2026 IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob 2026)
August 1-4, 2026. Edmonton, Canada

Speak2Move: A Vision-Language-Based Semantic Mapping Framework for Autonomous Navigation in Assistive Robots

Sadra Zargarzadeh[1*], Andor C. G. Siegers[1,2*], and Mahdi Tavakoli[1,3]

***Abstract*— Traditional control interfaces for powered assistive devices are often inaccessible to individuals with severe mobility impairments. Embodied AI, which integrates perception, reasoning, and action within robotic systems, offers a promising approach to improve accessibility and autonomy. This work presents Speak2Move, an embodied AI framework that enables intuitive voice-based control of assistive mobility devices. By integrating a large language model with object detection, semantic segmentation, SLAM, and autonomous navigation, Speak2Move creates a closed-loop pipeline connecting perception, reasoning, and action. The system incorporates a novel semantic mapping method and is deployed on a physical robot equipped with an RGB-D camera. Performance is evaluated through nine real-world navigation trials assessing spatial reasoning, obstacle avoidance, semantic understanding, and high-level task planning. A user study with five participants showed that Speak2Move consistently reduced cognitive workload during navigation, highlighting the potential of embodied AI to enhance accessibility, independence, and user experience in assistive mobility.**

I. Introduction

Mobility impairments caused by conditions such as stroke, spinal cord injuries, multiple sclerosis, amyotrophic lateral sclerosis, and cerebral palsy affect millions worldwide and significantly limit daily activities and independence. To mitigate these challenges, powered mobility devices including powered wheelchairs, walkers, scooters, and rollators have become essential for personal mobility and community participation [1]. Demand for these devices has increased in recent years due to population aging, rising disability prevalence, and advances in assistive technology [2].

Despite the wide range of control interfaces, from joysticks and switches [3] to tongue- and headrest-based systems [4] and brain-computer interfaces [5], many users continue to face difficulties operating these devices. Calibration complexity, fine motor requirements, and high cognitive effort render many interfaces inaccessible for individuals with severe impairments. Autonomous systems can reduce manual input but often struggle to interpret user intent flexibly or operate reliably in unstructured home and community settings.

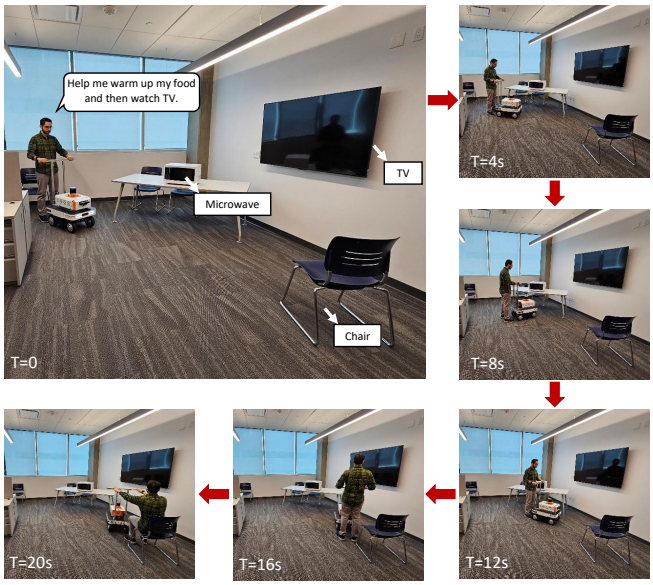


Fig. 1: Exemplary navigation from a natural speech prompt. Speak2Move uses the stored semantic and spatial maps, along with the user's spoken prompt, to identify a suitable goal pose. The system then autonomously navigates to that pose.

Each interface is generally designed for a specific residual-ability profile. Joysticks require fine upper-limb dexterity, switch scanning relies on a single consistent gross movement, tongue-based systems exploit preserved oral motion, and BCIs support users with near-total motor paralysis and intact cognition. The proposed speech-guided interface addresses a different subgroup: individuals with functional speech but insufficient manual or cranial motor control for existing methods.

Recent advances in embodied AI, which integrates multimodal perception, high-level reasoning, and physical action within robotic systems, provide a promising avenue to overcome these limitations. Incorporating foundation models, particularly large language models, enables more natural and adaptive human-robot interaction by allowing systems to interpret ambiguous commands, infer intent, and plan context-aware actions across diverse environments. These capabilities are especially relevant for assistive mobility, where rigid interfaces and precise input requirements often pose significant barriers.

*Sadra Zargarzadeh and Andor C. G. Siegers have contributed equally to this work and share first authorship.

This research was supported by the Canada Foundation for Innovation (CFI), the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Canadian Institutes of Health Research (CIHR), Alberta Innovates, and the Government of Alberta's grant to Centre for Autonomous Systems in Strengthening Future Communities

[1]A. C. G. Siegers, S. Zargarzadeh, and M. Tavakoli are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. {andor, sadra.zar, mahdi.tavakoli}@ualberta.ca

[2]A. C. G. Siegers is with the Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON, Canada.

[3]M. Tavakoli is with the Department of Biomedical Engineering, University of Alberta.

