Smart Walking Cane for Indoor Navigation for the Visually Impaired

Dr. Abdul Basit Memon
Associate Professor, ECE
Habib University
Karachi, Pakistan
basit.memon@sse.habib.edu.pk

Lyeba Abid
Computer Engineering
Habib University
Karachi, Pakistan
la07309@st.habib.edu.pk

Ali Muhammad Asad
Computer Science
Habib University
Karachi, Pakistan
aa07190@st.habib.edu.pk

Sadiqah Mushtaq
Computer Engineering
Habib University
Karachi, Pakistan
sm07152@st.habib.edu.pk

Syed Muhammad Ali Naqvi
Computer Science
Habib University
Karachi, Pakistan
sn07590@st.habib.edu.pk

***Abstract*—The development of an indoor wheel mobile robot navigation system presents a unique challenge due to the complexity of indoor environments and the presence of obstacles. While autonomous robots are increasingly operating in indoor human environments, the uneven and unstructured, especially dynamic environments still pose a big problem. Yet, the development of such navigational robots having the potential to overcome some of the limitations of traditionally used navigational aids for the visually impaired, specifically in unfamiliar territories. This paper presents an opportunity for an autonomous mobile robot designed in the shape of a walking cane - however, modelled through a differential drive robot for now - to navigate through an indoor environment, effectively guiding a user to a set destination while avoiding obstacles on the path. Our focus is on enhancing the mobility of robots in indoor settings that are traditionally designed for human use but are often unstructured. Our research not only presents an approach to obsacle avoidance, but our approach also integrates path planning algorithms for the autonomous robot to efficiently calculate the shortest path to the destination given a map, and a path following algorithm to ensure that the robot follows the path generated by the path planning algorithm, while avoiding obstacles on the way.**

Through this paper, we aim to contribute to the field of robotics by addressing critical challenges in indoor navigation, emphasizing the importance of adaptability and safety in shared human-robot environments.

I. INTRODUCTION

Visual impairment encompasses a spectrum of conditions affecting an individual's ability to see. It ranges from mild blurry vision, requiring corrective lenses, to complete blindness. This can involve reduced sharpness, decreased field of view, difficulty processing details, or even total light perception loss. The causes are diverse, ranging from congenital disabilities to age-related diseases like cataracts or glaucoma.

Visual impairment presents a profound challenge to the fundamental human ability to navigate, particularly within complex indoor environments. Reliance on visual cues for spatial awareness and path planning is disrupted, necessitating a shift towards alternative sensory inputs like tactile perception and auditory echolocation. This re-mapping of the world requires significant cognitive effort, often resulting in reduced mobility and increased dependence on others. Traditional walking canes offer basic support for obstacle detection, but lack the ability to provide comprehensive information about surroundings and facilitate efficient navigation. This limitation can hinder independence, restrict mobility, and significantly impact the quality of life for many visually impaired individuals. Addressing the navigational needs of visually impaired individuals demands innovative solutions that go beyond traditional assistive tools, fostering independence, confidence, and a renewed sense of control over their surroundings.

In response to this critical need, this paper explores the development and potential of a smart walking cane specifically designed for indoor navigation by the visually impaired. We move beyond a mere obstacle detection tool, proposing a multi-functional assistive walking cane that, given the map of the indoor environment and a destination point within the map, finds a path and navigates the person within the indoor environment to the desired destination while avoiding any obstacles.

Our walking cane makes use of A-star path planning algorithm to find the optimum path to a destination point in an indoor setting. It then utilizes pure pursuit algorithm to follow the path while simultaneously avoiding unseen obstacles that the robot might encounter along the way. Using odometric calculations and the sensor readings obtained from lidar sensors, the robot localizes itself and estimates its pose for efficiently following the path. Additionally, the robot also maintains its velocity according to the pace of the user holding the walking cane to maintain user friendliness.

Following the discussion of the related work, the paper will explain the overall architecture of the robot and its detailed explanation in the Functioning section. Moreover, the relevant algorithms used are also discussed. The technical details are followed by the simulation results in different indoor environments to analyze the performance of the robot.

II. Related Work

The development of autonomous mobile robots capable of assisting people in human environments, particularly in real-time navigation for the visually impaired, has been a significant research focus [1] for research in robotics. A substantial amount of research and commercial solutions [2] [3] [4] [5] have been dedicated to this purpose, with designs often tailored to the specific needs and behaviors of visually impaired individuals, as well as the specifications of navigation assistance technologies. One of the approaches is to use navigation assistance technologies such as wearable devices that provide auditory feedback for their user to navigate with [6] [7], [8] including smartphones, which convey the location of the relevant points of interest [9], [10]. Computer vision approaches guide their user to a specific object whence detected [11], [12].

Another approach aims to guide mobile robots to cater the diverse user needs, mainly in two types; wheeled and quadruped [13], [14]. The wheeled robot navigates using wheels and is equipped with sensors like cameras, LIDAR, and ultrasonic sensors for obstacle detection and pathfinding [15] [16] [17]. It communicates with users via speakers and microphones [18] [19] [20].

The quadruped robot, on the other hand, mimicks a dog's behavior and movement, offering a more intuitive user interface. Its sensor systems are akin to the wheeled robot's [21], [22]. These robots comprehend their environment through mapping [23], [24], path planning [25], and speed control [26], ensuring collision-free navigation [27]. The advent of guide dog robots first began with the MELDOG project [28], with subsequent studies [29], [30] developing a guide dog robot that could lead and recognize a visually impaired person using a laser range finder sensor. An outdoor navigation system was built using GPS and tactile-foot feedback, which was worn by blind pedestrians [31].

Further advancements in guide dog projects [32], [33] utilized cameras to perceive the world. Some systems streamed video from a smartphone to a laptop on the robot for processing [32], [33], while others sent images from the smartphone to a cloud service for processing [34]. One system used an embedded system with two cameras among other sensors, but image processing was done on a remote server, which took longer than an edge computing server [35].

