Reinforcement Learning Approaches for Autonomous Wheelchair Navigation and Docking

Dev Patel, Mechatronics and AI – B.Eng
Western University

Abstract

Autonomous docking and navigation technologies hold great potential to enhance the mobility, safety, and independence of powered wheelchair users. For individuals with limited mobility small barriers such as precisely aligning a wheelchair for charging, seating transfers, or entering confined spaces, can significantly hinder daily tasks. Traditional navigation methods often depend on external infrastructure or global positioning systems, which are unreliable or unavailable in indoor and dynamic human environments. Thus, intelligent solutions capable of perceiving and adapting to their surroundings are essential.

This work explores reinforcement learning (RL) as a pathway toward fully autonomous wheelchair navigation and docking. We present a progressive study investigating RL-based control policies across increasing levels of environmental complexity. In the initial phase, we implemented Twin Delayed Deep Deterministic Policy Gradient (TD3) privileged goal information, achieving high docking success with no/minimum obstacles. To move closer to real-world autonomy, the second phase replaced explicit goal coordinates with ArUco marker–based visual targets, introducing partial observability. Using Recurrent Proximal Policy Optimization (RPPO), the agent learned memory-based navigation, achieving 72% success in minimally cluttered scenes with no explicate goal information.

These findings highlight the promise of RL for enabling robust, vision-based docking behaviors in assistive mobility platforms. Ongoing work aims to extend these methods to cluttered and dynamic indoor spaces, leverage inverse reinforcement learning for reward discovery, and deploy the learned policies onto real-world wheelchair hardware to further promote user independence and accessibility

Introduction

Public transit systems often require powered wheelchair users to manually align their chair with designated docking stations for safety during travel. This process can be especially difficult for individuals with severe motor impairments, such as those with limited control over joystick inputs. Precise alignment within narrow tolerances is required, and even small errors may prevent docking from completing successfully. As a result, many users must rely on external assistance from drivers or attendants, reducing independence and increasing boarding times. An autonomous docking system that can position the wheelchair reliably and safely would greatly improve accessibility, efficiency, and user autonomy in transit.

While autonomous navigation is an active area of research, the docking problem poses unique challenges: it demands centimeter-level accuracy, must operate in crowded and cluttered environments, and must handle partial observability when the docking target is occluded. Reinforcement learning(RL) provides a promising framework, but reward shaping, perception under uncertainty, and safety constraints remain open research problems.

In this work we progressively increase the realism of the docking task. We begin with a privileged-information setting, where the agent is given the exact goal coordinates. Using Twin Delayed Deep Deterministic Policy Gradient (TD3), the wheelchair achieves approximately 91% success in an environment where the exact distance and angle to the goal are known. We then remove access to the true goal location, introducing an ArUco fiducial marker to provide vision-based goal detection. This transforms the problem into one of partial observability, where the goal may be out of view. To address this, we employ Recurrent Proximal Policy Optimization (RPPO), which leverages temporal memory to achieve reliable docking even when the goal is intermittently visible. Finally, to reduce reliance on manually engineered reward functions, we explore inverse reinforcement learning(IRL), which infers rewards from expert trajectories.

Objectives

The long-term objective of the proposed research program is to develop autonomous navigation and docking technologies for powered wheelchairs that enhance user mobility, independence, and safety in complex real-world environments. The solution aims to operate reliably without requiring infrastructure modifications or external localization systems, adapting dynamically to user preferences and environmental variations.

Objectives (Cont'd)

To achieve this vision, several key research challenges must first be addressed, leading to the following short-term objectives:

Objective 1: Develop a simulation environment and reinforcement learning based control framework for autonomous docking and navigation.

This framework will integrate motion planning, control, and perception into a unified learning process that allows the wheelchair to dock precisely and safely in diverse indoor environments, including narrow passages and cluttered spaces.

