General Concept of Guide-Dog Robot Assistant Development for Blind and Visually Impaired People Using Webot Simulator

Sarerusaenye Ismail

Taylors University Lake Campus Subang Jaya, Malaysia

DOI: https://doi.org/10.51244/IJRSI.2026.1313CS021

Received: 05 July 2026; Accepted: 10 July 2026; Published: 25 July 2026

ABSTRACT

Nowadays, many industries are reducing human labor by integrating robotic systems to improve productivity and lower operational costs. For individuals with visual impairments, service animals such as guide dogs require continuous care and maintenance, creating a need for alternative assistive technologies. This study presents the development of a guide-dog robot prototype designed to resemble a real guide dog, equipped with basic navigation capabilities and intelligent behavior. The system is developed and simulated using Webots to explore appropriate artificial intelligence techniques and fundamental robotics knowledge. The robot navigation system is implemented using a supervised feedback learning approach based on rule-based logic (if–else statements) for path searching and obstacle avoidance. The project aims to provide knowledge sharing for students and academicians on effective methods for developing assistive robotic systems. During testing and implementation, several challenges were identified, including inaccurate distance sensor readings, blind spots caused by limited sensor coverage, and camera processing delays. To address these limitations, future improvements may include applying simple filtering techniques, such as averaging multiple sensor readings to reduce noise and enhance obstacle detection accuracy. Additionally, integrating distance sensor data with camera input can minimize collision risks by improving environmental perception. Simplifying image processing algorithms and reducing camera resolution may further decrease computational delay and improve real-time responsiveness. This study demonstrates the feasibility of developing a rule-based guide-dog robot simulation while highlighting practical challenges and potential enhancements for future research in assistive robotics.

Keywords: Supervised algorithms, Webots, autonomous mobile robot, obstacles avoidance, image-processing, pre-defined path, FSM.

INTRODUCTION

Many people experience visual impairment at some point in their lives, which can severely limit their activities. They may find it difficult to leave the house without assistance or require a companion. In some public places particularly in Malaysia, regulations prohibit bringing dogs inside. While dogs provide companionship, comfort, and assistance in work, sports, and shows, they are banned in certain countries, especially Islamic ones, because many Muslim scholars consider dog saliva ritually impure and recommend washing seven times after contact (Jabatan Mufti Negeri Perlis, 2026). Dogs also require long training periods and typically live only about ten to eleven years. To address these challenges, this research proposes a guide robot to assist visually impaired people using Webots, an open-source cross-platform robot simulator. Robot controllers in Webots can be programmed in several languages, including C, C++, Java, Python, MATLAB, and RUBI.

LITERATURE REVIEW

Feedback is an essential component of scaffolding for learning. Feedback provides insights into the assistance of learners in terms of achieving learning goals and improving self-regulated skills (Cavalcanti, A.P et al.,2021). The study implements a supervised feedback learning framework grounded in rule-based machine learning, where predefined logical rules are used to direct the robot's navigation and behavioral responses. In

INTERNATIONAL JOURNAL OF RESEARCH AND SCIENTIFIC INNOVATION (IJRSI)
ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue XIII July 2026
Special Issue on Emerging Paradigms in Computer Science and Technology

