Design and Implementation of Autonomous Sonar Based Vehicle Robot

MUHAMMAD ADIL ANSARI*, FAHIM AZIZ UMRANI**, AND KEHKASHAN ASMA MEMON***

RECEIVED ON 26.06.2009 ACCEPTED ON 18.11.2009

ABSTRACT

Autonomous robots are intelligent machines that are capable of performing desired tasks by themselves, without explicit human control. This paper presents design and implementation of the ASVR (Autonomous Sonar Based Vehicle Robot). ASVR is a microcontroller based, programmable mobile robot that can sense and react to its environment and can work in partially known and unpredictable environments. A novel algorithm based on ultrasonic sensors and simple calculations for real-time obstacle detection and avoidance that is intended for mobile robots is also outlined. Also a novel technique is proposed and implemented for steering referencing of vehicle. The design is implemented in air using ultrasonic sensors but can be adapted using sonar to underwater environments where it has important applications such as deep sea maintenance and reconnaissance tasks. The paper also presents performance results of a prototype developed to prove the design concept.

Key Words: **Ultrasonic Sensor; Obstacle Detection; Obstacle Avoidance, Autonomous Robots.**

1. INTRODUCTION

Mobile robotic vehicles are increasingly being proposed for high-risk, high-speed, rough terrain scenarios, such as planetary exploration, mining, forestry, hazardous site clean-up, and military applications [1-2]. Potential missions include logistics, surveillance, fire missions and soldier assistance [3]. According to UNIFR (United Nations and The International Federation of Robotics), the use of robots has increased from 86,200-106,300 operational units, from 2004-2007, a 19% increase in the matter of just three years [4]. A new area showing commercial promise is domestic robots, with a flood of small vacuuming robots. In order to be successful, such robotic systems must provide an effective and inexpensive service to their users.

Keeping this goal in mind, this paper presents the extended version of [5] which provides the simple and relatively inexpensive mapping and avoidance system with the ability to detect and avoid obstacles at various distances. In [5], the focus of the authors is on the proposed algorithm,

---

* M.Sc. Student, Hangyang University, South Korea.

** Lecturer, Department of Telecommunication Engineering, Mehran University of Engineering & Technology, Jamshoro.

*** Assistant Professor, Department of Electronics Engineering, Mehran University of Engineering & Technology, Jamshoro.

the robot uses ultrasonic transducers for gathering useful information about surroundings interfaced with microcontroller for making simple logic reasoning to calculate range and information required for navigation. While in this paper the concepts and techniques proposed in [5] are implemented and tested. The design steps taken to the control the movement of vehicle, steering reference section, and distance measuring with microcontrollers are presented in detail. Further, a novel technique for steering reference section is also detailed. Currently autonomous robots rely on lasers to determine optimum path and nearby obstructions, while DVL (Doppler Velocity Logs), inertial rate gyros, and magnetic compasses to estimate vehicle position [6]. Laser sensors are sensitive to ambient light, absorption, shorter range and color of object [7], posing many design challenges. Further many state of the art and complex algorithms such as DDF (Decentralized Data Fusion) and cooperative control algorithms are developed in the literature [8] for unmanned vehicles. While such algorithms are effective in performance but require the use of complex technology and expensive hardware, thus not suitable for the commercial use of autonomous robots. Therefore, the work described here uses ultrasonic transducers for gathering useful information about surroundings interfaced with microcontroller for making simple logic reasoning. We name our robot ASVR. A novel iterative algorithm developed in C-language [9] is employed to find the position of obstacles in real time and enabling the robot to decide upon how it will go about avoiding the obstructions. Robot's mobility is provided by a gear headed DC motor and stepper motor which provides steering. While the level of intelligence is not high in this system, it can navigate over wide areas and pilot in tight situations around homes or open environment using non-contact sensors.

The remainder of this paper is organized as follows. In the next section (Section 2), robot design model of ASVR and its structure is elaborated with detailed description of distance measurement using ultrasonic sensors. Section 3 discusses the proposed algorithm for detection and avoidance of the objects. This section also demonstrates the method of terrain mapping. Finally conclusions and future recommendations are mentioned in Section 4.

2. ROBOT DESIGN

The following subsections outline the basic methods and techniques employed in the design process. The system is mainly divided into five blocks as shown in Fig. 1, namely MCU (Microcontroller Unit), DMU (Distance Measuring Unit), DC motor drive, steering referencing section and stepper motor drive. The detailed functioning of these blocks is described in following sub-sections.

2.1 Microcontroller Unit

The heart of our designed system is MCU. We have used AT89C52 for its simplicity, low cost and easy availability [10]. There are five major sections of ASVR and all are controlled by MCU as shown in Fig. 1.

The DC motor drive section is responsible to drive the vehicle in forward and backward direction with adjustable speeds (slow, medium or fast) depending on the particular scenario. Both DC motor drive and stepper motor drive are connected with microcontroller via port 0 [10]. DC motor drive has three control lines (i.e., P0_0-P0_2) reserved to control DC motor and stepper motor drive has four control lines (i.e. P0_0-P0_2) to control Stepper motor. Steering reference section has only one control line which sends signal to microcontroller to tell the reference point of the steering. P1_0 is connected to steering reference section which is input to microcontroller. DMU is connected to one external interrupt and P2_0, Here both act as input and output of microcontroller respectively.