While most researches cater to, and build upon existing traditional navigational aids, some researches have also proposed white canes as a viable solution to increase the sensing range [36]. The GuideCane [37] and Co-Robotic Cane [38] prototypes are capable of actively changing their direction for obstacle avoidance.

Stride Senze [39] by the company NOK - NextOfKin is one such implementation of walking canes, featuring self balancing technology, multiple sensors to detect and avoid obstacles, and behaviour learning algorithms, equipped with the ability to adapt to the user's gait.

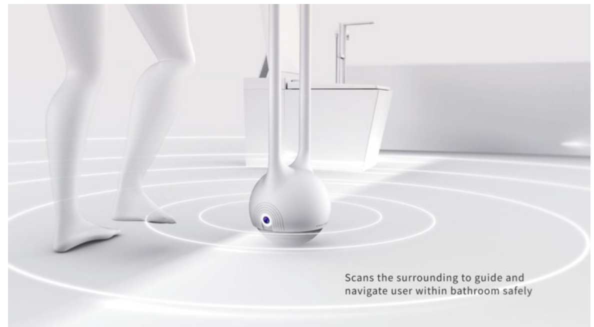

Figure 1: Stride Senze by NOK - NextOfKin

Lately, Stanford researchers have also been able to develop a relatively cheaper, self navigating smart cane [40] which incorporates sensing and way-finding approaches, inspired from self-driving vehicles, using a LIDAR sensor to measure the distance to nearby obstacles and then directs users around those areas [40]. The cane also includes additional sensors such as the accelerometer, GPS, gyroscopes and magnetometers. The cane uses algorithms like Simultaneous Localization and Mapping (SLAM) and visual servoing to guide the user effectively to a detected object [40].

Figure 2: Self-Navigating Smart Cane - Stanford University

Building on the prior comprehensive foundations laid by existing research and available technologies, our work seeks to expand the capabilities of such technologies, mainly the smart walking cane. Our work goes beyond the scope of existing smart canes focused primarily on outdoor scenarios. By addressing the unique challenges of indoor environments, such as cluttered spaces, and dynamic obstacles as one would feel in an office based setting. We provide a simulation on Gazebo to demonstrate the working of our smart walking cane through improved sensory feedback and an intuitiv design. Unlike existing solutions that only focus on object detection, our approach also tends to utilize the A* algorithm for path planning to find the optimum path to a set destination in an indoor setting. Then using the pure pursuit algorithm, the robot follows the path while simultaneously avoiding unseen obstacles that the robot might encounter along the way. Using odometric calculations and the sensor readings obtained from lidar sensors, the robot localizes itself using EKF-based localization and estimates its pose for efficiently following the path. Additionally, the robot also maintains its velocity according to the pace of the user holding the walking cane to maintain user friendliness.

III. Functioning

A. Simulation

We use the Gazebo simulator engine for simulating the ROS based robot. For experimentation and testing purposes, we have specifically utilized Gazebo Turtlebot environments in the ROS virtual machine - which is pre-configured to support the examples in ROS Toolbox

1) Modelling the cane: For the purpose of simulation, the navigation cane is modelled as a wheel mobile robot. The Gazebo Turtlebot serves as a model which acts as a assistive cane in our case. The idea is that a differentail drive or omni directional walking cane can be developed like the one in figure 3 as an extension to the turtlebot model.

Figure 3: Figure shows an example of wheeled walking cane that can be modelled as an extension to turtlebot simulation model

The turtle bot is a ROS based differential drive wheeled mobile robot equipped with a LiDar sensor. The robot can receive velocity commands using the publisher through cmd_vel topic as well as send the LiDar sensor readings using the receiver through scan topic in ROS simulator. Additional information about publishing and receiving data (communicating with ROS) can be found in [https://www.mathworks.com/help/ros/ug/exchange-data-with-ros-publishers-and-subscribers.html].

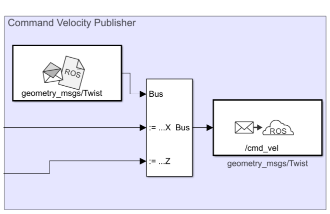

Figure 4: Publisher for sending velocity commands to walking cane

2) Modelling the person: Since the person which will be guided by the walking cane will be holding the cane, we need to take into account the person's moving speed to make the mobile robot more user friendly. Therefore, we have modelled the person as an external factor which

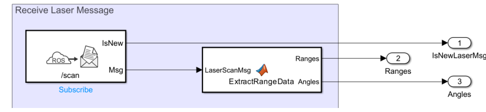

Figure 5: Receiving sensor readings from the cane

```
function [distance, angle] = simulatePerson()

    % Generate random values using Perlin noise for distance and angle
    noiseScale = 10;  % Adjust the scale of the Perlin noise
    distance = imnoise(zeros(1, 1), 'perlin', noiseScale, 0.5);
    angle = imnoise(zeros(1, 1), 'perlin', noiseScale, 0.5) * 180;
```

Figure 6: Modelling the LiDar sensor to simulate the person

would be affecting the movement of the walking cane. Although the person and the second LiDar sensor measuring his/her distance and angle from the robot is not simulated on Gazebo simulator, it is modelled using a matlab script which generates a distance and orientation of the user with respect to the robot using perlin noise. The later part of the paper discusses how this is handled by the robot.

B. System Architecture

Our system comprises of four main components.

- Localization
- Path Planning
- Path Following
- Obstacle Avoidance

Figure 7 shows the architecture of the robot. The Path Planning and Localization block, takes a destination within the indoor environment and makes use of A-star path planning algorithm to generate a set of way points for the robot to follow. The robot uses the sensor readings and odometric calculations and apply localization techniques to estimate the robot pose. This information is then used by the path following control block to generate velocities. This block also adjusts the velocity according to the movement of the user. This data is then fed to Obstacle Avoidance block which avoids any unforeseen obstacles which are not a part of the map. Finally the control velocities are generated and published to the robot motors. These blocks are discussed further.

