2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

A Trajectory Planner For Mobile Robots Steering Non-Holonomic Wheelchairs In Dynamic Environments

Martin Schulze[1] **, Friedrich Graaf[1] **, Lea Steffen[1], Arne Roennau[1], Rüdiger Dillmann[1]

***Abstract*— Motion planning for mobile robot platforms is one of the long-established research fields in robotics. In this paper, we propose a trajectory planner for mobile holonomic robots to steer non-holonomic conventional passive wheelchairs in dynamic environments. The challenges to overcome when steering a wheelchair are to find smooth feasible trajectories, maintain a fast reactive response to dynamic obstacles and to satisfy a set of additional constraints such as limiting physical forces acting on the wheelchair occupants. Our approach is a variant of the timed-elastic-bands (TEB) planner, which includes a footprint of the wheelchair during optimization, and generates a steering angle which is then consumed by an arm controller to actuate the relative orientation between the wheelchair and the mobile platform. This is realized by posing new non-holonomic and kinodynamic constraints on the TEB planner and an implementation of a suitable real-time dual-arm controller for executing steering commands. We demonstrate our results based on a TEB baseline comparison in simulation using functional models of our robot HoLLiE and a wheelchair.**

I. INTRODUCTION

Facing demographic declines over the last few decades and an already precarious multi-causal situation of short-staffed nursing wards, the need for more rapid progress in care robotics becomes increasingly eminent. The development of care robotics for task-specific applications in nursing environments, however, is practically still in its infancy. This is particularly true for humanoid-like mobile manipulator systems aiming to facilitate the physical work of caregivers. Besides tasks involving direct physical contact-based interactions with patients, there is also a great deal of work with the potential to be automatized. Such tasks include pick and place services of empty wheelchairs or following patients with an empty wheelchair during walking exercises supervised by caregivers. In this scenario, the robot would relieve one of the caregivers for other patient-related tasks.

In this work, we present an approach of a reactive local planner for feasible kinodynamic trajectories of robotic systems consisting of self-propelled mobile manipulators and non-holonomic constrained passive wheelchairs, as they are present in healthcare facilities. Autonomous wheelchairs are much simpler to realize but their range of potential applications is limited in comparison to prospective humanoid robots.

*This research received funding as the project HoLLiEcares from the German Federal Ministry of Education and Research (BMBF) under the grant no. 16SV8406.

**The authors contributed equally to this paper.

[1]FZI Research Center for Information Technology, 76131 Karlsruhe, Germany schulze@fzi.de, graaf@fzi.de, steffen@fzi.de, roennau@fzi.de, dillmann@fzi.de

Our main contribution is an extension for the timed elastic-bands (TEB) planner, enabling feasible kinodynamic trajectories for wheelchairs. Secondly, we provide additional constraints to improve comfort for wheelchair occupants. Moreover, we implemented a real-time two-arm controller in ROS control to guarantee synchronous movements of both end effectors. We intend to release our software as open-source packages in the near future.

The remainder of this paper is structured as follows: related work is summarized in section II. Section III introduces our robot models and simulation setup. The wheelchair motion planner and arm controller are presented in section V and IV, respectively. Finally, we discuss evaluation results in section VI and give a conclusion and outlook in section VII.

Fig. 1: HoLLiE coupled to a wheelchair.

II. RELATED WORK

Motion planning for mobile robots with mounted manipulators for steering non-holonomic wheeled movable objects, such as carts, exists for more than two decades. An example of an early work can be found in [1] but their trajectory planning approach was only limited to simple open-loop trajectories. A consecutive work has presented a whole-body zero-moment-point (ZMP) [2] but no trajectory planning following a global path was involved. A study of fetch-and-give tasks, particularly for applications in elderly care, has been conducted by Pyo et al. [3]. Algorithmic details on the employed trajectory planning approach and its performance

are not explicitly stated, but they do mention being close to [4] in this regard. To our knowledge, the most similar overall approach to this work has been presented by Scholz et al. [4]. Their state lattice planner discretizes the search space based on hand-crafted motion primitives, which then allows to reduce the optimization problem to a graph-search problem. Our work relies on a repeated non-linear optimization task, whereas adding constraints is simple and does not lead to exponentially growing search spaces. Lattice state approaches can also serve as a warm-start for optimal control formulations [5] which we did not consider in our work so far.

