Hz
Max Range	700 cm, stable 500 cm
Min Range	2 cm
Trigger Input Signal	10μS TTL pulse
Echo Input Signal	Input TTL lever signal
Dimension	46x20.5x15 mm

Ultrasonic sensors are widely used in automation industries as well as in medical practices. They are commonly used as either proximity sensors or level sensors. Proximity sensors can often be found used in the automobile industry for self-parking and anti-collision systems, as well as in robotics for obstacle detection/avoidance systems. Level sensors detect, monitor, and regulate liquid levels.

In general, an ultrasonic sensor contains two main components: a transmitter and a receiver. A transmitter emits ultrasound using piezo crystals and a receiver encounters the sound after it has travelled to and from the target object.

Fig. 5. SF-SR02 Ultrasonic sensor[2]

An ultrasonic sensor works by transmitting an ultrasound pulse at 40 kHz, which is higher than the frequency of sound audible to human beings (20Hz to 20 kHz). This pulse travels through air and it will bounce back to the sensor if there is an obstacle or object within its operational range. The time taken for the pulse to be emitted and received is recorded, and the distance of the object from the sensor is then found through the recorded time by referring to the speed of sound in air. This type of measurement is commonly known as a *Time of Flight* measurement (Fig.6).[3]

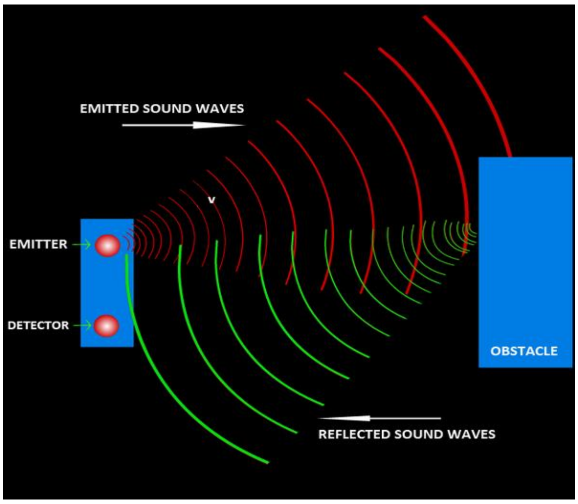

Fig. 6. Time of flight measurement[4]

IV. PYTHON SCRIPTS

This system makes use of 3 python scripts that run on the Raspberry Pi. In order to obtain the distances read by all 8 sensors (Fig.7), 2 python programs have been created. The first program reads the data of sensors 1-4, and the second program reads the data of sensors 5-8, and both programs run recursively with virtually no delay.

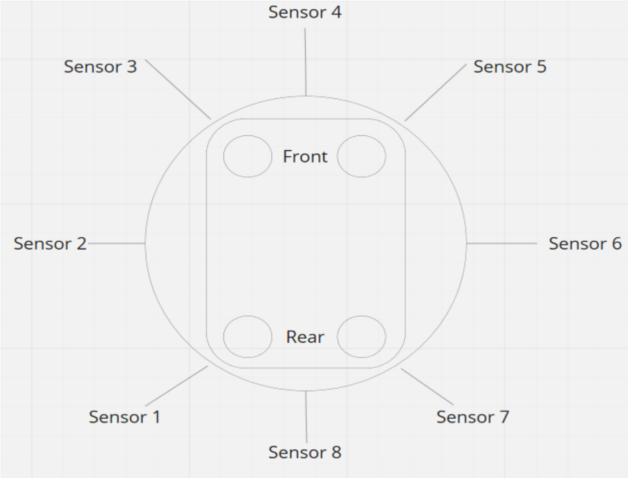

Fig. 7. Eight-sensor configuration for the AI algorithm

Two programs have been used instead of one program for all 8 sensors so that the data is read from the sensors faster. This is because, with one program, each sensor has to wait for 7 other sensors before it can read values again, whereas with two programs, each sensor only waits for 3 other sensors. These programs are almost identical to each other, and they each serve to read the value of their respective sensors before sending the data to their own text files, in the form of string values. Each text file contains data read by 4 sensors (first text file is for sensors 1-4, and the second, 5-8), and the text files are overwritten at each iteration of the program.

