Coupled Mobile Manipulation via Trajectory Optimization with Free Space Decomposition

Spahn, Max; Brito, Bruno; Alonso-Mora, Javier

DOI
10.1109/ICRA48506.2021.9561821

Publication date
2021

Document Version
Final published version

Published in
Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA)

2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

2021 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9077-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRA48506.2021.9561821

Coupled Mobile Manipulation via Trajectory Optimization with Free Space Decomposition

Max Spahn, Bruno Brito and Javier Alonso-Mora

***Abstract*— This paper presents a real-time method for whole-body trajectory optimization of mobile manipulators in simplified dynamic and unstructured environments. Current trajectory optimization methods typically use decoupling of the mobile base and the robotic arm, which reduces flexibility in motion, does not scale to unstructured environments, and does not consider the future evolution of the environment, which is crucial to avoid dynamic obstacles. Given a goal configuration, such as waypoints generated by a global path planner, we formulate a receding horizon trajectory optimization minimizing the distance-to-target while avoiding collisions with static and dynamic obstacles. The presented method unifies the control of a robotic arm and a non-holonomic base to allow coupled trajectory planning. For collision avoidance, we propose to compute three convex regions englobing the robot's major body parts (i.e., base, shoulder-link and wrist-link) and thus reducing and limiting the number of inequality constraints, regardless of the number of obstacles in the environment. Moreover, our approach incorporates predicted trajectory information to smoothly, and in advance, avoid dynamic obstacles. The presented results show that trajectory optimization for the coupled system can reduce the total execution time by 48% and that applying the convex region generation for individual links allows keeping the computational costs low, even for complex scenarios, enabling onboard implementation.**

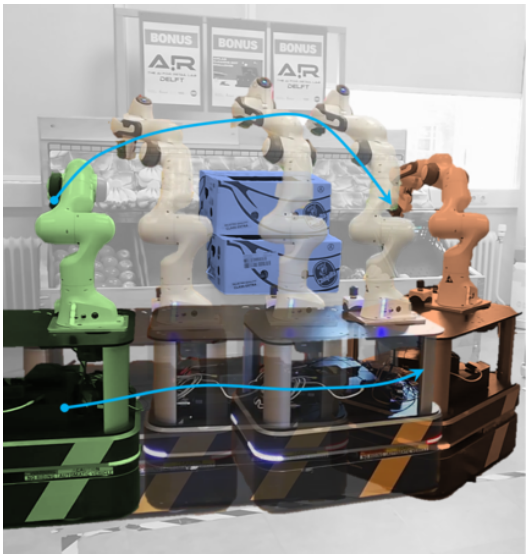


Fig. 1: Mobile manipulator performing a pick & place task using whole-body trajectory optimization with our coupled MPC formulation.

I. Introduction

Mobile manipulation is the field of robotics in which a mobile robot's locomotion ability is combined with the manipulation ability of a robotic arm. However, conventional trajectory optimization methods dealing with such systems often dramatically reduce flexibility by decoupling planning for the base and the arm. Motion for both parts then need to be synchronized or are executed sequentially. Both synchronization and sequencing limit the ability of the robot to perform complex tasks, and it was shown that decoupled and sequenced approaches show significantly higher operational times [1], [2]. Yet, solving the whole-body trajectory optimization problem is challenging due to a large number of degrees of freedom (e.g., 10 for the robot used in our experiments and shown in Fig. 1). Moreover, dynamic and unstructured environments have not been addressed in a coupled approach yet. Such environments have been extensively investigated for autonomous vehicles (e.g., mobile robots navigating through human crowds [3], [4], drones in cluttered environments [5], [6]). In the context of autonomous vehicles, model predictive control (MPC) can effectively incorporate future evolutions of the environment [7]. A coupled MPC scheme for mobile manipulators was introduced in [8], where locomotion and manipulation were softly decoupled through dynamic weight-setting. Dynamic obstacles were not considered, and the method suffered from increasing computational costs as the environment becomes more densely populated [9]. More specifically, collision avoidance in an unstructured environment typically results in many inequality constraints that scale with the number of obstacles.

In this work, we propose a whole-body trajectory optimization, sometimes referred to as MPC, using convex region decomposition of the free space on multiple kinematic chain links for collision avoidance with static obstacles. As a result, the number of inequality constraints remains constant regardless of the number of obstacles, allowing continuous control of the arm and the base to navigate through unstructured environments. Dynamic sphere-shaped obstacle avoidance is included using stage-dependent sphere-to-sphere inequality constraints.

Experimental results demonstrated that operational time is reduced considerably when using whole-body trajectory optimization. Computational costs for the optimization solver were independent of the number of obstacles present in the environment. Single dynamic obstacles moving at a constant

*This research was supported by Ahold Delhaize. All content represents the opinion of the author(s), which is not necessarily shared or endorsed by their respective employers and/or sponsors.

The authors are with the Department of Cognitive Robotics, Delft University of Technology, 2628 CD, Delft, The Netherlands {m.spahn, bruno.debrito, j.alonsomora}@tudelft.nl

velocity could be avoided successfully.

II. Related Works

Past works devoted to motion planning for mobile manipulators can be divided into two main categories: optimization-based and sampling-based methods [10], [11]. Sampling-based methods, such as rapidly exploring random trees (RRT's) [12]–[14] and probabilistic roadmaps (PRM's) [15], [16] are highly efficient at generating paths for systems with high-degree of freedom. However, these approaches consider static environments requiring a complete replanning if the conditions change and hence, are not applicable for dynamic environments [9].