Motion planning problems of truck-trailer systems in mobile robotics and autonomous driving [6] are conceptually similar to cart pushing problems. One of the early truck-trailer approaches [7] deforms an initially collision free global path with respect to non-holonomic, and in particular, obstacle potential-field constraints until the path is collision free. Their elastic-bands approach differs from ours by the choice of optimization method and do not optimize time intervals explicitly. Moreover, the TEB maintains representative samples of each homotopy class. Sampling-based approaches such as various RRT variants are very popular candidates for non-holonomic trajectory planning issues [8]. Drawbacks are usually high computational costs or restricted search spaces.

III. ROBOTIC PLATFORM AND SIMULATION

For the scope of this study we employ a functional model of our custom build multi-functional service robot HoLLiE (figure 1) which we integrated into the Gazebo Classic simulator [9]. The HoLLiE robotic platform is based on [10], but apart from a light-weight spring-loaded parallelogram structure with two articulated joints and parts of the original shell, all hardware has been renewed. The upper body is mounted centered on the omnidirectional mobile *Clearpath Ridgeback* platform which is available in Gazebo. This includes virtual ROS controllers for its Mecanum drive, odometry and a plugin for a virtual IMU. Two virtual Sick microScan3 planar LIDAR scanners are mounted on front and back, respectively. Merging both laser scans yields a 360° field-of-view, due to the construction of the chassis. The head of HoLLiE is mounted on a 2 DOF joint (Schunk Powerball). A model of a Microsoft Azure Kinect (RGB-D) is placed on-top of the Schunk Powerball and emulated in Gazebo using the Gazebo-ROS-Openni-Kinect sensor plugin. Two 6 DOF PILZ PRBT-6 (non-compliant) industry robotic arms are mounted on the upper part of the torso and Schunk PG70 1 DOF grippers as end effectors. The force-accumulation in Gazebo simulations of multi-contact forces, grasping in particular, is inherently unstable using the ODE or Bullet physics engine. Therefore, we decided to write a Gazebo plugin to dynamically attach both grippers to the wheelchair handles, which can also be released via a ROS service call. This is similar to the planned setup on the robot. A fixed gripper will be used, as available grippers usually do not have enough grasping force to steer a wheelchair without slip.
The simulated platform possesses a tare weight of roughly $165\,\mathrm{kg}$ and can reach a maximum speed of $1.1\,\mathrm{m/s}$. The upper body has a total weight of roughly $62\,\mathrm{kg}$. The wheelchair model for Gazebo is designed based on Bischoff S-Eco 300 using 6 continuous hinge joints. 2 for the 24" rear wheels and 4 for the 2 DOF 8" supporting front wheels. The wheelchair weights approximately $18\,\mathrm{kg}$.

Mapping the environment in our work is achieved with Google's Cartographer [11] while traversing the environments (see VI-A) with HoLLiE to capture the necessary sensor data and to assemble a 2D occupancy grid-map representation. This occupancy gridmap then is kept frozen. Pre-mapped environments are not strictly required, but eliminate side effects such as changing coordinate frames for evaluation purposes. Dynamic obstacles are still captured by the local-planner, which directly receives sensor-data from the range scanners and the Azure Kinect camera. The wheelchair (and potentially the patient's legs) strongly obscures the view of frontal laser scanners of the Ridgeback platform. Thus, without prior filtering, the cartographer and TEB would register the wheelchair and the person constantly as obstacles. Since the point clouds recorded by the Azure

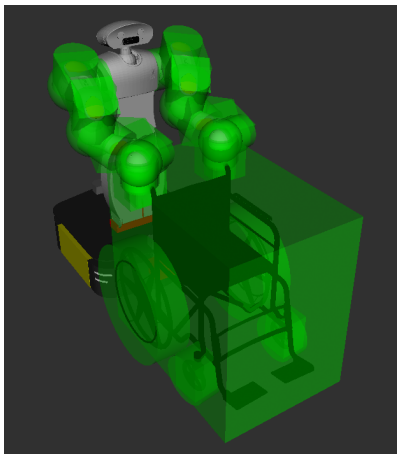


Fig. 2: Illustration of filter areas around HoLLiE and the wheelchair. All points of the point clouds and laser scans that fall into the green areas are removed by the filter.

Kinect in HoLLiE's head also contain the wheelchair, they must also be filtered out. To cover a possible passanger, a rectangular bounding box is placed around the coordinate frame of the wheelchair. To have the bounding box always in the right place in real-time, the position of the wheelchair is determined by proprioception. Hence, the pose of the wheelchair frame can be determined directly from the angle and the assumption that the wheelchair and the gripper are firmly connected. As HoLLiE's arms can appear in the field of view of the Azure Kinect, the self-body-filter filters the point cloud for all collisions of points with links of the arms and grippers. Our approach, as visualized in figure 2, is almost identical to the one in [4]. Finally, the ground plane

is filtered out by a PCL pass-through filter.