C. Path Planning and Localization

In the realm of robotic systems and autonomous navigation, the effective integration of diverse inputs is crucial

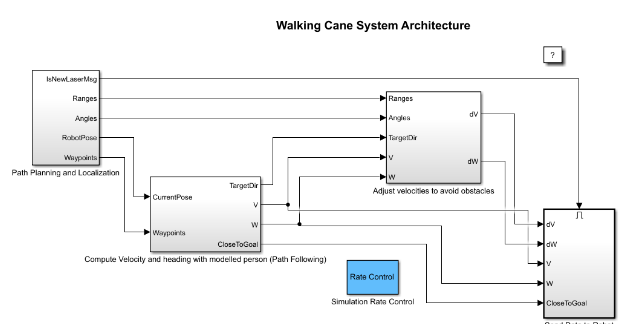

Figure 7: Robot System Architecture

for enabling intelligent decision-making and precise control. This section delves into the intricacies of our implemented algorithms and methodologies for path planning and localization. Our primary focus lies on the utilization of the A* pathfinding algorithm for finding a path given the math of an indooer environment, providing a comprehensive overview of strategic starting and goal points, as well as the dynamically computed optimal paths for our robotic trajectory. Furthermore, we explore the integration of sensor measurements, detailing the processing steps involved in extracting vital information for robust control systems. The estimation of the robot's pose, a fundamental aspect for navigation, is discussed extensively, encompassing the use of wheel speed sensors, kinematic equations, and the integration of an Extended Kalman Filter (EKF) to enhance accuracy in pose estimation. This intricate combination of algorithms and sensor fusion underscores our commitment to developing a sophisticated and reliable assistive technology for indoor navigation, particularly tailored for scenarios involving visually impaired individuals using a Smart Walking Cane.

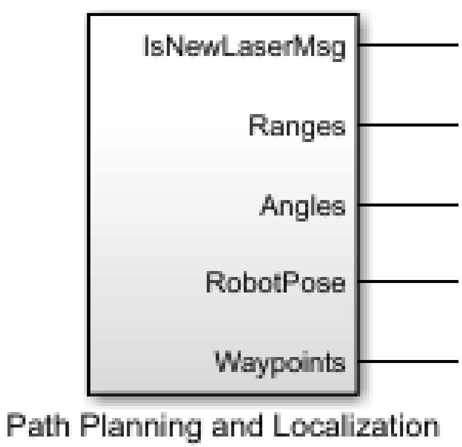

Figure 8: Path Planning and Localization Block

1) Path Finding Algorithm: A* star algorithm is a widely used path finding algorithm used to find an optimum path between two points. It takes a map in form of a graph, the starting and ending points as nodes and finds a path. The algorithm is given in Algorithm 1 listing.

In our case, we have employed MATLAB and the Robotics System Toolbox for path planning scenario utilizing the A* algorithm. In the context of our robot, within the grid-based Gazebo environment we use binary occupancy maps, given as a 'mat' file and use it to generate a path given a starting and destination point. We set starting and goal points within the Cartesian coordinate system, delineating the commencement and destination of our robotic trajectory. The plannerAStarGrid class, driven by the A* algorithm, computes an optimal, collision-free path on the Gazebo office map as shown in Figure 9. The generated path is then interpolated with suitable intervals to generate way points for the robot to follow.

Algorithm 1: A* Algorithm

Data: Graph G, start node $start$, goal node $goal$
Result: Optimal path from $start$ to $goal$
Initialize:
- $openSet$ with $start$ (priority queue with initial cost)
- $closedSet$ as an empty set
- $cameFrom$ as an empty map
- $gScore$ map with $start$ initialized to 0
- $fScore$ map with $start$ initialized to heuristic estimate from $start$ to $goal$

while *$openSet$ is not empty* **do**
  $current \leftarrow$ node in $openSet$ with the lowest $fScore$;
  if $current = goal$ **then**
    return reconstructPath($cameFrom, goal$);
  Remove $current$ from $openSet$;
  Add $current$ to $closedSet$;
  foreach *neighbor n of $current$* **do**
    if *n is in $closedSet$* **then**
      continue;
    $tentative_gScore \leftarrow$ $gScore[current]$ + distance($current, n$);
    if *n is not in $openSet$ or $tentative_gScore < gScore[n]$* **then**
      Add n to $openSet$;
      $cameFrom[n] \leftarrow current$;
      $gScore[n] \leftarrow tentative_gScore$;
      $fScore[n] \leftarrow$ $gScore[n]$ + heuristic estimate($n, goal$);

return failure (no path found);

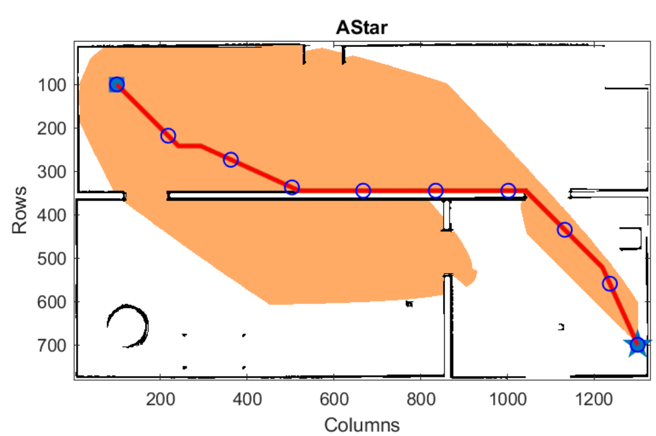

Figure 9: An example of the shortest path between two points on the Gazebo office map

2) Sensor Measurements: We have leveraged the Simulink modeling environment to integrate lidar sensor data seamlessly into our control systems. The initial step involves the Read Scan block, which interfaces with the Robot Operating System (ROS) network, allowing us to acquire laser scan messages (angles and ranges) from a lidar sensor. This block takes the ROS input and produces the laser scan message as its output.