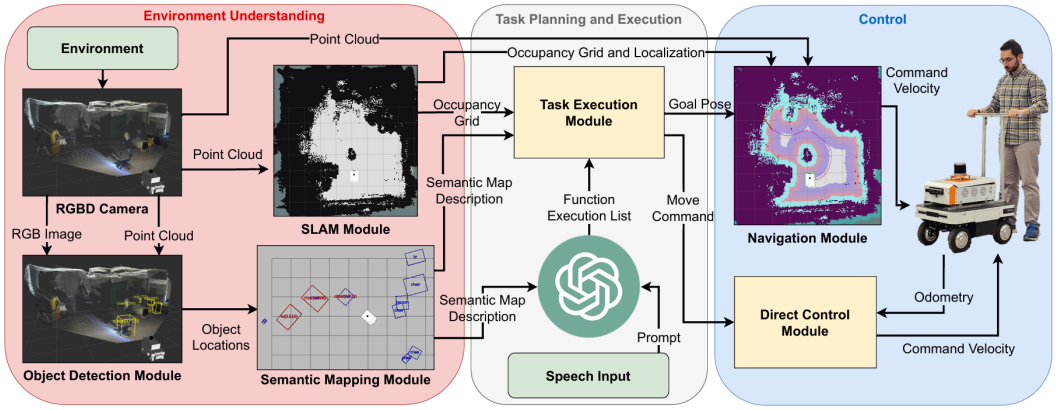


Fig. 2: *Speak2Move system architecture.* The integrated RGB-D camera streams an RGB image and a depth point-cloud. Perception modules (object detection and SLAM) produce a semantic-polar map and a metric map, respectively. These are passed to the high-level task planner, which outputs a symbolic action list. The task execution module parses that plan and invokes navigation (Nav2) and direct control modules for fine, continuous manoeuvres.

In this context, LLMs serve not just as language engines, but as bridges between human intent and robot control. For instance, the ability of an LLM to interpret "take me to the kitchen" without requiring joystick input allows users with limited motor abilities to issue intuitive, natural commands. When combined with robot perception and control systems, LLMs create an opportunity to reimagine how users interact with powered mobility devices, shifting from rigid low-level control to flexible, goal-directed autonomy.

We introduce Speak2Move, a modular embodied AI framework designed to address these needs. Speak2Move links user speech to autonomous navigation using a pipeline that combines YOLO-based visual perception, semantic-polar mapping, GPT-4o-based high-level planning, and Nav2 for low-level control. The system enables users to move through their environments by speaking simple commands, which are grounded in the robot's visual world and executed without requiring physical manipulation of a control interface.

The present approach presupposes that users possess adequate expressive speech and basic language comprehension. Individuals with profound aphasia, anarthria, or comparable speech-production deficits would require alternative input channels (e.g., eye-gaze, switch scanning, or AAC devices). Consequently, the prototype targets users whose mobility is severely limited while their spoken communication remains functional.

This work contributes to three intersecting research domains: foundation-model-based planning in robotics, system-level coordination of vision, language, and action, and assistive human-robot interaction. Through nine autonomous trials and a NASA-TLX user study, we show that Speak2Move reduces user cognitive load and improves navigation performance over traditional control methods. Unlike prior navigation pipelines, our novelty lies in adapting and integrating proven modules into a safety-conscious, semantically aware, and user-centered framework explicitly designed for assistive robotics. The framework also provides a blueprint for applying embodied AI to healthcare, extending beyond mobility aids to service, rehabilitation, and home assistance.

The remainder of the paper is organized as follows: Section II reviews related work in assistive robotics, embodied AI, and LLM-driven planning. Section III details the Speak2Move framework architecture, including semantic mapping, polar representation, LLM planning, and system integration. Section IV presents experimental results. Section V discusses limitations, and Section VI concludes the paper.

II. Related Work

Recent advances in large language models (LLMs) have motivated their integration into robotic systems, leveraging their reasoning and planning capabilities across multiple control levels, from high-level task planning to low-level control [6]–[8]. Parallel efforts have explored LLMs in assistive robotics to enhance human-robot interaction, user feedback, and system adaptability.

LLMs have been widely investigated for autonomous navigation [9]. Zhong et al. [10] employ an LLM as a low-level controller by encoding semantic and spatial information as text and directly generating velocity commands. Other approaches integrate multiple foundation models, combining LLMs with Vision Language Models (VLMs), Vision Navigation Models (VNMs), and perception networks. For example, Shah *et al.* [11] integrate an LLM, VLM, and VNM for outdoor navigation, while Qiao et al. [12] propose an open-source navigation system using chain-of-thought reasoning with spatial understanding and object recognition. Zhou et al. [13] combine a Vision-Language-Action model and a navigation policy network with an LLM to achieve state-of-the-art performance. In contrast to approaches that tightly couple LLMs with low-level control, our work adopts a modular design in which the LLM serves as a high-level task planner, interfacing with established perception and navigation modules.

LLMs have also been applied in assistive robotics to enable more flexible and intuitive interaction. Hu et al. [14] use LLMs to generate executable robot code from natural language instructions, simplifying the programming

of assistive mobile robots. Other studies emphasize safety and usability [15], such as the design guidelines proposed by Padmanabha et al. [16] for integrating LLMs into physically assistive systems. Wang et al. [17] combine object detection models with LLMs to provide navigation assistance for visually impaired users. Building on these efforts, Speak2Move integrates language, vision, and navigation in a safety-conscious, modular framework tailored specifically to assistive mobility.

III. System Architecture

The Speak2Move framework enables assistive robots to respond to natural language commands by autonomously guiding users based on spoken input. After receiving a command, the system evaluates the environment and selects an appropriate action sequence, including navigating to known objects, performing simple movements, scanning for unseen objects, or waiting. Actions are executed in a seamless, context-aware manner.

