

Article

Design of an Assisted Driving System for Obstacle Avoidance Based on Reinforcement Learning Applied to Electrified Wheelchairs

Federico Pacini , Pierpaolo Dini  and Luca Fanucci 

Department of Information Engineering, University of Pisa, 56122 Pisa, Italy; luca.fanucci@unipi.it (L.F.)

* Correspondence: federico.pacini@phd.unipi.it

Abstract: Driving a motorized wheelchair is not without risk and requires high cognitive effort to obtain good environmental perception. Therefore, people with severe disabilities are at risk, potentially lowering their social engagement, and thus, affecting their overall well-being. Therefore, we designed a cooperative driving system for obstacle avoidance based on a trained reinforcement learning (RL) algorithm. The system takes the desired direction and speed from the user via a joystick and the obstacle distribution from a LiDAR placed in front of the wheelchair. Considering both inputs, the system outputs a pair of forward and rotational speeds that ensure obstacle avoidance while being as close as possible to the user commands. We validated it through simulations and compared it with a vector field histogram (VFH). The preliminary results show that the RL algorithm does not disruptively alter the user intention, reduces the number of collisions, and provides better door passages than a VFH; furthermore, it can be integrated on an embedded device. However, it still suffers from higher jerkiness.

Keywords: assistive technology; wheelchairs; assisted driving; artificial intelligence; reinforcement learning; simulation



Citation: Pacini, F.; Dini, P.; Fanucci, L. Design of an Assisted Driving System for Obstacle Avoidance Based on Reinforcement Learning Applied to Electrified Wheelchairs. *Electronics* **2024**, *13*, 1507. <https://doi.org/10.3390/electronics13081507>

Academic Editors: Victor M. Becerra, Elías Revestido Herrero and Francisco Jesus Velasco

Received: 27 March 2024

Revised: 11 April 2024

Accepted: 12 April 2024

Published: 16 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Motivation

The World Health Organization has calculated that approximately 1.3 billion people, constituting 16% of the world's population, confront substantial disability [1]. Within this group, more than 200 million individuals encounter difficulties with mobility. The Convention for the Rights of Persons with Disabilities (CRPD) of 2007 fostered a greater appreciation for personal autonomy, as is evident in the evolving disability policies [2]. Numerous studies emphasized the influence of mobility on social interactions, emphasizing that limited mobility leads to a reduction in social engagement, subsequently adversely affecting overall well-being [3–13]. In this regard, people with motor skill impairments use wheelchairs for daily activities. Among them, some users are not able to self-propel a manual wheelchair. For this group, a power wheelchair is often the solution to increase their independence. However, driving a motorized wheelchair is not without risk and requires high cognitive effort to obtain a good environmental perception to complete tasks such as obstacle avoidance and path planning. A study reported that more than 54% of people with wheelchairs have had at least one accident in the previous 3 years [14–20]. Therefore, individuals experiencing low vision, visual field limitations, spasticity, tremors, or cognitive impairments encounter significant challenges when operating a power wheelchair. Consequently, they either face restrictions from operating the wheelchair, leading to a substantial reduction in mobility and social interactions, or they become more susceptible to accidents. Robotic assistive solutions have emerged as a promising avenue to enhance the safety of wheelchair navigation. Exploiting the significant advancements in autonomous vehicles,

the adaptation of analogous techniques to treat wheelchairs as mobile robots enables the feasible development of fully autonomous or cooperative systems.

Nevertheless, electric-powered wheelchairs come with additional requisites, such as ensuring the safety and comfort of the users they carry. However, completely autonomous wheelchairs release the users from any cognitive and muscular effort, resulting in a progressive degradation of their residual capacities. Therefore, a cooperative system is introduced, wherein the system responds to user input, which is typically provided through a joystick (with potential extension to other input devices), unless a hazardous situation is detected. Upon identifying danger, the system activates a Deep Reinforcement Learning (DRL)-based motion policy. This policy guides the wheelchair to navigate away from the perceived danger while striving to align with the user's input as much as possible. The rationale behind minimizing the disparity between user input and policy output is to create a system that encourages users to autonomously handle risky situations. Simultaneously, the system maintains a secure environment, allowing users to experiment and enhance their driving skills with a safety net in place.

1.2. Related Works

Numerous attempts have been made to devise shared driving systems for wheelchair navigation. Traditional approaches encompass both exclusively obstacle avoidance systems and hybrid variants.

Control systems based on fuzzy algorithms are widely used in a wide range of applications, including mobility assistive devices, such as electric wheelchairs for the disabled. These fuzzy algorithms offer an effective method for managing the complexity of dynamic and unpredictable systems, such as those encountered in assisted mobility. In the context of electrified wheelchairs for the disabled, the objective of the control system is to ensure safe and comfortable driving for the user. Fuzzy algorithms allow for the incorporation of expert knowledge and linguistic rules to flexibly interpret input variables, such as the wheelchair position, environmental conditions, and user preferences, to generate appropriate control outputs. An example of the application of fuzzy algorithms could be the control of the speed and direction of the wheelchair based on the presence of obstacles in the surrounding environment. Using sensors to detect obstacles, the fuzzy system can evaluate the degree of danger and adjust the speed and trajectory of the wheelchair to avoid collisions or dangerous situations. Furthermore, fuzzy algorithms allow for the easy integration of feedback from the user, for example, through intuitive interfaces, such as joysticks or voice commands. This allows for the customization of control based on user capabilities and preferences, further enhancing the driving experience [21–25].

In cases where a global navigation satellite system (GNSS) suffers a partial loss of information, such as during transit in densely wooded urban environments or inside buildings, active detection and localization using LiDAR sensors can become fundamental to ensure safe and effective navigation. LiDAR (light detection and ranging) sensors are devices that use laser pulses to precisely measure the distance between the sensor and surrounding objects. In the context of UAVs, LiDAR sensors can be mounted on the aircraft itself to map the surrounding environment and detect obstacles in real-time. The concept of "active depth cluster detection" refers to the LiDAR system's ability to identify and distinguish between different objects in the environment, thus forming a "cluster" of data points associated with each object. This information is extremely valuable for the control of electric wheelchairs, as it allows the system to accurately perceive its surroundings and make informed navigation decisions. When a GNSS is available, data from this system can be integrated with that of LiDAR to further improve the accuracy and robustness of localization and navigation. However, in the case of partial information loss in a GNSS, the localization system must rely mainly on LiDAR data to determine the position and orientation of the UAV. Using advanced data fusion algorithms, such as extended Kalman filters or particle filters, information from different sensors (such as LiDAR and a GNSS)

can be efficiently integrated to obtain accurate estimates of the position and orientation of the UAV despite the partial loss of information [26–29].

Regenerative braking control is an advanced technology used in electric wheelchairs to improve safety and efficiency when driving on downhill roads. This system uses the principle of energy regeneration, allowing the wheelchair to convert kinetic energy during braking into electrical energy that can be stored and reused to extend the battery life. When the electric wheelchair is on a downhill road, the regenerative braking control comes into action to regulate the speed and prevent excessive acceleration. This is particularly important to ensure user safety and prevent dangerous situations, such as tipping over or losing control of the wheelchair. The new regenerative braking control system is designed to be highly sensitive and responsive, dynamically adapting to changes in terrain and road conditions. Using advanced sensors to detect slopes, inclinations, and wheelchair speeds, the system can modulate braking in real time to maintain a safe and comfortable speed during descent. Furthermore, regenerative braking control can be integrated with other safety and driver assistance systems, such as rollover prevention systems or imminent collision warnings. This integration allows for coordinated and synergistic management of the various functions, guaranteeing an optimal level of safety and performance when driving on downhill roads. Finally, the new regenerative braking control system can be customized to the user's specific preferences and needs, allowing for more intuitive and comfortable driving. This includes the ability to adjust the braking sensitivity, the maximum speed allowed when going downhill, and other custom settings [30–33].

In purely obstacle avoidance systems, the objective is for the system to navigate around obstacles while adhering to the user's input according to a specified policy [34–40]. In hybrid systems, the primary goal of obstacle avoidance is combined with supplementary tasks, such as target tracking, wall following, door crossing, and reaching points of interest [41–44]. In principle, numerous algorithms designed for robot navigation can be adapted to wheelchair navigation by considering the wheelchair as a differential robot. However, the domain of wheelchairs inherently introduces additional factors that necessitate adjustments to these algorithms. Notably, one crucial factor is the user's comfort, which must be taken into account. One widely employed approach is the dynamic windows approach [45–50]. This method addresses constraints arising from limited velocities and accelerations. Specifically, it periodically evaluates a short time interval during which the robot can move and calculates trajectory approximations within this interval by considering circular curvatures. This results in a two-dimensional search space comprising translational and rotational velocities. The search space is further refined by including only velocities that enable the robot to stop safely and be reachable in the next interval. The resultant velocities form a dynamic window centered around the current velocities of the robot in the velocity space. From the admissible velocities within this dynamic window, the combination of translational and rotational velocities is selected by maximizing an objective function. While there have been a few attempts to apply this approach to wheelchairs [51–58], notable drawbacks include increased computational costs and latency, particularly as the environment becomes more complex. Another widely employed approach for mobile robots is the potential fields method (PFM), with a prominent implementation known as the virtual force field (VFF). In this method, each obstacle generates a repulsive force directed toward the wheelchair. While this approach has demonstrated effective obstacle avoidance behavior in open spaces, challenges arise when dealing with close obstacles or navigating through narrow paths between walls. When the wheelchair moves precisely in the middle of an aisle, there are no issues. However, upon nearing one of the adjacent walls, it generates a repulsive force that pushes the wheelchair toward the opposite wall, which, in turn, produces a similar effect. Under specific conditions, these actions can initiate a vicious cycle, leading to system instability [59–63]. The vector field histogram (VFH) algorithm, which succeeded the VFF, gradually addresses some of its limitations in its original form and its evolution (VFH+ and VFH*). Thanks to its inherent representation of intermediate data in the polar histogram, the VFH allows the robot to navigate through narrow passages

and doorways. Additionally, VFH+ introduces a safety distance, enabling the robot to maintain a secure distance from obstacles.

Furthermore, the VFH+ algorithm eliminates the fluctuations in the steering direction that affected the VFF, thanks to a smoothing approach applied to the polar histogram itself [64–71]. However, the VFH also has its drawbacks. The algorithm requires information about the robot's footprint, which, in our case, is for a wheelchair. To guarantee user safety, the footprint must surpass the actual size. As the algorithm requires a circular shape and the wheelchair possesses a rectangular footprint, a circular footprint is adopted for the algorithm. The diameter of this circular footprint is set to be greater than the diagonal of the rectangle, with the specific ratio depending on the desired safety level to be ensured. This increased footprint poses challenges when navigating through narrow spaces, like doorways, as the wheelchair's larger footprint may make it seem unable to pass through the gap. Moreover, a notable strength of the VFH method with autonomous and semi-autonomous mobile robots becomes a drawback when a user is onboard. The VFH enables fast traversal through cluttered environments by avoiding obstacles with minimal speed reduction. However, when a user is onboard, this behavior is perceived as a sudden and unpredictable change in direction, potentially leading to uncomfortable and even dangerous situations, such as overturning. Consequently, significant modifications are necessary for traditional algorithms to address these issues, adding complexity to the solution [72–77]. With the increasing popularity and capabilities of deep learning methods, there has been a development of robot navigation methods utilizing neural networks. Convolutional neural networks (CNNs) have been employed in various attempts, leveraging their effectiveness in image processing. Specifically, CNNs excel in visual obstacle detection, yielding successful outcomes in this context [78–84]. When integrated into a larger software ecosystem, their outputs can serve as inputs to traditional motion planners, enabling the computation of a path that avoids the detected obstacles. Despite the theoretical effectiveness of this approach, the adoption of a CNN-based wheelchair navigation system has not been widespread. This limitation derives from the inherent nature of CNNs, which require a large dataset of images for training. Given the scarcity of suitable datasets for wheelchair navigation, researchers often need to collect their samples before conducting training. While promising results have been demonstrated [85,86], the labor-intensive data collection process has impeded widespread adoption. Another approach involves the utilization of recurrent neural networks (RNNs). RNNs serve as optimization approximation solvers for complex constrained optimization problems [87] or can be combined with traditional methods, such as artificial potential fields (APFs) [88,89]. In the context of wheelchair navigation, a system with a discrete action space was implemented using RNNs [90].