supervised learning, tasks are broadly categorized into classification and regression. Classification involves predicting discrete and categorical output labels. This means that when the computer is fed with a dataset as input to learn, it must be supplied with the correct answer-sheet that contains the possible output of every instance of the input dataset. In ML, the input data to the computer is known as the features and the answer-sheet is termed the output labels/targets (Choudhury et al., 2022). This algorithm may apply to the robot development such as for autonomous mobile robot. Due to the increasing population of visually handicapped people, guide dogs have shown an enormous demand in recent years. However, few guide dog robots have been discovered to service visually handicapped people worldwide (Hong,B., et al., 2022). Guide dog robots have the potential to significantly improve mobility and safety for the visually impaired, with continued research driving their advancement (Wang, Wei et al, 2025). For example, the first guide dog robot development is 'MelDog' who started develop in 1977 where the objective was to enhance mobility aids for the blind by providing them with the functions of guide dogs such as obedience in navigating a blind master, intelligent disobedience in detecting and avoiding obstacles in his or her path, and well-organized man-machine communication which does not interfere with his or her remaining senses (American Heritage,2026). Key design features of this type of robot include a movable platform, human-machine interface, and capability of avoiding obstacles. In other research, a novel algorithm enabling the robot to recognize its follower's locomotion as well to detect the center of corridor has been proposed and implemented in the robot's human-machine interface (Saegusa, Shozo et al,2010). Guide dogs enable remarkable autonomy and mobility for their handlers. However, only a fraction of people with visual impairments has one of these companions (University of Texas College of Natural Sciences, 2024). The guide dog robots not only guide the visually impaired person safely to their destination, but also providing explanation services of street scenes, enabling the visually impaired person to enjoy visual scenes (Ichikawa,R. et al, 2023). A small robot that could guide humans with a leash could potentially solve such an issue. The ability of the leash to become slack allows the robot to change the internal dimensions of the human-robot system and thus allows the robot to guide the human through narrow spaces, such as a doorway. However, utilizing a leash could involve a hybrid system switch, like for example the leash could be taut or slack, which makes this motion planning more challenging (Xiao et al., 2021). Most previously developed systems for guiding blind individuals are based on rigid robotic arms that detect obstacles and help humans to circumvent them. Researchers at the University of California Berkeley's Hybrid Robotics Group have recently created a quadrupedal robot with a leash that could take on the role of a guide dog (Phys.org,2021). This robot, presented in a paper pre-published on arXiv, can help humans to safely navigate indoor environments without crashing into objects, walls and other obstacles. On the other hand, this research paper discovered the challenges of Guide-dog robot using the simulator and discussed the insightful findings for students' feasibility study.

METHODOLOGY

This research has three components; (1) process to avoid the obstacles, (2) set up the robot equipment's, (3) location detection, and (4) proposed the scenario. Firstly, research created and designed an environment that looks like a city environment that consists of few buildings and few trees, and the 'Mybot' robot has been selected from category of robots in Webots to represent a 'Guide-dog' robot. Figure 1 shows the steps in how all the bounding objects in this environment have been set up.

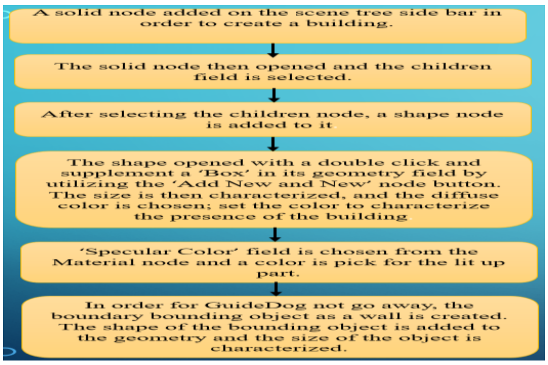

Figure 1: City environment set up via Webot

Next, the detection instruments need to be set up by using a few steps because this is the only way the Guide-dog may be able to detect any nearby obstacles. Here, two cameras and two distance sensors are used where the camera is used to detect the object and able to calculate the distance between the obstacles and robots. Figure 2 shows the attributes for camera and distance sensor set up.

CAMERA Scale: x=1m; y=1m; z=1m Field of view: 1.047 Width: 100 Height: 100	SIZE OF THE ROBOT Scale: x=8m; y=8m; z=8m DISTANCE SENSOR Scale: x=1m; y=1m; z=1m

Figure 2: Attributes of camera, robot body and distance sensor

There are possible maneuvers that the programmer should consider; (1) the Guide-dog robot walks towards the desired building by analyzing the environment through its cameras, (2) there is an obstacle in the middle of movement. Next, Guide-dog robot uses two distance sensors to avoid the obstacles by wheels out or distance itself from the object, and lastly (3) the Guide-dog robot rotates slightly, so that it can proceed towards the building without bumping on the objects. Next, the prototype of environment or layout is simulated, and four different color paths have been chosen. The goal of Guide-dog robot is to detect a spot of color that is used as a path for the robot to identify and reach the destined location. Thus, to achieve that, Guide-dog robot required analyze the image pixels of the center of the footage recorded through the cameras and sums up the color of the components. If one color component is represented more than the other colors, therefore the destined path location is identified. This is the implementation method where research utilizes multi-color detection using RGB color sensors. If there is no correct destination detected, the robot will keep rotating until it finds the correct path. This Guide-dog robot simulates designed to follow a path such as the development by Rajee et al(2024), however the path had marked by three different RGB colors on a contrasting white surface, demonstrating its ability for self-guided movement. Equipped with an array of Infrared Ray (IR) sensors, the robot detects the line, which enables it to maintain its designated course. These sensors enhance both the precision and flexibility of its navigation (Rajee, A. et al, 2024). Here, once the correct path has been detected, Guide-dog robot needs to stop at the current path position, then it captured and stored the image and sent the output or statement that it has found thru the command prompt. Finally to simulate and test the Guide-dog robot, a few scenarios have been created for the analysis and testing purposed as the following conditions:

a. There is a blind person in the city who is waiting to go to a particular place.

