2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Nonlinear Model Predictive Control of a BMI-guided Wheelchair for Navigation in Unknown Environments

Davide De Lazzari, Piero Simonetto, Niccolò Turcato, Luca Tonin, Ruggero Carli

Abstract— **The ability to discern human intentions from brain signals has opened the possibility of leveraging Brain-Machine Interfaces (BMIs) for the control of robotic devices, especially benefiting individuals with severe motor disabilities. In this work, we present a novel approach for navigating a semiautonomous wheelchair towards targets generated by a BMI, all while ensuring collision avoidance. Our approach employs Nonlinear Model Predictive Control (NMPC) for real-time trajectory generation in unknown and dynamic environments. The empirical results obtained from real-world experiments clearly demonstrate the advancements of our solution over current state-of-the-art techniques. Our implementation is proven to outperform well-established methods in terms of both smoothness and alignment with the user's intended behavior.**

I. Introduction

The direct decoding of user intentions from brain signals offers the opportunity to employ Brain-Machine Interfaces (BMIs) as tools for seamlessly integrating humans into the control loop of robotic devices [1], [2]. This extends the potential of Human-Robot Interaction (HRI) to a broader range of usages and applications. In particular, reducing dependence on physical actions holds significant promise for individuals with severe motor disabilities.

Previous research has investigated the usage of brain signals to interface with robots [3], [4]. The results are promising; however, direct and continuous control of the robot remains a challenge due to the inherent instability of biological signals. While there are some demonstrated advantages associated with extended user training [5], a purely BCI-driven robotic device remains impractical.

Hereby, we present a novel method for navigating a semiautonomous wheelchair in unknown environments using a model predictive controller for real-time trajectory generation. Our approach relies on the user's ability to generate a spatial target with a BMI, and on the wheelchair sensors to detect nearby obstacles. A depiction of the experimental setup is presented in Fig. 1.

Safe navigation of mobile robots and autonomous vehicles is a major area of research. A common approach involves addressing the path planning and path following problems separately.

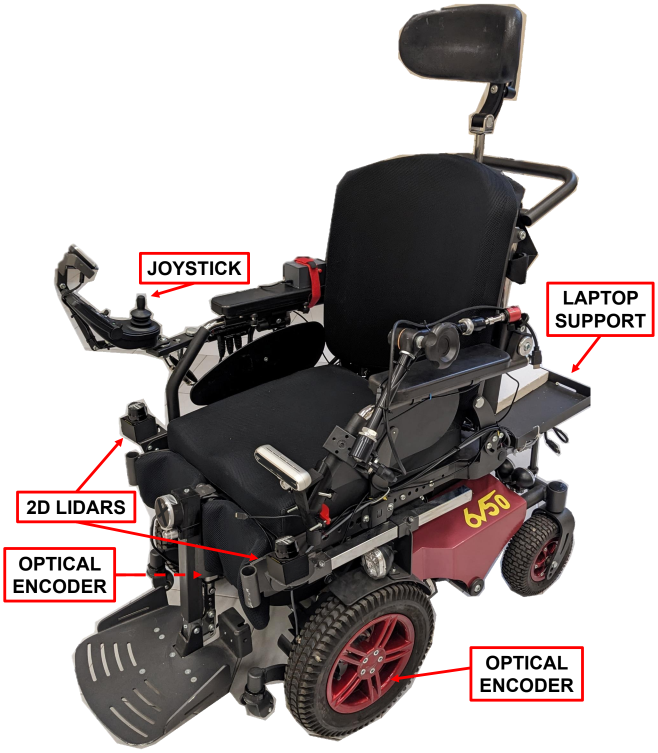

Fig. 1: The powered wheelchair employed in the experiments

Considering path planning [6], various algorithms can be found in the literature. Artificial Potential Field (APF) methods represent the target with an attractive field and obstacles with repulsive fields. The formulation is simple but they often get stuck in local minima. Other approaches include graph search-based algorithms, which perform a global search after discretizing the space. Their main drawback relies on the intrinsic tradeoff between discretization accuracy and computational time. Incremental search methods, like, the RRTs algorithms, aim to address this limitation by progressively refining the discretization of the configuration space. Nevertheless, handling dynamic obstacles while maintaining real-time performances remains an active research area.