The necessity for mathematical modeling of the environment, the limited real-time performance of algorithms, local locking issues, and other challenges associated with previous methods have prompted the exploration of new approaches. In 2013, the concept of DRL was introduced, demonstrating the capability of a system to learn to play Atari games by inputting the environment and training it with positive or negative rewards based on chosen actions [91–94]. Subsequently, numerous studies have utilized DRL approaches, with some successful attempts focusing on obstacle avoidance [95]. These attempts involved the application of Double Deep Q Learning (DDQN), which is the successor of Deep Q Learning (DQN), which addresses the issue of action value overestimation. However, due to DDQN's limitation in handling discrete action spaces, other methods based on the actor–critic paradigm have been proposed. In [96], the Deep Deterministic Policy Gradient (DDPG) was successfully applied to robot obstacle navigation. An improved version of DDPG, namely, twin delayed deep deterministic policy gradients (TD3), was applied in [97], while [98] presented an asynchronous advantage actor–critic (A3C), showcasing the possibility of training parallel agents with lower computational costs compared with traditional DQN methods. The strength of DRL approaches lies in not needing to create an accurate model because, under certain circumstances, after a large amount of training, the

network will be able to map an input to the acceptable output. However, DRL techniques are not without drawbacks [99].

Issues such as sparse rewards, poor generalization, and the simulation-to-reality gap can impact the algorithms and necessitate specific countermeasures [100].

A common aspect across various techniques is the utilization of sensors to acquire information about the surrounding environment. This aspect becomes particularly crucial for DRL. Training agents in a physical environment is often time-consuming and poses risks. Consequently, researchers commonly opt to train agents in a simulation environment and subsequently transfer this learning to the physical realm. However, a substantial gap typically exists between the two domains. To mitigate this gap, various techniques can be employed [100]. Among these techniques, the use of sparse laser-ranging data was shown to minimize the disparity between simulations and reality [101]. Additionally, to broaden the distribution of the state space in the pool of samples and enhance the adaptability of the agent to new situations, one approach involves randomizing the starting point and the goal with an incremental variance at the onset of each episode [102]. Given the promising results demonstrated by DDPG in obstacle avoidance, our work was founded on TD3. TD3 builds upon DDPG but introduces enhancements to address certain drawbacks, such as potential instability and the significant reliance on identifying optimal hyper-parameters for a given task, which can be attributed to an overestimation of the Q-value. This improvement was achieved by incorporating a second critic network, delaying the update of the actor, and introducing action noise regularization. In terms of sensor selection, we opted for 2D LiDAR, striking a suitable balance between sensor reliability and cost.

1.3. Author Contributions

The primary contributions of this study are outlined as follows:

- Formulated a cooperative system that prioritizes adhering to the user's input while ensuring safety conditions and providing a way to escape from dangerous situations;
- Developed a neural network architecture based on the twin delayed deep deterministic policy gradient (TD3) for wheelchair navigation in a continuous action space;
- Established an infrastructure for hyper-parameter optimization and parallel agent training in deep reinforcement learning based on the Robotic Operating System (ROS) and Gazebo.

The rest of the article is organized into the following sections: Section 2, which describes the simulation environment; Section 3, in which the structure of the system based on the reinforcement learning paradigm is presented; Section 4 presents the simulation results in the various tested scenarios; Section 5 reports the discussion and critical review of the results obtained. At the end, Section 6 reports our conclusions and ideas for future developments.

2. Simulation Environment and Setup

2.1. Wheelchair

In previous research, we performed applied research to develop a plug-and-play kit for transforming a manual wheelchair into an automated one with the constraint of no irreversible modification on the chassis. Hence, we started with a physical wheelchair and we concentrated our efforts on designing and validating the mechanical and electronic model of the system in charge of electrifying it. On such occasions, we did not include any human driving safety mechanism. For this reason, in this work, we wanted to design a system that could be eventually placed on top of our previous system to cover the missing driving safety features.

2.2. System Definition

From a robotic point of view, a wheelchair can be thought of as a bicycle system if the effect of the motion of the pivoting front wheels is neglected. By assuming this, the wheelchair can be treated as a bicycle with two controlled drive wheels. Hence, the kinematic equations ruling the wheelchair movement are

$$\Omega_o = (w_A - w_B) \frac{R}{2d}, \quad (1)$$

$$V_o = (w_A + w_B) \frac{R}{2} \quad (2)$$

where Ω_o and V_o are, respectively, the rotational and linear speeds of the point O , which is assumed to be the barycenter of the wheelchair; R is the wheelchair driving wheel radius; and d is the distance between the driving wheels and the point O .

The geometrical representation of the wheelchair can be found in Figure 1.

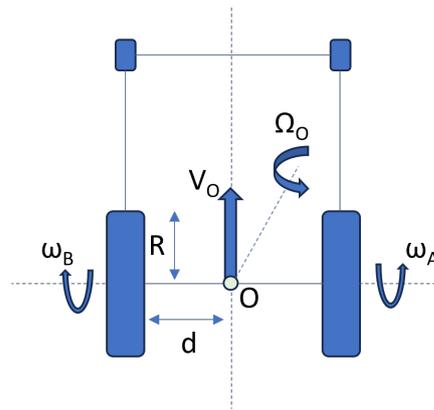


Figure 1. Geometrical representation of the wheelchair.

In electrical wheelchairs, the user sets a target linear speed V_{ref} and a target rotational speed Ω_{ref} via a mapping between the user input, often expressed via a joystick and the controller. In our configuration, the agent, based on the perceived obstacle distribution and the user input, computes the forward V_{ref} and angular Ω_{ref} target speeds. At a lower level, the motors need a current to spin. To translate a reference value to an actual motor command, the controller performs a control loop by regulating the motor current until the actual value is not equal to the target one.

For the sake of simulation, we needed a 3D model that can be used in the simulation environment. For this reason, a 3D model was designed. Because there is no standard dimension of a wheelchair, we decided to model our wheelchair according to the dimensions in Table 1.

Table 1. Dimensions of the wheelchair components.

Parameter	Value
Width	0.7 m
Length	1.1 m
Drive wheel radius	0.27 m
Drive wheel width	0.05 m
Castor wheel radius	0.17 m

Thanks to our advanced implementation in the wheelchair model within the ROS/Gazebo environment (Figure 2), we could ensure an incredibly realistic and functional simulation. A crucial part of this implementation concerns the consideration of actuator dynamics and drive control. Regarding actuator dynamics, we integrated accurate models that take into account the intrinsic characteristics of electric motors and other components. Each actuator was precisely modeled to reflect its response to the applied voltage, maximum achievable speed, available torque, and inertia. Additionally, we considered phenomena such as energy loss due to friction and component inertia to ensure a faithful simulation [103–107].

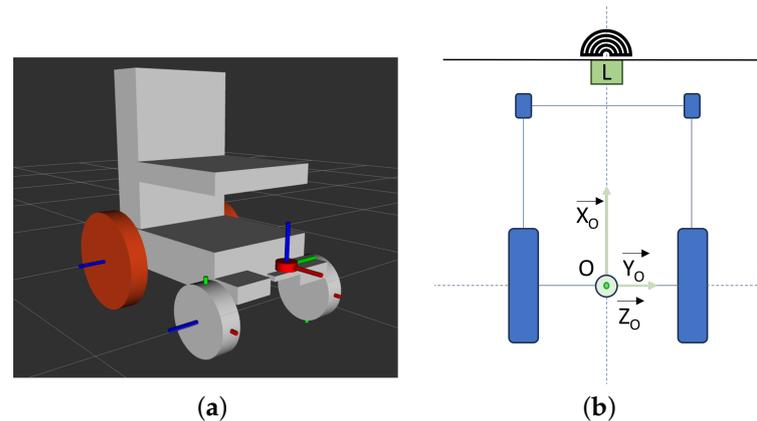


Figure 2. Wheelchair representation (a). In the mathematical representation (b), O is the origin of the system reference; L is the LiDAR source; and \bar{X}_O , \bar{Y}_O , and \bar{Z}_O are the system axes.

In terms of the drive control, we implemented a sophisticated system that translates user commands, such as those from a joystick, into physical actions. Using advanced control algorithms, like PID control and model predictive control, we could ensure a quick and stable response of the model to user requests. This control not only handles acceleration, deceleration, and turning but also deals with trajectory maintenance and obstacle management to ensure safe and smooth driving [108–113]. All of this was seamlessly integrated into the ROS and Gazebo ecosystem, enabling efficient communication between the wheelchair model and other components of the robotic system. Thanks to this integration, we could simulate and evaluate the performance of the model under a wide range of environmental conditions and operational situations, ensuring that our virtual wheelchair is at the forefront in terms of realism and functionality [114,115]. Our advanced implementation of actuator dynamics and drive control within ROS/Gazebo allowed us to offer an exceptionally accurate and functional simulation of the wheelchair, furthering our commitment to creating innovative and cutting-edge solutions in the field of robotics and mobility assistance.

2.3. Setup

The navigation system logic was implemented by making use of the Robotic Operating System (ROS). The ROS is a well-established set of software libraries and tools for building robot applications. For simulation purposes, Gazebo was selected. Gazebo is an open-source, well-established simulator in the field of robotics that provides a robust physics engine, as well as convenient programmatic and graphical interfaces. The neural networks were implemented by making use of PyTorch, which is a machine learning framework used in applications that span from natural language processing to robotics. The joystick used during the simulation was a simple two-axis joystick. The learning of local navigation through DRL and some further tests were performed on a computer equipped with an NVIDIA GTX 2060 graphics card, 16 GB of RAM, and an Intel Core i7-9750 H CPU. For the sake of collecting the inference time of the actor network, an NVIDIA JetsonNano 2 GB Developer Kit was employed. For testing purposes, a single user without disabilities was selected as the driver. The testing phase was articulated in three experiments: the first and the second one, both comprising three tests, aimed to compare our algorithm with a well-known obstacle avoidance algorithm (Vector Field Histogram (VFH)); the third one aimed to evaluate its usability on embedded, namely, constrained, devices. Each test was executed at least three times and the shown values are the mean of the collected ones. When significant, the standard deviation is also shown. For the collection of the inference time, the test was executed 100 times.

3. Reinforcement Learning Algorithm Architecture

Considering the environment as the union of the wheelchair, the user, and the surrounding obstacles, we considered the system we wanted to design as the components in

charge of yielding the correct physical commands to let the wheelchair follow the user input while avoiding the obstacles. Referencing Figure 3, conventional local robot navigation frameworks typically feature a local mapping component that takes sensor readings as the input and produces a local map. Another component, which is commonly known as local path planning, utilizes this local map to generate references guiding the robot to its destination while avoiding obstacles. Subsequently, these references are translated into low-level commands, such as phase current values, through an additional module often referred to as the controller. In our system, both the local mapping and local path planning were replaced by an agent utilizing a DRL method.

