Autonomous Navigation and Real Time Mapping Using Ultrasonic Sensors in NAO Humanoid Robot

Suresh Kumar
Department of Computer Systems Engineering
Sukkur IBA University
Sukkur, Pakistan
suresh@iba-suk.edu.pk

Malak Majeedullah
Department of Electrical Engineering
Sukkur IBA University
Sukkur, Pakistan
malakmajeedullah.be17@iba-suk.edu.pk

Abdul Baseer Buriro
Department of Electrical Engineering
Sukkur IBA University
Sukkur, Pakistan
abdul.baseer@iba-suk.edu.pk

Rohibullah
Department of Electrical Engineering
Sukkur IBA University
Sukkur, Pakistan
rohibullah.be17@iba-suk.edu.pk

Received: 5 July 2022 | Revised: 18 July 2022 | Accepted: 20 July 2022

Abstract-Mapping is an essential and basic requirement for a mobile robot in order to be able to navigate autonomously. This paper proposes a solution for autonomous navigation and real-time mapping using the virtual humanoid robot called NAO. The robot navigates through its environment using ultrasonic sensors only and develops a 2-D map of the environment. For implementation and testing, the Webots simulator was used. It provides real-time values, modification and designing of the 3-D world arena, and plugins for other parameters control. We test autonomous navigation in differently shaped environments. The proposed mathematical algorithm allows the autonomous navigation of the robot and calculates the position of the robot and the obstacles (if any). The results indicate that the algorithm can localize the robot within the environment whereas the accuracy in localization can be increased by adding a control constant to the orientation of the robot. The results demonstrate that the algorithm is more effective in the rectangular arena than in the triangular and pentagon arenas.

Keywords-robotics; mapping; humanoid; navigation

I. INTRODUCTION

Humans can navigate through unknown areas while avoiding collisions with other humans or objects. We are, also, capable to develop maps in our cognition of the known areas, which help us navigate. If we want robots to navigate around the world as humans do, then they must be capable of autonomous navigation, avoid unnecessary contacts with objects or humans, and develop maps. For navigation in outdoor environment, mobile robots are preferred. However, humanoid robots, due to their symmetrical structure to human beings, are preferred for indoor environments. This structure eliminates many singularity constraints and enables the robots to act in an environment designed specifically for humans [1]. In other words, one can say that humanoid robots can take the place of humans in many workspaces as these days robots are performing very well in assisting humans in health, education, research, etc. [2].

For a humanoid robot to perform a task successfully in a real-world environment, stable dynamic biped locomotion is required [3]. Also, in order to work efficiently, a robot must know about its environment through maps, its own and obstacles' location in the environment. However, due to limited sensing capabilities, it is challenging for a robot to localize itself and detect obstacles in its environment. GPS has been used to determine the location of the robot, however, the GPS does not work in indoor environments or produces an error in location due to the absence of line of sight or reflected signal [4]. Furthermore, other sensors are required to detect the objects/obstacles in the environment. By adding more and more sensors, the complexity of the algorithms increases, adding delay in real-time response during navigation. In this paper, we propose an algorithm for obstacle detection and autonomous navigation and mapping of a humanoid robot by using ultrasonic sensors. For this purpose, we used a virtual humanoid, the NAO robot, for our experiments [5]. NAO is a humanoid robot with 25 degrees of freedom, dual vision cameras, force sensors on its feet, touch sensors in the head and arms and a pair of ultrasonic sensors on its chest. Apart from that, the robot has internal temperature and position sensors to detect joint motors' temperature and joint positions. The official vendor of the robot provides a simulator, Choregraphe, for the robot. However, for the experiments, we have used a third-party open-source simulator, Webots [6]. In comparison to Choregraphe, Webots provides extra features such as adding obstacles in the environment and tracking the odometry of the robot. The simulator also contains motion libraries to access robot joints and move them through high-level commands. During the motion, the motion library of the robot also controls physical the stability of the robot, thus, the robot does not fall

Corresponding author: Suresh Kumar

while walking. The robot can be programmed to reach a specific point in its environment by walking, in its frame of reference [7].