In the last decades, Model Predictive Control (MPC) has been largely employed in mobile robotics and automotive research, due to its prediction abilities and capability of handling constraints. However, due to the high computational

This study was carried out within the PNRR research activities of the consortium iNEST (Interconnected North-Est Innovation Ecosystem) funded by the European Union Next-GenerationEU (Piano Nazionale di Ripresa e Resilienza (PNRR) – Missione 4 Componente 2, Investimento 1.5 – D.D. 1058 23/06/2022, $ECS\_00000043$). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

All authors are with the Department of Information Engineering, University of Padova, Italy. The authors' e-mails are respectively: delazzarid@dei.unipd.it, piero.simonetto@phd.unipd.it, turcatonic@dei.unipd.it, luca.tonin@unipd.it, carlirug@dei.unipd.it

demand in the case of nonlinear systems, MPC has mostly been used for trajectory tracking and path following [7]. Nonetheless, with the advancement of nonlinear programming tools, Nonlinear MPC (NMPC) has been employed in recent years also for trajectory generation [8]. Specifically, in our setup, NMPC can serve as a powerful tool for real-time trajectory generation. We can formulate a single optimization problem that takes into account both the system model and the presence of obstacles to obtain a trajectory.

Employing a model predictive controller to guide an autonomous wheelchair has already been exploited [9], [10]. In [9] the authors employ MPC for path following using a linearized kinematic model. While, in [10], they formulate an MPC problem for trajectory generation by considering linear kinematics and convex obstacles.

Other solutions for trajectory planning of powered wheelchairs employ mostly APF-based methods [11], [12]. However, APF planners provide greedy solutions and do not take into consideration the system dynamics.

In our approach, a spatial target is periodically generated with a non-invasive BMI. Considering the unreliability of brain signals over time and the long-term physical strain on the individual, we deliberately update the target slowly to improve robustness and alleviate prolonged stress on the person. Therefore, our ultimate goal is to generate the optimal trajectory in real time while guaranteeing comfort and safety without any prior knowledge of the environment. Following the receding-horizon principle, our controller applies only the first input, thus resulting in a responsive controller able to adapt to environmental changes. Our problem formulation accounts for both velocity and acceleration constraints and incorporates arbitrary obstacles as soft constraints within the cost function. We carry out experiments both in a Gazebo [13] simulation and on the physical system. It's worth noting that, in these experiments, spatial targets are generated by the user via a graphical interface rather than through a BMI. We assess our implementation by benchmarking it against established mobile robot controllers found within the ROS [14] stack, specifically DWA [15] and TEB [16], thereby demonstrating the efficacy of our approach.

Hence, the primary contribution of this paper lies in the development of an innovative method for navigating a BMI-guided wheelchair in unknown environments, along with its validation through real-world experiments. Our solution represents a significant advancement over current state-of-the-art techniques. It specifically considers time-varying arbitrarily-shaped obstacles, enabling navigation in dynamic and cluttered environments. Additionally, the inclusion of kinodynamic constraints and real-time trajectory generation results in better real-world performance, that closely resembles human behavior.

The paper is organized as follows. In Section II we describe the system model and formalize the control problem. Subsequently, in Section III we present our method and define the optimization problem. Lastly, in Section IV we describe our experimental setup and show the results of our experiments. Section V concludes the paper.

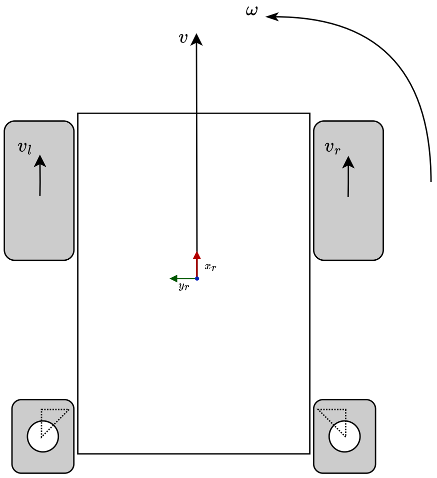

Fig. 2: Wheelchair kinematics.

II. Modeling and Problem Formulation

In this section, we will address the control problem by introducing the wheelchair model and formalizing safety requirements for collision avoidance.

A. Wheelchair model

Dynamical models for rubber-tired vehicles have been widely studied in the literature, each offering distinct qualities and weaknesses. On one side, a precise mathematical model offers more accurate system behavior predictions. Conversely, it may lead to excessive computational demand and hinder real-time feasibility. Furthermore, highly detailed models often necessitate accurate parameter estimation and precise sensing of physical parameters such as wheel orientations and velocity.

With this in mind, the wheelchair model adopted in this work is the so-called unicycle model, as depicted in Fig. 3. This model is simple but effective enough to capture the behavior of most electric wheelchairs in the market, namely those of the differential-drive type.