b. But as he or she is not familiar with the place, thus he or she cannot go any further.

c. It is also a place or spot where the Guide-dog robot is not allowed to move.

d. The Guide-dog robot is sent to the person who is visually impaired.

e. The person holds on to the Guide-dog robot and walks towards the destined location.

In 2007, the definition of simulation according to the Pearson education means to try to duplicate the characteristics of a real system (Liliana I-F et al, 2025). Notwithstanding mirroring procedures to perceive how they act under distinctive conditions, reenactments are likewise used to test new speculations (Swol, Lyn, 2003). In the wake of making a hypothesis of causal connections, the scholar can arrange the connections as a machine program. Ahmed, F.K et al (2024) reported that on the off chance that the project then carries on in the same path as the genuine methodology, there is a good chance if the proposed connections are right while the most important aspect is the correct decision modelling techniques implementation. It should be used to stimulate, and it must be a clear cut of view on what the model is up to. Below Figure 3 shows the functions of stimulation in order.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue XIII July 2026
Special Issue on Emerging Paradigms in Computer Science and Technology

Figure 3: Supervised feedback learning steps for Guide-dog robot (self-drawn)

The reason why simulation is used is because of its flexibility and it can handle large and complex systems. This does not interfere with the real system but allows the study of interaction among variables (Wei,C.L et al.,2022). Simulation is generally divided into two components: (1) the discrete event simulation and, (2) the continuous systems simulation. However, according to Francois E. Cellier, combined system simulation is also a type of simulation which has the properties suitable to continuous system simulation and discrete event simulation, although the term is not well understood (Joshua D.C & Gautam B.,2016). Discrete Event Simulation (DES) is the methodology of arranging the conduct of a complex system as a requested arrangement of overall characterized occasions. In this setting, an occasion contains a particular change in the system's state at a particular point in time. Continuous system simulation are systems depicted by a settled set of differential comparison with state variables and first state induction both being consistent over the entire recreation run. Another hands, in this research paper, programmer needs to understand about robot system which the rationale behind it such as to characterize the grouping of robot movements and gear operation. This research is composed of an individual acquainted with robot sort, the robot cell which the parts to be handled and the operations to be performed. Figure 4 shows the libraries required in Webots simulator. It shows the header file imported such as webots/robot.h, webots/differential_wheels.h, webots/camera.h, webots/robot.h, webots/distance_sensor.h, and stdio.h are come from the file name "js.h". The SPEED the value is 100 so the robot can run with fast enough and the TIME_STEP is 64.