The framework comprises seven modules organized into three groups, as shown in Figure 2. Environment modeling includes a SLAM module for spatial mapping and an object detection module that supplies semantic object locations to a semantic mapping module. Task planning is handled by an LLM module that generates function calls executed sequentially by a task execution module. Control is provided through a navigation module for autonomous movement and a direct control module for low-level commands. Inter-module communication is managed via ROS2, with coordinate transformations handled by tf2.

For evaluation, the system was deployed on an AgileX Ranger Mini 3 robot equipped with a ZED 2i RGB-D camera and an NVIDIA Jetson Orin AGX. The platform supports double Ackermann and holonomic drive modes, with autonomous navigation using double Ackermann to avoid disorienting lateral motion. Full mobility remains available through direct control commands. The robot supports both manual and autonomous operation, and rear-mounted handles provide physical support and assist user guidance.

A. SLAM Module

Simultaneous Localization and Mapping (SLAM) enables autonomous robots to build maps of unknown environments while localizing within them. We integrate RTAB-Map, a widely used SLAM framework compatible with both RGB-D cameras and LiDAR sensors. Although LiDAR offers accurate 360° sensing, its cost can be prohibitive for individual users or affordable assistive devices. We therefore adopt an RGB-D camera-only solution to improve accessibility and reduce cost. The modular design of Speak2Move nonetheless allows LiDAR to be integrated when available.

Using the onboard RGB-D camera, the SLAM module generates a 2D occupancy map that identifies obstacles and free space while simultaneously localizing the robot within the environment. RTAB-Map ensures consistent transformation of sensor data into a global frame, and the resulting occupancy grid supports motion planning and execution by the task execution and navigation modules.

B. Object Detection Module

Object detection provides critical semantic awareness, enabling the framework to act as an intelligent navigational assistant. A You Only Look Once (YOLO) model is used to detect and segment objects in rectified RGB images captured by the RGB-D camera. For each detected object, the model outputs a segmentation mask, confidence score, and class label, which are processed to estimate the object's position in the global reference frame.

The segmentation mask is first morphologically eroded using OpenCV to mitigate over-segmentation by slightly reducing mask boundaries while preserving overall shape. The refined mask is then applied to the corresponding point cloud to extract 3D points on the object's surface in the camera coordinate frame. A Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [18], implemented via scikit-learn, filters spatial outliers caused by segmentation errors or sensor noise. From the resulting point cloud, a 3D bounding box is computed using the extrema along the Cartesian axes and transformed from the camera frame to the global frame. Finally, the bounding box is projected onto the 2D XY-plane and passed to the semantic mapping module together with the object's class label and confidence score.

C. Semantic Mapping Module

The semantic mapping module manages semanto-spatial information by integrating semantic and spatial outputs from the object detection module and transforming them into the robot's local coordinate frame for LLM interpretation.

For each new detection, the centroid and area of the 2D bounding box are computed. Objects with an area below $0.01\,\mathrm{m}^2$ or a confidence score below 0.7 are discarded. Remaining detections are compared with existing semantic map entries to determine whether they correspond to previously detected objects or represent new ones.

A match is declared if three conditions are met: (1) class labels are identical, (2) centroid distance is below 0.2m, and (3) the Intersection over Minimum (IoM) [19] between bounding boxes exceeds 0.2. IoM is defined as

$$IoM = \frac{|P \cap Q|}{\min\{|P|, |Q|\}} \tag{1}$$

where $|P|$ and $|Q|$ denote the areas of the new and existing bounding boxes, and $|P \cap Q|$ is their intersection. Matched objects are updated, while unmatched detections are added as new semantic entries.

To maintain accuracy in dynamic environments, semantic objects within 0.1–3 m of the robot and inside the RGB-D camera field of view are removed before adding new detections, preventing the accumulation of stale or transient objects. Finally, object centroids are transformed from the global frame to the robot's local frame and converted to polar coordinates. This compact, robot-centric representation

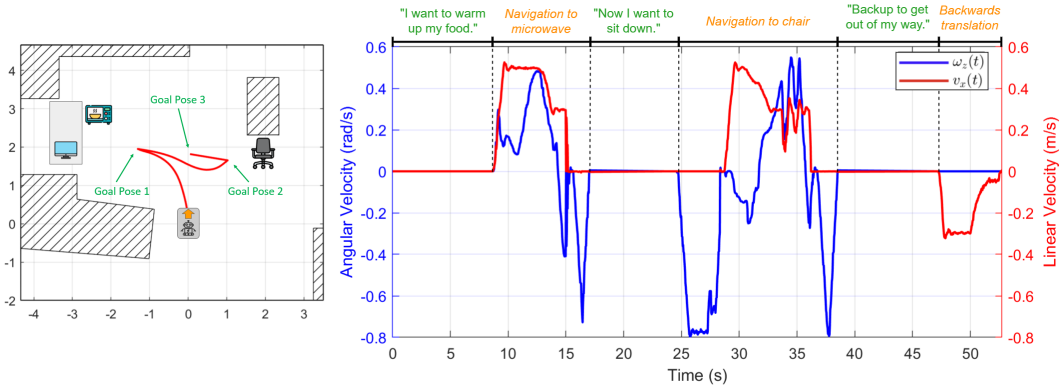
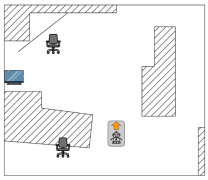
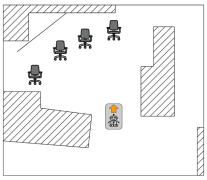


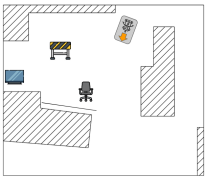
Fig. 3: Position and velocity data collected during Trial 5 (see Table I for more details). *Left* - Robot path plotted in the layout diagram along with the goal poses. All units are in meters. *Right* - Linear and angular velocity plotted vs. time. Linear velocity is plotted in red while angular velocity is plotted in blue. Prompting periods, where the user is giving input, are annotated in green, while robot actions are shown in orange.