The purpose of this study is to propose a mathematical algorithm for the NAO robot to detect objects using ultrasonic sensors and localize the detected objects in the environment and localize itself using odometry. Furthermore, the algorithm creates a 2-D map of the space/environment in which the robot navigates.

II. LITERATURE REVIEW

Different researchers have worked on the mapping of NAO humanoid robots using the fusion of the different sensors, i.e. IR sensors, RGB camera, tactile sensors, and ultrasonic sensors [8]. Authors in [9] proposed a method for autonomous navigation of the NAO robot using a fusion of visual and robot odometry. Their approach improves robot pose estimation using the fusion of odometry compared to robot odometry. However, fusion makes the algorithm computational expensive. Furthermore, the experiment was conducted in a rectangular environment where the robot followed a predefined path, either straight or L-shaped, only, thus limiting the environment and walking length and direction. Authors in [10] demonstrated the navigation of indoor environment by NAO robot using robot's odometry. They used the vision of the robot only to confirm the target position and adjust in pose at any other location than the target position. The robot was tasked to reach for a door that contains a specific symbol. The robot navigates straight using odometry until it reaches the target. Although the experiment was performed using a real robot, however, it was performed in a controlled environment where no active object was present. This limits the capability of the algorithm to navigate and map in an environment with active objects present. Authors in [11] proposed a fuzzy logic-based obstacle avoidance algorithm for the virtual NAO robot. The algorithm decides the orientation and walking speed of the robot based on the data from ultrasonic sensors. They performed the experiment using a virtual robot in Webots simulator, as we do in this study. However, the scope of their study is limited to obstacle avoidance only, whereas our study focuses on autonomous navigation 2-D mapping, which contains obstacle avoidance as a part of the algorithm.

Apart from humanoid robots, mobile robots also have been used for testing algorithms for autonomous navigation and obstacle avoidance. Authors in [12] designed a fuzzy logic control system to steer a mobile robot for autonomous navigation and obstacle avoidance. However, they considered a simple two-wheeled mobile robot, which reduces the complexity of motion. Furthermore, the authors used simulated sensors to detect the position and angle of the obstacles with respect to the robot.

The above-mentioned studies either use a fusion of multiple sensors [9, 10] or limited scope of the robot is considered using ultrasonic sensors [11, 12]. Furthermore, using multiple sensors increases the complexity in getting data and computational cost. Also, dependency on many sensors can cause failure while performing a task, whereas the battery life is affected adversely [13].

III. METHODOLOGY

Two ultrasonic sensors, along with the ultrasonic transmitters, are placed on the chest of the robot. The sensors are in a distance of 0.0832m (8.32cm) apart from each other. Each sensor is able to detect 40KHz frequency with an effective cone angle of 60^o and range from 0.25m to 2.55m [8]. The simulated robot is able to detect any object within this cone angle and range. It returns the distance of the object from the sensors, irrespective of its position. If there is no object within the range and cone angle, then the simulator returns the maximum distance, i.e. 2.55m. Since the two ultrasonic sensors return the distance from the object, thus it is challenging to locate the object within the space.

The proposed autonomous navigation algorithm lets robots make movements autonomously based on the sensor data. If the sensors detect any object at distance less than 0.8m, then the robot makes a turn, otherwise, it makes a forward move. The robot makes a left or right turn if an object is at a distance less than 0.8m on either the right or the left side respectively. If both sensors detect an object at a distance less than 0.8m then the robot makes a u-turn, i.e., a 180^o turn. The distance, 0.8m, is selected for the robot's safety purpose as it should not strike the object while extending its arms. Figure 1 shows the flow diagram of the autonomous navigation.