2.2 DC Motor Drive

DC motor is used to control the direction and speed of the designed vehicle. The mechanical arrangement of DC motor connected with the wheels and microcontroller unit is illustrated in Fig. 2. ASVR is driven in forward and backward direction with variable speeds depending on the environment the vehicle encounters. The different speeds are produced by PWM (Pulse Width Modulation) technique [11]. PWM varies speed of DC motor by varying its duty cycle. A multiplexer is used to select the desired speed and the selected speed fed to two AND gates. These AND gates are switched by direction control line coming from controller. PWM signal passing from AND gate enters into the H-Bridge, and accordingly DC motor is derived.

This DC motor drive unit is interfaced with the MCU via three lines as described earlier; two for speed control (C_0 and C_1) and one for the direction control (C_3) as shown in

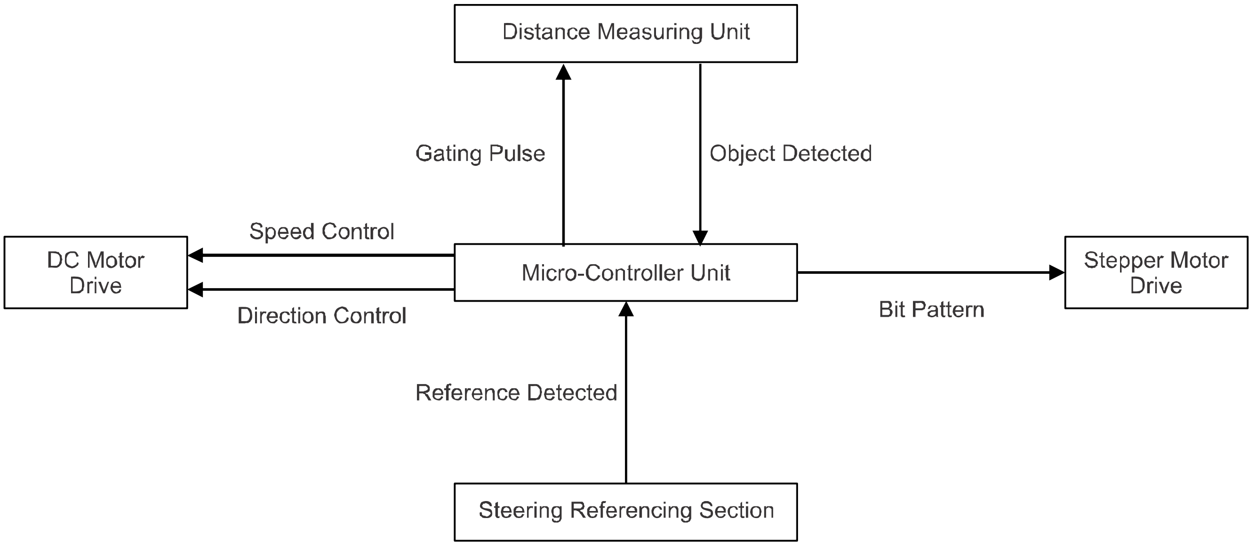

FIG. 1. BLOCK DIAGRAM OF ASVR

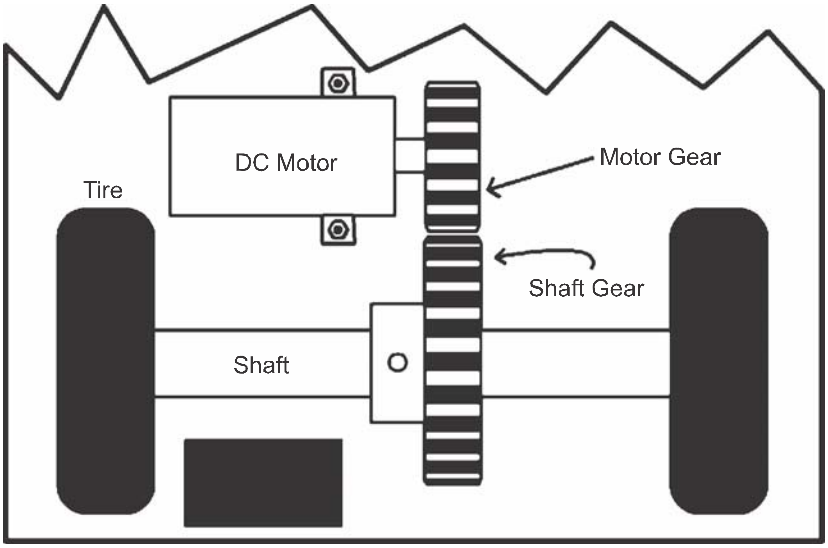

FIG. 2. MECHANICAL ARRANGEMENT SHOWING CONNECTIVITY OF DC MOTOR WITH SHAFT

Fig. 3. Here three different speeds are fixed (fast, medium and slow) for different conditions as shown in Table 1. When the ASVR finds a clear path without any obstructions, fast speed is selected, however, when the robot detects obstructions in its path it travels at moderate speed. While to turn the vehicle left or right, slow speed is selected by the controller. To drive ASVR in slow and medium speeds, two pulse-width modulated signals, PWM_1 and PWM_2, at frequencies of 15Hz having duty cycles of 65 and 85% are generated, respectively. To drive the ASVR in full speed a +5V signal is multiplexed along with PWM_1 and PWM_2 into the input of direction control lines as shown in Fig. 3.