IV. ARM CONTROL

The real-time capable dual-arm controller receives target angle from the TEB local controller and translates them into actual joint commands, to guarantee accurate and synchronized movements of both arms. It implements all necessary ROS Control joint-position interfaces for an accurate position control. Due to budget constraints, HoLLiE (and her digital twin) does not have force-torque sensors mounted on the arms, wrists or grippers. Unfortunately, this circumstance limited this study to a rigid wheelchair-coupling and prohibits this system from being adaptive to various wheelchair weights, terrain slopes, friction-profiles and practical certification in commercial applications. Cartesian compliant controller [12], for instance, could be considered in potential future work. The controller update routine is outlined in algorithm 1 and *readJointValues* as well as *writeJointCommands* imply direct interaction with the hardware abstraction layer of ROS Control. *ComputeBeta* first performs a simple forward-kinematics pass of both arms to compute the positions and orientations of both grippers. Then it computes their midpoint, projected onto the xy-plane (ground), in order to finally measures the relative planar angle between the midpoint and the frame origin sitting at the endpoint of $link_0$ (position of the virtual pivot point). *ComputeDelta* is supposed to compute the absolute difference of target and current theta angles and feed them into a PID controller. *addAndLimitBeta* in our implementation is as simple as $\beta_{next} = \min(\beta_{current} + \beta_{\Delta}, \beta_{\Delta max})$ whereas $\beta_{\Delta max}$ must match the achievable deltas of joint values per control cycle. The limit needs to be small enough to avoid deviation of both arms in Cartesian space. Next, the expected gripper poses at the steering angle β_{next} are computed and transformed to the joint space via *TracIK* [13] for inverse kinematics.

Algorithm 1: Dual-Arm Control Loop Pseudocode

Input: $\beta_{target} \in [-\pi, +\pi]$
Data: $\theta_{\Delta max}$ `// HW joint limit per cycle`
$\phi_{left}, \phi_{right} \leftarrow readJointValues()$
$\beta_{current} \leftarrow computeBeta(\phi_{left}, \phi_{right})$
$\beta_{\Delta} \leftarrow computeDelta(\beta_{target}, \beta_{current})$
$\beta_{next} \leftarrow addAndLimitBeta(\beta_{current}, \beta_{\Delta}, \beta_{\Delta max})$
/* $P_{left}, P_{right} \in SE(3)$ */
$P_{left}, P_{right} \leftarrow computeGripperPoses(\beta_{next})$
/* $\tilde{\phi}_{left}, \tilde{\phi}_{right} \in [-\pi, +\pi] \cup \{\varnothing\}$ */
$\tilde{\phi}_{left}, \tilde{\phi}_{right} \leftarrow computeInvKin(P_{left}, P_{right})$
if $\tilde{\phi}_{left}$ *is* $\varnothing$ *or* $\tilde{\phi}_{right}$ *is* $\varnothing$ **then**
  $writeJointCommands(\tilde{\phi}_{left}, \tilde{\phi}_{right})$

V. MOTION PLANNING

The motion planning problem is to find an optimal and collision free trajectory with respect to a set of configuration-space, (non-)holonomic and kinodynamic constraints. The TEB planner [14] used in this work cast the motion planning problem into a least-square optimization problem, expressed as a sparse hyper-graph in the g2o-framework [15], which then is efficiently solved by the Levenberg-Marquardt method using a sparse solver. Obstacles are associated with nodes in the optimization graph using costmap discretization, either on cell level or clustered as polygon representations.
During the optimization procedure, TEB planner alternates the elements of a path $Q = \{s_k\}_{k=0,\dots,N}$ of poses $s_k = [x_k, y_k, \theta_k] \in \mathbb{R}^2 \times \mathcal{S}^1$ and the intermediate time differences $\tau = \{\Delta T_k \in \mathbb{N}\}_{k=0,\dots,N}$ until it has converged to a (local) minima. The path Q is initialized with the result of a global planner. After optimization, the output is consecutively checked for (kinodynamic) validity and no occurring collisions. A violation of kinodynamic constraints or collisions triggers re-planning or abortion.
Let $\xi = \{\beta_k \in \mathcal{S}^1\}_{k=0,\dots,N}$ denote the wheelchair steering angles at every time step k (see V-A). The formal definition of a TEB is originally given by the tuple $B := (Q, \tau)$ which we in our work then extend to $\tilde{B} := (Q, \tau, \xi)$.
The scalarized multi-objective optimization is formally given by (1) with γ_i denoting the relative weight of component function f_i. We follow [14] and use error cost functions as component functions as described in V-B.

$$f(\tilde{B}) = \sum_i \gamma_i f_i(\tilde{B})$$
$$B^* = \arg\min_{\tilde{B}} f(\tilde{B}) \tag{1}$$

Note that the entire optimization loop runs continuously at high-frequency rates, which makes it suitable for our use-case.