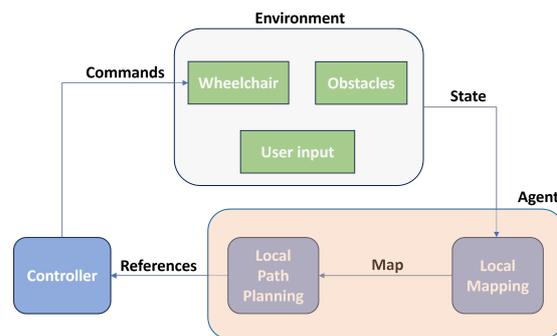


Figure 3. Traditional and DRL wheelchair navigation frameworks. In DRL, the agent learns by experience how to generate the appropriate references depending on the received state.

RL, which is inspired by animal learning in psychology, learns optimal decision-making strategies from experience. RL defines any decision maker as an agent and everything outside the agent as the environment. The agent aims to maximize the accumulated reward and obtains a reward value as a feedback signal for training through interaction with the environment. The interaction process between the agent and environment can be modeled as a Markov decision process comprising the essential elements S , A , R , and P ; S is the state of the environment, A is the action taken by the agent, R is the reward value obtained, and P is the state transition probability. The agent's policy τ is the mapping from the state space to the action space. When in the state $s_t \in S$, the agent takes action $a_t \in A$, and then transfers to the next state s_{t+1} according to the state transition probability P while receiving reward value feedback $r_t \in R$ from the environment. Although the agent receives instant reward feedback at every time step, the goal of RL is obtaining the largest long-term cumulative reward value R_t rather than short-term rewards. By introducing the discount factor $\gamma \in [0, 1)$, we can express the return value as follows:

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad (3)$$

In our scenario, the agent takes the form of a twin delayed deep deterministic policy gradient (TD3) architecture, which was trained within a simulated environment to execute user commands while navigating around obstacles. The TD3 operates as an actor–critic network, enabling actions in a continuous action space. An actor–critic model comprises two networks: the actor and the critic.

The actor determines the action to be taken, and the critic provides feedback to the actor regarding the quality of the action and suggestions for adjustment. The actor's learning is based on a policy gradient approach. In contrast, the critic assesses the action's quality by computing the value function. In the TD3 network, there are two critics to mitigate Q-value overestimation. In reinforcement learning terms, the critics evaluate the Q-value of the state–action pair $Q(s, a)$. Both critic networks share the same structure, but their parameter updates are delayed, allowing for divergence in parameter values. According to the TD3 architecture, the final critic's output selects the minimum Q-value from both critic networks to curb the overestimation of the state–action pair value. The local environment is described by a LiDAR placed in front of the wheelchair. Its 180° field-of-view range data

are condensed into buckets to reduce the size for efficiency. This information, combined with the current forward and angular speed, as well as the user’s desired speeds, forms the state S . This state is then input to the actor, which aims to produce the actual forward and rotational speeds. To enable the critics to evaluate the actor’s performance, both the actor’s output and the locally collected information are provided as input to the critics. Detailed information about the neural networks can be found in Figure 4.

As a design choice, for optimization reasons, we opted to confine the actor’s output values a_1 and a_2 to the range of $[-1, 1]$. These values are scaled within the range of forward and rotational speed values before being emitted by the agent. However, due to the LiDAR’s limited 180° field-of-view, as a safety precaution, we refrained from generating backward movements since we lack information about that specific area. Consequently, the ultimate output a from the actor is as follows:

$$a = \left[\frac{(a_1 + 1)}{2}, a_2 \right] \tag{4}$$

To allow the evaluation of the actor’s performance by the critics, the policy is rewarded based on the following function:

$$r(s_t, a_t) = \begin{cases} -K_c & \text{if collision happens} \\ \frac{\alpha R_s + \beta R_v + \theta R_\omega}{\alpha + \beta + \theta} & \text{otherwise} \end{cases} \tag{5}$$

where

$$R_s = K_s * \tanh(K_{ss} * (d - D_s)), \tag{6}$$

$$R_v = \frac{K_v}{|v_u - v_a|}, \text{ if } |v_u - v_a| \geq \epsilon \text{ else } \frac{K_v}{\epsilon}, \tag{7}$$

$$R_\omega = -K_\omega * |\omega_u - \omega_a| \tag{8}$$

with the convention that K_c is the reward for a collision, d is the distance between the wheelchair and the closest obstacle, D_s is a constant indicating the safety distance that the wheelchair should maintain to the closest obstacle, K_{ss} is a constant indicating the distance safety strictness, K_s is a constant to adjust the scale value of the distant safety reward R_s , v_u is the linear speed required by the user, v_a is the linear speed outputted by the actor, K_v is a constant to adjust the scale value of the forward speed reward R_v , ϵ is a small constant to limit the R_v value, ω_u is the angular speed required by the user, ω_a is the angular speed outputted by the actor, and K_ω is a constant to adjust the scale value of the angular speed reward R_ω . The objective is to ensure that $R_s + R_v + R_\omega \ll K_c$. This is crucial because if the penalty for collisions is not sufficiently substantial, there is a risk that the accumulated reward value in a single episode might surpass the penalty value. Such a scenario could lead to a positive total reward for the episode, which is undesirable. In this situation, the critic networks might incorrectly interpret that, despite a collision occurring, the overall action received a positive reward, potentially downplaying the significance of the collision.

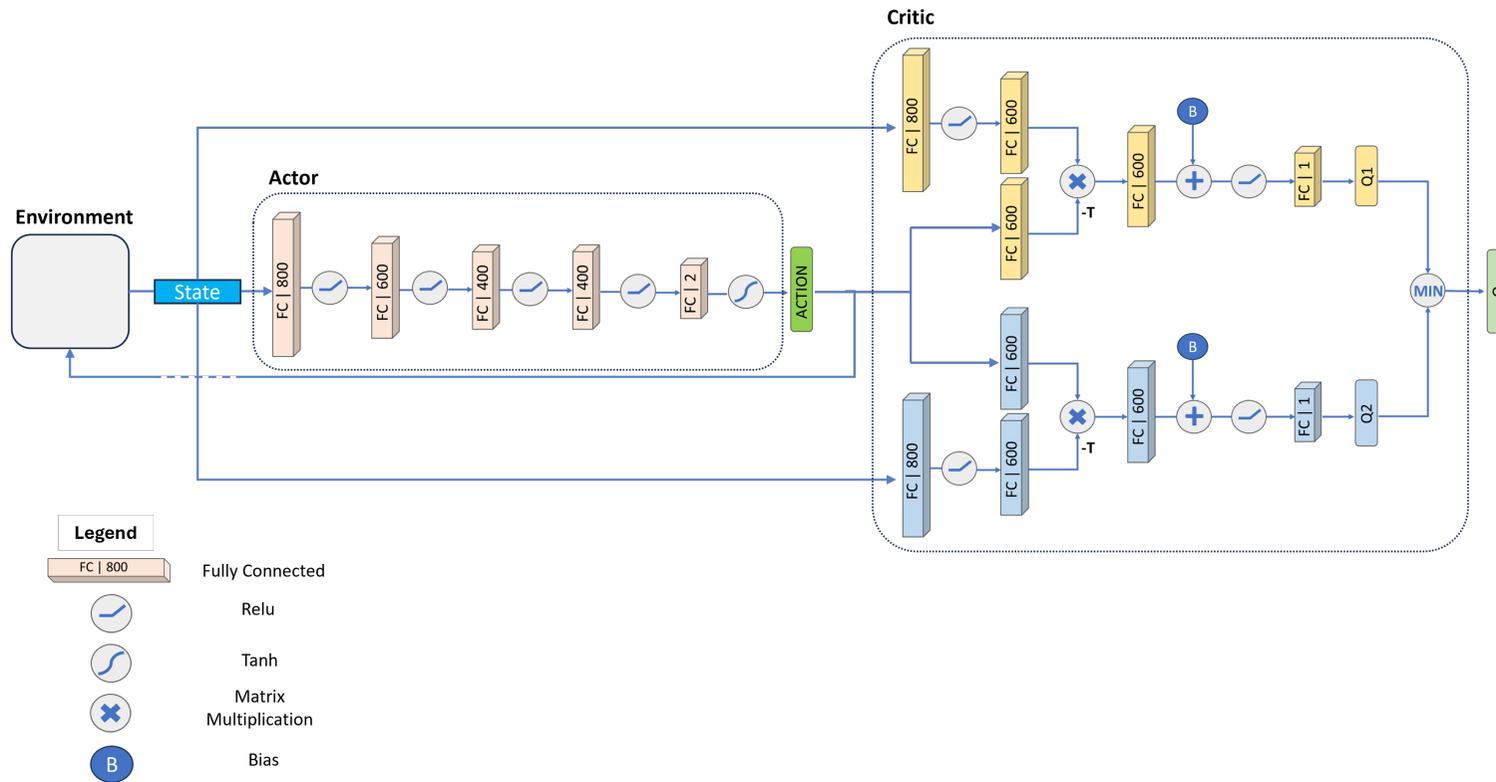


Figure 4. Implementation details of TD3 neural networks.

This misinterpretation could cascade into subsequent decisions, prioritizing strict adherence to user commands at the expense of collision avoidance, which is a failure to meet one of the agent's requirements. The reward R_ω increases as the difference between the desired angular speed and the calculated one decreases. Similarly, R_v increases with an asymptotic behavior at K_v/ϵ as the difference in forward linear speed diminishes. The behavior of R_s was designed such that it yields a positive value as the wheelchair stays farther from obstacles than D_s , and a negative value as the distance becomes smaller than D_s . By adjusting the constant values K_s , K_{ss} , and D_s , the value of R_s can be tailored to specific needs. In particular, tuning the value of K_s allows for the inflation or deflation of the reward value, K_{ss} regulates the "strictness" by controlling the gradient of the shape, and D_s determines the distance at which the reward should discourage further approach to the obstacle. The impact of these constant values on the \tanh function can be observed visually in Figure 5. To regulate the relative importance of each reward component among the others, the parameters α , β , and θ are used.

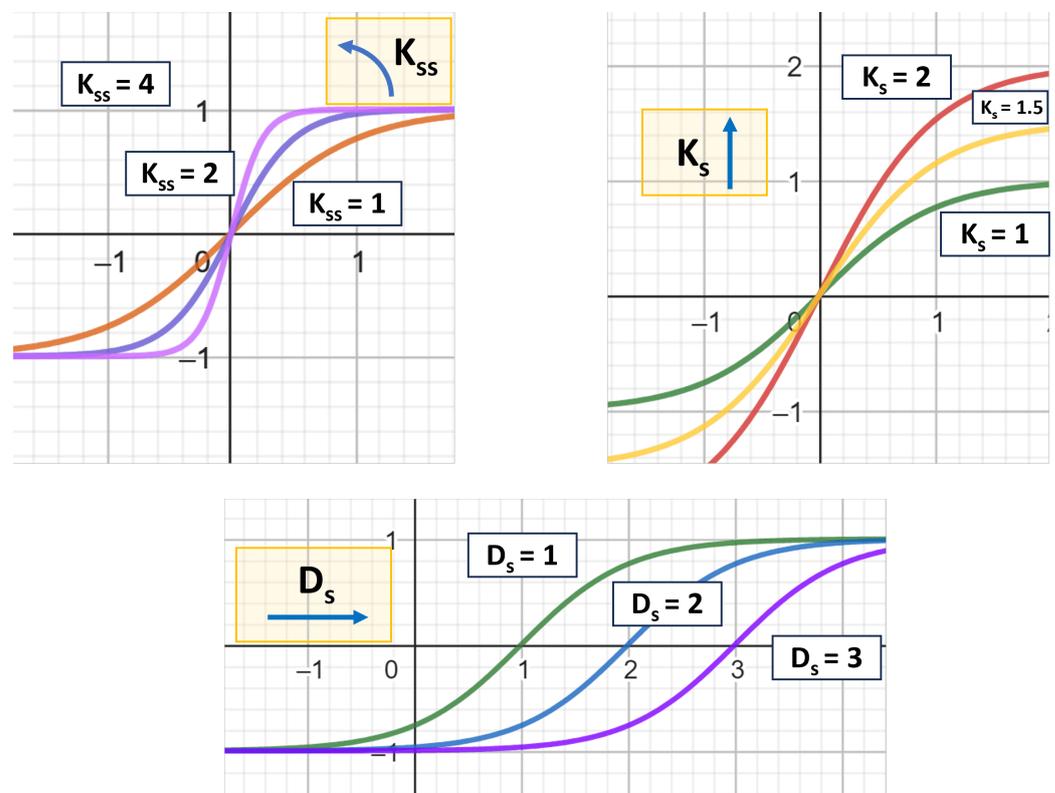


Figure 5. Visual representation of the independent effect of constants K_{ss} , K_s , and D_s on the reward function $R_s = K_s * \tanh(K_{ss} * (d - D_s))$.