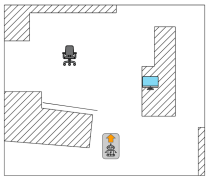
(a) **Trial 1** - Prompt:
1) *Go to the chair next to the TV.*
2) *Now turn all the way around.*



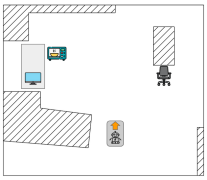
(b) **Trial 2** - Prompt:
1) *Go to the right-most chair.*



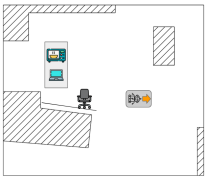
(c) **Trial 3** - Prompt:
1) *Help me get to the TV.*
2) *Now go to the chair.*



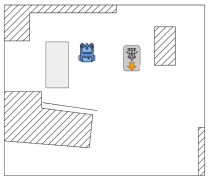
(d) **Trial 4** - Prompt:
1) *I need to sit down.*
2) *Turn around so I can sit.*
3) *Go to the TV quickly.*



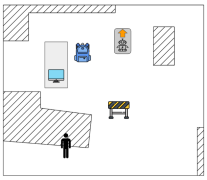
(e) **Trial 5** - Prompt:
1) *I want to warm up my food.*
2) *Now I want to sit down.*
3) *Backup to get out of my way.*



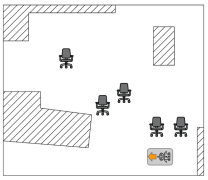
(f) **Trial 6** - Prompt:
1) *Look around.*
2) *I want to warm up my food.*
3) *Now I want to sit down.*



(g) **Trial 7** - Prompt:
1) *Find my backpack.*
2) *Take me to it.*



(h) **Trial 8** - Prompt:
1) *I need to grab something from my bag.*
2) *Great, now bring me to my friend.*



(i) **Trial 9** - Prompt:
1) *Bring me to the furthest chair.*

Fig. 4: Diagrams illustrating the nine trials carried out (see Table I for experimental results).

is provided to the LLM, enabling direct interpretation of navigation commands without additional spatial reasoning.

D. LLM Module

Recent studies show that LLMs are highly effective at generating complex plans from natural language input, motivating their use as high-level task planners. GPT-4o was selected due to its strong performance in task planning tasks [20].

The LLM module converts user speech into text using the SpeechRecognition Python library with the OpenAI Whisper API [21]. The resulting prompt is sent to GPT-4o via the OpenAI Python API, along with background context and a semantic environment description from the semantic mapping module. For reproducibility, the full set of supported functions is defined in a JSON schema in the Appendix, ensuring constrained and transparent LLM outputs. The OpenAI function calling mechanism enforces adherence to predefined function definitions (see appendix[1]). After a function sequence is produced, users may optionally review and confirm it before execution. This configurable "Safety Check" allows intervention if the output is undesirable. The planner also recognizes high-priority override commands such as "stop" and "cancel," immediately aborting any active motion or navigation task.

The LLM is intentionally limited to high-level task interpretation and semantic reasoning, while low-level execution is handled by safety-proven modules such as Nav2 and direct control. Obstacle avoidance, dynamic replanning, and path feasibility are managed by Nav2's Hybrid A* and MPPI planners rather than the LLM. Although prior work has explored richer LLM-driven planning or direct control, such approaches remain prone to hallucinations and unpredictable behaviour. In assistive robotics, even a single erroneous command can pose safety risks to users, bystanders, or the device. Constraining the LLM, therefore, improves reliability and trust while preserving natural language autonomy. This

[1]Link to Appendix and Video: `https://drive.google.com/drive/folders/1viCDhoseLOYyu1I65lvBzjDWlmBWYxqJ?usp=sharing`.

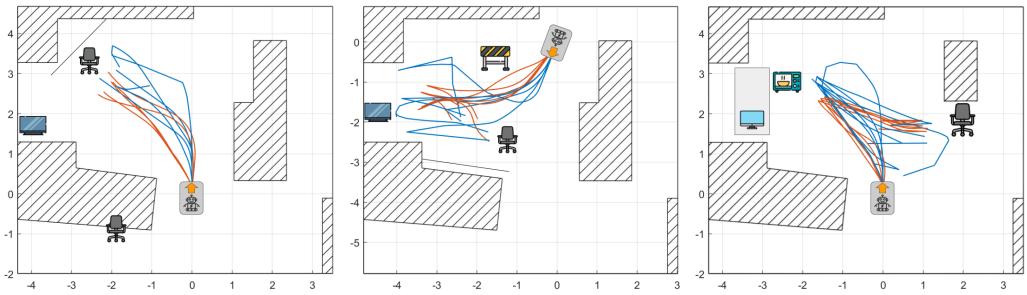


Fig. 5: User trial paths. Paths shown in blue correspond to manual control mode, while paths in red indicate autonomous operation using the Speak2Move system. All units are in meters. *Left*: Trial 1. *Center*: Trial 3. *Right*: Trial 5.