A. COUPLED KINEMATICS

The coupled kinematics of our system restricts the previously unconstrained Ridgeback platform by introducing non-holonomic constraints. The kinematics of the wheelchair resemble differential drives, which can be easily understood, as the support wheels on the front do not restrict any movements or the rear wheels. We consider motion planning of the platform and the control problem for the arms separately.

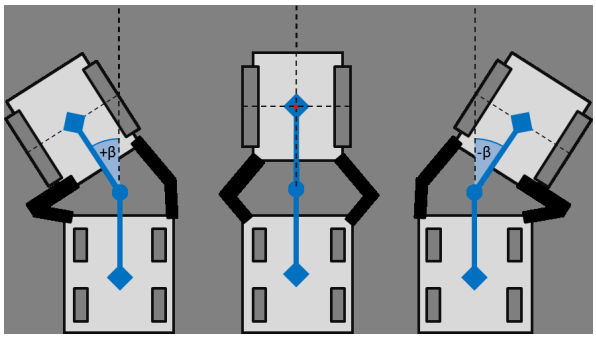


Fig. 3: The coupled kinematics and their abstract representation as virtual links and pivot point as a center of rotation.

The configuration space of the coupled system is a 6-tuple $(x_{pl}, y_{pl}, \theta_{pl}, x_{whl}, y_{whl}, \theta_{whl})$. We follow the approach

in [4] and simplify the arm control by allowing the origin of the wheelchair frame only to be rotated around a virtual pivot point (figure 3) on a circular arc by a limited angle β. We denote the virtual link rooted on the platform by $link_0$ and the other rooted on the wheelchair $link_1$, respectively. While the TEB planner would not suffer from an additional dimension due to its sparse optimization graph structure, our intention of restricting the problem is simply to reduce the complexity of the arm controller, see IV.

B. OPTIMIZATION CONSTRAINTS

Table I gives a selection of (holonomic) constraints, already pre-defined in the ROS reference implementation of the TEB planner. All listed constraints are fairly local, as they only consider the state and time-differences ΔT at or around a single time-step k. The constraint for avoiding obstacles is also local due to the restriction to the nearest obstacle. The only exception is the *Path Duration* constraint, which is a hyper-edge spanning across all time delta variables.

TABLE I: A list of constraints as pre-defined in the reference TEB implementation.

Variable	Error Function
Translational Velocity	
$v_k = \frac{\Delta s_k}{\Delta T_k}$	$e_{v_k} = \begin{cases} 0, & \text{if } \lvert v_k\rvert < v_{max} \\ \lvert v_k\rvert - v_{max} & \text{else} \end{cases}$
Angular Velocity	
$\omega_k = \frac{\Delta \phi_k}{\Delta T_k}$	$e_{\omega_k} = \begin{cases} 0, & \text{if } \lvert\omega_k\rvert < \omega_{max} \\ \lvert\omega_k\rvert - \omega_{max}, & \text{else} \end{cases}$
Translational Acceleration	
$a_k = \frac{2(v_{k+1}-v_k)}{\Delta T_k+\Delta T_{k+1}}$	$e_{a_k} = \begin{cases} 0, & \text{if } \lvert a_k\rvert < a_{max} \\ \lvert a_k\rvert - a_{max}, & \text{else} \end{cases}$
Angular Acceleration	
$\vec{a}_k = \frac{2(\omega_{k+1}-\omega_k)}{\Delta T_k+\Delta T_{k+1}}$	$e_{\vec{a}_k} = \begin{cases} 0, & \text{if } \lvert\vec{a}_k\rvert < \vec{a}_{max} \\ \lvert\vec{a}_k\rvert - \vec{a}_{max}, & \text{else} \end{cases}$
Path Duration	
ΔT_k	$e_T = \sum_{k=0}^{N-1} \Delta T_k$
Obstacles	
$d_k := \lvert\lvert s_k - o_{nn}\rvert\rvert_2$, $o_{nn} :=$ closest obstacle	$e_O = \begin{cases} 0, & \text{for } d_k \geq d_{min} \\ d_{min} - d_k, & \text{else} \end{cases}$