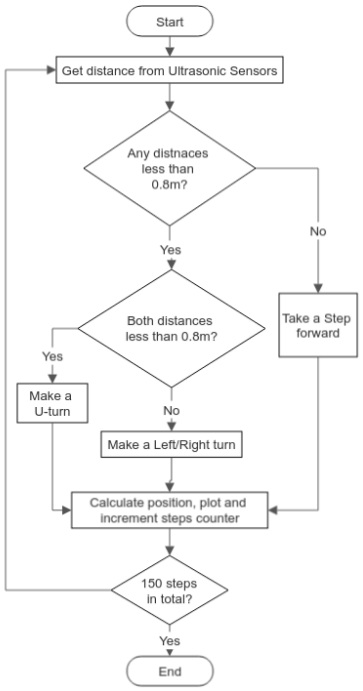

Fig. 1. Flow diagram of the autonomous navigation.

In order to locate the object, we assume that both sensors detect the same object at the same distance. In Figure 2, considering "O" as the object position, with respect to the center of the robot, L and R are the distance between the object and the robot detected by the sensors on either side and "D" is the distance between the object "O" and centre of the robot is given by (1):

$$D_L = \sqrt{L^2 - 0.0416^2}$$
$$D_R = \sqrt{R^2 - 0.0416^2} \quad (1)$$
$$D = \frac{D_L + D_R}{2}$$

where D_L, D_R the object's distance from the left and right sensor and D the object's distance from the robot.

The constant, 0.0416, in (1) is the distance (in meters) from the center of the robot to the center of the ultrasonic sensor on either side of the chest. In order to locate the object on the map, we need to consider the current position of the robot and its distance from the object. It should be noted that the robot is initially placed along the X-axis and Z-axis is on its left side whereas the Y-axis is perpendicular to the robot. In order to determine the position of the detected object, we use the distance calculated by both sensors and then add it to the existing position of the robot. The group (2) of equations provides the calculation details.

$$\begin{aligned} OL_x &= R_x + D_L \cos(R_0) \\ OR_x &= R_x + D_R \cos(R_0) \\ OL_z &= R_z + D_L \sin(R_0) \\ OR_z &= R_z + D_R \sin(R_0) \\ O_x &= Average(OL_x + OR_x) \\ O_z &= Average(OL_z + OR_z) \end{aligned} \quad (2)$$

where OL_x, OR_x the object's x coordinates with respect to sensors, OL_z, OR_z the object's z coordinates with respect to sensors, R_x, R_z, R_o the robot's x, z coordinates, initially 0, and O_x and O_z the object's x, z coordinates.

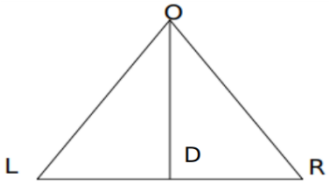

Fig. 2. Object localization using ultrasonic sensors.

The calculation is only made when the object is at a distance less than 0.8m. Objects beyond this range are ignored. For an object to be considered in the map, it must be detected by both sensors and ignored if detected by a single sensor. A nearby object is likely to be detected by both sensors.

In order to calculate the position of the object, we must know the current position of the robot. The robot's new x and z coordinates after making a motion, i.e. one step, left turn, right turn, or u-turn are given by:

$$R_x = (R_x) + S_s \cos(R_0 + \theta) \quad (3\alpha)$$

$$R_z = (R_z) + S_s \sin(R_0 + \theta) \quad (3b)$$

where S_s is the step size of the robot and θ the current angle, *+ve* for left anf *–ve* for right.

The position of a robot in its environment's Cartesian system is usually expressed in the position coordinates (x,y), and the orientation θ [14]. Initially, the robot's orientation and position in x and z (instead of y here) are 0. We conducted an experiment for calculating the step size and the angle of the robot when it walks straight or makes a turn as we observed deviation from the GPS position provided by the simulator and the actual location of the robot in the environment. We experimentally found the step size and effective angle of the ultrasonic sensors for the calculations in our algorithm.