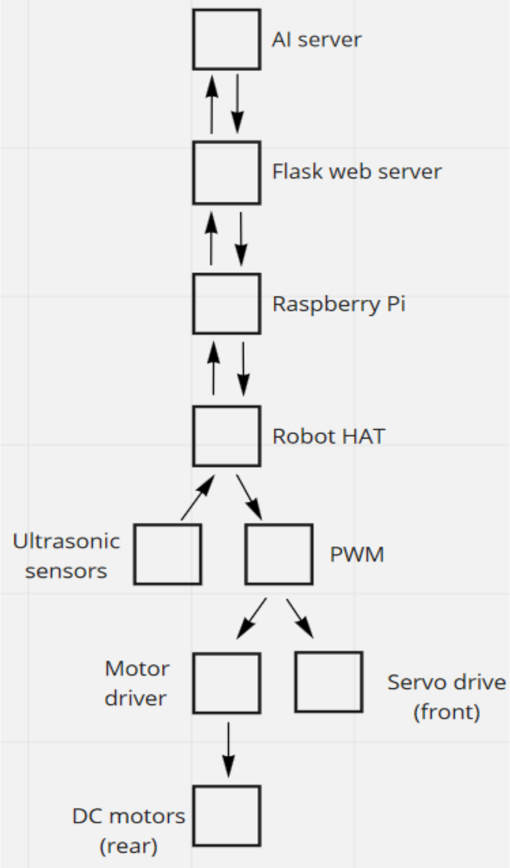

Fig. 8. Mobile Robot and AI server communication flow-chart

The programs make use of the “*Ultrasonic Avoidance*” module, which is a default module given by SunFounder that calculates the distance of objects read by the sensor. This is accomplished by calculating the time taken to receive a pulse signal once emitted.

The distance is first read by the sensor and then assigned to a variable. Next, in order to avoid logical errors, distances read by sensors are considered only if they are positive. Once the positive values are taken, they are then checked if they fall within the range of 0-80 centimeters. Any value that is above 80 centimeters is cast back to 80 centimeters, which is the upper limit set in the system. If the value falls within the set range, then it is converted to a string value with a space (“ “) and a comma (“,”) added to it. If it has been cast to 80, then the variable takes 80 as its value and is then converted to a string value with the space and comma suffixed.

Once all four sensors read a value and have a string that represents the value read, they are then concatenated and sent to the text file.

While the sensors read data and send them to text files as string values, a third script will also run simultaneously that serves as a medium of communication between the mobile robot scripts and the web server. This program first reads the two text files and stores each value as a string. The string will be of the format "*12,34,56,78,90,12,34,56*". It is then concatenated to a string that contains the URL of the web server. The final string after concatenation will resemble:"IP_address:9999/predict/?sonar=12,34,56,78,90,12,34,56". The value returned by this URL determines the direction in which the robot will move in response to the readings of its environment (see the flowchart presented in Fig.8). The program executes the aforementioned commands recursively.

V. WEB SERVER AND AI

The web server is made with a Python script that employs the Flask framework. The server script receives data that the Raspberry Pi's Python scripts send to it by calling its IP address. The web server is coded to then relay the data to an AI server, which processes the data to calculate directional commands. The commands are then sent through the web server back to the Raspberry Pi (sent to its IP address), where it is received by a Python script that instructs the motors' directional output (refer to the flowchart in Fig.8). The directional data being sent back to the module are in the form of strings, and the three possible values are "0", "1", and "2". The "0" instructs the module to go forwards, whereas "1" and "2" instruct the module to move in the left and right directions, respectively.

Explicit details of the mathematics and code behind the AI algorithm cannot be fully divulged due to copyright concerns. In essence, the web server receives data sent from the eight ultrasonic sensors, the LiDAR, and the camera mounted onto the cobot. It then converts said data to string values, and then sends them to the AI server. The AI server processes the data and sends it back to the server in the form of directional commands.

VI. LIDAR

Once the ultrasonic-sensor-equipped mobile robot is able to avoid obstacles autonomously by communicating with the web server and AI, a Light Detection and Ranging (LiDAR) sensor is added to the model. The ultrasonic sensors are useful for detecting low-level objects closer to the floor, but do not detect at a height sufficient for optimal object avoidance. A LiDAR mounted at the top of the mobile robot fulfills this requirement (as visible in Fig. 9 to the right of the camera).