All constraints mentioned in table I are applied on the Ridgeback platform and the wheelchair separately (except for the global *Path Duration*). In case of the wheelchair, we can compute its absolute pose s_w and orientation θ_w by a simple coordinate transform at each time step k. Additionally, we apply the following non-holonomic constraint (2) [16] only on the wheelchair to enforce the differential kinematics during planning. The Ridgeback platform is allowed to slide sideways as long as this constraint is satisfied. We assign this particular error term a high relative scalar weighting to lower the chance of an invalid trajectory after optimization.

$$e_{kin} = \left(\begin{bmatrix} \cos(\theta_{w_k}) \\ \sin(\theta_{w_k}) \end{bmatrix} + \begin{bmatrix} \cos(\theta_{w_{k+1}}) \\ \sin(\theta_{w_{k+1}}) \end{bmatrix} \right) \times (s_{w_{k+1}} - s_{w_k}) \quad (2)$$

In the remainder of this section we discuss our novel constraints.

$$e_\beta = \begin{cases} 0, & \text{for } \lvert\beta_k\rvert < \beta_{max} \\ \lvert\beta_k\rvert - \beta_{max}, & \text{else} \end{cases} \quad (3)$$

To limit the allowed radius of the arms, we add (3) to our set of constraints. If necessary, limits on its derivates can be applied similarly to the constraints in table I.

Next, we aim to improve the riding comfort by posing penalties when overshooting the maximum allowed centrifugal forces $a_{cen_{max}}$ as given in equation (5).

$$\begin{aligned} \Delta\theta_{w_k} &= \theta_{w_{k+1}} - \theta_{w_k} \\ a_{cen} &= \frac{\lvert\lvert s_{w_{k+1}} - s_{w_k}\rvert\rvert_2}{\left\lvert 2 * \sin\left(\frac{\Delta\theta_{w_k}}{2}\right)\right\rvert} * \frac{\Delta\theta_{w_k}^2}{\Delta T_k^2} \end{aligned} \quad (4)$$

$$e_{cen} = \begin{cases} 0, & \text{for } \lvert a_{cen}\rvert < a_{cen_{max}} \\ \lvert a_{cen}\rvert - a_{cen_{max}}, & \text{else} \end{cases} \quad (5)$$

Nothing prevents the optimization procedure so far to toggle the platform's (angular) acceleration values in very short time intervals. To further reduce stress on the wheelchair, we punish rapid changes in the form of (angular) acceleration derivative constraints (7). The acceleration values a_k represent the accelerations that occur during the time span $\Delta T_{k+1} + \Delta T_k$. A constraint for the angular accelerations can be derived in analogy to (6) and (7) by substituting the acceleration values and limit with their angular variants.

$$j_k = \frac{3(a_{k+1} - a_k)}{\Delta T_k + \Delta T_{k+1} + \Delta T_{k+2}} \quad (6)$$

$$e_{j_k} = \begin{cases} 0, & \text{for } \lvert j_k\rvert < j_{max} \\ \lvert j_k\rvert - j_{max}, & \text{else} \end{cases} \quad (7)$$

Lastly, a particular corner case whereas the steering signal right at the start of a planned trajectory approaches zero, often coinciding with narrow passages, needs to be intercepted. In equation (9) we ensure a pair-wise minimum velocity among the first 5 trajectory poses.

$$v_{(ij)_k} = \frac{\Delta s_{w_{(ij)_k}}}{\Delta T_{(ij)_k}}, \quad 0 \leq i < j \leq 5 \quad (8)$$

$$e_{firstVel} = \begin{cases} 0, & \text{for } \lvert v_{(ij)_k}\rvert > v_{min} \\ \lvert v_{(ij)_k}\rvert - v_{min}, & \text{else} \end{cases} \quad (9)$$

VI. RESULTS AND EXPERIMENTS

The experimental results are structured as follows: success rates, path lengths and durations are shown in VI-A. The forces acting on the wheelchair during the evaluation are discussed in VI-B. The baseline for all experiments is the reference implementation of the TEB with one additional constraint on the turning radius, which has been set to $1.14\,\mathrm{m}$ to match our setup. This effectively forces the platform into a kinematic that resembles the Ackermann kinematic. No

further constraints on the baseline do apply. All experiments are performed on a desktop PC using an Intel Core i7 6700K with $4.0\,\mathrm{GHz}$ and $64\,\mathrm{GiB}$ of RAM without a dedicated GPU. Collision-avoidance will not be benchmarked in this work but is inherited feature of the original TEB planner due to its pre-defined set of constraints which are evaluated multiple times each second.