Subsequently, we employ the Extract Range Data block, embodying the MATLAB code presented in this study. This block serves as a crucial processing step, where the laser scan message obtained from the Read Scan block is used as input. The Extract Range Data block processes this input, extracting two vital pieces of information: the range measurements and the corresponding angles. The Ranges output is a double-precision array representing the processed range data, while the Angles output encapsulates the wrapped angles within the $(-\pi, \pi]$ interval. Both outputs are pivotal for subsequent robotic functionalities.

3) Robot Pose: To estimate the pose of our differential-drive robotic platform, we employed a wheel speed sensor in Gazebo, utilizing the recorded wheel speeds for subsequent odometry calculations. We implemented a robust odometry model based on kinematic equations governing the relationship between wheel speeds and the robot's linear and angular velocities. These equations are crucial for accurately determining the robot's pose in our simulation of an indoor navigation for the visually impaired using a Smart Walking Cane. These equations, denoted as:

$$v = \frac{r}{2} \cdot (\text{wheelspeeds}(1) + \text{wheelspeeds}(2))$$

$$\omega = -\frac{r}{L} \cdot (\text{wheelspeeds}(1) - \text{wheelspeeds}(2))$$

are fundamental to our navigation strategy, providing continuous estimates of the robot's pose in the x-y plane and its orientation (ϕ).

The odometry model integrates these velocities over discrete time steps (Δt), where $\Delta t = 0.01$, which is same as the time set in gazebo. The updated pose of the robot is calculated using equations that consider its current position, velocity, and orientation:

$$x_{\text{next}} = x_k + \Delta t \cdot v \cdot \cos(\phi_k)$$

$$y_{\text{next}} = y_k + \Delta t \cdot v \cdot \sin(\phi_k)$$

$$\phi_{\text{next}} = \phi_k + \Delta t \cdot \omega$$

However, it's crucial to note that in real-world scenarios, the results obtained from odometry alone may not be highly accurate. The inherent limitations of odometry include wheel slippage, uneven surfaces, and accumulated errors over time, which can lead to discrepancies between the estimated and actual robot poses.

4) Pose Estimation: To address these challenges and enhance the accuracy of our pose estimation, we incorporate an Extended Kalman Filter (EKF) into our localization framework. The EKF equations are outlined as follows:

5) State Prediction::

$$\bar{x}_k = g(\hat{x}_{k-1}, u_k)$$

6) Covariance Prediction::

$$\bar{\Sigma}_k = G_k \Sigma_{k-1} G_k^T + R_k$$

7) Kalman Gain Calculation::

$$K_k = \bar{\Sigma}_k H_k^T (H_k \bar{\Sigma}_k H_k^T + Q_k)^{-1}$$

8) State Update::

$$\hat{x}_k = \bar{x}_k + K_k(z_k - h(\bar{x}_k))$$

9) Covariance Update::

$$\Sigma_k = (I - K_k H_k)\bar{\Sigma}_k$$

Here, $\bar{x}_k$ is the predicted state, $\bar{\Sigma}_k$ is the predicted covariance, G_k and H_k are Jacobian matrices representing partial derivatives, R_k is the process noise covariance, Q_k is the measurement noise covariance, and z_k is the sensor measurement.

The iterative integration of these EKF equations within our project ensures a refined and accurate estimation of the robot's pose, mitigating inaccuracies introduced by the limitations of odometry in real-world environments. This integrated approach showcases our commitment to developing an effective assistive technology for precise navigation in challenging indoor scenarios.

D. Path Following with Pure Pursuit Controller

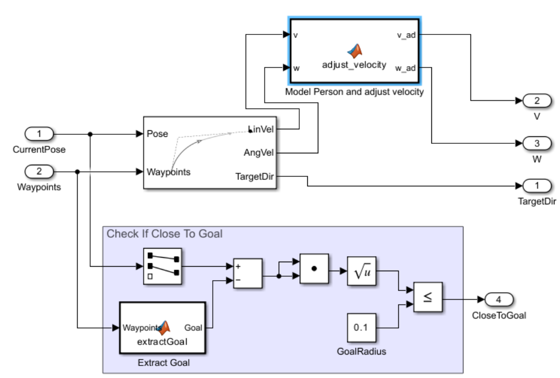

Figure 10: Breakdown of Path Following Block

Figure 11 shows a detailed breakdown of the main path following block of the robot. In the pursuit of precise path following for our robotic system in the indoor navigation project, we employ the Pure Pursuit algorithm—a well-established method for tracking a desired trajectory. The primary objective of this algorithm is to guide the robot along a predefined path, optimizing its trajectory to closely match the intended course. The core idea revolves around determining a point on the path, known as the "lookahead point," and directing the robot to navigate towards it. As displayed in figure 11, the pure pursuit algorithm uses the current robot pose and the waypoints obtained through filtering and path planning from the previous block and generates the velocity commands.

The key mathematical expressions governing the Pure Pursuit algorithm are derived from geometric relationships within the robot's coordinate system. The equations presented here elucidate the calculation of the lookahead distance (r) as a function of the lateral displacement (y) from the desired path and a constant parameter (L). The relationship is defined by the following sequence of equations:

$$
\begin{aligned}
& r = |y| + d \\
& d^2 + x^2 = r^2 \\
& (r - |y|)^2 + x^2 = r^2 \\
& r^2 + y^2 - 2r|y| + x^2 = r^2 \\
& r^2 + L^2 - 2r|y| = r^2 \\
& r = \frac{L^2}{2|y|}
\end{aligned}
$$

These equations provide a systematic derivation of the lookahead distance, highlighting the intricate relationship with lateral displacement (y). The final expression, $r = \frac{L^2}{2|y|}$, stands as a concise representation of the Pure Pursuit algorithm's mathematical foundation. This formula serves as a key component in our implementation, contributing to the overall efficiency and accuracy of our robotic path following in complex indoor scenarios.