The wheelchair actuation inputs are the linear and angular velocities. Nevertheless, the kinematic equations have been extended to consider velocities as states and their derivatives as input. By doing so, we can reduce the wheelchair acceleration and better estimate the real system evolution.

Thereby, the model input $\mathbf{u} = \begin{bmatrix} a & \alpha \end{bmatrix}^T \in \mathbb{R}^2$ consists of the linear and angular accelerations respectively. The outputs include the wheelchair position and orientation in the global reference system, together with the measured linear and angular velocities, namely, $\mathbf{x} = \begin{bmatrix} x & y & \theta & v & \omega \end{bmatrix}^T \in \mathbb{R}^5$.

The continuous-time model $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t))$ becomes

$$\begin{cases} \dot{x}(t) = v(t)\cos(\theta(t)) \\ \dot{y}(t) = v(t)\sin(\theta(t)) \\ \dot{\theta}(t) = \omega(t) \\ \dot{v}(t) = a(t) \\ \dot{\omega}(t) = \alpha(t) \end{cases} . \tag{1}$$

The model is considered also subject to input and state constraints, namely maximum accelerations and velocities.

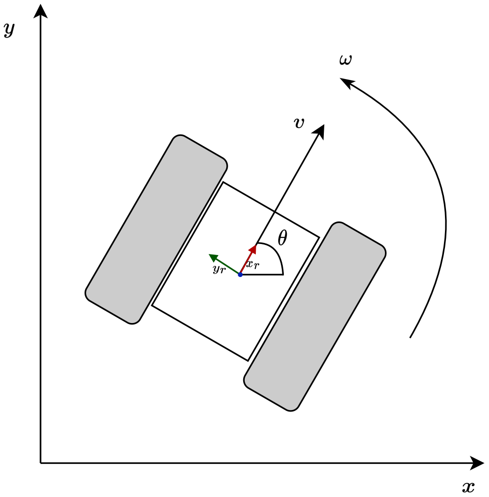

Fig. 3: Unicycle model.

B. Presence of obstacles

The sensors mounted on the wheelchair serve the purpose of detecting the nearby obstacles to avoid collisions. In this section, we assume that the sensing system provides an occupancy grid of the neighboring area at each time step. Each occupied grid cell is mapped to a small circular obstacle with radius $r_{obst} > 0$ equal to half of the cell size. Eventually, each obstacle i is entirely defined by its spatial coordinates $\boldsymbol{\xi}_i = \begin{bmatrix} x_{\xi_i} & y_{\xi_i} \end{bmatrix}^T \in \mathbb{R}^2$.

Therefore, in order to avoid collisions, the wheelchair position should respect the constraints

$$\sqrt{\|x - x_{\xi_i}\|_2^2 + \|x - x_{\xi_i}\|_2^2} - r_{wc} - r_{obst} > 0 \tag{2}$$

$\forall i$ at each timestep. Where $r_{wc} > 0$ is the radius of the wheelchair by approximating it with a circle.

C. Problem Statement

Taking into account all the previous assumptions on the system and obstacles, the control problem to solve becomes the following. Given a target spatial position, we aim to reach this goal while avoiding collisions. Furthermore, the trajectory should be sufficiently smooth in order to ensure comfort and be close to the user's expected behavior. Namely, we want to avoid abrupt changes in speed and direction while reaching the target in a reasonable amount of time.

III. Nonlinear Model Predictive Control

The synthesis of a Model Predictive Controller requires the formulation of a receding-horizon optimal control problem, which involves the system evolution, a cost function, and possibly additional constraints.

A. Cost function

Due to the nonlinearity of the system model, the final optimization problem will result in a Nonlinear Programming (NLP) problem. Nevertheless, a smooth cost function is desirable since most solvers for real-time applications rely on Sequential Quadratic Programming (SQP). In consideration of this, we adopted a quadratic cost with respect to the distance from the target, the wheelchair velocities and accelerations. Obstacles are included in the cost function as soft constraints, following a similar approach to traditional Artificial Potential Field (APF) methods. Specifically, we employed a sigmoid-like function with respect to the distance of the wheelchair from the obstacles.