The evaluation maps, shown in figure 4, are chosen to cover complicated situations that the planner is confronted with in real-life scenarios. They include a narrow place to turn $180°$, three types of turnings at different angles ($45°$, $90°$ and $135°$) as well as a doorway.

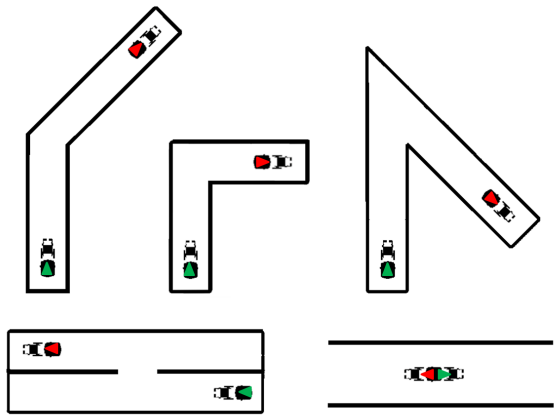


Fig. 4: Evaluation maps, covering versatile scenarios, for comparing the presented planner to the baseline TEB. The start poses are marked in green and the target poses in red.

A. Comparative experiments on evaluation maps

The success rate of the experiments on all three maps was 100% for the newly proposed planner. For the comparative evaluation, two metrics are employed, first, the success weighted by path time (SPT) and, second, the accuracy with which the target was reached. The SPT is given by

$$\frac{1}{N}\sum_{i=1}^{N} = S_i \frac{t_r}{max(t_p, t_r)}, \tag{10}$$

whereby $S \in [0, 1]$ indicates success and N the number of runs. t_p denotes the time required by the planner and t_r the time by the reference planner. The more runs were faulty or longer than the reference, the lower the value for SPT. Equation 10 is based on the success weighted by path length (SPL) [17], but slightly adapted. Thus, the original form refers to the length of the path, but here it has been replaced by its duration. This better suits the *Path Duration* constrain than the path length, which is not being explicitly optimized for. Note that the success rates $SPT, SPL \in [0, 1]$ are bounded particularly from above. The results of this evaluation are given in table II.

As expected, the results strongly suggest that the additional constraints in V-B do slow down the navigation task (except in the doorway scenario) particularly in narrow situations. Besides this exception, the results were surprisingly close to that of the reference planner. In turnaround-like situations, the non-holonomic wheelchair constraints seem even to be beneficial in terms of SPT. It is noteworthy, that the basic TEB planner was unable to finish the course on the easiest route, the $135°$ corner, as its behavior isn't as robust in narrow spaces. All recorded values are available in table III. The accuracy with which the target was reached is measured with respect to three factors:

- distance from target Δxy
- orientation $\Delta\theta$
- angular position of the wheelchair $\Delta\beta$

The amount of the deviations depends on the tolerance, which is set to $0.2\,\mathrm{m}$ for the Euclidean distance to the target pose. The tolerance range for the orientation θ, as well as the angle β is set to $0.2\,\mathrm{rad}$. In table III, the presented approach is compared to the basic TEB regarding path length, duration, roughness and accuracy. The roughness was calculated by the equation

$$R = \int_{t_0}^{t_1} \left| \frac{1}{L} \frac{d\kappa}{dt} \right|^2 dt \tag{11}$$

as shown in Palmieri et al. [8]. As this calculation heavily depends on the used path discretization for the integral, and the used units of measurement, the results can vary heavily between papers. For our calculation we used the Menger curvature κ with the absolute path length L in meters and the time t in milliseconds. All experiments were performed with the same starting position and the same target for both planners. These results show that the presented planner is, even with the more complex kinematic structure, able to compete with the basic TEB planner. One additional reason for the better behavior in path duration of the basic TEB planner is the lack of constraints on the jerk of HoLLiE, as this constraint was added to the new planner. This allows HoLLiE to accelerate faster, covering the same distances in less time, but at the expense of comfort for the occupant as described in the following section. The value of the roughness is, as described above, only meaningful by comparing it to the roughness of the reference planner, which is similar in most of the scenarios. Only the doorway and narrow turn have a significant difference, which originates from the additional maneuvering needed in these cases. The W-TEB, for example, has to make more turns and wider curves to position the wheelchair for it to fit through the doorway, while in the narrow turn case the

TABLE II: SPT and SPL for our planner relative to the baseline.