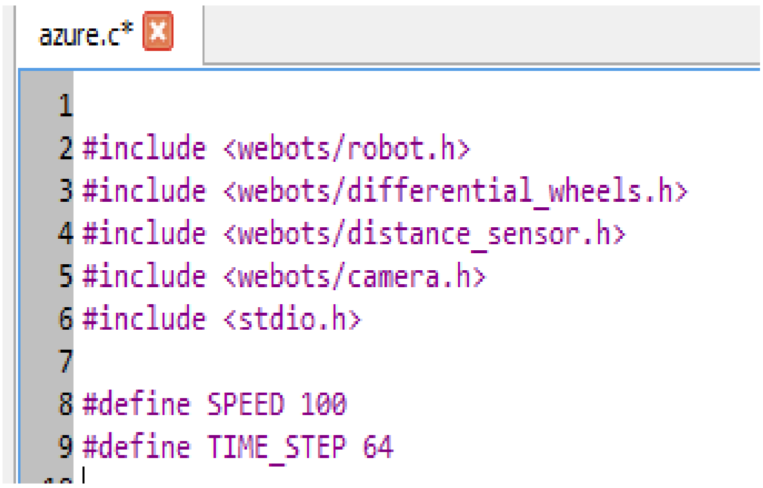

Figure 4: The libraries of Guide-dog robot using Webots

Next, Figure 5 shows the declaration of the distance sensors and the camera. "ds0" as the name of the first distance sensor, "ds1" as the second distance sensor, and "camera" to represent the two camera. This declaration used to access the device and enable it. And the last 2 codes it used for storing the height and width of the camera location on the robot's body.

```
WbDeviceTag ds0 = wb_robot_get_device("ds0");
WbDeviceTag ds1 = wb_robot_get_device("ds1");
wb_distance_sensor_enable(ds0, TIME_STEP);
wb_distance_sensor_enable(ds1, TIME_STEP);

camera = wb_robot_get_device("camera");
wb_camera_enable(camera, TIME_STEP);
width = wb_camera_get_width(camera);
height = wb_camera_get_height(camera);
```

Figure 5: Example source of program for camera and distance sensors set up

In previous discussion, the distance measurement and detection medium from the distance sensors have been addressed. Here, the distance sensor node for the Guide-dog robot allows the robot to move without colliding with walls, barriers, trees and other robots. It is bound with an 'INFRARED' node which is to detect distance by using Infrared ray. There is two single ray set to each distance sensor and one of the rays functions is to detect a wider angle of the robot's distance.

```
    if (pause_counter > 0)
      pause_counter--;

    if (pause_counter > 10) {
       left_speed  = 30;
         right_speed =  -10;
    }

    else if (pause_counter > 0) {
         left_speed  = 30;
       right_speed =  -10;
    }

    else {

      red = 0;
      green = 0;
      blue = 0;
```

Figure 6: Example of program for camera and distance sensors set up

Figure 6 shows the code that is used for path detection. When the path has been detected, Guide-dog robot in the 'waiting' pose until pause_counter condition is decremented enough. If the path has been found quite recently, the Guide-dog robot starts to turn around but it does not analyse the image or footage. Then, in the right position based on this rule-based statement such as the left and right wheels have been calculated, the Guide-dog robot may do the turning and analysing the camera's image or footage to find the destination (prompt from the command function in controller). Figure 7 shows the code to analyse the image or footage from the camera where the goal is to detect a path (a spot of color) of pre-defined condition in the middle of the screen. To achieve that, it simply parses the image pixels of the centre of the image (current_blob) and sum the color components (Red, Green and Blue).

```
    for (i = width/3; i < 2*width/3; i++) {
      for (j = height/2; j < 3*height/4; j++) {
        red += wb_camera_image_get_red(image, width, i, j);
        blue += wb_camera_image_get_blue(image, width, i, j);
        green += wb_camera_image_get_green(image, width, i, j);
      }
    }

    if ((red > 3*green) && (red > 3*blue))
      current_blob = RED;
    else if ((green > 3*red) && (green > 3*blue))
      current_blob = GREEN;
    else if ((blue > 3*red) && (blue > 3*green))
      current_blob = BLUE;
    else
      current_blob = NONE;
```

Figure 7: Code of camera to detect the colors comparison using RGB

Figure 8 shows the prototype layout of a city that includes many components such as buildings, different types of trees, oil barrels, speed limit sign, pedestrian crossing, round table, round table, floor light, chair, potted tree and others. There are a few blocks such as the building, floor decoration set as boundary objects, walls and obstacles. For example, the Cyber tower is circled by green color to direct the robot following the path, the red path as the way to Museum and blue path as the way to Hotel. The Guide-dog robot will not be able to go through the building as there is a distance sensor installed inside the body. The green, blue and the red floors are the path colors which the robot will be able to detect through its camera. After detecting the path colors, the robot will move towards the building following the indicates path.

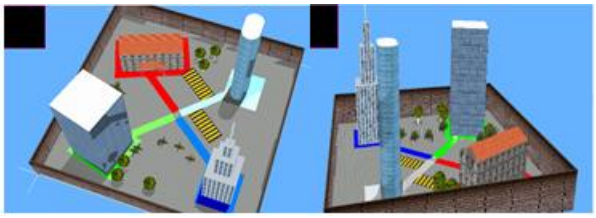

Figure 8: Protoype of proposed environment (self-drawn)