Software Infrastructure

The software infrastructure is composed of various components distributed across different layers of the technology stack. The DRL code was developed using PyTorch and consists of three primary classes. Specifically, one class encapsulates the actor network structure, the second encapsulates the critic network structure, and the third encapsulates the TD3 network structure. For the training of neural networks and initial testing in a secure environment, a simulator was employed. In particular, due to the utilization of the Robot Operating System (ROS) for tasks related to wheelchair navigation, Gazebo was selected as the simulator. For collecting samples of a policy execution in the environment and later using them for training purposes, the well-known replay buffer technique was employed. Following the TD3 architecture [116], a "soft update" is implemented. This implies that the actor and critic entities are characterized by a base network and a target network. During training, only a batch of sampled steps from various episodes is used to

update the base critic networks. This update involves a gradient descent process, with the loss calculated between the target Q (which is computed through Bellman equations using the minimum Q-value from target critic networks) and the Q-values outputted by the base critic networks. The update for the target critic network occurs less frequently and involves infusing only a small portion of the updated base critic network parameters. This approach aims to stabilize the learning process toward the optimal policy.

Following the actor–critic paradigm, the base actor network is trained based on feedback from the base critic network. However, unlike the base critic networks, the base actor network is not trained at every step but at predefined intervals, justifying the “Delayed” term in the TD3 acronym. Similar to the critics, the target actor network receives only a portion of the updated base actor network. As shown in Figure 6, both actors and critics have a base and target networks. The target networks are updated using the so-called soft update strategy, namely, a small amount of parameters from the base networks is infused into the target ones. The target actor network receives $State^I$, which is the state recorded during the collection of training episodes and is the resulting state of the application of the action yielded by the base actor network. Hence, the target actor network computes the $Action^I$, which is evaluated by the target critic networks. The two results $Q1^I$ and $Q2^I$, which are the expected cumulative rewards for taking that particular action in that particular state, are compared and the minimum is selected. This Q-value is then used as the ground truth for evaluating the goodness of the estimations done by the two base critic networks. By trying to minimize the error, the base networks are updated. According to the actor–critic paradigm, the actor networks are updated thanks to the feedback provided by the critic networks.

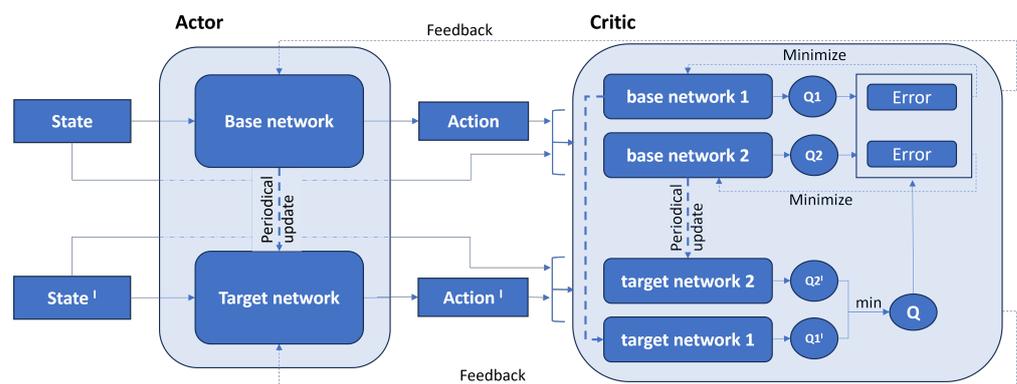


Figure 6. Schematic representation of TD3 architecture.

The hyper-parameter tuning phase stands out as one of the most time-consuming aspects in general for deep learning but especially for DRL algorithms. Experimenting with various hyper-parameter combinations in parallel accelerates this process. Due to the impracticality of manually running and tracking each combination, hyper-parameter tuning frameworks have been developed, and one such framework is Optuna. Optuna requires a range of values as input and an objective function to execute. It iteratively runs the objective function, exploring different combinations of input parameters each time. The objective function is expected to return a metric that allows for comparison, and thus, determining the ranking of the best combinations. In our context, the objective function should conduct training of our TD3 network with a specific set of hyper-parameters. Unfortunately, the combination of ROS and the Gazebo simulator does not provide native support for multiple instances running in parallel. To address this limitation, we decided to encapsulate them within containers. Each container is built on top of a base Ubuntu 20.04 image enhanced with virtual network computing (VNC) capabilities, enabling remote desktop connections to the instances for easy monitoring of the training phase. To minimize the memory footprint and required resources, a lightweight desktop environment (LXQt) was chosen. Consequently, the base image is extended with the installation of ROS Noetic,

which already includes the standalone versions of Gazebo and Rviz. Additionally, PyTorch 1.2.1 with CUDA 12.1 was installed. As of the time of writing, Docker cannot execute a container by attaching an Nvidia GPU by default, which is recommended for expediting training. Therefore, the NVIDIA Container Toolkit must be installed in advance and utilized during the instantiating of a container [117].

With a container capable of receiving DRL input parameters, the adaptation of the Optuna objective function involves initiating a container with the specified hyper-parameters. The container then carries out the training phase and, upon reaching a termination condition (such as a maximum execution time, steps, or early stopping), provides the average reward value over the last 10 episodes. Optuna considers this value and continues the typical hyper-parameter tuning process.

4. Simulations

The TD3 network was trained in the Gazebo simulator for 700 episodes, which took approximately 4 h. Each training episode concluded when a collision was detected or 500 steps were taken. V_{max} and ω_{max} were set as 1 m per second and 1 rad per second, respectively. The delayed rewards were updated over the last $n = 10$ steps and the parameter update delay was set as two episodes. The training was carried out in the simulated 10×10 m-sized environment depicted in Figure 7.

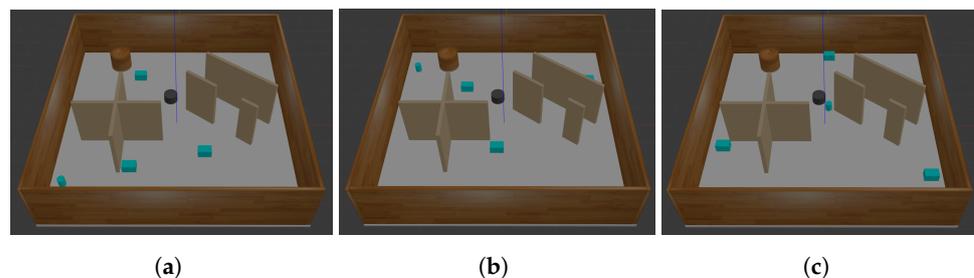


Figure 7. The 10×10 maze used for the DRL training phase. Light blue obstacles were randomly placed at the beginning of each episode to help the generalization process. Different positions can be observed in (a–c).

For the network to work not only in a simulation but also in real life, it needs to learn generic obstacle avoidance behavior from laser data. The data from the LiDAR were bagged into 21 groups, where the minimum value of each group created the final laser input state of 21 values. To facilitate generalization and policy exploration, Gaussian noise was added to the sensor and action values. To help with sim2real transfer, the environment was varied in each episode by randomly changing the locations of the box-shaped obstacles. Examples of their changing locations are depicted in Figure 7a–c. The ROS Melodic version managed the packages and control commands. To validate the obtained agent, we decided to compare it against the vector field histogram plus because of its good reputation for cooperative obstacle avoidance and against bare metal usage (no obstacle avoidance system). The tests were decided by taking into consideration the pitfalls of the VFH, as stated in [72]. We refer to our proposed method as a Cooperative-DRL-Driving-System (CDDS), vector field histogram (VFH) system for VFH+ obstacle avoidance, and a Bare-Metal-System (BRS) for the absence of assistive methods. The parameters used for CDDS and VFH can be found in Appendix A. The system performance is a function of quantitative measures and subjective ratings of comfort and safety. We tried to grasp both with the following metrics:

- Average forward speed.
- Jerkiness, which measures how “smooth” the wheelchair handling is, as represented by the mean and standard deviation value of the gradient generated by subsequent motor command variations (the lower, the better).
- Collision risk, which is the number of occurred collisions.

For each test, the metrics were calculated over three executions. According to our jerkiness definition, the mean value was calculated according to the following formula:

$$\text{Jerkiness} = \frac{1}{2N} \left(\sum_{i=0}^T \frac{l_{i+1} - l_i}{t_{i+1} - t_i} + \sum_{i=0}^T \frac{r_{i+1} - r_i}{t_{i+1} - t_i} \right) \quad (9)$$

where $i = T$ is the N^{th} concluding timestamp of the simulation, and l_i and r_i are, respectively, the commands given to the left and right motor at the i th timestamp.

The first experiment consisted of three tests executed in the environments depicted in Figure 8. For all of them, the aim was to commute from the starting point to the ending point while avoiding any collisions. The first test consisted of following the shape of an obstacle, trying to stay as close as possible without colliding, simulating the need to follow the shape of a wall or a piece of furniture to grasp objects on top of it. The second test consisted of traversing an environment disseminated with obstacles, following the path highlighted in Figure 8b by the orange arrows. The third test consisted of navigating in narrow corridors, simulating the daily challenges of living in a house. The results of the experiment are described in Table 2. Trajectory samples can be found in Figure A1.

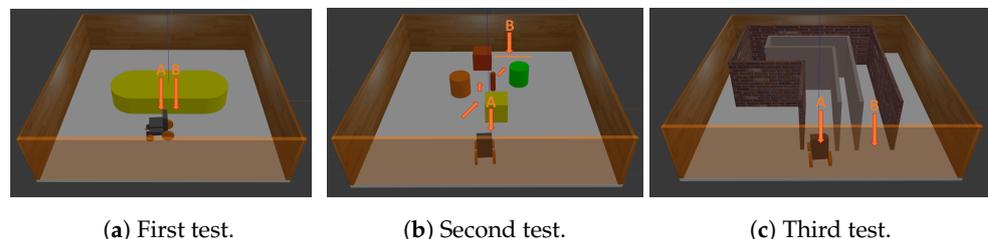


Figure 8. Tests belonging to the first experiment. For all of them, point A is the starting point and point/line B is the ending. The aim was to commute from A to B without colliding. (a) First test: follow the shape of the obstacle as close as possible. (b) Second test: navigate through obstacles. (c) Third test: navigate through narrow corridors.

Table 2. Results of first experiment.

		CDDS	VFH	BRS
First test	Average forward speed [m/s]	0.59	0.35	0.64
	Jerkiness	8.3 ± 58.8	3.6 ± 21.2	6.5 ± 57.4
	Max collisions	0	0	0
Second test	Average forward speed [m/s]	0.29	0.14	0.37
	Jerkiness	13.8 ± 71.1	4.9 ± 24.7	10.6 ± 69.2
	Max collisions	0	0	1
Third test	Average forward speed [m/s]	0.39	0.31	0.42
	Jerkiness	12.1 ± 83.4	5.7 ± 29.6	11.8 ± 85.9
	Max collisions	0	0	2

The second experiment consisted of three tests executed in the environments depicted in Figure 9.