LiDAR sensors are used to provide more accurate remote sensing and object distances while also being able to visually map an environment by means of image detection software.

In addition to the ultrasonic sensors, LiDAR sensors are another source of information for the AI algorithm which would make obstacle detection more precise.

LiDAR sensors are similar to ultrasonic sensors, except that LiDAR sensors use pulses from a laser to collect measurements.

LiDar sensors also have a transmitter and a receiver. The transmitter normally sends several pulses per second and the receiver reads the light waves that bounce off an object. The LiDAR sensors, like the ultrasonic sensors, also use *Time of Flight* measurements, except that the speed of sound in air is replaced by the speed of light in air, since lasers are employed (Fig. 11).

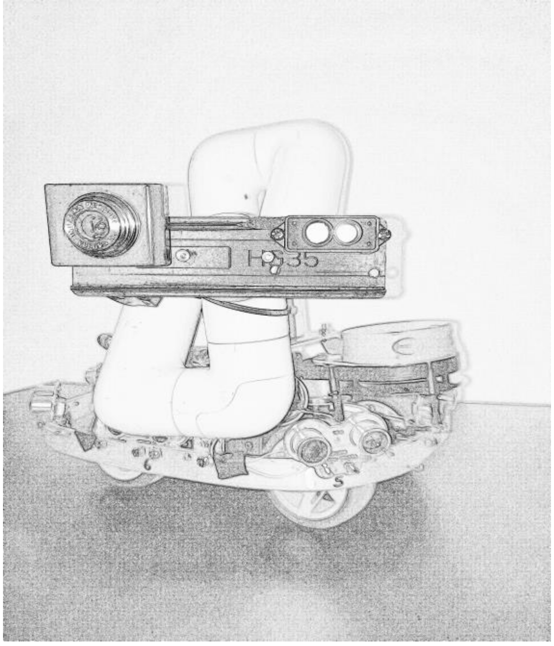

Fig. 9. Full robot with TF Mini LiDAR mounted on top-right of support slab.

The system currently uses the TF Mini LiDAR unidirectional laser sensor (Fig.10) as a prototype in order to test the functionality of the AI algorithm. It resembles the ultrasonic sensors employed in the system, however, it is much smaller.

Fig. 10. TF Mini LiDAR (ToF) Laser Range Sensor[5]

TABLE III: SPECIFICATIONS OF TF MINI LIDAR

Working Voltage	4.5-6V
Average Power	0.6W
Communication Interface	UART(TTL)
Refresh Frequency	100Hz
Max Range	12m
Min Range	0.3m
Field Of View Angle	2.3°
Ranging Accuracy	1%(<6m), 2%(6-12m)
Dimension	42x15x16 mm

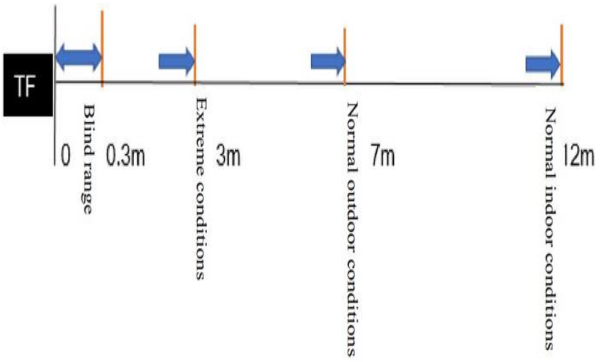

Fig. 11. TF Mini LiDAR range diagram[5]

VII. MYCOBOT-PI

The myCobot-Pi is a 6-axes collaborative robot (Fig.12). It adopts a Raspberry Pi 4 microprocessor with Raspbian by integrating it into its base, and it is used to program the movement of the cobot. In order to command the movement of the cobot, Python code is used. The cobot also allows programming through Robotic Operating System (ROS) which is integrated into its Raspberry Pi microprocessor.

The cobot serves as the arm of the mobile robot system, and it is mounted onto the mobile base. For the purposes of this product, the cobot is programmed to extend itself out horizontally with it's end effector aimed downwards. It rotates clockwise and counterclockwise between two user-defined positions, akin to a pendulum swinging to a sideways gravitational pull. A camera is mounted at the end of the cobot, which allows for environmental scanning.