The other part of a DC motor drive, which is used for direction control, contains H-Bridge that is quite popular circuit arrangement for direction control of DC motor [12]. The control lines C_2 and C_3 allow us to select the direction of motor. Fig. 3 illustrates the arrangement of two AND gates, H-Bridge and control lines C_2 and C_3 (Table 2) for direction selection.

TABLE 1. TRUTH TABLE FOR SPEED SELECTION

C_0	C_1	Speed
0	0	PWM_1 (Slow Speed)
0	1	PWM_2 (Medium Speed)
1	0	High Level Signal (Fast Speed)
1	1	Not Used

2.3 Stepper Motor Drive

For the DMU placed on the wooden platform, it is essential that, both the platform and DMU must be able to maneuver their direction. Both the DMU and wooden platform can be controlled via single stepper motor, but this would restrict the capability of DMU to scan the field independently before advancement. Therefore, we have used two 4-bit stepper motors control the steering of ASVR [11]. They are controlled via port 0 of the microcontroller. The push pull arrangement of BJT transistors (2N3055) are used as current amplifiers for the required 0.5A current per phase by the stepper motors as shown in Fig. 4.

This push pull pair uses 2N3055 BJT transistor, which is widely used power transistor and easily available. As we require 0.5A the minimum β of 2N3055 transistor is selected 20, so we can find the value of R_b to limit the current in the desired range i.e. 0.5A. Using the following relation:

$$\beta = \frac{I_c}{I_b} \tag{1}$$

TABLE 2. TRUTH TABLE FOR DIRECTION SELECTION

C_2	C_3	Direction
0	0	Free
0	1	Clockwise
1	0	Anti Clockwise
1	1	Break

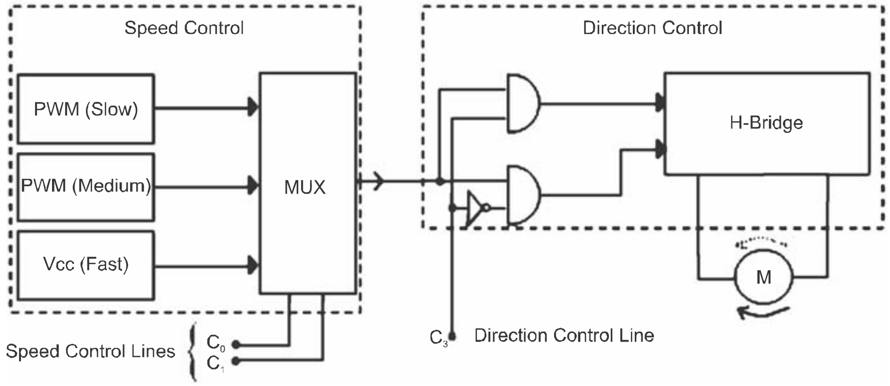

FIG. 3. BLOCK DIAGRAM OF DC MOTOR DRIVE

$I_c = 0.5$A and $I_c \cong I_e$, therefore,

$$I_b = \frac{I_c}{\beta} = \frac{0.5}{20} = 25mA \tag{2}$$

$$R_b = \frac{V_b}{I_b} = \frac{5}{0.025} = 200\Omega \tag{3}$$

We use the resistor of 220Ω as the one with 200Ω is not available. Alternatively, one can use two 100Ω resisters in series. Fig. 4 is used just for one bit and therefore, for four bits we need to cascade four push-pulls BJT transistors.

2.4 Steering Reference Section

When ASVR starts its operation it must know the current heading of the steering. For this purpose, we propose a novel technique which creates a reference or zero point by drawing a white line in the black coloured gear as shown in Fig. 5.

When the robot starts its operation, it resets the position of steering to a straight head. This zero point can be detected through an arrangement of LDR (Light Detection Resistor) and LED (Light Emitting Diode) as shown in Fig. 6. A white light LED is used which is installed in such a manner that it illuminates on the surface of steering gear as shown in Fig. 5. A LDR is installed pointing right above the area where the white light falls. When the white light emitting from LED illuminates the white line on the gear, it is reflected back and collected by the LDR. Consequently, the resistance of LDR is drastically reduced resulting in a