Within our system architecture, we have also integrated the "Check If Close To Goal" block, utilizing a sequence of mathematical operations. These operations involve the extracted goal, waypoints, and relevant parameters to determine the system's proximity to its intended destination. The output, labeled "CloseToGoal," succinctly communicates whether the system is near its destination or not. This information serves as a critical input to our control logic, guiding the system's behavior based on its proximity to the goal.

1) Adjust Velocity for user friendliness: The second LiDar sensor that we have modelled within Matlab (as described in subsection A) tells us the distance and orientation of the user with respect to the robot. Figure 12 shows the orientation and distance of the user from the walking cane.

We want the velocity of the walking cane and the person to be the same. We have used a simple PD controller to adjust the velocity of the robot based on user's velocity. As depicted in figure 11, it takes input the velocity outputs of pure pursuit controller and adjusts the velocity of the robot based on the second LiDar sensor measurements.

Let v, w be the linear and angular velocities generated by the block in figure 11 as outputs. Let v_{pure} and w_{pure} be the velocity commands generated by the pure pursuit block. We set upper and lower threshold t_1 and t_2 for the distance and θ_1 , θ_2 for the orientation. If the person is within these limits, the error remain zero and the velocity $v = v_{pure}$ and

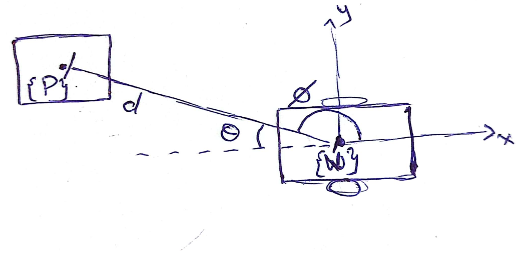

Figure 11: The distance (d) and angle (phi) of the user with respect to the robot

$w = w_{pure}$. If the distance and orientations are not within the threshold limits, i.e the person tends to come too close or too far from the cane, then this means that the velocity of the person differ from the cane, then through PD control we adjust the velocity as follows.

$$
e_d = \begin{cases} t_1 - d & \text{if } d > t_1 \\ t_2 - d & \text{if } d < t_2 \\ 0 & \text{if } t_2 \leq d \leq t_1 \end{cases}
$$

$$
e_\theta = \begin{cases} \theta_1 - \theta & \text{if } \theta_1 > \theta \\ \theta_2 - \theta & \text{if } \theta_2 < \theta \\ 0 & \text{if } \theta_1 \leq \theta \leq \theta_2 \end{cases}
$$

$$
\begin{aligned}
v_p &= k_{p_v} \cdot e_d - k_{d_v} \cdot \dot{d} \quad \text{(Linear Velocity adjustment)} \\
w_p &= k_{p_w} \cdot e_\theta - k_{d_w} \cdot \dot{\theta} \quad \text{(Angular Velocity adjustment)} \\
v &= v_{pure} + v_p \\
w &= w_{pure} + w_p
\end{aligned}
$$

In our experiments, the linear velocity control adjusts the velocity, however the angular velocity control does not produce desired results and hence needs to be remodelled.

E. Adjust velocities to avoid obstacles

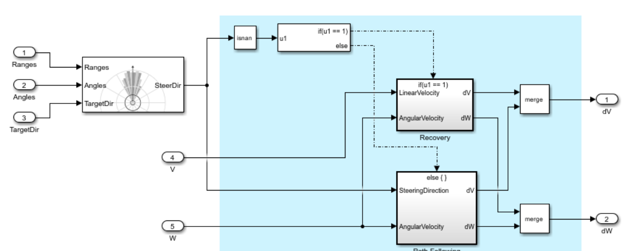

Figure 12: Adjust velocities to avoid obstacles

Our robot employs the VFH algorithm for obstacle avoidance, analyzing range sensor data to navigate through environments. Seamlessly integrated into our path-following system, VFH optimizes steering directions for effective obstacle avoidance and precise target pursuit. The system's adaptive velocity adjustments accommodate both clear and ambiguous steering scenarios, ensuring efficient navigation in diverse environments.

1) Vector Field Histogram (VFH): The VFH algorithm is a key aspect of our path-following robot's obstacle avoidance strategy. Using range sensor data, VFH calculates obstacle-free steering directions by generating polar density histograms, providing insights into obstacle locations.

VFH transforms polar density histograms into binary histograms, defining valid steering directions based on specified thresholds. Essential robot parameters, such as `RobotRadius`, `SafetyDistance`, `MinTurningRadius`, and `DistanceLimits`, influence accurate steering direction calculation. `RobotRadius` represents the smallest encircling circle, `SafetyDistance` adds a safety margin, `MinTurningRadius` considers turning capabilities, and `DistanceLimits` sets obstacle avoidance range.

VFH seamlessly integrates with our path-following system, utilizing angles, ranges, linear velocity, and angular velocity. This integration ensures optimal steering directions, balancing obstacle avoidance with target pursuit.

2) Velocity Evaluation and Recovery: In the intricate process of determining velocities based on the proposed steering direction, the system exhibits a nuanced approach. When a discernible steering direction is identified, the system engages in real-time adjustments to both linear and angular velocities. This adaptability ensures that the system seamlessly aligns with the intended direction of motion.

Conversely, in scenarios where the steering direction remains ambiguous or indiscernible, a sophisticated recovery mechanism is set into motion. The mechanism, activated through an `else` statement, orchestrates a thoughtful interplay of adjustments. By modifying velocities, particularly `dV`, and incorporating additional input labeled `1 dV`, the system crafts a responsive solution denoted as `2 dW`. This intricate dance of computations empowers the system to efficiently navigate challenges arising from the absence of a viable steering direction.