With a slight abuse of notation, we identify as $\mathbf{x}(k)$ and $\mathbf{u}(k)$ the state and input at time kT_s, where $k \in \mathbb{Z}$ is the discrete-time variable and T_s is the sampling time. Therefore, the NMPC cost function over the prediction horizon NT_s is defined as

$$\begin{aligned} J = \sum_{k=0}^{N-1} \|\mathbf{x}(k) - \mathbf{x}_{ref}\|_{\mathbf{Q}}^2 + \|\mathbf{u}(k)\|_{\mathbf{R}}^2 + J_\xi(\mathbf{x}(k), \mathbf{p}) + \\ + \|\mathbf{x}(N) - \mathbf{x}_{ref}\|_{\mathbf{Q}_N}^2 + J_\xi(\mathbf{x}(N), \mathbf{p}) \ . \end{aligned} \tag{3}$$

J_ξ represents the cost function associated with the obstacles, namely,

$$J_\xi(\mathbf{x}, \mathbf{p}) = \max_{i=1,\dots,M} \tilde{J}_\xi(\mathbf{x}, \boldsymbol{\xi}_i) \ , \tag{4}$$

where

$$\tilde{J}_\xi(\mathbf{x}, \boldsymbol{\xi}_i) = \kappa \frac{1}{1 + \exp(\lambda \, d(\mathbf{x}, \boldsymbol{\xi}_i))} \ , \tag{5}$$

$$d(\mathbf{x}, \boldsymbol{\xi}_i) = \sqrt{\|x - x_{\xi_i}\|_2^2 + \|x - x_{\xi_i}\|_2^2} - r_{wc} - r_{obst} \ . \tag{6}$$

$\mathbf{Q}$, $\mathbf{Q}_N \in \mathbb{R}^{5\times 5}$, $\mathbf{R} \in \mathbb{R}^{2\times 2}$ are weight matrices, with $\mathbf{Q} \succcurlyeq 0$, $\mathbf{Q}_N \succcurlyeq 0$, $\mathbf{R} \succ 0$.

The state reference $\mathbf{x}_{ref} \in \mathbb{R}^5$ is constant and equals to $\begin{bmatrix} x_{ref} & y_{ref} & \theta_{ref} & v_{ref} & \omega_{ref} \end{bmatrix}^T$. Bearing in mind that in our setup the target is going to be inferred from a BMI, we won't consider any reference in the angular position, therefore θ_{ref} is arbitrary and its relative weight is set to zero. Moreover, in the experiments, we set $v_{ref} = \omega_{ref} = 0$.

$\mathbf{p}$ is the vector comprising all the currently detected obstacles, i.e., $\begin{bmatrix} \boldsymbol{\xi}_1^T & \boldsymbol{\xi}_2^T & \cdots & \boldsymbol{\xi}_M^T \end{bmatrix}^T$. The obstacles are treated as static throughout the entire prediction horizon. Nevertheless, they are re-measured at each time step in order to ensure reactiveness in changing environments.

Equation (5) is a scaled sigmoid function where $\kappa > 0$ and $\lambda > 0$ determine its maximum value and steepness (see Fig. 4). $d(\mathbf{x}, \boldsymbol{\xi}_i)$ represents the distance of the wheelchair from an obstacle minus the wheelchair and obstacle radii (as in Eq. (2)).

The dependence of J on $\mathbf{x}_{0:N} = \begin{bmatrix} \mathbf{x}(0)^T & \mathbf{x}(1)^T & \cdots & \mathbf{x}(N)^T \end{bmatrix}$, $\mathbf{u}_{0:N-1} = \begin{bmatrix} \mathbf{u}(0)^T & \mathbf{u}(1)^T & \cdots & \mathbf{u}(N-1)^T \end{bmatrix}$, and $\mathbf{p}$ is omitted for clarity. The matrices $\mathbf{Q}$, $\mathbf{Q}_N$ are set to penalize states far from the goal position, and matrix $\mathbf{R}$ is set to minimize the linear and angular acceleration of the system, thus having smoother changes in velocity. Instead of imposing strict constraints as in Eq. (2), we opt for soft constraints as defined by Eq. (4). This approach still ensures safety provided that appropriate weights are selected, while also reducing computational burden.

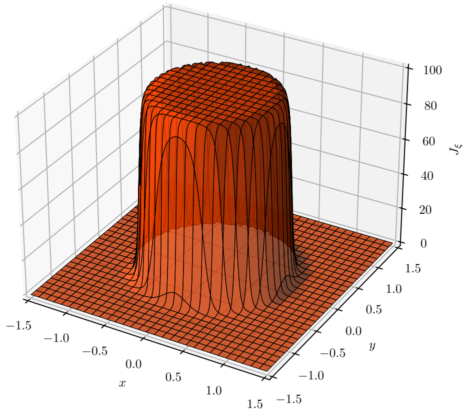

Fig. 4: The cost function $\tilde{J}_\xi(\mathbf{x}, \boldsymbol{\xi}_i)$ w.r.t. (x, y), with $x_{\xi_i} = y_{\xi_i} = 0$, $\lambda = 50$, and $\kappa = 100$.