Figure 9 shows the code for the two distance sensors which when Guide-dog robot facing any obstacles such as a wall, boundary object, or buildings, it automatically moves to backward direction and turns to another direction to avoid the collusion. It turns proportionally following the sensors' value because the closer Guide-dog robot from the wall, the more the robot needs to turn and distance the position from the obstacles. The distance between Guide-dog robot and the detected obstacles can be measured around 300 to 500 meters. For example, in the situation that Guide-dog robot has been set to go to Cyber tower which this Cyber tower building has been pre-defined using green path of color. If the path with the green color reads is 'NONE' in the indicator, it means the robot will keep turn until the robot find another path with the green color. If it is red or blue color of path, Guide-dog robot stops for three to four seconds and analyse the environment before detecting the location of current place where it stands on unlikely, if the robot detects the green path, it will stop and detect the place destination then continue with walking and moving to the destination, Cyber tower. This method applied to others destination such as if the Guide-dog robot has been instructed to go to Museum or Hotel.

```
    if (ds1_value > 500) {
        if (ds0_value > 500) {
            left_speed = -SPEED;
            right_speed = -SPEED / 2;
        } else {
            left_speed = -ds1_value / 10;
            right_speed = (ds0_value / 10) + 5;
        }
    } else if (ds0_value > 500) {
        left_speed = (ds1_value / 10) + 5;
        right_speed = -ds0_value / 10;
    }
```

Figure 9: Code to set the distance sensors to avoid the obstacles

ANALYSIS AND RESULTS

This research implementation uses the collision avoidance algorithm of supervised feedback learning. It detects distance between the robot and the object nearby. If the distance sensor detects Guide-dog robot is too close to the obstacles, then the robot will then move backward and slightly turn to different directions. The wheels both turn backward for the robot to move backward. To decide which side to turn, the wheel for the sides (left_speed and right_speed) spin in backward direction following the proportion values in the controller. For example, if the robot needs to turn left then the left wheel will need to spin backward and the right wheel will need to turn forward. Guide-dog used the supervised algorithm because there is a controller. The wheels for the robot will receive command from the controller where the user input the direction that the robot should go. Camera function transfers the image recorded and display it in the provided box. User can see what visual the robot is getting on and what is the object in front of it. During testing phase, Guide-dog robot moves around finding the three colors path (red, blue and green). It must detect the place that has been set through the command prompt by user. If the item picker cannot find the place, then it cannot receive another new command until the first place is reached. There are a few flaws research found from the testing and implementation phase. Guide-dog robot sometimes lose control due to too many obstacles in many sides around the environment as shown in Figure 10. This will lead Guide-dog stuck in the corner and couldn't move out or do the turning either to backward or in forward direction.

Figure 10: The lost position of Guide-dog

With only two distance sensors, it may be unable to do the perfect turning because of a few factors such as narrow field of view, no rear sensing and large bind zones. In other words, Guide-dog robot unable to move out in this position and failures to complete the given instructions. On the other hand, this research paper also found that with only two distance sensors both face forward and are mounted at small angles, then Guide-dog robot cannot detect walls or edges on the side or back, thus once it reached a corner or edge, it does not know how to escape. To fix this problem, Guide-dog robots require side or rear distance sensors, increase sensor field of view and add at least another 3 sensors as a common that imitates the maze navigation. Besides, another problem that must be addressed is about the camera limitations where it required proper lighting, correct thresholding or color detection. It has no ability to misclassify textures and has limited frame rate because Guide-dog in this research paper used a line following algorithm and path detection. It does not include the edge detection. In addition, all lighting differences or shadows also break the detection process. To fix this issue, the camera resolution needs to be adjusted and use better image filtering such as HSV instead RGB. Another issue is the physics and edge problem in Webots where the wheel friction is too high, and the edge collision geometry is incorrect. This condition only can be fixed by adjusting the wheel friction. The round wall corners also should be start by sharp 90-degree physics traps. Research encoded the small piece of program to adjust no reverse behavior to set negative velocity as the following additional code:

left_motor.setVelocity(-speed);

right_motor.setVelocity(-speed);