IV. Experimental Results

In our experimental evaluation, we conducted three test cases to assess the performance of our indoor navigation system using the Smart Walking Cane. The test cases represent scenarios ranging from controlled environments with known maps to more dynamic situations involving unknown obstacles. The integration of A* path planning, localization, Pure Pursuit for path following, and obstacle avoidance through the VFH algorithm is evaluated in these diverse scenarios.

1) Test Case 1: Room-to-Room Navigation in Gazebo Office: In the first test case, the robot was tasked with navigating from one room to another within the Gazebo office environment. The map of the office was known to the robot, and the starting and destination coordinates were provided. The robot successfully generated an A* path, followed the path using Pure Pursuit, and adapted its velocity based on user movements. Unknown obstacle avoidance was not a significant factor in this scenario, as the obstacle location were priorly known by the robot.

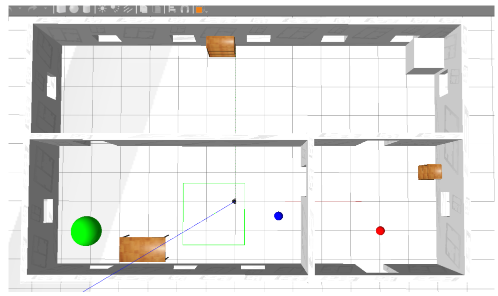

Figure 13: Gazebo office simulation environment

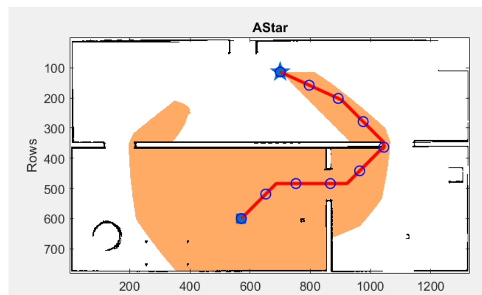

Figure 14: Waypoints generated by A-start algorithm for navigation

Figure 14 shows the path generated by the A-start algorithm to generate waypoints for the robot to follow.

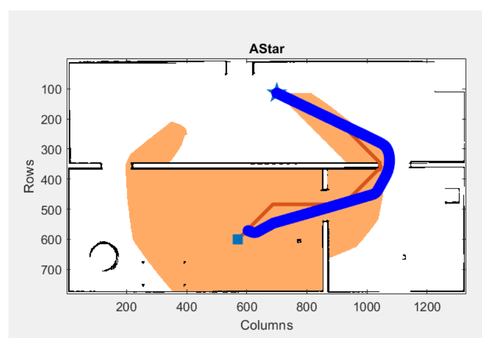

Figure 15: Actual path (blue) followed by robot

Figure 15 shows the actual path that the robot took. The difference in paths is due to the interference of obstacle avoidance block to maintain a certain radius from obstacles while following the path as well as the movement randomness introduced by the noise in movement produced by the modelled user. The system demonstrated accurate path planning and execution, showcasing the seamless integration of algorithms. The robot successfully reached its destination while adjusting its velocity based on the simulated user's movement. The results indicate the effectiveness of our navigation system in controlled environments.

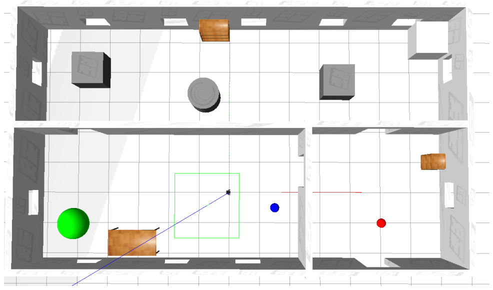

Figure 16: Unknown obstacles in environment

2) Test Case 2: Navigation with Unknown Obstacles in Gazebo Office: In the second test case, we introduced unknown obstacles in the Gazebo office environment which are not known to the path follower in the map to reach the same destination as in Case 1. The A* path planning algorithm generates the path same path as in Figure 14. However, due to the unknown obstacle, the robot should adjust its route accordingly. Figure 16 shows the unknown block obstacle placed in the path.

The robot is equipped with LiDAR sensors to detect obstacles in real-time. The VFH algorithm plays a crucial role in obstacle avoidance, allowing the robot to adapt its steering directions based on real-time LiDAR data. The system successfully navigated through the environment, avoiding obstacles and adjusting velocities as needed. Figure 17 shows the path adjustment the robot made to avoid the obstacle. This test demonstrates the robustness of our navigation system in handling unforeseen obstacles in a dynamic environment.

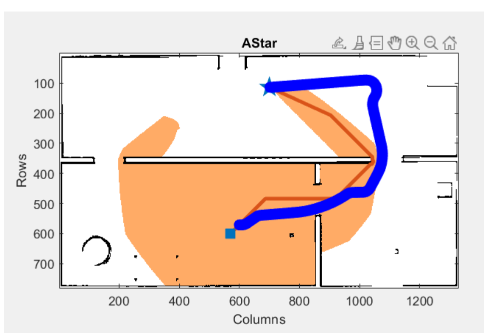

Figure 17: Path followed by robot (blue) in contrast to the planned path (orange)

3) Test Case 3: Navigation in moderately crowded environment: In the third test case, we extended our evaluation to a more complex environment with more unknown obstacles. The unknown obstacles can be seen as a mildy crowded environment with people inside it as shown in figure 18.