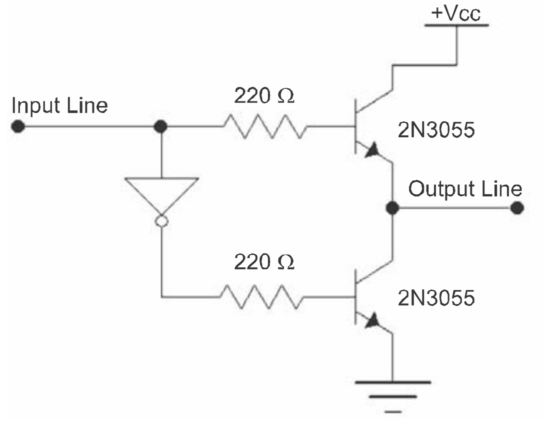

FIG. 4. PUSH-PULL ARRANGEMENT OF BJT FOR ONE PHASE OF STEPPER MOTOR

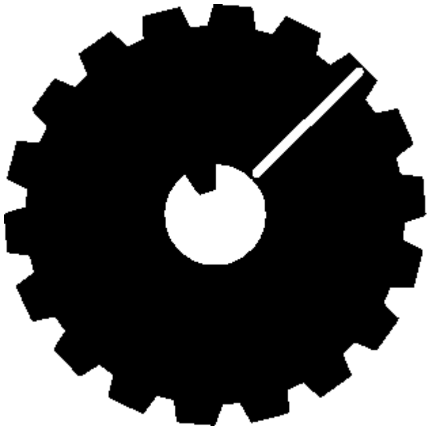

FIG. 5. TOP VIEW OF GEAR HIGHLIGHTING ZERO POINT

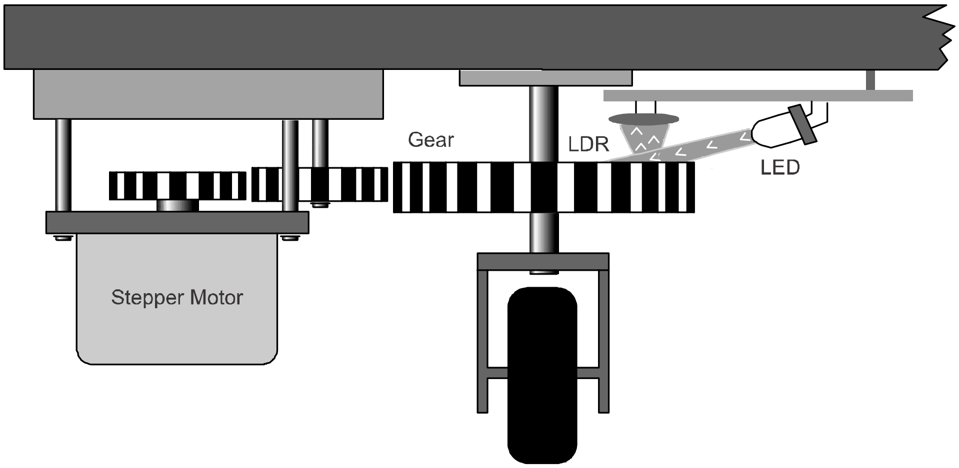

FIG. 6. MECHANICAL ARRANGEMENT SHOWING CONNECTIVITY OF STEPPER MOTOR WITH STEERING

high signal to the microcontroller unit which indicates the detection of white light. Once this white line or reference point is detected it is considered that the wheel is front faced. After some observations we have come to know that stepper motor with gear we used rotates 0.75^o per step. So, if the robot has to move 45^o left, its steering section will be rotated to left up to 45^o and then back to zero, similarly as we do in car driving. Now to move 45^o left, microcontroller unit will need to send 60 (45/0.75=60) bit transitions.

2.5 Distance Measuring Unit

Fig. 7 shows the block diagram of DMU. In DMU, we have used RADAR principles to measure the distance [13]. A pair of ultrasonic transducers is used for transmitting the ultrasonic pulses and receiving their echo. The transmission of the pulses is initiated and the echo is processed by a microcontroller (AT89C52) to calculate the distance. Since the center frequency of ultrasonic transducers is 40KHz, short bursts of 40KHz signals after an interval are sent to ultrasonic speaker.

The mechanical arrangement of the DMU is shown in Fig. 8. MU sends a gating pulse to DMU which is fed into the oscillator through one-shot multi-vibrator to generate a 40KHz signal. This signal is amplified and then transmitted through ultrasonic speaker. If the transmission is not intercepted by any obstruction, no echo is received;