IV. EXPERIMENTS AND RESULTS

In order to observe the robot's motion and behavior in the simulation, we controlled the robot manually by sending commands through a keyboard. First, we calculated the deviation in forwarding motion as mentioned in [10], hence, we designed an arena without boundary walls and obstacles, and allowed the robot to move straight forward on the x-axis only. After running the simulation for 150 steps, we observed that the robot did not follow the straight line but deviated from the desired path and the average deviation was 0.1856^o per step. The deviation was calculated by taking the difference between the initial value of the inertial unit, yaw, and the final value. The deviation did not form a regular pattern but was random at every stage. One can get a different deviation value by running the simulation for more than 150 steps.

A. Step Size Experiment

The distance covered by the robot depends on the step size of the robot. We did an experiment to calculate the step size, which is mentioned in the reference manual as 0.088m. While performing the step size experiment, the initial coordinates were noted down, and then we sent the move forward command through the keyboard and noted down the final coordinates. The change in the distance gives us the step size of the robot and was found to be 0.086553m per step. The average of 150 steps was found to be 0.088m as claimed by the reference manual of the robot.

B. Rotation Angle Experiment

NAO robot can make a turn on either its left or right side depending upon the information received from the sensors. These turns are made using specific motion files designed by the developer in which the joints and their values, which are changed during the turn, are defined. In this experiment, we only considered 3 angles for the turns: 60^o turn to the left side, 60^o turn to the right side and 180^o turn to the right side. The deviation in the 180^o rotation was found to be 12^o, which means the robot rotates by 192.56^o rather than 180^o while making a U-turn. For 60^o rotation, the deviation was found to be 3^o in both left and right turns. The rotation performed was at 57^o rather than 60 degrees. The deviations can be observed in Figure 3. We tested the algorithm for autonomous navigation, localization and 2-D mapping in 3 different arenas. In the experiments, the robot position provided by the simulator GPS and the object position, set by the user, were considered ground truths. The graphs were plotted with calculated positions against the ground truths. The autonomous navigation was limited to 150 steps. The measurements on axes in all the Figures are given in meters.

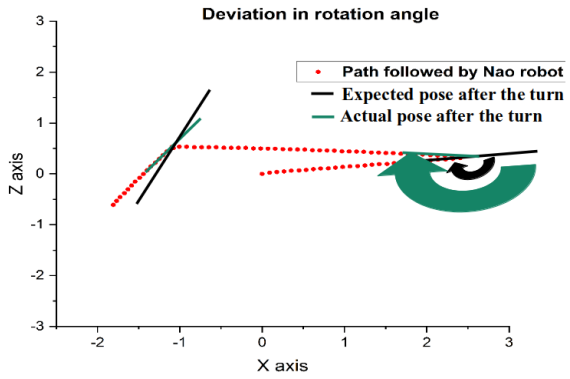

Fig. 3. Deviation in the angle while making turns.

C. Rectangular Arena

The rectangular arena was created with four walls, whereas the height of the walls was kept the same to that of the robot. The arena is 2.5m×2.5m area in size. The robot started from different points, i.e. the upper right corner, the lower left corner, and the center to observe if there were any changes. No significant change was observed in the results, thus we considered the center point as the starting point of the navigation. Figure 4 shows the simulated rectangular arena and the robot. The results of autonomous navigation and localization are plotted in Figure 5. The grey color shows the calculated path using the proposed algorithm and the red color path shows the simulator's GPS path followed by the robot. The green color represents the wall bounding the robot in the rectangular arena and the blue points on the map show the parts of the wall detected by the ultrasonic sensors of the robot as objects and plotted on the map with respect to the robot's position using the proposed algorithm. It is evident that the robot's position and orientation are very accurate while moving in a straight direction. However, a deviation can be seen after the robot is taking turns. This confirms the results shown in Figure 3. We found that the deviation varies in each turn. In order to remove this error, we calculated an average of the deviations of the turns the robot made during the 150 steps. The results improved after considering this offset in turns, as shown in Figure 6.