However, although the Guide-dog robot successfully escapes corners, in fact at certain times the turning still collides and has infinite stuck loop because it hasn't equipped a backward motion device or escapes routine program. Most robot simulators nowadays enable the program such as a random recovery movement, but it is not available in Webots simulator. In mobile robot navigation experiments conducted using Webots, minimal yet robust hardware and control configuration is essential to ensure stable and repeatable performance. An overly simplified sensing system often leads to deadlocks, oscillatory motion, or failure to recover from edge and corner conditions. Therefore, a carefully designed minimal setup can significantly improve navigation reliability without increasing system complexity excessively. A practical minimal configuration consists of three to five distance sensors strategically positioned at the front, front-left, and front-right of the robot chassis. This arrangement reduces blind spots and provides sufficient spatial awareness for obstacle avoidance and wall-following behaviors. The inclusion of angled side sensors improves corner detection and allows smoother turning transitions. While a single forward-facing camera may assist in path or line detection tasks, distance sensors should remain the primary mechanism for collision avoidance due to their lower computational cost and faster response time. From a control perspective, a finite state machine (FSM) architecture is recommended over purely reactive conditional logic. FSM, a tool in model-driven engineering, is a computational model based on a system a finite number of *states()*. The navigation controller should include and available in Webots distinct behavioral states such as forward motion, obstacle avoidance, wall following, recovery, and reverse maneuver. This structured approach enables the robot to transition intelligently between behaviors based on sensor feedback rather than relying on simple threshold comparisons. In particular, the integration of a reverse recovery state is critical for escaping corner traps and local minima situations. Additionally, implementing a wall-following strategy, such as the right-hand or left-hand rule, provides consistent directional guidance in maze-like environments. This strategy ensures systematic exploration while minimizing indecisive movements. To further enhance robustness, a timeout or stagnation detection mechanism can be incorporated. If no positional progress is detected within a specified time interval, the robot may execute a short backward motion followed by a randomized turn to escape potential deadlock conditions. Mechanical and simulation parameters also contribute to stable navigation. Proper tuning of wheel friction, motor torque, and collision boundaries in Webots ensures realistic physical interaction with the environment. Rounded wall geometries and appropriate timestep selection within the control loop further reduce the likelihood of simulation-induced instability.

CONCLUSION

In conclusion, stable navigation in a simulated mobile robot does not require a large number of sensors or complex perception systems; rather, it depends on strategic sensor placement, structured control architecture, and robust recovery behaviors. A minimal configuration comprising three to five distance sensors, an optional camera, a finite state machine controller, and an explicit reverse recovery mechanism provides a balanced trade-off between simplicity and reliability. Path planning is the step that utilizes localization, mapping, and perception information to determine the optimal path in subsequent movement epochs, guiding the automated mobile robot from one location to another location (Zheng,s. et al, 2023). Accurate localization and perception information is essential for correct planning and controlling and in this research paper, it is related to precision of camera detection. Andrew Y. et al 92024) reported the future improvements may include sensor fusion techniques, adaptive threshold tuning, and simultaneous localization and mapping (SLAM) integration to enhance autonomy and environmental awareness. Since combining the readings into a unified coordinate frame requires knowledge of the robot's position and finding the robot's pose requires a map of the environment, SLAM algorithms must use probabilistic and iterative methods to model the sensor and motion noise to generate cohesive map and trajectory estimates.

REFERENCES

1. Anirban Dutta Choudhury, Rohan Banerjee, Sanjay Kimbahune, Arpan Pal *New Frontiers of Cardiovascular Screening Using Unobtrusive Sensors, AI, and IoT* , 2022 pp 61-89.
2. Rajee, Alimul & Marof, Mohammad Rifatul Islam. (2024). Project Report on Line Follower Robot: A System Design Using Microcontroller. 10.13140/RG.2.2.31415.33448.
3. Anderson Pinheiro Cavalcanti, Arthur Barbosa, Ruan Carvalho, Fred Freitas, Yi-Shan Tsai, Dragan Gašević, Rafael Ferreira Mello (2021), Automatic feedback in online learning environments: A systematic literature review, Computers and Education: Artificial Intelligence, Vol 2, ISSN 2666-920X.
4. Andrew Yarovoi, Yong Kwon Cho (2024) Review of simultaneous localization and mapping (SLAM) for construction robotics applications, Automation in Construction, Vol.162, ISSN 0926-5805,https://doi.org/10.1016/j.autcon.2024.105344.
5. Jabatan Mufti Negeri Perlis (2026). A Simple Look at Muslims and Dogs. https://muftiperlis.gov.my/index.php/en/minda-mufti/1298-a-simple-look-at-muslims-and-dogs .
6. Bin Hong, Zhangxi Lin, Xin Chen, Jing Hou, Shunya Lv, Zhendong Gao (2022), Development and application of key technologies for Guide Dog Robot: A systematic literature review, Robotics and Autonomous Systems, Vol 154, ISSN 0921-8890.
7. Wang, Wei & Jing, Bin & Yu, Xiaoru & Zhang, Wei & Wang, Shengyu & Tang, Ziqi & Yang, Liping. (2025). YOLO-Extreme: Obstacle Detection for Visually Impaired Navigation Under Foggy Weather. Sensors. 25. 4338. 10.3390/s25144338.
8. Saegusa, Shozo & Yasuda, Yuya & Uratani, Yoshitaka & Tanaka, Eiichiro & Makino, Toshiaki & Chang, Jen-YuanJames. (2010). Development of a Guide-Dog Robot: Leading and Recognizing a Visually-Handicapped Person using a LRF. Journal of Advanced Mechanical Design Systems and Manufacturing - J ADV MECH DES SYST MANUF. 4. 194-205. 10.1299/jamdsm.4.194.