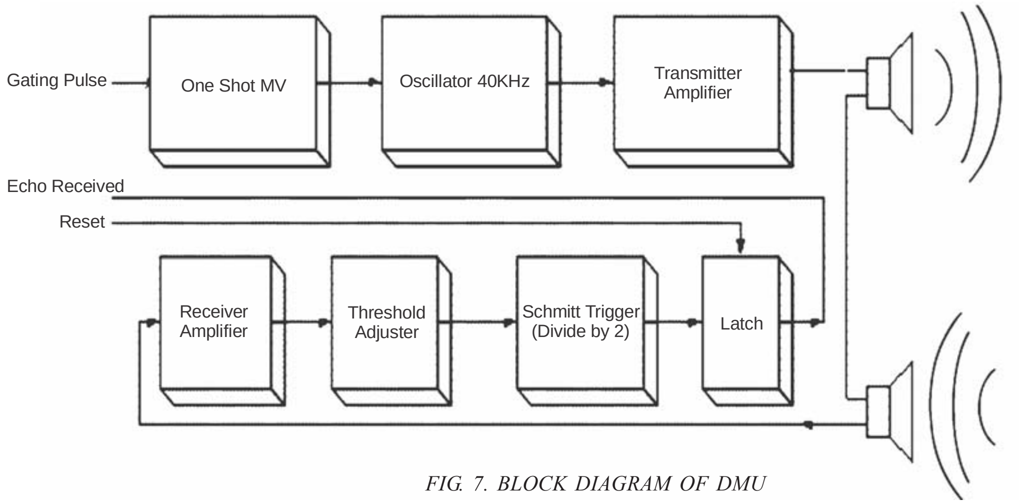

FIG. 7. BLOCK DIAGRAM OF DMU

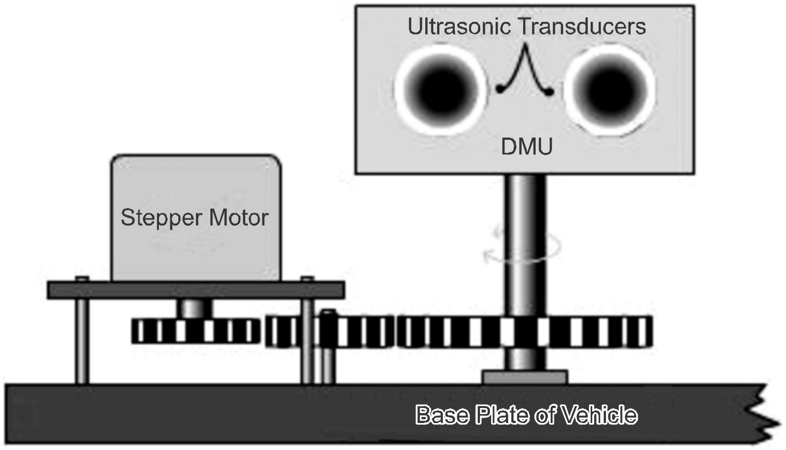

FIG. 8. MECHANICAL ARRANGEMENT OF DMU AND STEPPER MOTOR

otherwise a bounced back signal is picked by the sensor. The received echo is amplified and then fed to the threshold adjuster which is used for calibration purpose. The amplified signal is latched to the MU. Due to two-way travel of the ultrasonic pulse i.e. from DMU to the object and back to DMU, the time elapsed between transmission and reception is divided by 2 within microcontroller.

After receiving the echo and calculating the distance of the obstacle the DMU is rotated with small angle using stepper motor to the left or right. At this point the robot again transmits the signal and look, for the clear path. This process is repeated until a clear path is found.

For the DMU placed on the wooden platform, it is essential that, both the platform and DMU must be able to maneuver their direction. Both the DMU and wooden platform can be controlled via single stepper motor, but this would restrict the capability of DMU to scan the field independently before advancement. Therefore, we have used two 4-bit stepper motors for this purpose. When the robot starts its operation it must know the current orientation of the steering. This information is vital to us if we desire to construct a map out of the information. For this purpose, a steering reference section is designed.

3. OBJECT DETECTION AND AVOIDANCE

Proposed technique is carried out in two phases: obstacle detection phase, and obstacle avoidance phase.

3.1 Obstacle Detection

In detection phase, to detect the object in its range the robot transmits a modulated pulse of 40KHz. In this design one of the internal timers of the microcontroller is used to measure the time t during which ultrasonic wave is transmitted and an echo is received. MU starts the timer when gating pulse is sent and stops it when it receives an echo. If no echo is received until time out, the robot continues its advance in the current direction and the timer is reset after which another gating pulse is sent. This procedure is repeated until any echo is received, in other words, an object is detected. When the object is detected, the contents of timer register are used to calculate the distance between the DMU and the object using simple distance and time relation, d=(txc)/2, where d is the distance between object and the DMU, c is the speed of sound in air (344 m/s), and t is the time taken by sound wave to bounce back. The operating clock speed of the microcontroller used is 12MHz and the timer frequency is 1/12th the crystal frequency i.e. 1MHz, therefore, the smallest time interval that the internal timer can measure is 1µs. Once the object is detected it enters into obstacle avoidance phase.

3.2 Obstacle Avoidance

In obstacle avoidance phase, an optimum path is determined for ASVR to move further. Once the object is detected, the ASVR stops moving and the MU directs the stepper motor to rotate in the left direction with small steps up to 90^o. On each step an ultrasonic pulse is transmitted. During this scanning, if no object is found, the ASVR is directed to go ahead from its current location. Otherwise, the DMU calculates the distance of newly detected object. Next, the angle between two detected objects is calculated exploiting the well known expression:

$$\text{Angle} = \text{Number of Pulses Applied x Angles Per Step} \tag{4}$$

Once the angle is found we can calculate the distance between the two objects based the proposed algorithm that is described in next section. The calculated distance information between the two obstacles is used to decide whether the ASVR can go through it or not.

3.3 Proposed Algorithm

We assume that the shape of ASVR is square and s indicates the length of each side. This algorithm is applicable for only static objects in the arena. ASVR moves in its arena and DMU continuously sends short bursts of ultrasound waves for scanning, and waits for the return of the echo for a predetermined period of time, which is set in the microcontroller program.

Let us assume that the ASVR is placed in a terrain with two objects represented by Obj_1 and Obj_2 in Fig. 9. Here we are interested in calculating the distance represented by lines a_1 and b_1 to check that ASVR can take a path in between the two objects or not.