Fig. 4. The robot in the simulated rectangular arena.

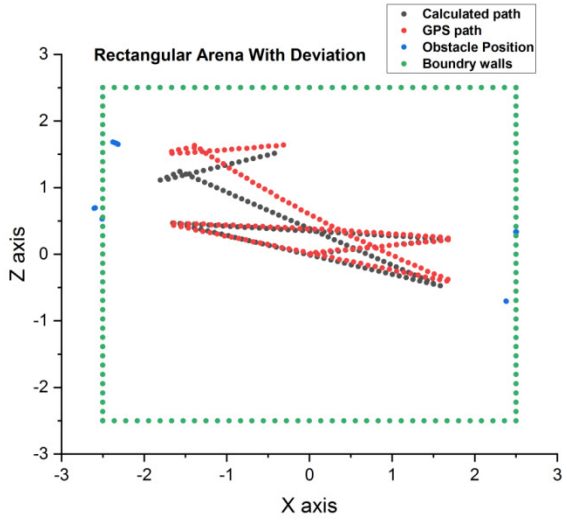

Fig. 5. Autonomous navigation in a rectangular arena.

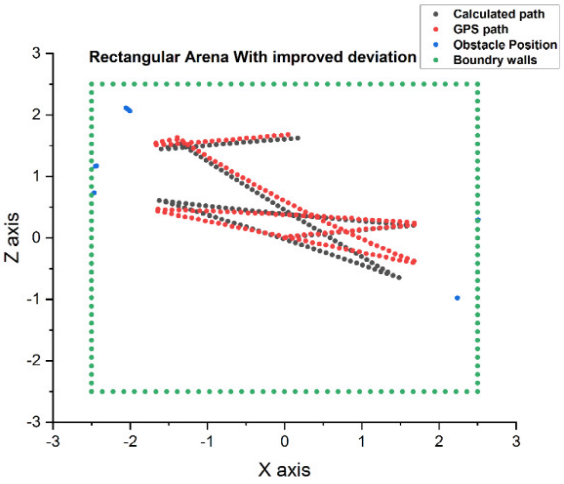

Fig. 6. Autonomous navigation after adding an offset.

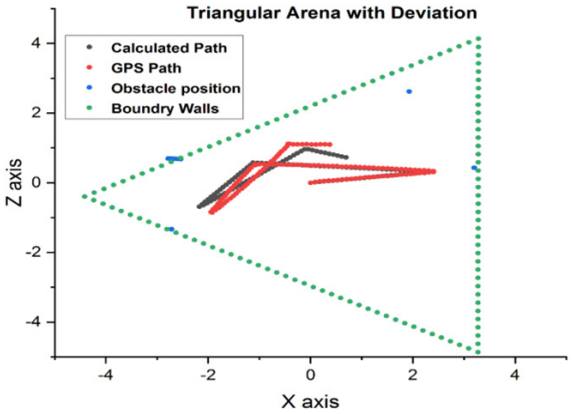

Fig. 7. Autonomous navigation in the triangular arena.

D. Triangular Arena

The experiment was repeated for the same number of steps in a triangular arena. The arena's area size was kept almost the same as that of the rectangular in the previous experiment.

Figure 7 shows the results obtained while navigating through the triangular arena. Figure 7 shows that the calculated path and GPS path of the robot has no change while moving in a straight direction. However, again, deviation occurred when the robot made turns.

E. Pentagon Arena

The arena of the third experiment was pentagon-shaped. In this arena, the robot was also allowed to autonomously navigate up to 150 steps in total. The results are shown in Figure 8.