B. Optimization Problem

In this work, the Nonlinear Optimal Control Problem (NOCP) is solved with MATMPC [17], an open-source software built in MATLAB for real-time NMPC applications. In MATMPC, a Nonlinear Programming (NLP) problem is formulated by applying direct multiple shooting over the prediction horizon, i.e.,

$$\begin{aligned} \min_{\mathbf{x}_{0:N}, \mathbf{u}_{0:N-1}} \quad & J \\ \text{s.t.} \quad & \mathbf{x}(0) = \mathbf{x}_0 \\ & \mathbf{x}(k+1) = f_d\left(\mathbf{x}(k), \mathbf{u}(k)\right) \\ & \mathbf{x}_{min} \leq \mathbf{x}(k) \leq \mathbf{x}_{max} \\ & \mathbf{u}_{min} \leq \mathbf{u}(k) \leq \mathbf{u}_{max} \quad . \end{aligned} \tag{7}$$

f_d refers to the system discrete-time evolution obtained by numerical integration.

$\mathbf{x}_{min}$ and $\mathbf{x}_{max}$ represent the state constraints, comprising $v_{min} \leq v(k) \leq v_{max}$ and $\omega_{min} \leq \omega(k) \leq \omega_{max}$.

$\mathbf{u}_{min}$ and $\mathbf{u}_{max}$ are the input constraints, namely, $a_{min} \leq a(k) \leq a_{max}$ and $\alpha_{min} \leq \alpha(k) \leq \alpha_{max}$.

Problem (7) is solved using Sequential Quadratic Programming (SQP) with line-search strategies and the `HPIPM` [18] solver.

IV. Experiments

In this section, we provide a description of the experimental setup, and outline the results of our experiments.

The experiments are conducted first in a Gazebo simulation and finally on the physical system. We evaluate our approach by comparing it to well-established algorithms for mobile robot navigation available within the ROS framework, specifically DWA and TEB. Moreover, to assess the different methods, spacial references have been selected using a graphical interface instead of a BMI. The DWA planner employs a Dynamic Window Approach to generate velocity commands based on a local costmap, while the Timed Elastic Band (TEB) local planner optimizes a trajectory through a sparse scalarized optimization problem considering execution time, separation from obstacles, and kinodynamic compliance. In the real scenario, we also tested a human user's performance by manually guiding the wheelchair with its onboard joystick controller.

In this section, we will refer to the controller introduced in Section III simply as NMPC. The parameters and weights for NMPC used in the experiments are reported respectively in Table I and Table II. DWA and TEB were used with their default parameters, with the only exception of the maximum velocities, which were capped at ± 0.3 m/s and ± 0.8 rad/s.. Moreover, in the NMPC, we limited the number of SQP iterations to 30 to ensure real-time performance.

A. Experimental setup

The physical setup is depicted in Fig. 1, with the wheelchair kinematics reported in Fig. 2. Motion is achieved by activating the front wheels, which are fixed and individually powered by motors. This configuration, commonly referred to as *differential drive*, allows the wheelchair to maneuver by differentially rotating the two wheels. The wheels are actuated by a non-tunable low-level controller, developed by the manufacturer, that takes as input the reference linear and angular velocities and applies the required torques to each wheel. The rear wheels are passive castor wheels, primarily serving as support for the wheelchair frame. An onboard odometry system measures the velocities and displacements of the front wheels with two optical encoders but it lacks sensors on the rear wheels. Furthermore, two lidar sensors are mounted at the front to measure obstacle distances. Lastly, during the experiments, a laptop computer responsible for system control was positioned at the rear of the wheelchair.

The wheelchair kinematics and sensors are reproduced in a Gazebo simulation using Gazebo's `Differential Drive` and `Laser` plugins.

In both the physical and simulated setups, the wheelchair is connected to a local ROS network. Inputs and outputs are forwarded in real time using ROS messages. NPMC was implemented within MATLAB and Simulink using the MATLAB ROS Toolbox. The `GazeboPlugin` was used for simulations.