For all of them, as it happened for the first experiment, the aim was to commute from the starting point to the ending one, trying to avoid any collisions. However, this time the testing strategy aimed to compare CDDS and VFH in the traversing of a door opening by varying the door size. The width of the hallway and the standard door size were set, respectively, to 0.91 m and 0.81 m, according to the American Disability Act [118]. The relevant metric was the percentage of successful door traversing against the door width. The first test consisted of traversing the door starting in an obstacle-free condition. Thus, the wheelchair had to follow a straight line to cross it. The second test consisted of traversing the door starting in a corridor. Thus, the wheelchair had to go through the corridor and

make a turn at a certain point. The third test consisted of entering a corridor through a door. Thus, the wheelchair had to go toward the door and make a turn to enter the corridor. The results of the experiment can be found in Figure 10.

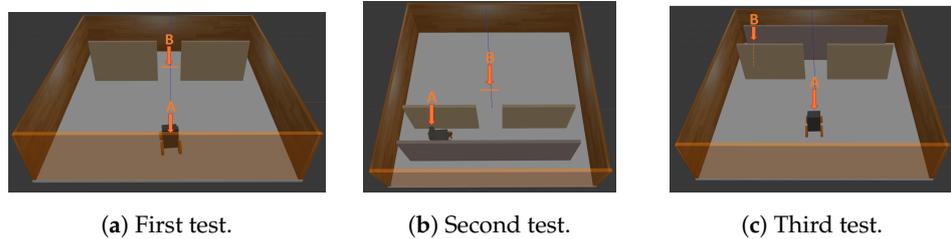


Figure 9. Tests belonging to the second experiment. For all of them, point A is the starting point and point/line B is the ending. The aim was to commute from A to B without colliding. (a) First test: go through the door opening. (b) Second test: drive out from a corridor through a door opening. (c) Third test: enter a corridor through a door opening.

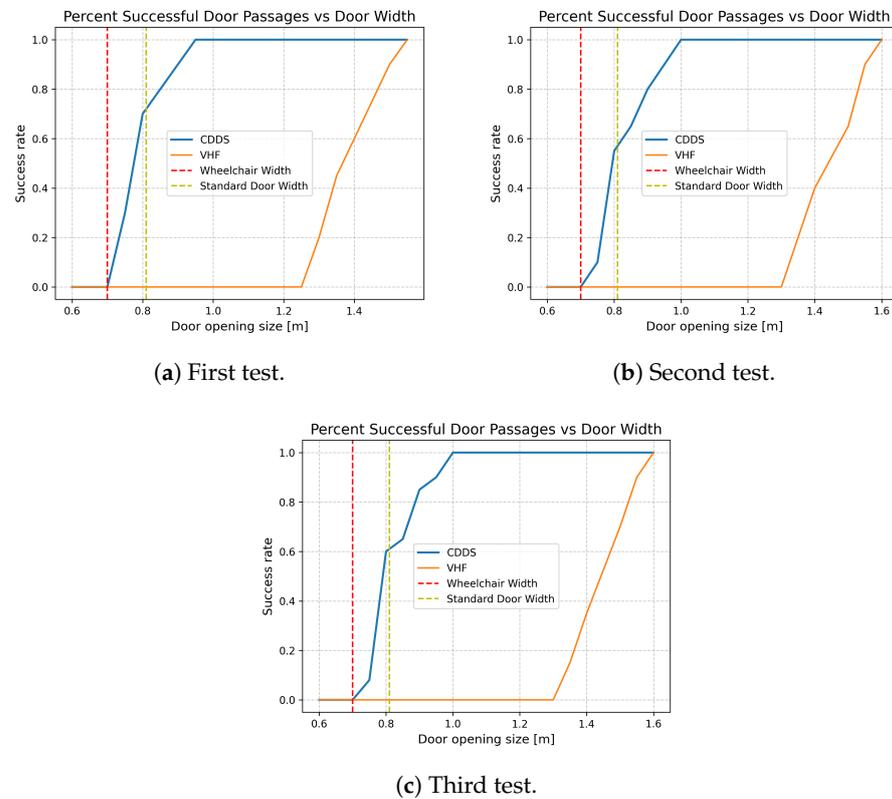


Figure 10. Results related to the second experiment. For all three, on the Y-axes, there is the percentage of successful door passages, namely, those reaching the ending point without collisions, whereas on X-axes, there is the varying door opening sizes.

Additionally, the CCDS was tested to retrieve the computational time for yielding a pair of control commands. It introduced a fixed delay in the joystick control loop. If this delay is longer than the users’ reaction time, they may feel out of control of the vehicle. Therefore, we empirically decided that the maximum possible delay introduced by the system was 100 ms. Having the final target in mind, namely, deploying the CDDS on an embedded system, we decided to collect the inference time running CDDS on a NVIDIA Jetson Nano 2 GB Developer Kit [119]. Because the execution time of the control unit and the joystick sampling were a few orders less than the actor network inference time, we decided to deploy only the actor network. The results of the tests can be found in Table 3.

Table 3. Results of inference execution on different platforms using TFLite framework.

Platform	Execution Time
PC (only CPU)	0.098 ± 0.038 ms
Jetson Nano	1.22 ± 0.73 ms

5. Results and Discussion

The experiments investigated various aspects of the CDDS. A comparison between the VFH and the CDDS revealed that the latter achieved a higher success rate in the indoor passage, particularly in narrower openings. This advantage stemmed from the CDDS not mandating the wheelchair footprint to fit precisely within a safety circumference, unlike the VFH. While the VFH strictly adhered to a safety distance boundary, the CDDS offered a flexible boundary, occasionally permitting the wheelchair to exceed it. Consequently, the VFH registered zero success until the door accommodated the wheelchair's footprint and safety distance, while the CDDS achieved roughly a 70% success rate at standard door widths, indicating room for improvement. Both systems excelled in obstacle avoidance, occasionally outperforming the BRS, as expert users may prioritize speed over obstacle vigilance. Additionally, jerkiness, which refers to non-smooth movements, warrants consideration. In electric wheelchair driving, jerkiness can affect maneuverability, which is influenced by factors like environment complexity and algorithm design. However, jerkiness does not necessarily correlate with low maneuverability, considering various controller attributes. To understand the relationship between jerkiness and maneuverability better, comprehensive analysis, including empirical evaluations and reinforcement learning algorithm assessments, is necessary. The CDDS exhibited poor jerkiness performance due to noise in the actor neural network output, suggesting potential solutions, such as implementing low-pass filters or introducing jerkiness penalties in the reward function. In terms of speed, the CDDS maintained a better average forward speed than the VFH by adjusting the speed based on user commands and obstacle detection, albeit lower than an expert using the BRS, which lacked speed regulation. Regarding the inference time, the CDDS met the maximum threshold, confirming its suitability for embedded devices. Overall, the CDDS effectively navigated diverse environments and avoided obstacles, leveraging a neural-network-based motion policy for rapid decision making. Although the CDDS showed promising performance compared with the VFH, it had limitations, particularly in jerkiness and sensor capabilities. Future research will explore additional components, sensors, diverse DRL architectures, and moving obstacles to effectively address these limitations.

6. Conclusions

Similar to many human-machine systems, the CDDS leverages the strengths of both users and machines by enabling shared control of the system output. Human users possess the adaptability to adjust control behavior based on environmental changes and functional needs. By facilitating user-machine collaboration, the system enhances adaptability, versatility, and robustness. Autonomous systems should complement human capabilities rather than replace them entirely, as humans possess untapped potential. The integration of DRL policies into cooperative driving systems for power wheelchairs has shown overall success, yielding improvements over traditional approaches, like the VFH. While DRL policies have provided basic mobility, unresolved issues remain. Jerkiness, which stems from noisy output in continuous DRL policies, poses a challenge. Additionally, limitations in obstacle detection arise from the use of 2D LiDAR, which fails to detect obstacles outside its plane. The selection of 2D LiDAR balances cost, robustness, and accuracy considerations. Future research will address these challenges by exploring diverse DRL architectures and sensor combinations. Integration into existing plug-and-play systems aims to furnish a comprehensive and safe electrification kit for wider application.

Author Contributions: Conceptualization, F.P., P.D. and L.F.; Methodology, F.P., P.D. and L.F.; Software, F.P., P.D. and L.F.; Validation, F.P., P.D. and L.F.; Formal analysis, F.P., P.D. and L.F.; Investigation, F.P., P.D. and L.F.; Resources, F.P., P.D. and L.F.; Data curation, F.P., P.D. and L.F.; Writing—original draft, F.P., P.D. and L.F.; Writing—review & editing, F.P., P.D. and L.F.; Visualization, F.P., P.D. and L.F.; Supervision, F.P., P.D. and L.F.; Project administration, F.P., P.D. and L.F.; Funding acquisition, F.P., P.D. and L.F. All authors contributed equally to this research work. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by Centro Nazionale di Ricerca in “High-Performance Computing Big Data and Quantum Computing CN1 SPOKE 6 Multiscale Modelling” and by MIUR FoReLab Project “Dipartimenti di Eccellenza”.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author (F.P.) upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

List of parameters used for the CDDS and VFH during testing.

Table A1. Parameters of CDDS used for simulation.

	Parameters	Value
Actor	Layer 1	25×800
	Activation 1	Relu
	Layer 2	800×600
	Activation	Relu
	Layer 3	600×400
	Activation	Relu
	Layer 4	400×400
	Activation	Relu
	Layer 5	400×2
	Activation	Tanh
Critic ₁	Layer 1	25×800
	Activation 1	Relu
	Layer 2	$600 \times 600, 600 \times 600$
	Activation	Matrix Mul
	Layer 3	600×600
Critic ₂	Layer 5	600×1
	Layer 1	25×800
	Activation 1	Relu
	Layer 2	$600 \times 600, 600 \times 600$
	Activation	Matrix Mul
Critic	Layer 3	600×600
	Activation	Relu
	Layer 5	600×1
	Activation	min
	- State	$[lidar_0, \dots, lidar_{20}, V_{user}, \omega_{user}, V_{wheelchair}, \omega_{wheelchair}]$
- Action	$[V_{wheelchair}, \omega_{wheelchair}]$	
- D_s (safety distance) *	0.1 m	

* Set during training.

Table A2. Parameters of VFH used for simulation.

	Parameters	Value
Wheelchair	Footprint radius	0.61 m
	Safety distance	0.1 m
	Minimum turning radius	0.61 m
Cost function	Target direction weight	5
	Current direction weight	2
	Previous direction weight	2
Histogram	Number of angular sectors	180
	Range distance limits	[0.05 m , 6 m]
	Histogram thresholds	[3, 10]

Appendix B

Additional material related to the execution of the first experiment.