Environment	SPT	SPL
corner 135°	1	0.94
corner 90°	0.72	0.98
corner 45°	0.80	1
doorway	0.44	0.96
narrow turn	1	0.75

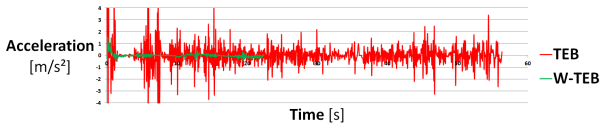
TABLE III: A comparison of the baseline planner, to our TEB variant, referred to as *Wheelchair-TEB* (W-TEB). The path length is given in l, the duration in t, the geometric roughness in r and the accuracy in regard to Δxy, $\Delta\theta$ and $\Delta\beta$.

		corner 135°		corner 90°		corner 45°		doorway		narrow turn	
		TEB	W-TEB	TEB	W-TEB	TEB	W-TEB	TEB	W-TEB	TEB	W-TEB
l	[m]	11.8	12.4	9.4	9.6	15.8	15.3	10.7	11.1	3.4	4.5
t	[s]	56.4	22.4	13.6	18.9	23.0	28.9	18.2	41.2	15.2	13.1
r		4630	3383	36.54	45.04	84.57	16.34	1.228	13 523	9338	125.8
Δxy	[m]	5.14	0.11	0.02	0.15 m	0.18	0.13	0.02	0.12	0.03	0.19
$\Delta\theta$	[rad]	0.11	0.02	0.00	0.02	0.02	0.02	0.004	0.03	0.007	0.13
$\Delta\beta$	[rad]	NA	0.04	NA	0.03	NA	0.03	NA	0.02	NA	0.09

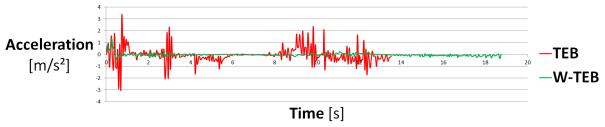
reference planner needs to reverse multiple times. This is due to the limited space to turn in one go with the given minimum turning radius.

B. *Comfortability over the course of acceleration*

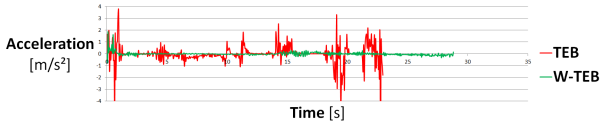
The comfort of the driving experience is important because it has a direct impact on the user's acceptance of the new system. To determine the comfort, the course of the acceleration is recorded and its rate of change analyzed. However, as there is no single general threshold at which a comfortable path can be distinguished from an uncomfortable one, a comparison with the paths recorded by the reference planner was carried out instead. The acceleration curve over the time is shown in figure 5. The accelerations occur in the reference measurements significantly stronger than in the wheelchair planner. The rates of change of acceleration are also significantly higher, which would make the path of the reference measurement much more uncomfortable for the person riding in the wheelchair. The acceleration trajectories are measured on HoLLiE, since the reference planner cannot push a wheelchair. However, via the wheelchair's connection to HoLLiE, it can be assumed that the wheelchair is subjected to either the same accelerations or even lower. The reduction of the acceleration change and thus of the maximum acceleration depends on the additional constraint inserted in the wheelchair planner for the jerk. It is noteworthy that the limit of the maximum acceleration is the same in both planners.



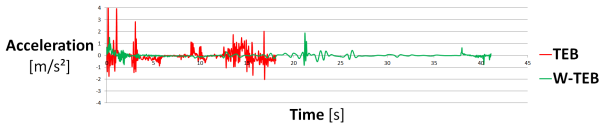
(a) Corner 135°



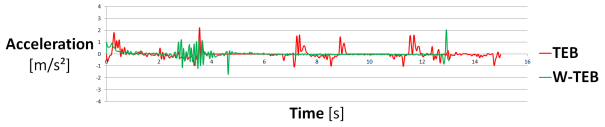
(b) Corner 90°



(c) Corner 45°



(d) Doorway



(e) Narrow turn

Fig. 5: Measurements of the acceleration curve over time on all five maps in figure 4 for the basic TEB and the presented planner.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a wheelchair compatible variant of the TEB planner and a compatible real-time controller for two robotic arms. We evaluated the entire system (including sensing and mapping) in a Gazebo simulation on multiple environments with varying specific difficulties and compared our results against the reference TEB implementation. Our results demonstrate that the proposed TEB variant is fully capable of planning smooth kinodynamically feasible trajectories for a non-holonomic coupled robotic system. Moreover, the acceleration profiles demonstrate with low stress acting on the wheelchair, particularly when compared to the baseline.