Algorithm 1 Find Goal Pose

Input:

- $\mathbf{c}_o = (c_{ox}, c_{oy})$: coordinates of the target object's centroid in the global frame.
- $\mathcal{M}$: occupancy grid produced by the SLAM module.
- $\mathcal{P}_r$: set of navigable grid cells (reachable points) in $\mathcal{M}$.
- $\mathbf{x}_R$: pose of the robot in the global frame.

Parameters:

- δ_m: side length of one grid cell in $\mathcal{M}$ (e.g., $0.05\,\text{m}$).
- r_{start}: minimum distance goal pose distance from object.
- r_{max}: maximum allowable distance for a valid goal pose.

Output:

- `GoalPose` or `None` if not found.

$\mathcal{C} \leftarrow \varnothing$ {Candidate point list}
$r_{\text{cand}} \leftarrow \infty$ {Closest candidate distance}
$\Theta \leftarrow [0,\ \delta\theta,\ 2\delta\theta,\ \ldots,\ 2\pi]$ {Where $\delta\theta = \frac{\delta_m}{r_{\text{start}}}$}
for each $\theta \in \Theta$ **do**
  $r \leftarrow r_{\text{start}}$
  while $r \leq r_{\text{max}}$ **do**
    $\mathbf{p} \leftarrow (c_{ox} + r\cos(\theta),\ c_{oy} + r\sin(\theta))$
    $\mathbf{p}_{\mathcal{G}} \leftarrow$ transform $\mathbf{p}$ to grid cell coordinates
    if $\mathbf{p}_{\mathcal{G}} \in \mathcal{P}_r$ **then**
      if $r = r_{\text{cand}}$ **then**
        Append $\mathbf{p}$ to $\mathcal{C}$
      else if $r < r_{\text{cand}}$ **then**
        $\mathcal{C} \leftarrow [\mathbf{p}]$
        $r_{\text{cand}} \leftarrow r$
      end if
      break
    end if
    $r \leftarrow r + \delta_m$
  end while
end for
if $\mathcal{C} = \varnothing$ **then**
  return `None`
end if
Compute distances $d(\mathbf{p}, \mathbf{x}_R)$ for each $\mathbf{p} \in \mathcal{C}$
$\mathbf{p}^* \leftarrow \arg\min_{\mathbf{p} \in \mathcal{C}} d(\mathbf{p}, \mathbf{x}_R)$
$\psi \leftarrow \arctan 2(c_{oy} - p_y^*,\ c_{ox} - p_x^*)$
return `GoalPose`$(\mathbf{p}^*, \psi)$

design choice serves as a safeguard rather than a limitation, as the modular architecture can expand LLM authority as verification and safety mechanisms mature.

E. Task Execution Module

The task execution module sequentially executes the function list output by the LLM module. There are four task categories that this module handles: 1) *navigation tasks*, 2) *direct control tasks*, 3) *wait tasks*, and 4) *velocity adjustments*.

1) Navigation Tasks: Before navigating to an object, a valid goal pose must be identified. All navigable points in the occupancy grid generated by the SLAM module are first extracted using a Breadth-First Search (BFS). Algorithm 1 then selects a feasible goal pose, which is passed to the navigation module if found.

2) Direct Control Tasks: For direct control tasks, the task execution module forwards the selected function and parameters, such as linear or angular distance and velocity, to the direct control module. Supported commands include `move_forward`, `move_backward`, `strafe_right`, `strafe_left`, `turn_right`, and `turn_left`. The `scan_environment` command is also available, enabling the robot to rotate in place to capture a panoramic view with the RGB-D camera and detect semantic objects in previously unmapped areas.

3) Wait Tasks: Wait tasks instruct the system to pause for a specified duration before proceeding. During this wait period, all subsequent commands are blocked, ensuring sequential execution of the task plan.

4) Velocity Adjusments: Navigation and movement commands use dynamically adjustable maximum velocities (defaults: 0.5m/s forward, 0.3m/s backward, and $\pi/2$rad/s angular). Users can modify these at runtime via internal task execution and navigation parameters. Requested velocities are validated against safety ranges ($[0.1, 1.0]$ m/s forward, $[0.1, 0.7]$ m/s backward, $[0.5, 2.4]$ rad/s angular); values outside these bounds are clamped to the nearest limit.

F. Navigation Module

Autonomous navigation is essential for guiding users to detected objects. The framework employs the Navigation 2 (Nav2) [22] stack to provide reliable autonomous navigation. Nav2 was selected for its proven safety, robustness across diverse platforms, and seamless integration with ROS2 and

RTAB-Map, reducing the risks associated with LLM-based control.

Navigation parameters were tuned to match the specific robot and application. The Hybrid A* global planner and Model Predictive Path Integral local planner [23] were used, as they support the robot's Ackermann motion model and provide dynamic obstacle avoidance required for operation in changing environments.

G. Direct Control Module

After autonomous navigation, the robot may not be optimally positioned. For example, when approaching a chair, it may stop directly in front of it and obstruct seating, motivating the need for user-directed repositioning. The direct control module enables forward and backward translation, lateral motion, and in-place rotation, giving users full access to the robot's mobility.

To ensure smooth and accurate execution of movement commands, motion profiling is applied to generate stable trajectories. Two intermediary functions, $u(t)$ and $\lambda(t)$, are used to represent the proportion of the total linear or angular displacement completed at time t, measured from the start of the motion command $t = 0$. These functions are defined as follows for linear and angular motion, respectively:

$$u(t) = \frac{\sqrt{(x(t) - x(0))^2 + (y(t) - y(0))^2}}{d_{target}} \tag{2}$$

$$\lambda(t) = \frac{|\theta(t) - \theta(0)|}{\theta_{target}} \tag{3}$$

where $(x(t), y(t))$ represents the robot's position in the global reference frame at time t, $\theta(t)$ is the yaw orientation of the robot in the global reference frame at time t, d_{target} is the user-specified target distance and θ_{target} is the user-specified target angle.