When the first object Obj_1 is detected by ASVR, it calculates its distance to Obj_1, i.e. x_1 and rotates the DMU to its left with small angle interval to find the left most edge of the Obj_1. The distance between DMU and the edge of Obj_1 is denoted by x'_1 and this line segment makes an angle α_1 with respect to the line x_1. To find the front end boundary of Obj_1, at each step top view image of that particular line is calculated and compared with the last value. This process will continue till the extreme edge of Obj_1 is found. The shortest value is saved in the variable x_1 that shows the nearest edge of Obj_1. The front end boundary will help to avoid any collision of ASVR with obstruction. This simple mechanism gives our design extra ability to deal with irregular objects.

Now that we have x_1 and x'_1 lines, using Pythagorean relation we can find h.

$$h = \sqrt{(x_1')^2 - (x_1)^2} \quad (5)$$

The horizontal dashed line in Fig. 9 is equivalent to x_1 line shifted left with h units after finding the left edge of Obj_1. Next the DMU is rotated further to its left up to 90° to find a clear path. Assume that there is another object Obj_2 detected at distance x_2 by ASVR during scanning process in the arena. x_2 is considered the rightmost edge of Obj_2. Next we calculate the angle α_2 that x_2 makes with respect to line x_1. As can be seen in Fig. 9 b'_1 is the top view image of line x_2, and can be calculated as:

$$b_1' = x_2 \cos\alpha_2 \quad (6)$$

and

$$b_1 = x_1 - b_1' \quad (7)$$

By Pythagoras Theorem

$$a_1' = \sqrt{(x_2)^2 - (b_1')^2} \quad (8)$$

Therefore,

$$a_1 = a_1' - h \quad (9)$$

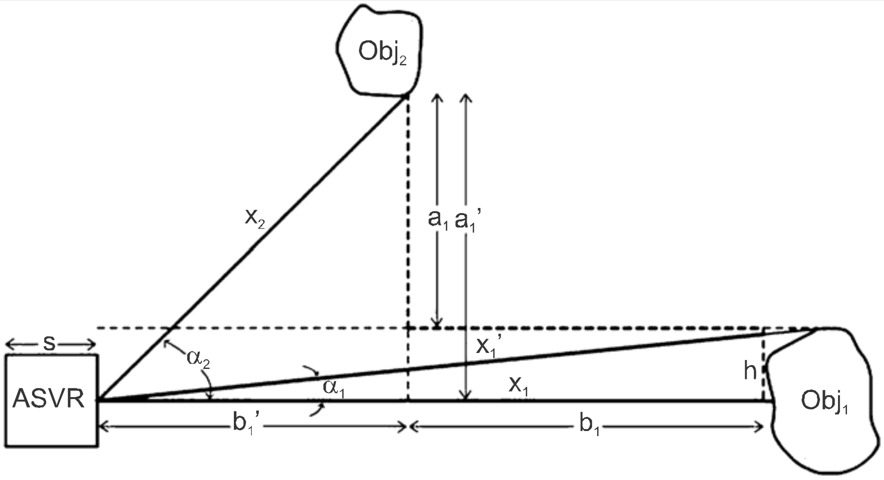

FIG. 9. TWO OBJECT CASE

Now the values of a_1 and b_1 are compared with the sides of ASVR i.e. s. If these values are sufficiently larger than s that means the ASVR can pass through the two objects.

Utilizing the principle described previously for two objects, Fig. 10 shows a general arrangement of ASVR with respect to the three objects. To avoid complexity, the line segments are representing the distances between DMU and objects and considered as extreme edges accordingly. Here b'_1 is the top view image of x_2, therefore,

$$b_1' = x_2 \cos\alpha \tag{10}$$

and

$$b_1 = x_1 - b_1' \tag{11}$$

Note from Fig. 10 that the lines a_1, b_1, and x_2 make a right angle triangle, therefore, a_1 can be calculated using Pythagoras theorem:

$$a_1 = \sqrt{(x_2)^2 - (b_1')^2} \tag{12}$$

Comparing values of a_1 and b_2 with s, it can be decided whether ASVR can pass through the gap between Obj_1 and Obj_2. If ASVR cannot pass through the gap then DMU is rotated further to its left to find the left most edge of Obj_2 and right most edge of Obj_3. The distance between DMU and Obj_3 is denoted as x_3 in Fig. 10. Here b'_1 forms the top view image of x_3, therefore, we have:

$$b_2' = x_3 \cos\beta \tag{13}$$

$$b_2 = b_1 - b_2' \tag{14}$$

Again note that the lines a'_1, b'_1 and x_3 form a right angle triangle, therefore,

$$a_2' = \sqrt{(x_3)^2 - (b_2')^2} \tag{15}$$

$$a_2 = a_2' - a_1' \tag{16}$$

Now by comparing values of a_2 and b_2 with s, appropriate decision can be made using proposed technique to move or not to move through the gap between Obj_2 and Obj_3. Generalizing this process for ASVR's path planning we are only concerned with the values of a_1, a_2, a_3 … and b_1, b_2, b_3 …. Comparing these values with s decisions are made as follows (Path patterns are depicted in Fig. 11):