With respect to the kinematical model introduced in Eq. (1), the state feedback is obtained from the onboard odometry. At each time step the ROS package `costmap_2d` computes a local costmap by merging the sensor outputs. The costmap is represented by an occupancy grid used by all planners indicating the obstacle locations. The grid cells are sufficiently small (on the order of 10^{-2} meters) to accurately represent intricate obstacle shapes. Specifically, as described in Section II, NMPC considers each occupied grid cell as a single obstacle. Both in the simulated and real experiments, the computation of the planners and the NMPC was performed on an Intel i7-11800H processor.

B. Simulated experiments

The Gazebo environment used for simulations replicates the layout of the real environment and the kinematics and dynamics of the wheelchair, featuring four obstacles arranged to resemble a typical room with everyday objects and furniture.

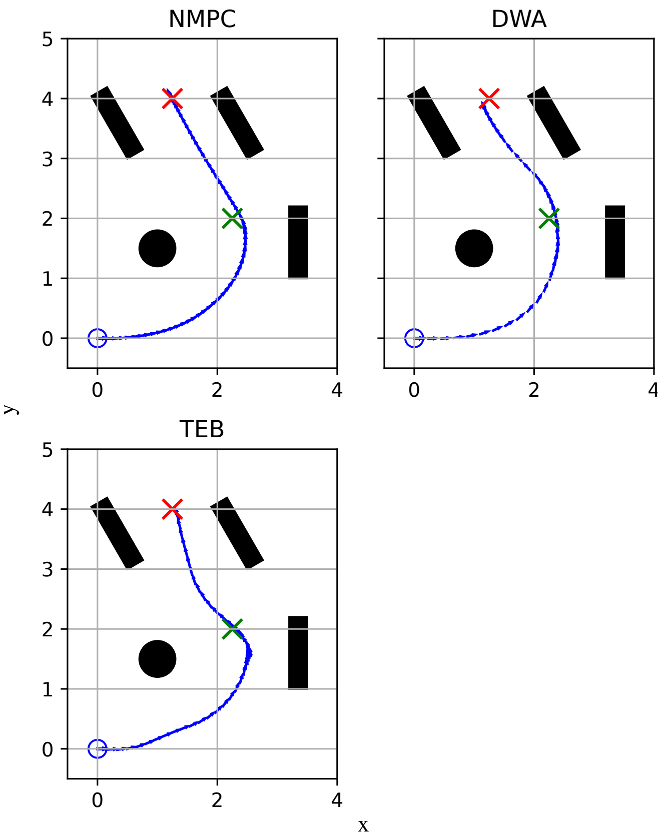

Fig. 5: Experimental results in the simulated environment. For each tested controller we plot with small blue arrows the downsampled onboard odometry. The black figures represent the objects in the scene. The center of the blue circle is the starting point of the experiment. The green cross marks the first target location, and the red cross the second one, which is provided to the planner only upon reaching the first. The global reference frame is unique and kept constant for all experiments.

In the simulated experiments we provide two goals sequentially to mimic a scenario where spatial targets are obtained through a combination of BMI and intention recognition, following this procedure: (i) The wheelchair model is spawned at the origin of the global reference frame, aligning its frame with the global one. (ii) The controller receives a first reference position and generates the velocity inputs until the goal is reached. (iii) Upon reaching the first target, the controller is supplied with a second and final reference and guides the wheelchair until reaching it.

In Fig. 5 we report the trajectories generated by NMPC, DWA, and TEB in the simulated room [1].

Each controller should maneuver the wheelchair from the initial position to the first target, making a left turn while avoiding collisions. Then, to get to the second location, it needs to navigate the corridor created by the two parallel obstacles. All reference positions were equal within each test. Moreover, it is important to emphasize that all controllers are unaware of the final goal while approaching the first one.

As can be seen in Fig. 5, successfully reached both targets while avoiding the obstacles. NMPC and DWA generated similar trajectories, with NMPC being slightly smoother in the second half. On the other hand, the path followed by the TEB-controlled wheelchair presents unnecessary changes of direction, resulting in a less natural and comfortable trajectory.