A trapezoidal deceleration ramp is used to reduce overshoot from abrupt velocity changes by gradually slowing the robot as it approaches its goal. Without this profile, deceleration would begin only at the target, allowing momentum to carry the robot past the intended stop. The motion profile is defined for linear and angular velocities as follows:

$$v_{cmd}(t) = \begin{cases} v_{max} & \text{if } u(t) \leq 1 - R \\ \frac{1-u(t)}{R} * v_{max} & \text{if } 1 - R < u(t) \end{cases} \tag{4}$$

$$\omega_{cmd}(t) = \begin{cases} \omega_{max} & \text{if } \lambda(t) \leq 1 - R \\ \frac{1-\lambda(t)}{R} * \omega_{max} & \text{if } 1 - R < \lambda(t) \end{cases} \tag{5}$$

where $v_{cmd}(t)$ and ω_{cmd} are the respective linear and angular command velocities at time t, v_{max} and ω_{max} are the user-defined maximum linear and angular velocities, and R is the proportion of the motion during which deceleration is undergone (e.g., when $R = 0.3$ deceleration only occurs in the final 30% of the movement). R can be adjusted depending on the desired motion and max speed. The "backwards translation" section of Figure 3 illustrates an exemplary linear movement.

TABLE I: Autonomous Navigation Trial Results

Trial #	Distance Travelled (m)	Angle Travelled (rad)	Drive Time (s)	Trial Time (s)	Safety Check
1	2.80	13.6	19.47	42.29	Enabled
2	1.79	0.52	5.01	12.64	Enabled
3	2.89	8.99	16.32	29.4	Enabled
4	2.46	11.2	15.95	38.9	Enabled
5	6.07	6.72	29.46	52.74	Disabled
6	4.93	24.9	37.51	59.74	Disabled
7	0.59	14.5	24.61	33.03	Disabled
8	3.27	4.01	17.64	38.55	Disabled
9	5.10	3.66	16.61	24.05	Disabled

IV. Experiments and Results

To assess the capabilities of the Speak2Move system, two real-world experiments were conducted: *autonomous navigation trials*, which assessed the system's detection and navigational performance, and a *human-robot comparative usability study*, which examined the system's impact on users' perceived task load during interaction.

A. Autonomous Navigation Trials

To assess detection, mapping, and navigation performance, structured trials were conducted in a controlled lab environment with predefined object arrangements forming distinct semantic scenes, as shown in Figure 4. The scenarios were intentionally kept simple, as the goal was not to maximize environmental complexity but to evaluate whether an LLM-driven navigation system is suitable for assistive use and whether it can improve user experience. This design prioritized feasibility and usability over stress-testing the navigation pipeline.

In Trials 1, 2, 3, 4, 5, 8, and 9, semantic and spatial mapping were enabled, and the robot was manually driven beforehand to build a comprehensive semantic map. Trials 6 and 7 omitted this preliminary mapping phase. Each trial began with a natural language prompt provided by the experimenter, with all prompts shown in Figure 4. When the optional Safety Check was enabled in Trials 1 through 4, the proposed execution plan was reviewed and approved before execution. Prompts were issued sequentially until all actions were completed.

The results demonstrate that Speak2Move supports spatial reasoning in Trials 1, 2, and 9, obstacle avoidance in Trials 3, 8, and 9, and task planning across all trials. The system also adapts dynamically in Trials 4 and 8 and successfully grounds natural language commands in semantic understanding to perform object-based navigation in Trials 5, 6, and 8.

B. Human-Robot Comparative Perceived Workload Study

To evaluate the effect of Speak2Move on cognitive load and navigation performance, a comparative usability study was conducted with five able-bodied male participants (age = 26±4.6 years, height = 177.6±6.2 cm, weight = 76.8±8.8). All participants had intact spoken English and normal or corrected-to-normal vision and hearing. The study was approved by the University of Alberta Research Ethics Board

TABLE II: Results of User Trials. The *manual* control mode refers to trials in which participants operated the robot using a remote controller, while *Speak2Move* control mode refers to trials conducted with our Speak2Move system enabled. The Raw Task Load Index (RTLX) measures subjective workload across six categories: Mental Demand (MD), Physical Demand (PD), Temporal Demand (TD), Performance (Pe), Effort (Ef), and Frustration (Fr), each rated on a 20-point scale. TLX Total denotes the aggregated and normalized task load score for each trial.