9. University of Texas (2024), To Optimize Guide-Dog Robots, First Listen to the Visually Impaired. https://cns.utexas.edu/news/research/optimize-guide-dog-robots-first-listen-visually-impaired.
10. Xiao, A., Tong, W., Yang, L., Zeng, J., Li, Z., & Sreenath, K. (2021). *Robotic guide dog: Leading a human with leash-guided hybrid physical interaction*. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA) (pp. 11470–11476). IEEE. https://doi.org/10.1109/ICRA48506.2021.9561786.
11. Phys.org (2021), Researchers develop a robotic guide dog to assist blind individuals. https://techxplore.com/news/2021-04-robotic-dog-individuals.html.
12. American Heritage (2026), Guide Dog Robot (MELDOG). https://tachilab.org/en/projects/meldog.html.
13. Hochul Hwang, Hee-Tae Jung, Nicholas A Giudice, Joydeep Biswas, Sunghoon Ivan Lee, Donghyun Kim. Towards Robotic Companions: Understanding Handler-Guide Dog Interactions for Informed Guide Dog Robot Design. *CHI 2024: Conference on Human Factors in Computing Systems*, 2024 DOI: 10.1145/3613904.3642181
14. Liliana Ionescu-Feleagă, Voicu D. Dragomir, Sînziana-Maria Rîndașu, Oana-Cristina Stoica, Ștefania-Cristina Curea, Mariana Bunea, Laura-Eugenia-Lavinia Barna (2025), Business simulation games from the perspective of accounting and management professors: Implications for sustainability education in universities, The International Journal of Management Education, Vol 23(2), ISSN 1472-8117.
15. Ichikawa, Reiya & Zhang, Bin & Lim, Hun-ok. (2023). Research on a Guide Dog Robot for Expressing Visual Environment by Voice視覚環境を音声で表現可能な盲導犬ロボットに関する研究. IEEJ Transactions on Electronics, Information and Systems. 143. 562-568. 10.1541/ieejeiss.143.562.
16. Swol, Lyn. (2003). The Effects of Nonverbal Mirroring on Perceived Persuasiveness, Agreement with an Imitator, and Reciprocity in a Group Discussion. Communication Research - COMMUN RES. 30. 461-480. 10.1177/0093650203253318.
17. Ahmed Farouk Kineber, Nehal Elshaboury, Ayodeji Emmanuel Oke, John Aliu, Ziyad Abunada, Mohammad Alhusban (2024), Revolutionizing construction: A cutting-edge decision-making model for artificial intelligence implementation in sustainable building projects, Heliyon,Vol 10 (17), ISSN 2405-8440.
18. Chung-Lun Wei, Yu-Min Wang, Hsin-Hui Lin, Yi-Shun Wang, Jun-Lin Huang (2022), Developing and validating a business simulation systems success model in the context of management education, The International Journal of Management Education,Vol 20 (2), ISSN 1472-8117.
19. Cellier, François. (1977). Combined continuous/discrete system simulation languages: usefulness, experiences and future development. ACM Sigsim Simulation Digest. 9. 18-21. 10.1145/1102505.1102514.
20. Zheng, S., Wang, J., Rizos, C., Ding, W., & El-Mowafy, A. (2023). Simultaneous Localization and Mapping (SLAM) for Autonomous Driving: Concept and Analysis. Remote Sensing, 15(4), 1156. https://doi.org/10.3390/rs15041156
21. Joshua D. Carl and Gautam Biswas (2016), Parallel and Multi-Thread Programming, Ordinary Differential Equations, Simulation, Journal of Software Engineering and Applications, Vol.9 No.5.
22. Razeen, H et al., (2019),Finite state automaton based control system for walking machines, International Journal of Advanced Robotic Systems.