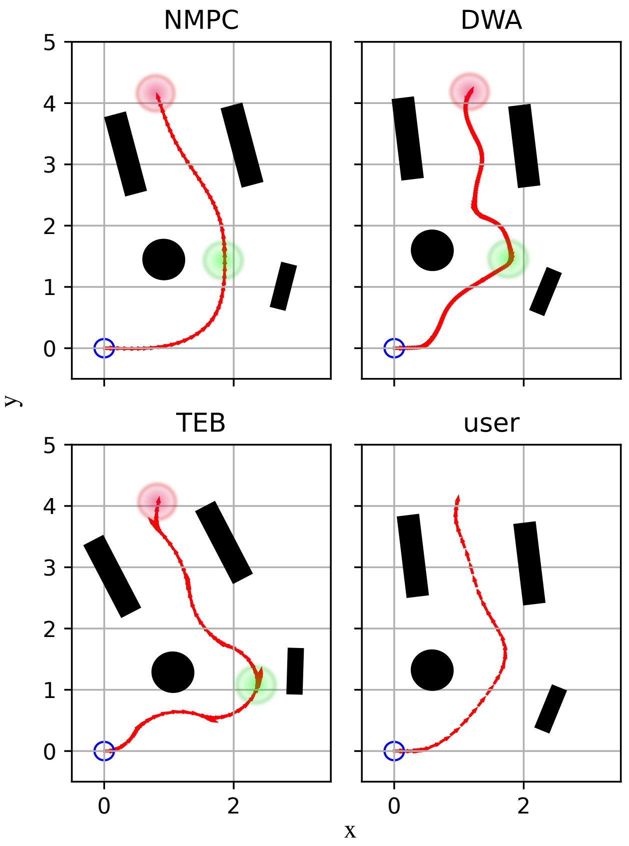

Fig. 6: Experimental results in real-world environment. For each tested controller we plot with small red arrows the downsampled onboard odometry. The black figures resemble the positions of the objects in the scene. The center of the blue circle is the starting point of the experiments. The green and red circles represent the first and second targets respectively, both chosen with a graphical interface. The *user* plot refers to a human driving wheelchair.

C. Real-world experiments

The real-world experiments follow a similar procedure as the ones conducted in simulation. As depicted in Fig. 6, the environment comprises four obstacles for the wheelchair to navigate among. As in the simulation, a first and a second target are provided to the controller sequentially.

Before each run, the wheelchair is manually positioned at the starting point, setting the initial pose as the global reference frame for the individual experiment. Thus, given the challenge of consistently reproducing the same starting position in every test, the experiments do not share a common reference frame. To address this issue, the two targets were manually selected using the RVIZ software [19], which provides a graphical interface for the ROS framework. As a result, the reference positions exhibit slight variations across the different experiments.

The experimental results are shown in Fig. 6, where the fourth plot refers to a human user manually driving the wheelchair with its onboard joystick controller.

[1] A recording of the simulated experiment is available at https://youtu.be/rCUJUujnWH0

While all planners demonstrate satisfactory performance in simulations, the real-world experiments paint a different picture. By examining the plots, we can observe that NMPC closely mirrors the simulation results. In contrast, the performances of DWA and TEB worsen significantly. NMPC offers a trajectory characterized by minimal changes of direction, with large, smooth turns, aligning well with the user demonstration. TEB's path still resembles the simulated one, but the small changes of direction become more pronounced, yet remain relatively smooth. In contrast, DWA struggles to follow a path similar to the one in the simulation, presenting multiple heading corrections. Furthermore, it is worth emphasizing that the cost function associated with the obstacles, as defined in Eq. (4) and illustrated in Fig. 4, is quite steep. As a result, NMPC does not penalize paths proximal to objects while ensuring safety. This characteristic makes the NMPC controller well-suited for navigating in cluttered environments, such as indoor settings.

Parameter	Value
r_{wc}	0.8 m
r_{obst}	0.05 m
λ	50
κ	100
N	100
T_s	0.08 s
v_{min}	0 m/s
v_{max}	0.3 m/s
ω_{min}	-0.8 rad/s
ω_{max}	0.8 rad/s
a_{min}	-0.2 m/s^2
a_{max}	0.2 m/s^2
α_{min}	-0.1 rad/s^2
α_{max}	0.1 rad/s^2

TABLE I: Parameters of the nonlinear model predictive controller used in all experiments.

Weight	Value
$\mathbf{Q}$	$\text{diag}(10, 10, 0, 1, 1)$
$\mathbf{Q}_N$	$\text{diag}(50, 50, 0, 50, 50)$
$\mathbf{R}$	$\text{diag}(10, 10)$

TABLE II: Weights of the nonlinear model predictive controller used in all experiments.

V. Conclusions

In this paper, we presented a new method for navigating a BMI-guided wheelchair in dynamic environments. We designed a Nonlinear Model Predictive Controller that drives the wheelchair to a spatial target generated by the user while avoiding collisions. We validated the effectiveness of our approach with real-world experiments by benchmarking it against the DWA and TEB planners. The proposed controller provides better overall performance in terms of smoothness and resemblance of the user's expected behavior.