- ON
- Calibrate sensors
- Record orientation

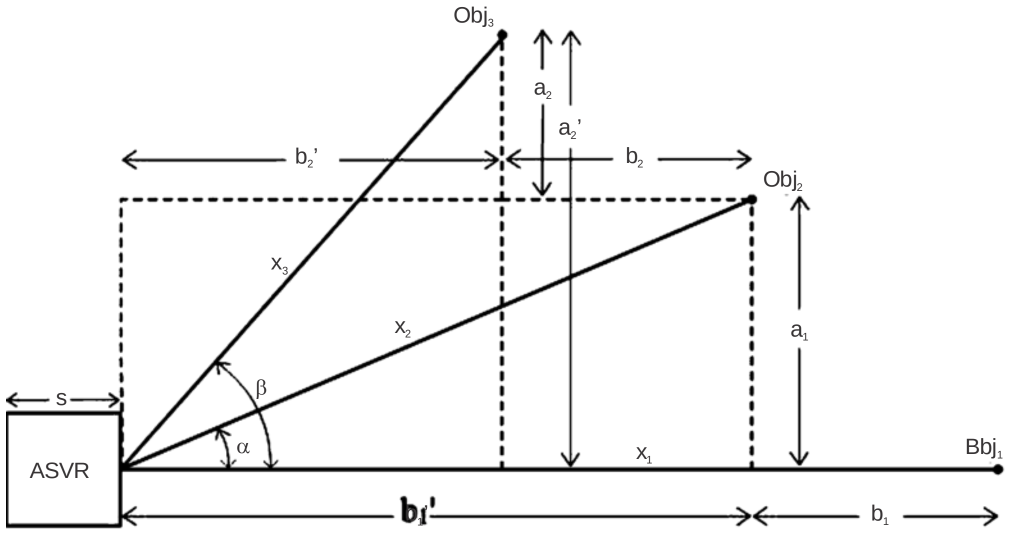

FIG. 10. GENERAL ARRANGEMENT OF OBJECTS

- Transmit pulse
- If Obj_1detected = no then keep moving forward
- If Obj_1detected = yes then stop
- Rotate DMU to left to detect Obj_2
- If Obj_2detected = no then followpath_1 //
- If Obj_2detected = yes then calculate (a_1, b_1)
- If $a_1>s$ and $b_1<s$ then follow_path_2
- If $a_1<s$ and $b_1<s$ then
    - If $a_1>$half of s then follow_path_3
- If $a_1>s$ and $b_1>s$ then
    - If $a_2>b_2$ then follow_path_2
    - If $b_2>a_2$ then follow_path_3
- If $a_2<s$ and $b_2<s$ then rotate DMU to left to check Obj_3
- If Obj_3detected = no then follow_path_4
- If Obj_3detected = yes then calculate (a_2, b_2)
- If $a_2>s$ and $b_2<s$ then follow_path_5
- If $a_2<s$ and $b_2>s$ then follow_path_6
- If $a_2>s$ and $b_2>s$ then
    - If $a_2>b_2$ then follow_path-5
    - If $a_2<b_2$ then follow_path_6

This process will continue until the DMU has rotated up to 90° with respect to x_1, i.e. when maximum rotation to the left has been achieved and still no appropriate path has been determined. In such case, a similar procedure is followed to the right side of the Obj_1. Once the appropriate path is determined and followed the DMU is placed front faced and again detection phase is started.

4. TERRAIN MAPPING

ASVR has implemented a simple but effective algorithm using ultrasonic sensors. It pings obstructions and measures the echo from the object, then using steering reference section it determines the orientation of the vehicle. ASVR's path planning is randomly determined in any unknown environment. Most autonomous robots rely on maps taken by satellite pictures, but in indoor

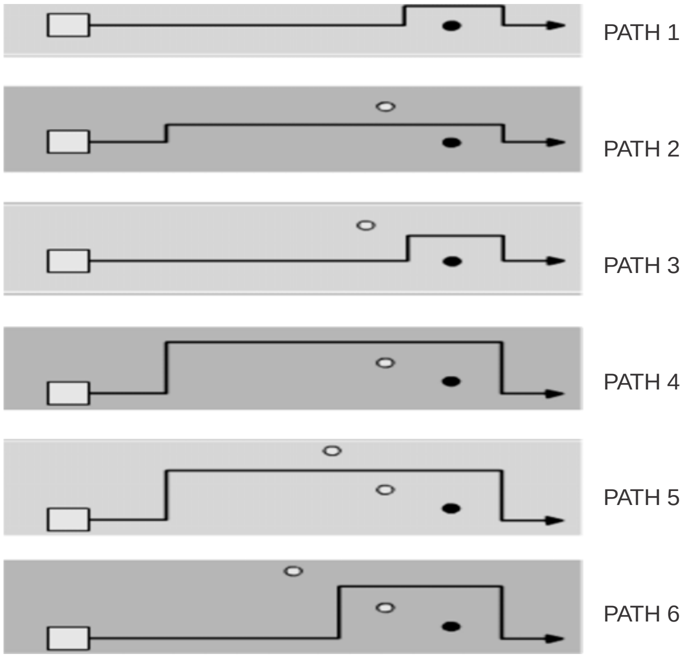

FIG. 11. PATH PATTERNS FOLLOWED BY ASVR

environment this is not possible. Therefore, the proposed design develops the two-dimensional map by processing the sensed information. Once the data is received in the specified format, the data can then be processed through an algorithm which will convert it into X and Y coordinates on a map. Fig. 12 displays the mapping process. The algorithm simply takes the current location of the robot as (0, 0) and records the distance the object was detected from and lastly the angle at which the object was detected and an XY map point can easily be constructed, indicating the location where the object is located.