Future work aims to run the experiments on the real HoLLiE hardware and study the acceptance of generated trajectories and their smoothness. We will also include test persons in this study. Extending the sensor pipeline and TEB variant for a tracking functionality is another priority we will aim to achieve.

REFERENCES

[1] J. Tan and N. Xi, "Integrated sensing and control of mobile manipulators," ser. IEEE International Conference on Intelligent Robots and Systems, vol. 2, pp. 865 – 870 vol.2.

[2] S. Nozawa, T. Maki, M. Kojima, S. Kanzaki, K. Okada, and M. Inaba, "Wheelchair support by a humanoid through integrating environment recognition, whole-body control and human-interface behind the user," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 1558–1563. [Online]. Available: http://ieeexplore.ieee.org/document/4650903/

[3] Y. Pyo, K. Nakashima, T. Tsuji, R. Kurazume, and K. Morooka, "Motion planning for fetch-and-give task using wagon and service robot," in *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, pp. 925–932. [Online]. Available: http://ieeexplore.ieee.org/document/7222657/

[4] J. Scholz, S. Chitta, B. Marthi, and M. Likhachev, "Cart pushing with a mobile manipulation system: Towards navigation with moveable objects," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, pp. 6115–6120. [Online]. Available: http://ieeexplore.ieee.org/document/5980288/

[5] K. Bergman, O. Ljungqvist, and D. Axehill, "Improved Path Planning by Tightly Combining Lattice-Based Path Planning and Optimal Control," vol. 6, no. 1, pp. 57–66. [Online]. Available: https://ieeexplore.ieee.org/document/9084267/

[6] J. David and P. V. Manivannan, "Control of truck-trailer mobile robots: A survey," vol. 7, no. 4, pp. 245–258. [Online]. Available: http://link.springer.com/10.1007/s11370-014-0152-z

[7] F. Lamiraux, D. Bonnafous, and O. Lefebvre, "Reactive Path Deformation for Nonholonomic Mobile Robots," vol. 20, no. 6, pp. 967–977. [Online]. Available: http://ieeexplore.ieee.org/document/1362692/

[8] L. Palmieri, S. Koenig, and K. O. Arras, "RRT-based nonholonomic motion planning using any-angle path biasing," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 2775–2781. [Online]. Available: http://ieeexplore.ieee.org/document/7487439/

[9] 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. IEEE, pp. 2149–2154. [Online]. Available: http://ieeexplore.ieee.org/document/1389727/

[10] A. Hermann, J. Sun, Z. Xue, S. W. Ruehl, J. Oberlaender, A. Roennau, J. M. Zoellner, and R. Dillmann, "Hardware and software architecture of the bimanual mobile manipulation robot HoLLiE and its actuated upper body," in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE, pp. 286–292. [Online]. Available: http://ieeexplore.ieee.org/document/6584106/

[11] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1271–1278. [Online]. Available: http://ieeexplore.ieee.org/document/7487258/

[12] S. Scherzinger, A. Roennau, and R. Dillmann, "Forward Dynamics Compliance Control (FDCC): A new approach to cartesian compliance for robotic manipulators," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4568–4575.

[13] P. Beeson and B. Ames, "TRAC-IK: An open-source library for improved solving of generic inverse kinematics," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, pp. 928–935. [Online]. Available: http://ieeexplore.ieee.org/document/7363472/

[14] C. Rosmann, W. Feiten, T. Wosch, F. Hoffmann, and T. Bertram, "Efficient trajectory optimization using a sparse model," in *2013 European Conference on Mobile Robots*. IEEE, pp. 138–143. [Online]. Available: http://ieeexplore.ieee.org/document/6698833/

[15] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G^2o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, pp. 3607–3613. [Online]. Available: http://ieeexplore.ieee.org/document/5979949/

[16] C. Rosmann, F. Hoffmann, and T. Bertram, "Kinodynamic trajectory optimization and control for car-like robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 5681–5686. [Online]. Available: http://ieeexplore.ieee.org/document/8206458/

[17] N. Yokoyama, S. Ha, and D. Batra, "Success Weighted by Completion Time: A Dynamics-Aware Evaluation Criteria for Embodied Navigation." [Online]. Available: https://arxiv.org/abs/2103.08022