The integration of A* path planning, localization, Pure Pursuit, and VFH obstacle avoidance proved effective in handling the complexities of the environment. The robot successfully navigated through narrow passages, avoided obstacles, and adapted its velocity to provide a user-friendly

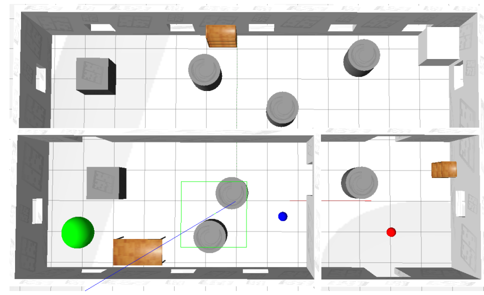

Figure 18: Moderately crowded indoor environment

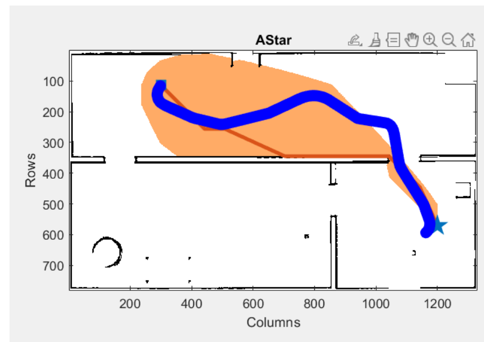

Figure 19: Path planned by Robot (orange) and the path followed by robot (blue) avoiding several obstacles

experience. This test demonstrates the versatility of our navigation system in diverse indoor settings.

V. Conclusion

Visual impairment encompasses a diverse range of challenges, from mild vision issues to complete blindness, impacting navigation within complex indoor environments. Traditional walking canes, while providing basic obstacle detection, fall short in offering comprehensive information for efficient navigation, limiting independence and quality of life for the visually impaired.

Addressing this critical need, our paper introduces a smart walking cane tailored for indoor navigation by the visually impaired. Going beyond obstacle detection, our multi-functional cane employs an A-star path planning algorithm to find optimal routes within indoor settings. The pure pursuit algorithm facilitates path following while avoiding unforeseen obstacles, enhancing user-friendly navigation.

Utilizing odometric calculations and lidar sensor data, the cane localizes itself and adjusts its velocity based on the user's pace, ensuring efficient path tracking. Our proposed system aims to empower visually impaired individuals with a renewed sense of independence, confidence, and control over their indoor surroundings.

VI. Future work

In future developments, our primary focus is on refining the smart walking cane's design and functionality to better cater to the needs of visually impaired individuals. A crucial step involves transitioning from the current differential drive robot model to a more realistic representation resembling a traditional walking cane. This shift is aimed at enhancing the device's usability and user acceptance by incorporating physical features like grip and ergonomic design.

Additionally, we plan to explore dynamic adaptations based on the user's orientation. While the system currently adjusts the robot's speed in response to the distance between the person and the cane, incorporating angular velocity adjustments according to the user's orientation could further improve the device's responsiveness and adaptability to various navigation scenarios.

Furthermore, our future work will include comprehensive testing of the system in dynamic environments with moving objects. This expansion is designed to assess the device's capability to navigate through scenarios where the surroundings are not static, contributing to the development of a more robust and versatile indoor navigation solution.

Lastly, to enhance the realism and applicability of the system, we aim to implement it on a platform beyond MATLAB. Conducting real-life tests and integrating the smart walking cane model into the Gazebo simulator will provide valuable insights into its real-world performance and effectiveness. This approach ensures the practical viability of the system for visually impaired individuals.

REFERENCES

[1] W. Burgard, A. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "The interactive museum tour-guide robot," in *Proceedings of the AAAI National Conference on Artificial Intelligence*, pp. 11–18, 01 1998.

[2] S. Azenkot, C. Feng, and M. Cakmak, "Enabling building service robots to guide blind people: A participatory design approach," in *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, HRI '16, pp. 3 – 10, IEEE Press, 2016.

[3] R. Golledge, J. Marston, J. Loomis, and R. Klatzky, "Stated preferences for components of a personal guidance system for nonvisual navigation," *Journal of Visual Impairment & Blindness*, vol. 98, 03 2004.

[4] M. Miao, M. Spindler, and G. Weber, "Requirements of indoor navigation system from blind users," in *Information Quality in e-Health* (A. Holzinger and K.-M. Simonic, eds.), (Berlin, Heidelberg), pp. 673–679, Springer Berlin Heidelberg, 2011.

[5] H. Nicolau, J. Jorge, and T. Guerreiro, "Blobby: How to guide a blind person," in *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, p. 3601–3606, Association for Computing Machinery, 2009.

[6] D. Dakopoulos and N. G. Bourbakis, "Wearable obstacle avoidance electronic travel aids for blind: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 1, pp. 25–35, 2010.

[7] R. K. Katzschmann, B. Araki, and D. Rus, "Safe local navigation for visually impaired users with a time-of-flight and haptic feedback device," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 3, pp. 583–593, 2018.

[8] B. Li, J. P. Muñoz, X. Rong, J. Xiao, Y. Tian, and A. Arditi, "Isana: Wearable context-aware indoor assistive navigation with obstacle avoidance for the blind," in *Computer Vision – ECCV 2016 Workshops* (G. Hua and H. Jégou, eds.), pp. 448–462, Springer International Publishing, 2016.

[9] J. R. Blum, M. Bouchard, and J. R. Cooperstock, "What's around me? spatialized audio augmented reality for blind users with a smartphone," in *Mobile and Ubiquitous Systems: Computing, Networking, and Services* (A. Puiatti and T. Gu, eds.), pp. 49–62, Springer Berlin Heidelberg, 2012.

[10] C. Gleason, A. Fiannaca, M. Kneisel, E. Cutrell, and M. Morris, "Footnotes: Geo-referenced audio annotations for nonvisual exploration," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, pp. 1–24, 09 2018.

[11] D. Ahmetovic, C. Bernareggi, A. Gerino, and S. Mascetti, "Zebrarecognizer: Efficient and precise localization of pedestrian crossings," in *2014 22nd International Conference on Pattern Recognition*, pp. 2566–2571, 2014.

[12] R. Manduchi and J. M. Coughlan, "The last meter: Blind visual guidance to a target," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pp. 3113 – 3122, Association for Computing Machinery, 2014.

[13] D. R. Bruno, M. H. de Assis, and F. S. Osório, "Development of a mobile robot: Robotic guide dog for aid of visual disabilities in urban environments," in *2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE)*, pp. 104–108, 2019.