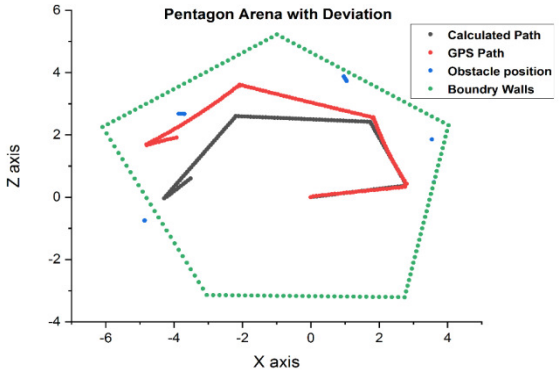

Fig. 8. Autonomous navigation in the pentagon arena.

Again, it is evident that the paths calculated through the algorithm and provided by the GPS are almost the same. However, deviation can again be observed after the robot takes a turn. The deviation kept increasing as the robot took more turns.

F. Adding a Kalman Filter

The Kalman Filter is a generic algorithm that is used to estimate a system variable while observing the measurement over time [15]. It is simple, consumes very small computational power, and is used with inaccurate or noisy measurements to estimate the state of that variable or another unobservant variable with greater accuracy. As the NAO robot moves, it encounters some deviations which do not form a regular pattern and hence in the long run will produce more deviations against the GPS values. We used a simple Kalman filter for one dimension which is very effective in computational cost and does not need any memory except the previous state.

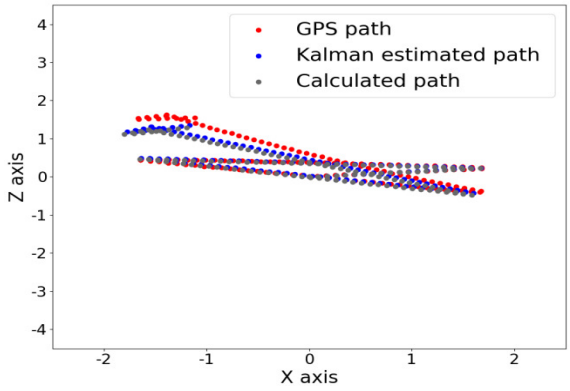

Fig. 9. Autonomous navigation using the Kalman filter.

The modified algorithm, with the Kalman filter, was tested in the rectangular arena. Figure 9 shows the results of this experiment. It is evident that the position calculation is not improved significantly after using the Kalman filter.

V. CONCLUSION

Using the Webots simulator, we designed a 2-D real-time mapping for a virtual NAO humanoid robot in different arenas using only ultrasonic sensors. All the graphs were plotted while the robot was navigating autonomously through the arena and were updated after each step. The results show that our algorithm is very simple in its calculations, yet it determines a more accurate position than the one provided by the GPS position of the simulator. Some deviation occurred in the position and orientation of the robot from the GPS values as the robot made turns. In Figure 3, we demonstrated that the robot is unable to make the turn at an exact angle and hence deviation is produced from the expected position.

The results demonstrate that the algorithm is able to localize the robot and obstacles in the environment while autonomously navigating through it. This localization is more accurate in the rectangular arena than in the triangular and pentagon arenas, maybe due to the acute angles and more angles in triangular and pentagon arenas respectively. However, further experiments may need to be conducted to build concrete reasoning.

We used the Kalman filter for one dimension variable to improve the results. However, it was evident that the filter did not significantly improve the results. The reason behind the negligible improvement from the Kalman filter is possibly the very small deviation in the calculated path compared to the simulator's GPS path. Furthermore, our experiment was limited to 150 steps. The filter may be effective for a higher number of steps. Anyhow, we do not consider the Kalman effective in our case.

Knowing the exact position of a robot, obstacles, computational cost, backup, and battery life are crucial for a robot to work in any environment. Hence we used only one sensor, the ultrasonic sensor, thus reducing computational cost and increasing battery life. This algorithm can also work as a backup if other sensors, such as the camera, fail. In the future, the work can be extended to improve the robot positioning further by improving the Kalman filter and its gain. Furthermore, the study will be extended to perform the experiments with real robots.