Fig. 12. myCobot-Pi [6].

VIII. CAMERA

The camera used to scan the environment is a Raspberry Pi USB Webcam (in the left side of Fig. 13). The camera's cable is plugged into one of the USB ports of the Raspberry Pi at the base of the myCobot-Pi. The lens itself is mounted on top of the end effector of the cobot, in such a manner that it points in the same direction as a gripper protruding outwards from the end effector would.

The camera streams visual data to a local web server configured through Motion software, running on the cobot's Raspberry Pi. The web server can be made publicly accessible through port forwarding, the configuration of which depends on the user's router setup and security necessities.

Fig. 13. Camera(left) and mini-testing LiDAR(right) attached to end effector of myCobot-Pi

The camera can be programmed to stream data continuously while the script for the motion of the robot runs. It can additionally be programmed to take a picture at a predesignated trigger event (such as a sudden change in colour) and to store the images into a local directory within the cobot's Raspbian system, or even send it to a server via the Raspberry Pi.

IX. POSITIONING USING TRIANGULATION

The mobile robot has at least 2 Raspberry Pi microprocessors, each of them with their own IP address. In order to accurately locate the position of the robot at all times, Wi-Fi positioning can be used.

This geolocation system works by reading the signal strength of the mobile robot through appropriately placed routers.

The first step in triangulation is to find the horizontal location of the mobile robot. This is achieved by considering two points and forming a triangle with the mobile robot as the third point. This will be used to find the position of the mobile robot between two routers. By comparing the signal strength of the mobile robot on both routers, we can find the proximity of the mobile robot with respect to each router. The distance can then be approximated from the signal strength. For example, in Fig. 14, the mobile robot is closer to *Router 1* than to *Router 2*, as indicated by the colour opacity representing the strength of the connection.

The problem with using only two routers is that the mobile robot can be on either side of the line that connects *Router 1* and *Router 2* (as shown in Fig.14). Hence, another router is used and the signal strength of the mobile robot with *Router 3* indicates the position with respect to the common line between *Router 1* and *Router 2*[7].

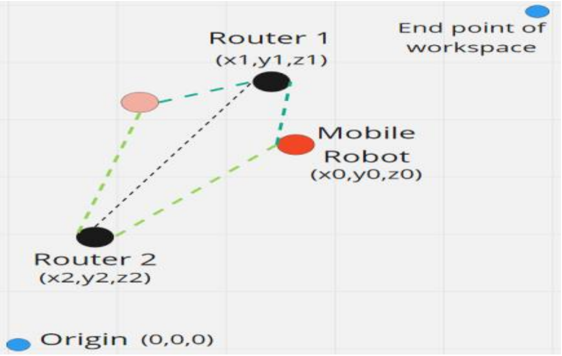

Fig. 14. Triangulation with 2 wifi routers. Colour of connection represents signal strength(green-good, orange-ok, red-weak)

The second step is to find the height of the robot and it is calculated in a similar manner as the horizontal distance, by using all three routers and their distance-signal strength approximation. A fourth router *Router 4* (Fig.15) is used for more precise horizontal and vertical distance approximation, as with three routers, the location approximation is only within 4-5 feet (1.2-1.5 m).

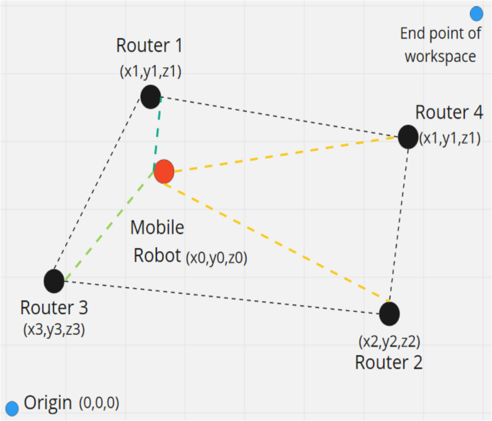

Fig. 15. Triangulation with four wifi routers. Colour of connections represent signal strength (green:good, orange:ok, red:weak)

In order to accurately calculate the distance of the robot from the router, the signal strength in dBm or RSSI from the router is used. Firstly, a fixed value of signal strength is assumed which will be used to interpolate the distance. Next, the robot is moved such that the routers provide the same reading. Lastly, the distance of the robot at the specified signal strength from each router is measured.