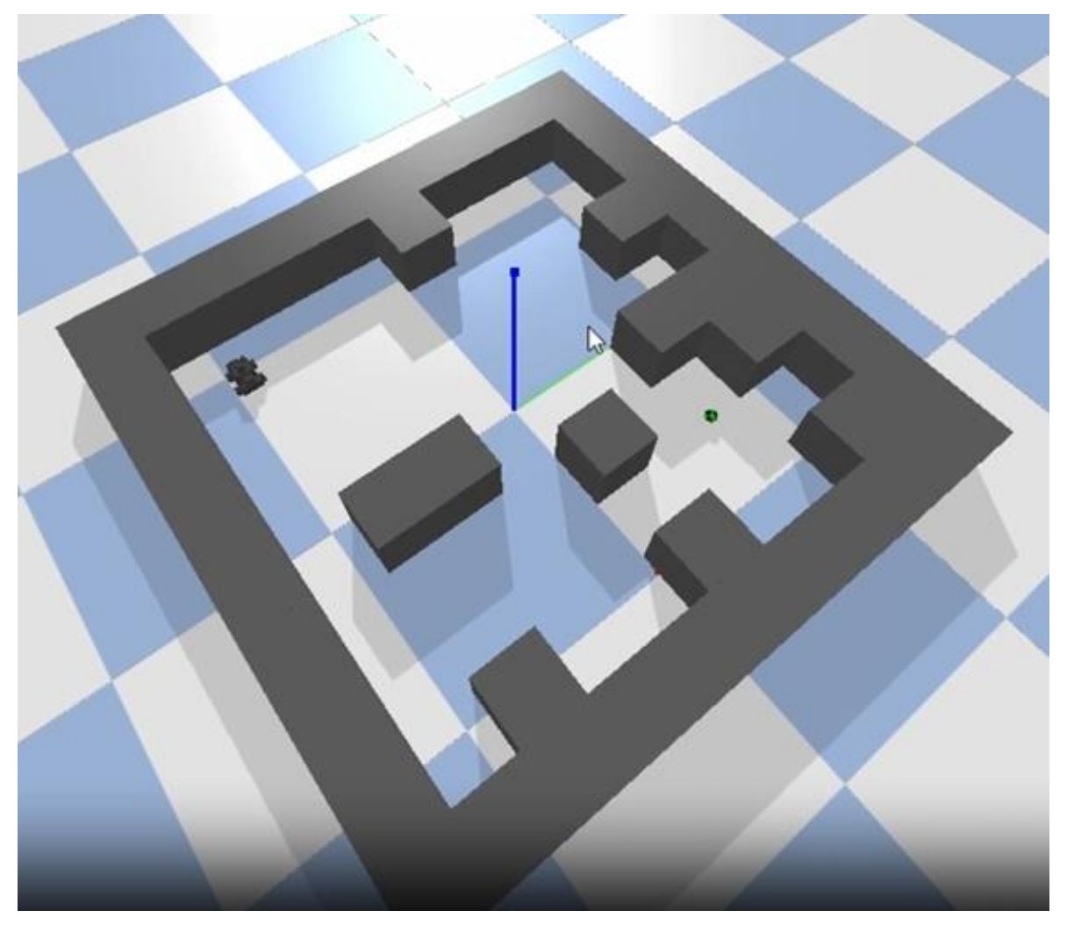
Objective 2: Design robust perception and environment understanding algorithms using low-cost sensors.

Sensor fusion methods combining LIDAR, camera, IMU, and encoder data will be developed to enable accurate environment representation and target detection (e.g., docking stations or ArUco markers) without reliance on expensive hardware or external infrastructure.

Objective 3: Establish a sim-to-real transfer pipeline for reliable real-world deployment.

A simulation-based training and validation framework will be built using Isaac Sim and PyBullet to bridge the gap between virtual learning and physical performance, ensuring the trained policies can generalize effectively to real wheelchair platforms under dynamic conditions.

Methods



Simulation Environment - We tested out two physics-based simulators (PyBullet / IsaacSim) to prototype navigation and docking behaviors. While Isaac Sim presents a more realistic graphic and physics simulation, pybullet allowed light-weight computation intended for prototyping. The map size is a 10x10m arena with walls surrounding the area. The wheelchair is modeled as a differential-drive platform, and maximum linear and angular velocities are restricted to satisfy the limits of safety and stability. The docking station consists of an ArUco tag mounted near a randomized target location and angle. Obstacles are represented as boxes/cylinders with randomized layouts and cluster densities. To improve robustness in real-world deployment, sensor drift was added to the IMU as well as actuator lag and wheel slip. With these additions, the model can build its own internal controller to compensate for sensor drift and accurately reach target speeds and rotations.