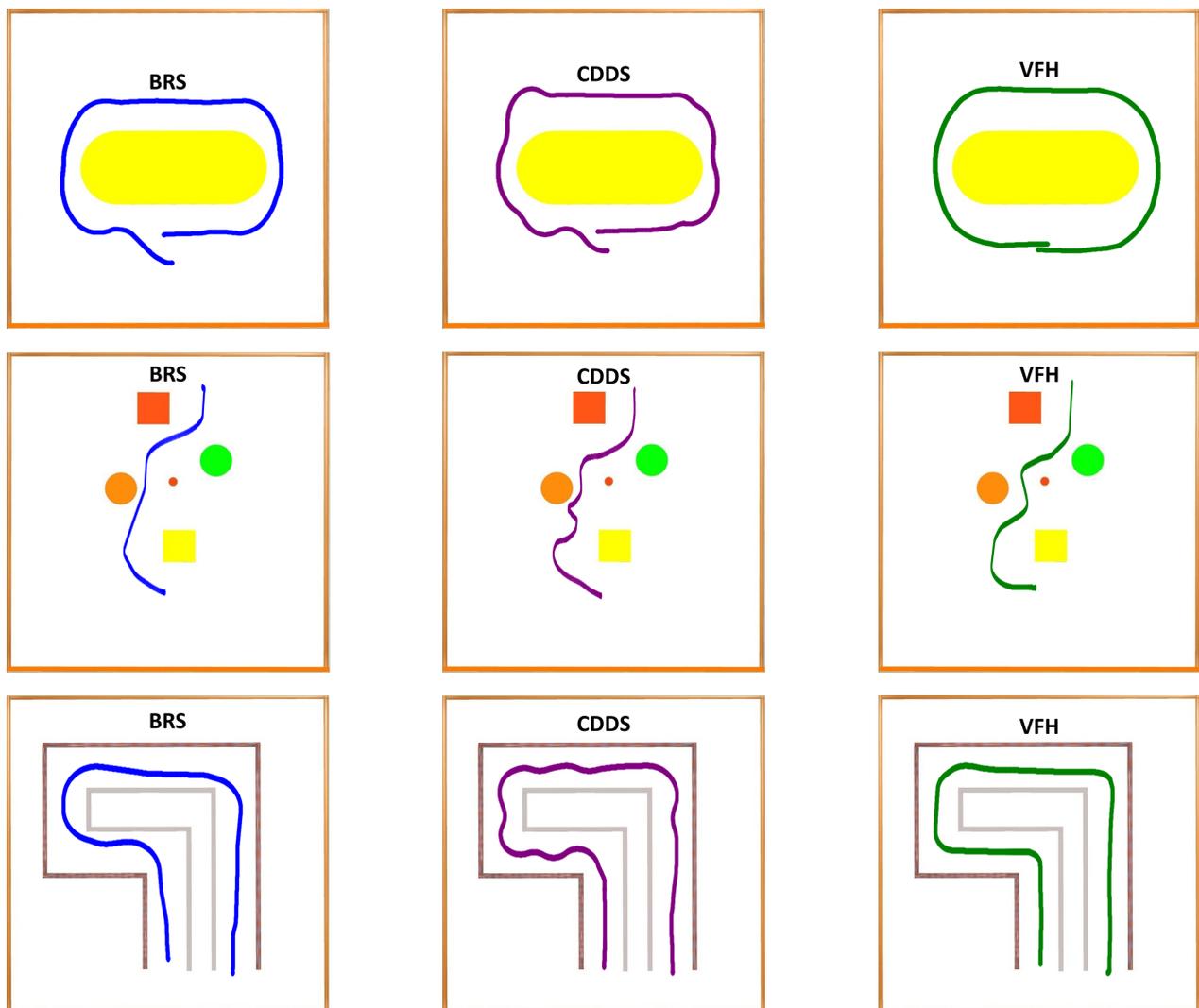


Figure A1. Trajectory samples during the first, second, and third experiments with the CDDS, VFH, and BRS driving systems.

References

1. World Health Organization; World Bank. *World Report on Disability 2011*; World Health Organization: Geneva, Switzerland, 2011.
2. UN General Assembly, Convention on the Rights of People with Disabilities, 24 January 2007, A/RES/61/106. Available online: <https://www.un.org/development/desa/disabilities/resources/general-assembly/convention-on-the-rights-of-persons-with-disabilities-ares61106.html> (accessed on 12 November 2023).
3. Mars, L.; Arroyo, R.; Ruiz, T. Mobility and wellbeing during the covid-19 lockdown. Evidence from Spain. *Transp. Res. Part A Policy Pract.* **2022**, *161*, 107–129. [[CrossRef](#)] [[PubMed](#)]
4. Landry, B.W.; Driscoll, S.W. Physical activity in children and adolescents. *PM&R* **2012**, *4*, 826–832.
5. Freedman, V.A.; Carr, D.; Cornman, J.C.; Lucas, R.E. Aging, mobility impairments and subjective wellbeing. *Disabil. Health J.* **2017**, *10*, 525–531. [[CrossRef](#)] [[PubMed](#)]
6. Arroyo, R.; Mars, L.; Ruiz, T. Activity Participation and wellbeing during the covid-19 lockdown in Spain. *Int. J. Urban Sci.* **2021**, *25*, 386–415. [[CrossRef](#)]
7. Mussone, L.; Changizi, F. The relationship between subjective well-being and individual characteristics, personality traits, and choice of transport mode during the first lock-down in Milan, Italy. *J. Transp. Health* **2023**, *30*, 101600. [[CrossRef](#)] [[PubMed](#)]
8. Checa, J.; Martín, J.; López, J.; Nel-Lo, O. *Those Who Cannot Stay at Home: Urban Mobility and Social Vulnerability in Barcelona during the COVID-19 Pandemic*; Asociacion Espanola de Geografia: Madrid, Spain, 2020.
9. Sánchez-Rodríguez, E.; Ferreira-Valente, A.; Pimenta, F.; Ciaramella, A.; Miró, J. Mental, physical and socio-economic status of adults living in Spain during the late stages of the state of emergency caused by COVID-19. *Int. J. Environ. Res. Public Health* **2022**, *19*, 854. [[CrossRef](#)] [[PubMed](#)]
10. Tsouros, I.; Tsimpa, A.; Pagoni, I.; Polydoropoulou, A. Activities, time-use and mental health during the first COVID-19 pandemic wave: Insight from Greece. *Transp. Res. Interdiscip. Perspect.* **2021**, *11*, 100442. [[CrossRef](#)] [[PubMed](#)]
11. Politis, I.; Georgiadis, G.; Nikolaidou, A.; Kopsacheilis, A.; Fyrogenis, I.; Sdoukopoulos, A.; Verani, E.; Papadopoulos, E. Mapping travel behavior changes during the COVID-19 lock-down: A socioeconomic analysis in Greece. *Eur. Transp. Res. Rev.* **2021**, *13*, 21. [[CrossRef](#)]
12. Mussone, L.; Changizi, F. A study on the factors that influenced the choice of transport mode before, during, and after the first lockdown in Milan, Italy. *Cities* **2023**, *136*, 104251. [[CrossRef](#)]
13. Politis, I.; Georgiadis, G.; Papadopoulos, E.; Fyrogenis, I.; Nikolaidou, A.; Kopsacheilis, A.; Sdoukopoulos, A.; Verani, E. COVID-19 lockdown measures and travel behavior: The case of Thessaloniki, Greece. *Transp. Res. Interdiscip. Perspect.* **2021**, *10*, 100345. [[CrossRef](#)]
14. Chen, W.Y.; Jang, Y.; Wang, J.D.; Huang, W.N.; Chang, C.C.; Mao, H.F.; Wang, Y.H. Wheelchair-Related Accidents: Relationship With Wheelchair-Using Behavior in Active Community Wheelchair Users. *Arch. Phys. Med. Rehabil.* **2011**, *92*, 892–898. [[CrossRef](#)] [[PubMed](#)]
15. Worobey, L.; Oyster, M.; Pearlman, J.; Gebrosky, B.; Boninger, M.L. Differences between manufacturers in reported power wheelchair repairs and adverse consequences among people with spinal cord injury. *Arch. Phys. Med. Rehabil.* **2014**, *95*, 597–603. [[CrossRef](#)] [[PubMed](#)]
16. Abou, L.; Rice, L.A. Risk factors associated with falls and fall-related injuries among wheelchair users with spinal cord injury. *Arch. Rehabil. Res. Clin. Transl.* **2022**, *4*, 100195. [[CrossRef](#)] [[PubMed](#)]
17. McClure, L.A.; Boninger, M.L.; Oyster, M.L.; Williams, S.; Houlihan, B.; Lieberman, J.A.; Cooper, R.A. Wheelchair repairs, breakdown, and adverse consequences for people with traumatic spinal cord injury. *Arch. Phys. Med. Rehabil.* **2009**, *90*, 2034–2038. [[CrossRef](#)] [[PubMed](#)]
18. Worobey, L.A.; Heinemann, A.W.; Anderson, K.D.; Fyffe, D.; Dyson-Hudson, T.A.; Berner, T.; Boninger, M.L. Factors influencing incidence of wheelchair repairs and consequences among individuals with spinal cord injury. *Arch. Phys. Med. Rehabil.* **2022**, *103*, 779–789. [[CrossRef](#)] [[PubMed](#)]
19. Toro, M.L.; Worobey, L.; Boninger, M.L.; Cooper, R.A.; Pearlman, J. Type and frequency of reported wheelchair repairs and related adverse consequences among people with spinal cord injury. *Arch. Phys. Med. Rehabil.* **2016**, *97*, 1753–1760. [[CrossRef](#)] [[PubMed](#)]
20. Hogaboom, N.S.; Worobey, L.A.; Houlihan, B.V.; Heinemann, A.W.; Boninger, M.L. Wheelchair breakdowns are associated with pain, pressure injuries, rehospitalization, and self-perceived health in full-time wheelchair users with spinal cord injury. *Arch. Phys. Med. Rehabil.* **2018**, *99*, 1949–1956. [[CrossRef](#)] [[PubMed](#)]
21. Seki, H.; Tanohata, N. Fuzzy control for electric power-assisted wheelchair driving on disturbance roads. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2012**, *42*, 1624–1632. [[CrossRef](#)]
22. Seki, H.; Kiso, A. Disturbance road adaptive driving control of power-assisted wheelchair using fuzzy inference. In Proceedings of the 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Boston, MA, USA, 30 August–3 September 2011; pp. 1594–1599. [[CrossRef](#)]
23. Sankardoss, V.; Geethanjali, P. Design and low-cost implementation of an electric wheelchair control. *IETE J. Res.* **2021**, *67*, 657–666. [[CrossRef](#)]
24. Callejas-Cuervo, M.; González-Cely, A.X.; Bastos-Filho, T. Design and implementation of a position, speed and orientation fuzzy controller using a motion capture system to operate a wheelchair prototype. *Sensors* **2021**, *21*, 4344. [[CrossRef](#)]
25. Seki, H.; Kuramoto, T. Fuzzy Inference-Based Driving Control System for Pushrim-Activated Power-Assisted Wheelchairs Considering User Characteristics. *IEEE Trans. Hum.-Mach. Syst.* **2022**, *52*, 1049–1059. [[CrossRef](#)]