For example, consider a fixed strength of - 50 dBm. The robot is then moved to a position where *Router 1* reads - 50 dBm. The distance of the robot from *Router 1* is now measured and noted. The same process is repeated with *Router 2*, *Router 3*, and Router *4*.

X. Scanning the Environment Using A 2D LiDAR Sensor

Another reason for the employment of LiDAR (Fig.16) sensors is their ability to emit pulses at 360°, which means that it can scan the entire environment surrounding the mobile robot, including potential blind spots that the ultrasonic sensors may fail to detect (Fig.19).

Before being assigned any performative tasks, the LiDAR must first familiarize itself with its surroundings. It does so by mapping the environment. The mobile robot is programmed to follow a trajectory approximately following the perimeter of the targeted workspace, as well as across the center of the room, the specific trajectory thereof depending on the size and shape of the workspace. This allows the LiDAR to be exposed to the entire environment and register all obstacles it detects, including the walls and boundaries of the room (Fig.17 and Fig.18).

Fig. 16. YDLiDAR X4 [8]

Once the map has been registered, the LiDAR device can be assigned target trajectory points, which it communicates to the motors by means of the AI server. The AI algorithm collaborates with the LiDAR device to find the optimal path to the destination from its then current location. Along the path the LiDAR scans its environment and sends data to the server in real time. Upon encountering an object obstructing its path, the AI server calculates the most efficient path to avoid the object while directing itself towards the target, and then sends back directional commands.

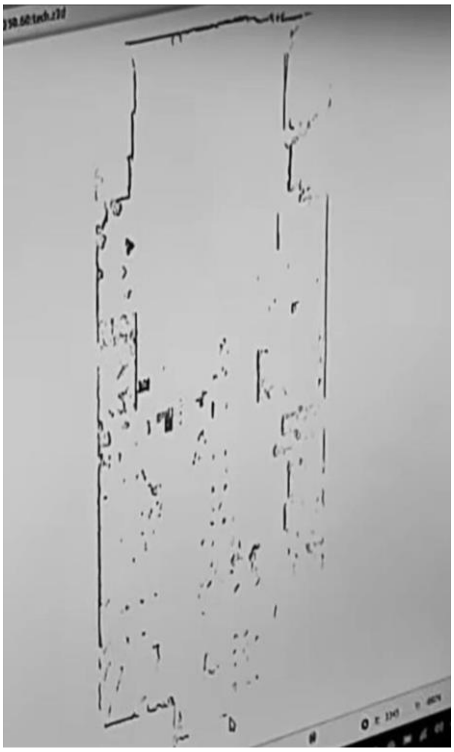

Fig. 17. Sample workspace map recorded by a LiDAR module, approx. 40m^2

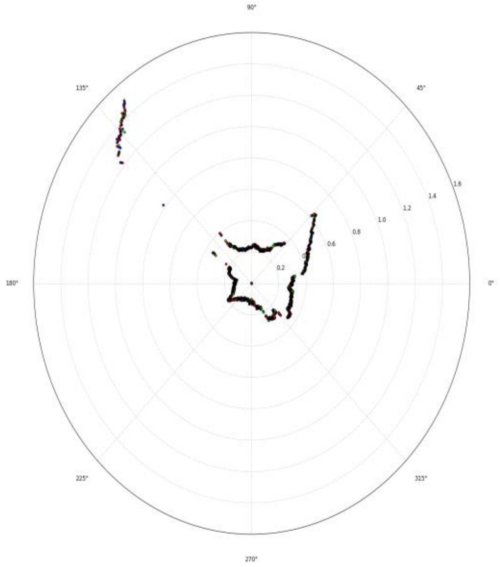

Fig. 18. Sample workspace map recorded by LiDAR module, approx. 40m^2[10]