REFERENCES

[1] K. P. Tee, R. Yan, Y. Chua, and Z. Huang, "Singularity-robust modular inverse kinematics for robotic gesture imitation," in *2010 IEEE International Conference on Robotics and Biomimetics*, Tianjin, China, Sep. 2010, pp. 920–925, https://doi.org/10.1109/ROBIO.2010.5723449.

[2] A. Choudhury, H. Li, C. M. Greene, and S. Perumalla, "Humanoid Robot-Application and Influence," *Archives of Clinical and Biomedical Research*, vol. 2, no. 6, pp. 198–226, Dec. 2018, https://doi.org/10.26502/acbr.50170059.

[3] N. L. A. Shaari, M. R. B. Razmi, M. F. Miskon, and I. S. M. Isa, "Parameter Study of Stable Walking Gaits for Nao Humanoid Robot," *International Journal of Research Engineering and Technology*, vol. 2, no. 9, pp. 16–23, Aug. 2013.

[4] O. M. Mubarak, "The Effect of Carrier Phase on GPS Multipath Tracking Error," *Engineering, Technology & Applied Science Research*, vol. 10, no. 5, pp. 6237–6241, Oct. 2020, https://doi.org/10.48084/etasr.3578.

[5] "NAO the humanoid and programmable robot," *SoftBank Robotics*. https://www.softbankrobotics.com/emea/en/nao.

[6] "Webots: robot simulator," *Cyberbotics*. https://cyberbotics.com/.

[7] "Cartesian control — Aldebaran 2.1.4.13 documentation," *Aldebaran*. http://doc.aldebaran.com/2-1/naoqi/motion/control-cartesian.html.

[8] "Sonars — NAO Software 1.14.5 documentation," *Aldebaran*. http://doc.aldebaran.com/1-14/family/robots/sonar_robot.html.

[9] Š. Fojtů, M. Havlena, and T. Pajdla, "Nao Robot Localization and Navigation Using Fusion of Odometry and Visual Sensor Data," in *Intelligent Robotics and Applications*, Montreal, Canada, Oct. 2012, pp. 427–438, https://doi.org/10.1007/978-3-642-33515-0_43.

[10] C. Wei, J. Xu, C. Wang, P. Wiggers, and K. Hindriks, "An Approach to Navigation for the Humanoid Robot Nao in Domestic Environments," in *Towards Autonomous Robotic Systems*, Oxford, UK, Aug. 2014, pp. 298–310, https://doi.org/10.1007/978-3-662-43645-5_33.

[11] O. Melinte, L. Vladareanu, and I.-A. Gal, "NAO robot fuzzy obstacle avoidance in virtual environment," *Periodicals of Engineering and Natural Sciences (PEN)*, vol. 7, no. 1, pp. 318–323, Apr. 2019, https://doi.org/10.21533/pen.v7i1.359.

[12] B. Kasmi and A. Hassam, "Comparative Study between Fuzzy Logic and Interval Type-2 Fuzzy Logic Controllers for the Trajectory Planning of a Mobile Robot," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 7011–7017, Apr. 2021, https://doi.org/10.48084/etasr.4031.

[13] S. Joshi and S. Talole, "An overview of energy systems in humanoid robots (Journal of Microelectronics and Solid State Devices)," *Journal of MIcroelectronics and Solid State Devices*, vol. 7, no. 3, pp. 12–19, 2020.

[14] H. Medjoubi, A. Yassine, and H. Abdelouahab, "Design and Study of an Adaptive Fuzzy Logic-Based Controller for Wheeled Mobile Robots Implemented in the Leader-Follower Formation Approach," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 6935–6942, Apr. 2021, https://doi.org/10.48084/etasr.3950.

[15] J. Humpherys, P. Redd, and J. West, "A Fresh Look at the Kalman Filter," *SIAM Review*, vol. 54, no. 4, pp. 801–823, Jan. 2012, https://doi.org/10.1137/100799666.