26. Sier, H.; Yu, X.; Catalano, I.; Queralta, J.P.; Zou, Z.; Westerlund, T. UAV Tracking with Lidar as a Camera Sensor in GNSS-Denied Environments. In Proceedings of the 2023 International Conference on Localization and GNSS (ICL-GNSS), Castellon, Spain, 6–8 June 2023; pp. 1–7. [\[CrossRef\]](#)
27. Antonopoulos, A.; Lagoudakis, M.G.; Partsinevelos, P. A ROS multi-tier UAV localization module based on GNSS, inertial and visual-depth data. *Drones* **2022**, *6*, 135. [\[CrossRef\]](#)
28. Ito, S.; Hiratsuka, S.; Ohta, M.; Matsubara, H.; Ogawa, M. Small imaging depth LIDAR and DCNN-based localization for automated guided vehicle. *Sensors* **2018**, *18*, 177. [\[CrossRef\]](#)
29. Deng, C.; Wang, S.; Wang, J.; Xu, Y.; Chen, Z. LiDAR Depth Cluster Active Detection and Localization for a UAV with Partial Information Loss in GNSS. In *Unmanned Systems*; World Scientific: Singapore, 2024.
30. Biao, J.; Xiangwen, Z.; Yangxiong, W.; Wenchao, H. Regenerative braking control strategy of electric vehicles based on braking stability requirements. *Int. J. Automot. Technol.* **2021**, *22*, 465–473. [\[CrossRef\]](#)
31. Zhang, J.; Yang, Y.; Qin, D.; Fu, C.; Cong, Z. Regenerative Braking Control Method Based on Predictive Optimization for Four-Wheel Drive Pure Electric Vehicle. *IEEE Access* **2021**, *9*, 1394–1406. [\[CrossRef\]](#)
32. Heerwan, P.; Shahrom, M.; Ishak, M.; Kato, H.; Narita, T. Investigation of the Performance of Plugging Braking System as a Hill Descent Control (HDC) for Electric-Powered Wheelchair. *Int. J. Automot. Mech. Eng.* **2023**, *20*, 10906–10916. [\[CrossRef\]](#)
33. Seki, H.; Ishihara, K.; Tadakuma, S. Novel Regenerative Braking Control of Electric Power-Assisted Wheelchair for Safety Downhill Road Driving. *IEEE Trans. Ind. Electron.* **2009**, *56*, 1393–1400. [\[CrossRef\]](#)
34. Baek, S.J.; Kim, A.; Kim, J.W. Implementation of Wheelchair Robot Applying SLAM and Global Path Planning Methods Suitable for Indoor Autonomous Driving. *IEMEK J. Embed. Syst. Appl.* **2021**, *16*, 293–297.
35. Somwanshi, D.; Bundele, M. Obstacle detection approach for robotic wheelchair navigation. In *Proceedings of the International Conference on Artificial Intelligence: Advances and Applications 2019: Proceedings of ICAIAA 2019*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 261–268.
36. Wahid, A.B.; Siraj, U.; Affan, M.; Ahmed, H.; Islam, F.; Ansari, U.; Naveed, M.; Ayaz, Y. Development of modular framework for the semi-autonomous RISE wheelchair with multiple user interfaces using robot operating system (ROS). *Int. J. Mech. Eng. Robot. Res.* **2018**, *7*, 515–520. [\[CrossRef\]](#)
37. Jung, Y.; Kim, Y.; Lee, W.H.; Bang, M.S.; Kim, Y.; Kim, S. Path Planning Algorithm for an Autonomous Electric Wheelchair in Hospitals. *IEEE Access* **2020**, *8*, 208199–208213. [\[CrossRef\]](#)
38. Green, J.; Clounie, J.; Galarza, R.; Anderson, S.; Campell-Smith, J.; Voicu, R.C. Optimization of an Intelligent Wheelchair: LiDAR and Camera Vision for Obstacle Avoidance. In Proceedings of the 2022 22nd International Conference on Control, Automation and Systems (ICCAS), Busan, Republic of Korea, 27–30 November 2022; pp. 313–318. [\[CrossRef\]](#)
39. de Paiva, F.P.; Cardozo, E.; Rohmer, E. A Path Tracking Control Algorithm for Smart Wheelchairs. In Proceedings of the 2020 International Symposium on Medical Robotics (ISMR), Atlanta, GA, USA, 18–20 November 2020; pp. 76–82. [\[CrossRef\]](#)
40. Maciel, G.M.; Pinto, M.F.; Júnior, I.C.d.S.; Marcato, A.L. Methodology for autonomous crossing narrow passages applied on assistive mobile robots. *J. Control Autom. Electr. Syst.* **2019**, *30*, 943–953. [\[CrossRef\]](#)
41. Levine, S.P.; Bell, D.A.; Jaros, L.A.; Simpson, R.C.; Koren, Y.; Borenstein, J. The NavChair assistive wheelchair navigation system. *IEEE Trans. Rehabil. Eng.* **1999**, *7*, 443–451. [\[CrossRef\]](#) [\[PubMed\]](#)
42. Messaoudi, M.D.; Menelas, B.A.J.; Mcheick, H. Review of navigation assistive tools and technologies for the visually impaired. *Sensors* **2022**, *22*, 7888. [\[CrossRef\]](#) [\[PubMed\]](#)
43. Callejas-Cuervo, M.; González-Cely, A.X.; Bastos-Filho, T. Control systems and electronic instrumentation applied to autonomy in wheelchair mobility: The state of the art. *Sensors* **2020**, *20*, 6326. [\[CrossRef\]](#) [\[PubMed\]](#)
44. Kim, E.Y. Wheelchair navigation system for disabled and elderly people. *Sensors* **2016**, *16*, 1806. [\[CrossRef\]](#) [\[PubMed\]](#)
45. Fox, D.; Burgard, W.; Thrun, S. The dynamic window approach to collision avoidance. *IEEE Robot. Autom. Mag.* **1997**, *4*, 23–33. [\[CrossRef\]](#)
46. Wieczorek, B.; Kukla, M.; Rybarczyk, D.; Warguła, Ł. Evaluation of the biomechanical parameters of human-wheelchair systems during ramp climbing with the use of a manual wheelchair with anti-rollback devices. *Appl. Sci.* **2020**, *10*, 8757. [\[CrossRef\]](#)
47. Abdulghani, M.M.; Al-Aubidy, K.M.; Ali, M.M.; Hamarsheh, Q.J. Wheelchair neuro fuzzy control and tracking system based on voice recognition. *Sensors* **2020**, *20*, 2872. [\[CrossRef\]](#) [\[PubMed\]](#)
48. Gao, J.; Ye, W.; Guo, J.; Li, Z. Deep reinforcement learning for indoor mobile robot path planning. *Sensors* **2020**, *20*, 5493. [\[CrossRef\]](#)
49. Wang, Z.; Liang, Y.; Gong, C.; Zhou, Y.; Zeng, C.; Zhu, S. Improved dynamic window approach for Unmanned Surface Vehicles' local path planning considering the impact of environmental factors. *Sensors* **2022**, *22*, 5181. [\[CrossRef\]](#)
50. Molinos, E.J.; Llamazares, A.; Ocaña, M. Dynamic window based approaches for avoiding obstacles in moving. *Robot. Auton. Syst.* **2019**, *118*, 112–130. [\[CrossRef\]](#)
51. Zhang, B.; Holloway, C.; Carlson, T. A hierarchical design for shared-control wheelchair navigation in dynamic environments. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 11–14 October 2020; pp. 4439–4446.
52. Demeester, E.; Nuttin, M.; Vanhooydonck, D.; Vanacker, G.; Van Brussel, H. Global dynamic window approach for holonomic and non-holonomic mobile robots with arbitrary cross-section. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems Edmonton, AB, Canada, 2–6 August 2005; pp. 2357–2362.

53. Devigne, L.; Narayanan, V.K.; Pasteau, F.; Babel, M. Low complex sensor-based shared control for power wheelchair navigation. In Proceedings of the 2016 IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016; pp. 5434–5439.
54. Zhang, B.; Barbareschi, G.; Ramirez Herrera, R.; Carlson, T.; Holloway, C. Understanding interactions for smart wheelchair navigation in crowds. In Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, Orleans, LA, USA, 29 April–5 May 2022; pp. 1–16.
55. Chiang, H.H.; You, W.T.; Lee, J.S. Shared driving assistance design considering human error protection for intelligent electric wheelchairs. *Energies* **2023**, *16*, 2583. [[CrossRef](#)]
56. Xi, L.; Shino, M. Shared control of an electric wheelchair considering physical functions and driving motivation. *Int. J. Environ. Res. Public Health* **2020**, *17*, 5502. [[CrossRef](#)]
57. Deng, X.; Yu, Z.L.; Lin, C.; Gu, Z.; Li, Y. Self-adaptive shared control with brain state evaluation network for human-wheelchair cooperation. *J. Neural Eng.* **2020**, *17*, 045005. [[CrossRef](#)]
58. Deng, X.; Yu, Z.L.; Lin, C.; Gu, Z.; Li, Y. A Bayesian Shared Control Approach for Wheelchair Robot With Brain Machine Interface. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2020**, *28*, 328–338. [[CrossRef](#)]
59. Koren, Y.; Borenstein, J. Analysis of Control Methods for Mobile Robot Obstacle Avoidance. In Proceedings of the IEEE International Workshop on Intelligent Motion Control, Istanbul, Turkey, 20–22 August 1990; Volume 2, pp. 457–461.
60. Koren, Y.; Borenstein, J. Potential field methods and their inherent limitations for mobile robot navigation. In Proceedings of the ICR, Sacramento, CA, USA, 9–11 April 1991; Volume 2, pp. 1398–1404.
61. Chen, W.; Chen, S.K.; Liu, Y.H.; Chen, Y.J.; Chen, C.S. An electric wheelchair manipulating system using SSVEP-based BCI system. *Biosensors* **2022**, *12*, 772. [[CrossRef](#)]
62. Matsuura, H.; Nonaka, K.; Sekiguchi, K. Model Predictive Obstacle Avoidance Control for an Electric Wheelchair in Indoor Environments Using Artificial Potential Field Method. In Proceedings of the 2022 IEEE/SICE International Symposium on System Integration (SII), Virtual, 9–12 January 2022; pp. 19–24. [[CrossRef](#)]
63. Sollehudin, I.; Heerwan, P.; Ishak, M.; Zakaria, M. Electric Powered Wheelchair Trajectory Planning on Artificial Potential Field Method. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Sanya, China, 12–14 November 2021; Volume 1068, p. 012012.
64. Borenstein, J.; Koren, Y. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Trans. Robot. Autom.* **1991**, *7*, 278–288. [[CrossRef](#)]
65. Gallo, V.; Shallari, I.; Carratù, M.; Laino, V.; Liguori, C. Design and Characterization of a Powered Wheelchair Autonomous Guidance System. *Sensors* **2024**, *24*, 1581. [[CrossRef](#)]
66. Adámek, R.; Bugeja, M.K.; Fabri, S.G.; Grepl, R. Enhancing the Obstacle Avoidance Capabilities of a Smart Wheelchair. In Proceedings of the 2022 20th International Conference on Mechatronics—Mechatronika (ME), Pilsen, Czech Republic, 7–9 December 2022; pp. 1–7. [[CrossRef](#)]
67. Li, K.; Ramkumar, S.; Thimmiraja, J.; Diwakaran, S. Optimized artificial neural network based performance analysis of wheelchair movement for ALS patients. *Artif. Intell. Med.* **2020**, *102*, 101754. [[CrossRef](#)]
68. Bolbhat, S.; Bhosale, A.; Sakthivel, G.; Saravanakumar, D.; Sivakumar, R.; Lakshmi pathi, J. Intelligent obstacle avoiding agv using vector field histogram and supervisory control. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2020; Volume 1716, p. 012030.
69. Zhang, G.; Zhang, Y.; Xu, J.; Chen, T.; Zhang, W.; Xing, W. Intelligent vector field histogram based collision avoidance method for auv. *Ocean. Eng.* **2022**, *264*, 112525. [[CrossRef](#)]
70. Kumar, J.S.; Kaleeswari, R. Implementation of Vector Field Histogram based obstacle avoidance wheeled robot. In Proceedings of the 2016 Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, India, 19 November 2016; pp. 1–6. [[CrossRef](#)]
71. Sary, I.P.; Nugraha, Y.P.; Megayanti, M.; Hidayat, E.; Trilaksono, B.R. Design of Obstacle Avoidance System on Hexacopter Using Vector Field Histogram-Plus. In Proceedings of the 2018 IEEE 8th International Conference on System Engineering and Technology (ICSET), Bandung, Indonesia, 15–16 October 2018; pp. 18–23. [[CrossRef](#)]
72. Bell, D.A.; Borenstein, J.; Levine, S.P.; Koren, Y.; Jaros, J. An assistive navigation system for wheelchairs based upon mobile robot obstacle avoidance. In Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, USA, 8–13 May 1994; pp. 2018–2022.
73. Fearn, T.; Labrosse, F.; Shaw, P. Wheelchair Navigation: Automatically Adapting to Evolving Environments. In Proceedings of the Towards Autonomous Robotic Systems: 20th Annual Conference, TAROS 2019, London, UK, 3–5 July 2019; Proceedings, Part II 20; Springer: Berlin/Heidelberg, Germany, 2019; pp. 496–500.
74. Sahoo, S.K.; Choudhury, B.B. AI advances in wheelchair navigation and control: A comprehensive review. *J. Process. Manag. New Technol.* **2023**, *11*, 115–132. [[CrossRef](#)]
75. Sahoo, S.K.; Choudhury, B.B. A review on smart robotic wheelchairs with advancing mobility and independence for individuals with disabilities. *J. Decis. Anal. Intell. Comput.* **2023**, *3*, 221–242. [[CrossRef](#)]
76. Sahoo, S.K.; Choudhury, B.B. Autonomous navigation and obstacle avoidance in smart robotic wheelchairs. *J. Decis. Anal. Intell. Comput.* **2024**, *4*, 47–66. [[CrossRef](#)]