Observation and Action Space

Phase 1 – In Phase 1, the observation vector o_t combines (i) the scalar goal distance d_g and relative bearing ϕ_g , (ii) the normalized linear And angular velocities $\frac{v}{v\,max}$, $\frac{\omega}{\omega\,max}$, and (iii) an N-beam planar LiDAR scan:

TABLE I
PHASE 1 OBSERVATION COMPONENTS

Component	Symbol	Dimension
Goal distance and bearing	$d_g,\ \phi_g$	2
Normalized velocities	$v/v_{\max},\ \omega/\omega_{\max}$	2
LiDAR ranges	$r_{1:N}$	N
Total	o_t	$N+4$

Phase 2 (ArUco-based,Partial Observability) - In Phase 1, the explicit goal coordinates are removed. Instead, the agent perceives the docking station through a fiducial ArUco tag and must rely on memory when the tag is not visible. The observation vector o_t concatenates four groups of features:

1) Tag state [d_{norm}, *sin* ϕ, f]: inverted normalized tag Distance (d_{norm} ∈ [0, 1] where 0 =far, 1 =near), sin of angle to tag bearing detected from position in camera frame, and a binary visibility flag f ∈ {0, 1} Where 0 represents no tag in view and 1 vice-versa. When the tag is not in value, these values default to [1, 0, 0] and memory features are relied upon. The tag is detected from camera footage using a pre-trained cv2 model to avoid training a new model using the camera as input.

TABLE II
PHASE 2 OBSERVATION COMPONENTS

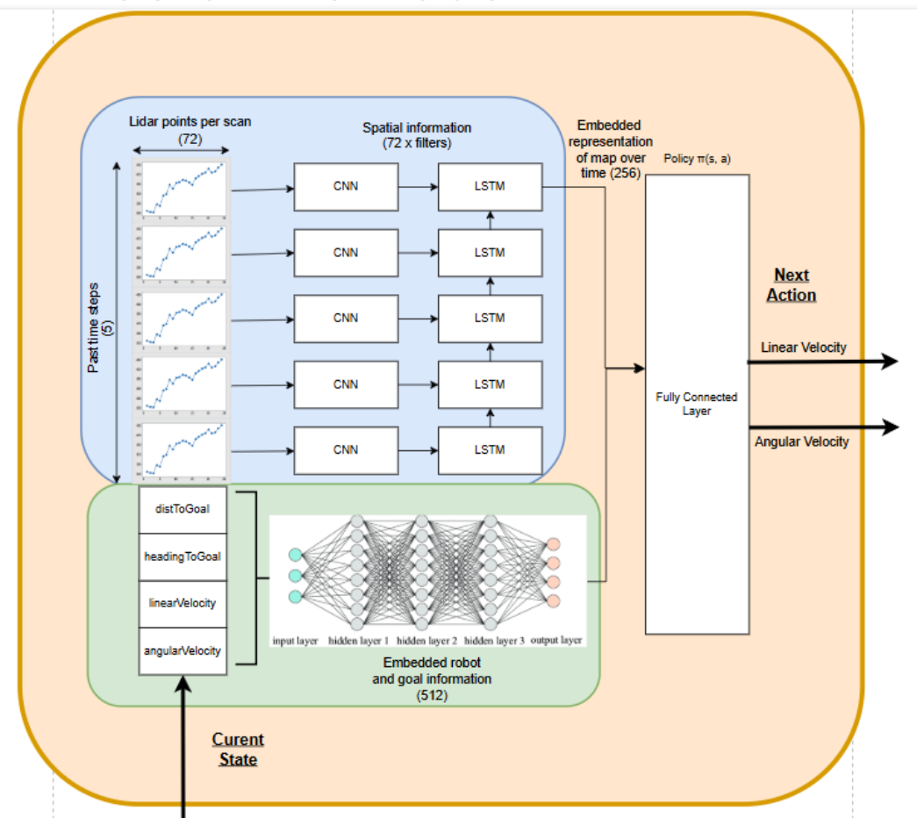
Component	Symbols	Dim.
Tag state	$d_{\mathrm{norm}}, \sin\phi, f$	3
Velocities	$v/v_{\max}, \omega/\omega_{\max}$	2
Heading + memory	$\sin\psi, \cos\psi, \Delta\psi,$ $\sin\phi_{\mathrm{last}}, \cos\phi_{\mathrm{last}},$ $d_{\mathrm{last}}, t_{\mathrm{seen}}$	7
LiDAR framestack	$r_{1:FN}$	FN
Total	o_t	$M = FN + 12$

Methods (Cont'd)

2) Velocity features: same as phase 1. **3) Heading and memory features:** *sin* ψ, *cos* ψ (Global robot angle), $\Delta\psi$ (angular velocity), last-seen tag bearing encoded as sin(ϕ_{last}), cos(ϕ_{last}), last-seen normalized tag distance d_{last}, and normalized time since tag was last observed t_{seen} ∈ [0, 1]. **4) LiDAR stack**: F recent LiDAR scans, each of N beams. A typical configuration uses N = 60 beams and F = 5 stacked frames, yielding M = 12 + 300 =312 dimensions.