[14] Y. Wei and M. Lee, "A guide-dog robot system research for the visually impaired," in *2014 IEEE International Conference on Industrial Technology (ICIT)*, pp. 800–805, 2014.

[15] A. Imadu, T. Kawai, Y. Takada, and T. Tajiri, "Walking guide interface mechanism and navigation system for the visually impaired," in *2011 4th International Conference on Human System Interactions, HSI 2011*, pp. 34–39, 2011.

[16] Q. Wang, K. Zhang, K. Zhao, and M. Liao, "Smart seeing eye dog wheeled assistive robotics," in *2021 3rd International Symposium on Robotics & Intelligent Manufacturing Technology (ISRIMT)*, pp. 104–108, 2021.

[17] K. Iwatsuka, K. Yamamoto, and K. Kato, "Development of a guide dog system for the blind people with character recognition ability," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 1, pp. 453–456 Vol.1, 2004.

[18] R. Ichikawa, B. Zhang, and H.-o. Lim, "Voice expression system of visual environment for a guide dog robot," in *2022 8th International Symposium on System Security, Safety, and Reliability (ISSSR)*, pp. 191–192, 2022.

[19] M. Elmzaghi, M. Fahad, and Y. Guo, "Implementing robust voice-control for human robot interaction for autonomous robot guides," in *2019 IEEE MIT Undergraduate Research Technology Conference (URTC)*, pp. 1–6, 2019.

[20] R. Hu, X. Yu, B. He, and Z. Li, "Implementation of voice interaction based on questions and answers for guide robot in exhibition hall," in *2017 2nd International Conference on Robotics and Automation Engineering (ICRAE)*, pp. 105–109, 2017.

[21] K. Izumi, K. Watanabe, M. Shindo, and R. Sato, "A sensor fusion technique using visual and ultrasonic information to acquire obstacle avoidance behaviors for quadruped robots," in *2006 SICE-ICASE International Joint Conference*, pp. 5120–5125, 2006.

[22] S. Woo, J. Shin, Y. H. Lee, Y. Hun Lee, H. Lee, H. Kang, H. R. Choi, and H. Moon, "Stair-mapping with point-cloud data and stair-modeling for quadruped robot," in *2019 16th International Conference on Ubiquitous Robots (UR)*, pp. 81–86, 2019.

[23] T. J. Son, A. H. Abu Hassan, and M. H. Jairan, "Optimized robot mapping and obstacle avoidance using stereo vision," in *2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pp. 279–284, 2021.

[24] N. WehChien, S. Gobee, and V. Durairajah, "Guided vehicle based with robot operating system for mapping and navigation task," in *2022 International Conference on Edge Computing and Applications (ICECAA)*, pp. 868–872, 2022.

[25] X. Liu, Z. Liu, F. Zan, W. Chen, and X. Zhang, "Research on path planning method of intelligent robot based on a* algorithm," in *2022 IEEE 5th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, pp. 536–539, 2022.

[26] J. Sparbert, J. Kümpel, and E. P. Hofer, "Motion control of a mobile robot with adaption to the environment," in *2001 European Control Conference (ECC)*, pp. 1145–1150, 2001.

[27] B. Zhang, M. Okutsu, R. Ochiai, M. Tayama, and H.-O. Lim, "Research on design and motion control of a considerate guide mobile robot for visually impaired people," *IEEE Access*, vol. 11, pp. 62820–62828, 2023.

[28] S. Tachi, K. Tanie, K. Komoriya, and M. Abe, "Electrocutaneous communication in a guide dog robot (meldog)," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 7, pp. 461–469, 1985.

[29] S. Saegusa, Y. Yasuda, Y. Uratani, E. Tanaka, T. Makino, and J.-Y. Chang, "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*, vol. 4, pp. 194–205, 02 2010.

[30] S. Saegusa, Y. Yasuda, and Y. Uratani, "Development of a guide-dog robot: human-robot interface considering walking conditions for a visually handicapped person," *Microsystem Technologies*, 2011.

[31] R. Velázquez, E. Pissaloux, P. Rodrigo, M. Carrasco, N. Giannoccaro, and A. Lay-Ekuakille, "An outdoor navigation system for blind pedestrians using gps and tactile-foot feedback," *Applied Sciences*, vol. 8, p. 578, 04 2018.

[32] Y. Wei, X. Kou, and M. Lee, "Development of a guide-dog robot system for the visually impaired by using fuzzy logic based human-robot interaction approach," in *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, pp. 136–141, 2013.

[33] Y. Wei and M. Lee, "A guide-dog robot system research for the visually impaired," in *2014 IEEE International Conference on Industrial Technology (ICIT)*, pp. 800–805, 2014.

[34] J. Jakob and J. Tick, "Concept for transfer of driver assistance algorithms for blind and visually impaired people," in *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pp. 000241–000246, 2017.

[35] W. M. Elmannai and K. M. Elleithy, "A highly accurate and reliable data fusion framework for guiding the visually impaired," *IEEE Access*, vol. 6, pp. 33029–33054, 2018.

[36] N. Fallah, I. Apostolopoulos, K. Bekris, and E. Folmer, "Indoor Human Navigation Systems: A Survey," *Interacting with Computers*, vol. 25, pp. 21–33, 01 2013.

[37] I. Ulrich and J. Borenstein, "The guidecane-applying mobile robot technologies to assist the visually impaired," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 31, no. 2, pp. 131–136, 2001.

[38] C. Ye, S. Hong, X. Qian, and W. Wu, "Co-robotic cane: A new robotic navigation aid for the visually impaired," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 2, no. 2, pp. 33–42, 2016.

[39] J. Neira, "Stride senze transforms a walker or a walking crane into a smart walking companion," 02 2022.

[40] P. Slade, A. Tambe, and M. J. Kochenderfer, "Multimodal sensing and intuitive steering assistance improve navigation and mobility for people with impaired vision," *Science Robotics*, vol. 6, no. 59, p. eabg6594, 2021.