This paper marks the initial phase of this research, with the potential to improve the quality of life for individuals with severe motor disabilities and contribute to the advancement of the field of Human-Robot Interaction.

Experiments seamlessly integrating an actual BMI and dynamic obstacles, while considering the precise dynamics and geometry of the wheelchair, are left for future work.

References

[1] T. Carlson and J. Del R. Millan, "Brain-Controlled Wheelchairs: A Robotic Architecture," *IEEE Robotics & Automation Magazine*, vol. 20, no. 1, pp. 65–73, Mar. 2013. [Online]. Available: http://ieeexplore.ieee.org/document/6476692/

[2] J. D. R. Millán, "Combining brain-computer interfaces and assistive technologies: state-of-the-art and challenges," *Frontiers in Neuroscience*, vol. 1, 2010. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnins.2010.00161/abstract

[3] R. Leeb, L. Tonin, M. Rohm, L. Desideri, T. Carlson, and J. d. R. Millán, "Towards independence: A bci telepresence robot for people with severe motor disabilities," *Proceedings of the IEEE*, vol. 103, no. 6, pp. 969–982, 2015.

[4] L. Tonin and J. D. R. Millán, "Noninvasive Brain–Machine Interfaces for Robotic Devices," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 191–214, May 2021. [Online]. Available: https://www.annualreviews.org/doi/10.1146/annurev-control-012720-093904

[5] A. Takai, D. Rivela, G. Lisi, T. Noda, T. Teramae, H. Imamizu, and J. Morimoto, "Investigation on the Neural Correlates of Haptic Training," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Miyazaki, Japan: IEEE, Oct. 2018, pp. 519–523. [Online]. Available: https://ieeexplore.ieee.org/document/8616093/

[6] B. Patle, A. Pandey, D. Parhi, A. Jagadeesh, *et al.*, "A review: On path planning strategies for navigation of mobile robot," *Defence Technology*, vol. 15, no. 4, pp. 582–606, 2019.

[7] K. Kanjanawanishkul and A. Zell, "Path following for an omnidirectional mobile robot based on model predictive control," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3341–3346.

[8] F. Micheli, M. Bersani, S. Arrigoni, F. Braghin, and F. Cheli, "Nmpc trajectory planner for urban autonomous driving," *Vehicle system dynamics*, vol. 61, no. 5, pp. 1387–1409, 2023.

[9] V. T. Nguyen, C. Sentouh, P. Pudlo, and J.-C. Popieul, "Path following controller for electric power wheelchair using model predictive control and transverse feedback linearization," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2018, pp. 4319–4325.

[10] G. Bardaro, L. Bascetta, E. Ceravolo, M. Farina, M. Gabellone, and M. Matteucci, "MPC-based control architecture of an autonomous wheelchair for indoor environments," *Control Engineering Practice*, vol. 78, pp. 160–174, Sept. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S096706611830220X

[11] L. F. Manta, C. F. Pană, D. Cojocaru, I. C. Vladu, D. M. Pătraşcu-Pană, and A. Dragomir, "Apf-based control for obstacle avoidance in smart electric wheelchair navigation," in *2021 22nd International Carpathian Control Conference (ICCC)*. IEEE, 2021, pp. 1–6.

[12] S. A. Kumar, J. Vanualailai, and A. Prasad, "Assistive technology: autonomous wheelchair in obstacle-ridden environment," *PeerJ Computer Science*, vol. 7, p. e725, 2021.

[13] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154 vol.3.

[14] S. A. I. L. et al., "Robotic operating system." [Online]. Available: https://www.ros.org

[15] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[16] C. Rösmann, F. Hoffmann, and T. Bertram, "Planning of multiple robot trajectories in distinctive topologies," in *2015 European Conference on Mobile Robots (ECMR)*. IEEE, 2015, pp. 1–6.

[17] Y. Chen, M. Bruschetta, E. Picotti, and A. Beghi, "Matmpc - a matlab based toolbox for real-time nonlinear model predictive control," in *2019 18th European Control Conference (ECC)*, 2019, pp. 3365–3370.

[18] G. Frison and M. Diehl, "Hpipm: a high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020.

[19] H. R. Kam, S.-H. Lee, T. Park, and C.-H. Kim, "Rviz: a toolkit for real domain data visualization," *Telecommunication Systems*, vol. 60, pp. 337–345, 2015.