5. CONCLUSIONS

This paper demonstrates the ability of ASVR to successfully detect major obstacles in its immediate environment. The proposed algorithm is based on simple trigonometric relations for making decision for optimum path and its implementation is also very cheap making the design commercially viable. Using the proposed algorithm, ASVR can efficiently avoid obstacles in its path by detecting their presence using trigger signals and is capable of determining the optimum path through the gap between obstacles or around them. This gives ASVR the ability to work in unknown environments.

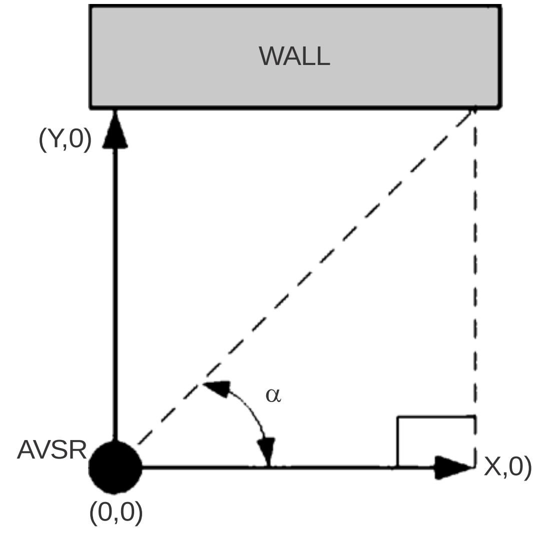

FIG. 12. OBJECT DISTANCE CALCULATION AND TERRAIN MAPPING

In our design the DMU is able to measure the distances up to 30 inches accurately. Although the proposed design proves to be a success it has its own limitations. The errors in accurate measurement of the distance are due to the slow rise time of the received pulses from the obstacles and error in estimating the speed of sound. The ultrasonic sensors are affected by the wind and the hardness of object. So, the range of DMU needs to be increased so that the autonomous vehicle can find its clear path while moving through the arena. In unknown environment moving agents also need to be detected and the extension of this work also include the detection and avoidance of dynamic agents.

ACKNOWLEDGEMENT

The authors would like to express their heartily gratitude to the Digital Electronics Laboratory, Department of Electronic Engineering, Mehran University of Engineering & Technology, Jamshoro, Pakistan.

REFERENCES

[1] Golombek, M., "Mars Pathfinder Mission and Science Results", Proceedings of the 29th Lunar and Planetary Science Conference, 1998.

[2] Shoemaker, C., and Bornstein, J., "Overview of the Demo-III UGV Program", Proceedings of the SPIE Conference on Robotic and Semi-Robotic Ground Vehicle Technology, Volume 3366, pp. 202-211, 1998.

[3] Dariusz, G., Karl, I., and Steven, D., "Probabilistic Modeling and Analysis of High Speed Rough-Terrain Mobile Robots", Proceedings of the 2004 IEEE International Conference on Robotics & Automation, New Orleans, LA, April, 2004.

[4] UN and IFRR (United Nations and The International Federation of Robotics), World Robotics, 2002. United NY, USA and Geneva.<http://www.asimovonline.com/>.

[5] Ansari, M.A., and Umrani, F.A., "SONAR Based Obstacle Detection and Avoidance Algorithm", International Conference on Signal Acquisition and Processing, Malaysia, 2009.

[6] Grenon, G., An, P.E., Smith, S.M., and Healey, A.J., "Enhancement of the Inertial Navigation System for the Morpheusautonomous Underwater Vehicles", IEEE Journal of Oceanic Engineering, Volume 26, No. 4, pp. 548-560, October, 2001.

[7] Katzberg, S.J., "Accuracy Limitations of Laser-Based Ranging Techniques in Robotic Applications", Southeastcon apos;90. Proceedings IEEE, Volume 2, No. 1-4, pp. 627-631, April, 1990.

[8] Cole, D.T., Goktogan, A.H., Thompson, P., and Sukkarieh, S., "Mapping and Tracking", IEEE Robotics & Automation Magazine, Volume 16, No. 2, pp, 22-34, June, 2009.

[9] Technical Staff, Advanced Training System, User Manual, RIMS Corporation.

[10] Mackenzie, S., "8051 Microcontroller", Prentice Hall, NY, USA, 1995.

[11] Umrani, F.A., and Palijo, T.S., "Automation and Robotics", Naseem Book Depot, Hyderabad, Pakistan, 2004.

[12] DC Motor Control Using an H-Bridge, ITP Physical Computing [URL: http://itp.nyu.edu/physcomp/Labs/DCMotorControl] (Last Accessed on 11th August, 2008).

[13] Technical Staff, "NIDA model 500EMT Series Microprocessor Trainer Lab/Text Manual, Block 5 Basic Microprocessor Applications, Lesson 3 Ultrasonic Sensor", NIDA Corporation, October, 2005.