Action Space - At each timestep the policy outputs a normalized 2D vector $a_t = [\hat{v}, \hat{w}]$, $\hat{v}, \hat{w} \in [-1,1]$, representing desired linear and angular velocities. These values are scaled to hardware limits and applied to the wheelchair motors.

Reinforcement Learning Algorithms - We evaluated two families of methods:



1) TD3 [1] with shaped rewards under privileged observations as a baseline for continuous control.
800x600 Fully Connected Layer
2) Recurrent PPO (RPPO) building on PPO [2]: an LSTM layer processes feature embeddings to cope with partial observability induced by occlusions and narrow field of view. Custom Network architecture:

Reward Shaping - We design a shaped reward that balances task progress, target visibility, motion smoothness, and collision avoidance, with terminal bonuses/penalties for goal completion and crashes.

$$r_t = \underbrace{w_p \alpha_t \Delta d_t}_{\text{progress}} + \underbrace{w_v \mathbb{1}_{\text{vis}}(1 - |\sin\theta_t|)}_{\text{target-in-view}} + \underbrace{w_s\left(v_t^{\text{lin}} - \tfrac{1}{4}|v_t^{\text{ang}}|\right)}_{\text{smoothness}} + \underbrace{w_\ell\left(\max\left(0, \frac{r_n - d_t^{\min}}{r_n}\right)\right)^2(-1)}_{\text{near-obstacle penalty}}$$

The first term rewards for decrease in distance to goal. When the ArUco tag is visible, the progress scale is $\alpha_t = 1$; otherwise, it is reduced to 0.3, discouraging blind movement when the goal is not observed. The second term for visibility peaks when the goal is centered on the camera and decreases as the goal leaves the field of view. The third term encourages smooth forward motion while discouraging excessive turning. The final term applies a quadratic penalty that activates only inside radius r_n growing rapidly as the shortest lidar point d_t^{min} approaches the collision radius to discourage traveling dangerously close to obstacles. $[w_p, w_v, w_s, w_l]$ are weights that are tuned to achieve desired performance.

Results and Conclusions

The **Phase 1 model**, trained with full goal information and shaped rewards, demonstrated strong performance in structured environments. It achieved a **100% docking success rate** over 1000 simulations in obstacle-free settings and **91% success** with a 10% obstacle probability. Failures primarily occurred when the wheelchair encountered large obstacles blocking the direct path to the goal. In such cases, the optimal behavior would be to reverse and replan; however, the agent instead continued forward due to the reward function's bias toward minimizing goal distance. This behavior highlights that the model prioritized **path efficiency** over **risk-aware decision-making**, a limitation arising from its reliance on immediate progress rewards rather than long-term spatial reasoning.

The **Phase 2 model** introduced **partial observability** by removing explicit goal coordinates and replacing them with **ArUco-based visual detection**. This increased task complexity, requiring the model to interpret visual input and maintain a memory of past observations. Under these conditions, it achieved a **92% success rate** in open environments and **66%** with a 10% obstacle probability. While it demonstrated reliable visual navigation when the target was visible, performance degraded once the goal was out of view. The model exhibited short-term planning tendencies, struggling to remember the environment layout or safely navigate around occluded obstacles, often leading to collisions or inefficient detours.

Overall, these results demonstrate the **feasibility of reinforcement learning–based control** for wheelchair docking but also emphasize the need for better **spatial reasoning and memory mechanisms**. Ongoing work focuses on encouraging **active exploration** when visual cues are lost and integrating **SLAM-based mapping** or learned memory modules to build a global understanding of the environment. Future iterations will explore **hybrid learning approaches**, combining reinforcement learning with imitation and visual-mapping techniques to improve **robustness, safety, and generalization** in dynamic indoor settings.

Impact

This research supports the development of intelligent powered wheelchairs capable of autonomous navigation and precise docking, addressing one of the most significant barriers to mobility and independence for individuals with physical disabilities. By leveraging reinforcement learning, vision-based sensing, and low-cost hardware, the project enables assistive technologies that adapt to user needs and operate safely in everyday indoor environments. The outcomes of this work have broad societal benefits, enhancing accessibility, reducing caregiver dependency, and improving quality of life for users. In the long term, the developed methods can extend to other forms of assistive and service robotics, contributing to more inclusive, equitable, and human-centered technological innovation in healthcare and community mobility.

References

[1] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 1582–1591.
[2] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.