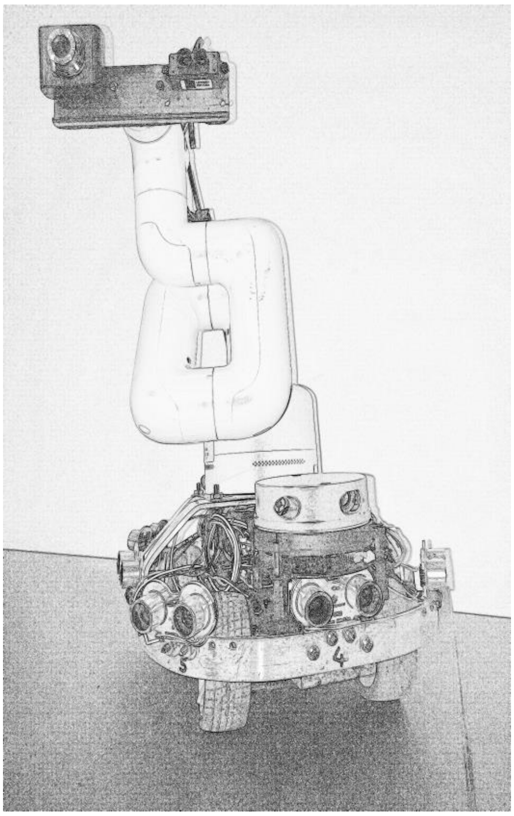

Fig. 19. Full view of robot with 360° LiDAR aimed frontwards

TABLE IV: SPECIFICATIONS OF YDLIDAR X4 [9]

Working Voltage	4.8-5.2V
Working Current	330-380 mA
Baud Rate	128000 bps
Range Frequency	5000 Hz
Range Resolution	<0.5 mm(for Range<2m) <1% of distance(Range>2m)
Scanning Frequency	6-12 Hz
PWM Frequency	10 kHz
Scanning Angle	0-360°
Max Range	10m
Min Range	0.12m
Angle Resolution	0.48°-0.52°
Ranging Accuracy	1%(<6m), 2%(6-12m)
Laser Wavelength	775-795 nm
Laser Power	3 mW
Dimension	102x71x63 mm

XI. FUTURE WORKS

The robot is of a very unique and innovative design, but still holds much room for improvement. For instance, the camera currently employed in the system is very basic and it does not produce high quality images and live feed which might be essential to the AI server, hence, the usage of better cameras that captures good quality images and live video feeds is necessary.

The mechanical structure currently used is not fully capable of withstanding the myCobot Pi at the top, and the motion of the mobile robot is not smooth. Moreover, parts of the structure use plastic, which is not durable. Hence an upgrade is due to be made to the structure that will use aluminum or steel in order to be able to withstand larger payloads whilst unhindering the motion of the robot.

While the sensors currently used for the system are effective, they are not durable and have a smaller field of view, which might increase susceptibility to blind spots during motion. Hence, larger and better sensors will be used in order to send data to the AI server more efficiently and accurately.

In order for the mobile robot to inspect fuel tanks, it requires the aid of another robot that is able to climb walls, as jet fuel tanks are partitioned, and the robot is required to move through the openings of each partition effortlessly.

The time taken for the mobile robot to communicate with the web and AI server is minimal at the moment. However, it can be made much lower. Using better sensors and motors that can send and receive commands much faster than the ones currently employed, and having better internet connectivity are ways of minimizing the latency.

The AI model that calculates the direction based on the input from the sensors is still under development and optimization to ensure that the processing of data takes less time and the directions commands are more accurate and instantaneous.

XII. EQUATIONS

The mathematical formula used by the ultrasonic sensors to calculate the distance of a target is as follows:

$$Distance = \frac{1}{2} \cdot Duration\ of\ pulse(s)\ .Speed\ of\ sound\left(\frac{m}{s}\right) \cdot 100\left(\frac{mm}{m}\right)$$

Distance is the product of *Duration of pulse*, which is the time taken by the pulse to be transmitted and received, and the *Speed* of *sound*. This equation is multiplied by 0.5 as the *Duration of pulse* provides the total time of flight, whereas *Distance* only gives the length of an object from the sensor.

The formula used to calculate the distance of a target object from a LiDAR sensor is as follows:

$$Distance = \frac{1}{2}.Duration\ of\ laser\ pulse(s)\ .Speed\ of\ light\left(\frac{m}{s}\right).100\left(\frac{mm}{m}\right)$$

Distance in this equation is found in a similar way to the previous equation, but in this case, the distance of an object from the LiDAR sensor is measured by using the *Speed of light*.