Trial	Participant	Control Mode	Distance Travelled (m)	Angle Travelled (rad)	Raw TLX MD	PD	TD	Pe	Ef	Fr	TLX Total	Task Load Reduction
1	1	Manual	3.95	**13.49**	17	11	11	16	18	16	0.742	70.4±14.3%
		Speak2Move	**3.47**	13.82	8	5	11	1	7	5	**0.308**	
	2	Manual	4.04	**13.44**	13	15	15	4	5	19	0.592	
		Speak2Move	**3.48**	13.64	1	2	1	1	1	2	**0.067**	
	3	Manual	5.30	14.42	9	6	9	8	9	13	0.450	
		Speak2Move	**4.16**	**5.06**	4	4	4	3	3	7	**0.208**	
	4	Manual	4.74	14.66	7	1	1	1	13	4	0.225	
		Speak2Move	**3.96**	**14.43**	1	1	1	1	1	1	**0.050**	
	5	Manual	4.93	13.73	5	3	6	1	7	8	0.250	
		Speak2Move	**3.51**	**13.64**	1	2	2	1	1	1	**0.067**	
3	1	Manual	**4.09**	**3.47**	17	15	11	3	16	17	0.658	53.3±24.6%
		Speak2Move	4.71	17.78	14	14	9	3	12	14	**0.550**	
	2	Manual	7.40	**4.65**	20	19	16	15	19	19	0.900	
		Speak2Move	**5.77**	13.82	3	2	5	2	4	3	**0.158**	
	3	Manual	8.43	**4.47**	11	10	7	7	10	13	0.483	
		Speak2Move	**4.25**	5.21	3	3	4	16	3	3	**0.267**	
	4	Manual	7.00	**17.98**	5	5	1	1	5	2	0.158	
		Speak2Move	**6.55**	20.11	1	2	1	1	1	1	**0.058**	
	5	Manual	6.21	**4.51**	8	8	6	3	9	8	0.350	
		Speak2Move	**5.05**	5.14	3	2	2	6	2	2	**0.142**	
5	1	Manual	8.39	13.43	20	20	12	16	18	20	0.883	64.4±14.0%
		Speak2Move	**6.96**	**4.77**	3	1	10	1	5	7	**0.225**	
	2	Manual	8.98	19.80	17	19	13	8	17	14	0.733	
		Speak2Move	**6.28**	**4.15**	3	4	3	3	3	4	**0.167**	
	3	Manual	7.81	**13.00**	8	9	10	9	9	10	0.458	
		Speak2Move	**6.47**	16.94	5	3	3	14	4	3	**0.267**	
	4	Manual	8.26	15.62	7	3	1	1	3	2	0.142	
		Speak2Move	**6.07**	**4.29**	1	1	1	1	1	1	**0.050**	
	5	Manual	7.61	**4.34**	4	4	4	2	6	2	0.183	
		Speak2Move	**6.78**	5.19	1	1	1	3	1	1	**0.067**	

(Pro00139670). Each participant completed Trials 1, 3, and 5, as described in Section IV-A, both with and without Speak2Move.

Participants were given three minutes to practice manual robot control at the start of the first trial. For each task, they were provided with the full prompt and instructed to complete it using manual control while standing behind the robot, leaning on it for support to simulate a smart walker. After completing each task, the Raw Task Load Index (RTLX) was administered to measure perceived workload across six dimensions: Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, and Frustration. Ratings on a 20-point scale were aggregated and normalized to produce a composite workload score between 0 and 1.

The environment was then reset, Speak2Move was activated, and a semantic map was generated through pre-scanning. After a brief overview of voice interaction, participants repeated the same tasks using Speak2Move, followed by a second RTLX assessment. This procedure was repeated for all participants and trials. Results are summarized in Table II, with corresponding robot trajectories shown in Figure 5.

Statistical analysis using the Wilcoxon signed-rank test showed that all participants reported lower NASA-TLX scores when using Speak2Move, yielding a highly significant one-tailed result ($W = 0$, $p < 0.001$). A repeated-measures ANOVA on TLX difference scores found no significant effect of task scenario, $F(1.01, 4.05) = 2.13$, $p > 0.05$, with mean TLX reductions of 0.70, 0.53, and 0.64 for Trials 1 through 3. These results indicate that Speak2Move consistently reduces cognitive workload across different navigation tasks.

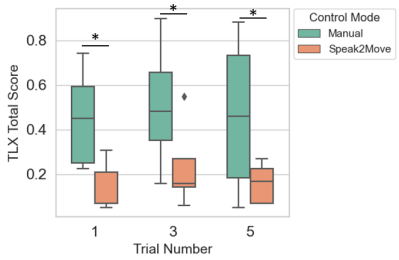


Fig. 6: Comparison of TLX Total values of manual control vs Speak2Move in the three trials.

V. Discussion and Limitations

A natural language-driven navigation system integrated into assistive devices, such as Speak2Move, can reduce cognitive load, improve navigation efficiency, and lessen reliance on caregivers. This capability may substantially benefit individuals with severe mobility or cognitive impairments by increasing independence and supporting broader social participation. The Speak2Move framework is adaptable to assistive platforms including smart walkers, powered wheelchairs, and lower-limb exoskeletons, as its LLM-based planning module is hardware-agnostic. Provided that a device exposes basic locomotion APIs, the system can generalize without retraining or reconfiguration.

The framework also offers potential for transparent decision-making. Although not fully explored in this study, the LLM planner can be prompted to verbally explain its reasoning, for example “I am navigating to the kitchen by

moving toward the detected refrigerator," which may improve user trust and interpretability.

Despite promising results, several limitations remain. The current system focuses on object-based navigation, limiting long-horizon autonomy in complex environments such as outdoor urban settings. However, exposing additional robot APIs (e.g., manipulation, elevator use, and door operation) could enable execution of multi-step tasks such as "pick up my medication, then escort me to the reception desk" without retraining. Manual joystick control remains the clinical standard for assistive mobility and was therefore used as the baseline. Few assistive systems jointly integrate vision and language as in Speak2Move, making direct comparisons difficult while highlighting the novelty of the approach. The closed-set object detector restricts recognition to predefined classes; open-set detection could support a broader range of real-world objects. Finally, reliance on spoken commands limits accessibility for users with significant speech or language impairments.