77. Grewal, H.S.; Jayaprakash, N.T.; Matthews, A.; Shrivastav, C.; George, K. Autonomous wheelchair navigation in unmapped indoor environments. In Proceedings of the 2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Houston, TX, USA, 14–17 May 2018; pp. 1–6. [\[CrossRef\]](#)
78. Dai, X.; Mao, Y.; Huang, T.; Qin, N.; Huang, D.; Li, Y. Automatic obstacle avoidance of quadrotor UAV via CNN-based learning. *Neurocomputing* **2020**, *402*, 346–358. [\[CrossRef\]](#)
79. Zhou, C.; Li, F.; Cao, W.; Wang, C.; Wu, Y. Design and implementation of a novel obstacle avoidance scheme based on combination of CNN-based deep learning method and LiDAR-based image processing approach. *J. Intell. Fuzzy Syst.* **2018**, *35*, 1695–1705. [\[CrossRef\]](#)
80. Chakravarty, P.; Kelchtermans, K.; Roussel, T.; Wellens, S.; Tuytelaars, T.; Van Eycken, L. CNN-based single image obstacle avoidance on a quadrotor. In Proceedings of the 2017 IEEE international conference on robotics and automation (ICRA), Singapore, 29 May–3 June 2017; pp. 6369–6374.
81. Yoon, H.Y.; Kim, J.H.; Jeong, J.W. Classification of the sidewalk condition using self-supervised transfer learning for wheelchair safety driving. *Sensors* **2022**, *22*, 380. [\[CrossRef\]](#)
82. Bakouri, M.; Alsehaimi, M.; Ismail, H.F.; Alshareef, K.; Ganoun, A.; Alqahtani, A.; Alharbi, Y. Steering a robotic wheelchair based on voice recognition system using convolutional neural networks. *Electronics* **2022**, *11*, 168. [\[CrossRef\]](#)
83. Zhang, Z.; Mao, S.; Chen, K.; Xiao, L.; Liao, B.; Li, C.; Zhang, P. CNN and PCA Based Visual System of A Wheelchair Manipulator Robot for Automatic Drinking. In Proceedings of the 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 12–15 December 2018; pp. 1280–1286. [\[CrossRef\]](#)
84. Sutikno.; Anam, K.; Saleh, A. Voice Controlled Wheelchair for Disabled Patients based on CNN and LSTM. In Proceedings of the 2020 4th International Conference on Informatics and Computational Sciences (ICICoS), Semarang, Indonesia, 10–11 November 2020; pp. 1–5. [\[CrossRef\]](#)
85. Tawil, Y.; Hafez, A.A. Deep Learning Obstacle Detection and Avoidance for Powered Wheelchair. In Proceedings of the 2022 Innovations in Intelligent Systems and Applications Conference (ASYU), Antalya, Turkey, 7–9 September 2022; pp. 1–6.
86. Ali, S.; Al Mamun, S.; Fukuda, H.; Lam, A.; Kobayashi, Y.; Kuno, Y. Smart robotic wheelchair for bus boarding using CNN combined with hough transforms. In Proceedings of the Intelligent Computing Methodologies: 14th International Conference, ICIC 2018, Wuhan, China, 15–18 August 2018; Proceedings, Part III 14; Springer: Berlin/Heidelberg, Germany, 2018; pp. 163–172.
87. Xu, Z.; Zhou, X.; Wu, H.; Li, X.; Li, S. Motion planning of manipulators for simultaneous obstacle avoidance and target tracking: An RNN approach with guaranteed performance. *IEEE Trans. Ind. Electron.* **2021**, *69*, 3887–3897. [\[CrossRef\]](#)
88. Yuan, J.; Wang, H.; Lin, C.; Liu, D.; Yu, D. A novel GRU-RNN network model for dynamic path planning of mobile robot. *IEEE Access* **2019**, *7*, 15140–15151. [\[CrossRef\]](#)
89. Savage, J.; Munoz, S.; Matamoros, M.; Osorio, R. Obstacle avoidance behaviors for mobile robots using genetic algorithms and recurrent neural networks. *IFAC Proc. Vol.* **2013**, *46*, 141–146. [\[CrossRef\]](#)
90. Haddad, M.J.; Sanders, D.A. Deep Learning architecture to assist with steering a powered wheelchair. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2020**, *28*, 2987–2994. [\[CrossRef\]](#)
91. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.5602.
92. Palumbo, A.; Gramigna, V.; Calabrese, B.; Ielpo, N. Motor-imagery EEG-based BCIs in wheelchair movement and control: A systematic literature review. *Sensors* **2021**, *21*, 6285. [\[CrossRef\]](#)
93. Shamseldin, M.A.; Khaled, E.; Youssef, A.; Mohamed, D.; Ahmed, S.; Hesham, A.; Elkodama, A.; Badran, M. A new design identification and control based on GA optimization for an autonomous wheelchair. *Robotics* **2022**, *11*, 101. [\[CrossRef\]](#)
94. Kocejko, T.; Matuszkiewicz, N.; Durawa, P.; Madajczak, A.; Kwiatkowski, J. How Integration of a Brain-Machine Interface and Obstacle Detection System Can Improve Wheelchair Control via Movement Imagery. *Sensors* **2024**, *24*, 918. [\[CrossRef\]](#)
95. Xue, X.; Li, Z.; Zhang, D.; Yan, Y. A deep reinforcement learning method for mobile robot collision avoidance based on double dqn. In Proceedings of the 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE), Vancouver, BC, Canada, 12–14 June 2019; pp. 2131–2136.
96. Tai, L.; Paolo, G.; Liu, M. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 31–36.
97. Cimurs, R.; Suh, I.H.; Lee, J.H. Goal-driven autonomous exploration through deep reinforcement learning. *IEEE Robot. Autom. Lett.* **2021**, *7*, 730–737. [\[CrossRef\]](#)
98. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In Proceedings of the International conference on machine learning. PMLR, New York City, NY, USA, 19–24 June 2016; pp. 1928–1937.
99. Ibarz, J.; Tan, J.; Finn, C.; Kalakrishnan, M.; Pastor, P.; Levine, S. How to train your robot with deep reinforcement learning: Lessons we have learned. *Int. J. Robot. Res.* **2021**, *40*, 698–721. [\[CrossRef\]](#)
100. Zhu, K.; Zhang, T. Deep reinforcement learning based mobile robot navigation: A review. *Tsinghua Sci. Technol.* **2021**, *26*, 674–691. [\[CrossRef\]](#)
101. Shi, H.; Shi, L.; Xu, M.; Hwang, K.S. End-to-end navigation strategy with deep reinforcement learning for mobile robots. *IEEE Trans. Ind. Inform.* **2019**, *16*, 2393–2402. [\[CrossRef\]](#)

102. Zhang, J.; Springenberg, J.T.; Boedecker, J.; Burgard, W. Deep reinforcement learning with successor features for navigation across similar environments. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 2371–2378.
103. Dini, P.; Saponara, S. Cogging torque reduction in brushless motors by a nonlinear control technique. *Energies* **2019**, *12*, 2224. [[CrossRef](#)]
104. Pierpaolo, D.; Saponara, S. Control system design for cogging torque reduction based on sensor-less architecture. In Proceedings of the Applications in Electronics Pervading Industry, Environment and Society: APPEPIES 2019 7, Pisa, Italy, 12–13 September 2019; Springer: Berlin/Heidelberg, Germany, 2020; pp. 309–321.
105. Bernardeschi, C.; Dini, P.; Domenici, A.; Saponara, S. Co-simulation and verification of a non-linear control system for cogging torque reduction in brushless motors. In Proceedings of the Software Engineering and Formal Methods: SEFM 2019 Collocated Workshops: CoSim-CPS, ASYDE, CIFMA, and FOCLASA, Oslo, Norway, 16–20 September 2019; Revised Selected Papers 17; Springer: Berlin/Heidelberg, Germany, 2020; pp. 3–19.
106. Dini, P.; Saponara, S. Design of an observer-based architecture and non-linear control algorithm for cogging torque reduction in synchronous motors. *Energies* **2020**, *13*, 2077. [[CrossRef](#)]
107. Bernardeschi, C.; Dini, P.; Domenici, A.; Palmieri, M.; Saponara, S. Formal verification and co-simulation in the design of a synchronous motor control algorithm. *Energies* **2020**, *13*, 4057. [[CrossRef](#)]
108. Cosimi, F.; Dini, P.; Giannetti, S.; Petrelli, M.; Saponara, S. Analysis and design of a non-linear MPC algorithm for vehicle trajectory tracking and obstacle avoidance. In *Applications in Electronics Pervading Industry, Environment and Society. ApplePies 2020*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 229–234.
109. Dini, P.; Saponara, S. Model-based design of an improved electric drive controller for high-precision applications based on feedback linearization technique. *Electronics* **2021**, *10*, 2954. [[CrossRef](#)]
110. Bernardeschi, C.; Dini, P.; Domenici, A.; Mouhagir, A.; Palmieri, M.; Saponara, S.; Sassolas, T.; Zaourar, L. Co-simulation of a model predictive control system for automotive applications. In Proceedings of the International Conference on Software Engineering and Formal Methods, Virtual Event, 6–10 December 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 204–220.
111. Dini, P.; Saponara, S. Processor-in-the-Loop Validation of a Gradient Descent-Based Model Predictive Control for Assisted Driving and Obstacles Avoidance Applications. *IEEE Access* **2022**, *10*, 67958–67975. [[CrossRef](#)]
112. Pacini, F.; Di Matteo, S.; Dini, P.; Fanucci, L.; Bucchi, F. Innovative Plug-and-Play System for Electrification of Wheel-Chairs. *IEEE Access* **2023**, *11*, 89038–89051. [[CrossRef](#)]
113. Pacini, F.; Dini, P.; Fanucci, L. Cooperative Driver Assistance for Electric Wheelchair. In Proceedings of the International Conference on Applications in Electronics Pervading Industry, Environment and Society, Genova, Italy, 28–29 September 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 109–116.
114. Dini, P.; Ariaudo, G.; Botto, G.; Greca, F.L.; Saponara, S. Real-time electro-thermal modelling and predictive control design of resonant power converter in full electric vehicle applications. *IET Power Electron.* **2023**, *16*, 2045–2064. [[CrossRef](#)]
115. Dini, P.; Basso, G.; Saponara, S.; Romano, C. Real-time monitoring and ageing detection algorithm design with application on SiC-based automotive power drive system. *IET Power Electron.* **2024**. [[CrossRef](#)]
116. Fujimoto, S.; Hoof, H.; Meger, D. Addressing function approximation error in actor-critic methods. In Proceedings of the International conference on machine learning. PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 1587–1596.
117. Nvidia Container Toolkit. Available online: <https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/latest/sample-workload.html> (accessed on 3 December 2023).
118. *Americans with Disabilities Act of 1990. Public Law 101–336. 1990. 104 Stat. 327, July 26, 1990*; U.S. Government Printing Office: Washington, DC, USA, 1990.
119. NVIDIA. *Jetson Nano 2GB Developer Kit*; NVIDIA: Santa Clara, CA, USA, 2020.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.