XIII. CONCLUSION

The mobile collaborative robot with obstacle avoidance is a revolutionary system that opens up a new world of completely autonomous systems that incorporate AI. The importance of such systems is crucial with rapid technological advancements, since the more technology progresses, the more time and resources are needed for creative thinking in order to further such progressions. Allocating human effort into tasks that can instead be automatized is a waste of resources over the course of time. Robots such as the one presented in this paper provide a means to optimize workflow and support sustainability.

AUTHOR CONTRIBUTIONS

Angad Singh Malhotra was involved in the development of the software, the electrical and wiring assembly, and the mechanical structure of the system, as well as in the composition of the paper.

Karthick Sivasubramanian was involved in the development of the software, the electrical and wiring assembly, and the mechanical structure of the system, as well as in the composition of the paper.

Niculae Mihai coordinated the research team and produced the design for the system, in addition to developing the mechanical structure and editing the paper.

Iulia Mihai was involved in a development research, programming architecture and algorithms solution.

ACKNOWLEDGMENT

The authors thank the technical support provided by Technoaccord Mechatronics Company from Laval, Quebec, Canada.

REFERENCES

[1] G. Niu, J. Wang, and K. Xu, “Model analysis for a continuum aircraft fuel tank inspection robot based on the Rzeppa universal joint,” *Adv. Mech. Eng.*, vol. 10, no. 5, p. 1687814018778229, May 2018, doi: 10.1177/1687814018778229.

[2] SF-SR02 Ultrasonic Module - Wiki. [Online]. Available: http://wiki.sunfounder.cc/index.php?title=SF-SR02_Ultrasonic_Module

[3] Understanding How Ultrasonic Sensors Work | MaxBotix Inc. [Online]. Available: https://www.maxbotix.com/articles/how-ultrasonic-sensors-work.htm

[4] What is an Ultrasonic Sensor?. *FierceElectronics*. [Online]. Available: https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor

[5] SEN0259 TF Mini LiDAR (ToF) Laser Range Sensor. [Online]. Available: https://abra-electronics.com/sensors/sensors-proximity-en/sen0259-tf-mini-lidar-tof-laser-range-sensor.html

[6] myCobot Pi- Raspberry Pi Powered 6 DOF collaborative robot. *ElephantRobotics*. [Online]. Available: https://shop.elephantrobotics.com/products/mycobot-pi-worlds-smallest-and-lightest-six-axis-collaborative-robot

[7] Android - Wi-Fi position triangulation. *Stack Overflow*. [Online]. Available: https://stackoverflow.com/questions/16485370/wi-fi-position-triangulation

[8] EAI YDLIDAR X4 360 Degree 2D LIDAR Lidar Range 10M Finder Sensor Module with UART Communication Interface: Amazon.ca: Industrial & Scientific. [Online]. Available: https://www.amazon.ca/-/fr/gp/product/B07DBYHJVQ/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&language=en_CA&th=1

[9] YDLIDAR X4 Lidar – 360-degree Laser Range Scanner (10 m). [Online]. Available: https://www.elektor.com/ydlidar-x4-lidar-360-degree-laser-range-scanner-10-m

[10] *Table of Contents*. Shenzhen Yuedeng Technology Co.,Ltd., 2021. Accessed: Aug. 09, 2021. [Online]. Available: https://github.com/YDLIDAR/YDLidar-SDK

Angad Singh Malhotra is a computer engineering undergraduate co-op student at Concordia University in Montreal, Canada. He has worked as a programmer at private mechatronics systems company Technoaccord in Quebec, Canada.

Karthick Sivasubramanian is an aerospace engineering undergraduate co-op student at Concordia University. He has worked as a programmer at private mechatronics systems company Technoaccord in Quebec, Canada.

Niculae Mihai graduated from the University Politehnica of Bucharest, Faculty of Machines and Manufacturing Systems, Romania. From 2006 he became Ph.D.Eng. in the field of robotics. Now, he is the manager of a private mechatronics systems company Technoaccord in Quebec, Canada.

Iulia Mihai is a mechatronics engineering undergraduate student the ETS University of Montreal, Quebec, Canada. She is the president of Iumirobo Inc. located in Laval, Quebec, Canada involved in advanced robotics systems.