VI. Conclusions and Future Work

This paper introduced Speak2Move, a modular framework that enables assistive robots to execute natural speech commands through integrated vision, language, and action. By combining semantic mapping, object detection, SLAM, navigation, and LLM-based planning, the system supports intuitive mobility-device control. A polar representation of object locations improves spatial reasoning, and the framework was validated through robot experiments and a user study.

Results showed reduced NASA-TLX workload and improved navigation efficiency, demonstrating the potential of Speak2Move to support independent mobility for individuals with physical or cognitive impairments. Future work will quantify caregiver burden reduction and mobility outcomes, compare LLM-generated plans against user- and expert-defined baselines, and extend autonomy through outdoor navigation, improved indoor localization, open-set object detection, and multi-camera perception. These advances will move Speak2Move toward a general-purpose embodied AI platform for assistive autonomy across diverse environments.

References

[1] V. Kumar, T. Rahman, and V. Krovi, "Assistive devices for people with motor disabilities," *Wiley Encyclopedia of Electrical and Electronics Engineering*, vol. 22, 1997.

[2] R. E. Cowan, B. J. Fregly, M. L. Boninger, L. Chan, M. M. Rodgers, and D. J. Reinkensmeyer, "Recent trends in assistive technology for mobility," *Journal of neuroengineering and rehabilitation*, vol. 9, pp. 1–8, 2012.

[3] D. A. Sanders, M. Langner, and G. E. Tewkesbury, "Improving wheelchair-driving using a sensor system to control wheelchair-veer and variable-switches as an alternative to digital-switches or joysticks," *Industrial Robot: An International Journal*, vol. 37, no. 2, pp. 157–167, 2010.

[4] N. Sebkhi *et al.*, "Evaluation of a head-tongue controller for power wheelchair driving by people with quadriplegia," *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 4, pp. 1302–1309, 2021.

[5] G. Kucukyildiz, H. Ocak, S. Karakaya, and O. Sayli, "Design and implementation of a multi sensor based brain computer interface for a robotic wheelchair," *Journal of Intelligent & Robotic Systems*, vol. 87, pp. 247–263, 2017.

[6] S. Zargarzadeh, M. Mirzaei, Y. Ou, and M. Tavakoli, "From decision to action in surgical autonomy: Multi-modal large language models for robot-assisted blood suction," *IEEE Robotics and Automation Letters*, 2025.

[7] Y.-J. Wang, B. Zhang, J. Chen, and K. Sreenath, "Prompt a robot to walk with large language models," in *2024 IEEE 63rd conference on decision and control (CDC)*. IEEE, 2024, pp. 1531–1538.

[8] S. Zargarzadeh, J. Okanlawon, M. Mirzaei, M. Mohammadi, and M. Tavakoli, "Enhanced reasoning and task planning for surgical autonomy using multi-modal large language models with gradual learning," *Biomimetic Intelligence and Robotics*, p. 100277, 2026.

[9] J. Lin *et al.*, "Advances in embodied navigation using large language models: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2311.00530

[10] Z. Zhong, Y. He, P. Li, F. Yu, and F. Ma, "A language-driven navigation strategy integrating semantic maps and large language models," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 9753–9760.

[11] D. Shah, B. Osiński, b. ichter, and S. Levine, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, K. Liu, D. Kulic, and J. Ichnowski, Eds., vol. 205. PMLR, 14–18 Dec 2023, pp. 492–504. [Online]. Available: https://proceedings.mlr.press/v205/shah23b.html

[12] Y. Qiao *et al.*, "Open-nav: Exploring zero-shot vision-and-language navigation in continuous environment with open-source llms," 2025. [Online]. Available: https://arxiv.org/abs/2409.18794

[13] G. Zhou, Y. Hong, Z. Wang, X. E. Wang, and Q. Wu, "Navgpt-2: Unleashing navigational reasoning capability for large vision-language models," in *Computer Vision – ECCV 2024*, A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, Eds. Cham: Springer Nature Switzerland, 2025, pp. 260–278.

[14] Z. Hu *et al.*, "Deploying and evaluating llms to program service mobile robots," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2853–2860, 2024.

[15] M. Chalaki, A. Zakerimanesh, A. Soleymani, V. Mushahwar, and M. Tavakoli, "Vision-based fuzzy control system with intention detection for smart walkers: Enhancing usability for stroke survivors with unilateral upper limb impairments," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 1997–2003.

[16] A. Padmanabha *et al.*, "Voicepilot: Harnessing llms as speech interfaces for physically assistive robots," in *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3654777.3676401

[17] H. Wang *et al.*, "Visiongpt: Llm-assisted real-time anomaly detection for safe visual navigation," 2024. [Online]. Available: https://arxiv.org/abs/2403.12415

[18] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, p. 226–231.

[19] F. Vogel *et al.*, "Utilizing 2d-region-based cnns for automatic dendritic spine detection in 3d live cell imaging," *Scientific Reports*, vol. 13, 11 2023.

[20] R. Mon-Williams, G. Li, R. Long, W. Du, and C. G. Lucas, "Embodied large language models enable robots to complete complex tasks in unpredictable environments," *Nature Machine Intelligence*, pp. 1–10, 2025.

[21] A. Zhang, "Speech recognition (version 3.11)," 2017, https://github.com/Uberi/speech_recognition).

[22] S. Macenski, F. Martin, R. White, and J. Ginés Clavero, "The marathon 2: A navigation system," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[23] S. Macenski, M. Booker, and J. Wallace, "Open-source, cost-aware kinematically feasible planning for mobile and surface robotics," *Arxiv*, 2024.