In contrast, trajectory planning methods focus on executing generated paths while avoiding collisions in dynamic environments. Despite the vast number of works developing motion planning methods for mobile manipulators, most approaches either tackle the motion planning problem for the robot's base or the robot's arm. Pioneer works on motion planning methods for statically mounted manipulators employed potential fields for collision avoidance [17]. Building on the previous, [18] introduced the Circular Field method to address dynamic collision avoidance. Finally, to ensure collision avoidance for the end-effector when grasping a moving obstacle, [19] employed a repulsive vector.

A. Collision Avoidance For Mobile Robots

The dynamic window approach [20] and its new variant proposed in [21] have proven to be efficient in generating smooth trajectories for mobile robots in static and dynamic environments. To navigate among pedestrians, [22] introduced the Social Forces model imitating the human navigation behavior and using it as navigation policy for the robot.Yet, Social Forces and its variants rely on handcrafted functions limiting their ability to handle more complex navigation scenarios. To deal with a large number of agents, *ORCA* was proposed in [23] and later extended for non-holonomic bases in [24]. However, these approaches demonstrate highly reactive behaviors because they only consider one step look-ahead predictions. MPC schemes were proposed for mobile robots and autonomous vehicles in [7] and [25] allowing to optimize over a prediction horizon and avoid, in advance, dynamic obstacles. To enable coupled control of a mobile manipulator, collision avoidance must be performed in the 3D space which is usually not necessary for ground vehicles. Several 3D MPC formulation were proposed for drones to enable safe motion through cluttered environments [5], [6].

B. Collision Avoidance For Mobile Manipulators

Despite abundant research in trajectory planning for mobile robots and robotic arms, few works focused on coupling both systems' control. It was shown that coupling the base and the robotic arm motion leads to a considerable reduction of total operational time and smoother motions [1], [2]. Nevertheless, these methods were designed for static environments and did not allow real-time collision avoidance. Furthermore, trajectory planning for the coupled system is a precondition for effective interactive navigation, including opening doors [26], [27] or moving obstacles out of the way [28].

In the context of mobile manipulation, less research focused on collision avoidance in dynamic environments, including changing scenes and moving obstacles. A real-time controller using MPC was presented in [29], in which either a holonomic or a non-holonomic base was combined with a two-degree-of-freedom robotic arm mounted onto the base. Although hardware constraints were respected, no collision avoidance was considered. An MPC formulation for mobile manipulators with holonomic bases that allows collision avoidance was presented in [8]. The perceived obstacles were translated into a set of spheres to be respected by the MPC scheme. The proposed approach used dynamically changing weights to change between arm motion and locomotion, resulting in a locked arm during navigation. A different weight setting was used to perform motion underneath a horizontal bar with an a priori position. The work is extended to non-holonomic bases and includes object detection in moving underneath the horizontal bar [9].

C. Contribution Of This Work

In this work, a 3D collision avoidance for the entire kinematic chain is proposed that allows safe arm motion during all phases of the locomotion process. In contrast to the work of [9], no dynamic weighting is required, as collision avoidance can be guaranteed for the entire kinematic chain. The proposed work does not require an object detection method to allow safe motion, as the raw point cloud can be processed using free convex region generation [5]. Moreover, our approach reduces overall operational time while introducing a convex region generation to keep the computational costs low.

III. Methods

A. Definitions

Throughout this paper, vectors are denoted in bold lowercase letters, $\mathbf{x}$, matrices in capital, M, and sets in calligraphic uppercase, $\mathcal{S}$. $\|\mathbf{x}\|_Q = \mathbf{x}^T Q \mathbf{x}$ denotes the weighted squared norm.

B. Mobile Manipulator's Representation

Let us consider a velocity controlled mobile manipulator consisting of a mobile base with a mounted robotic arm. The coupled system's dynamics are described by the discrete-time non-linear system

$$\boldsymbol{z}_{k+1} = f(\boldsymbol{z}_k, \boldsymbol{u}_k), \tag{1}$$

where $\boldsymbol{z}_k$ and $\boldsymbol{u}_k$ describe the state and the control inputs at the time-step k, respectively. The robot's state is the base position and orientation, and the manipulator's joint positions, $\boldsymbol{z} = [x, y, \theta, \mathbf{q}^{\text{arm}}]$. The robot's control inputs are the left u^l and right u^r wheel velocity, and joint velocities $\dot{\mathbf{q}}^{\text{arm}}$, hence $\mathbf{u} = [u^l, u^r, \dot{\mathbf{q}}^{\text{arm}}]$. We denote $\mathcal{Z}$ and $\mathcal{U}$ as the corresponding state and control commands admissible sets,

respectively. The space occupied by the robot is denoted as $\mathcal{B}(\boldsymbol{z}) = \bigcup_{i\in\{1,\dots,n_{\text{links}}\}} \mathcal{B}_i(\boldsymbol{z})$ and $\mathcal{B}_i(\boldsymbol{z})$ denotes the space occupied by the i-th robot's link with $i \in [0, N_{\text{links}}]$. We approximate the state occupied by each link by spheres with radius r_i. The space occupied by the static obstacles and dynamic obstacles is represented as $\mathcal{O}^{\text{static}}$ and $\mathcal{O}^{\text{dynamic}}$, respectively. To limit the complexity of the problem we only consider a limited number $n_{\text{links}} \leq N_{\text{links}}$ of the robot's links and the n_{dyn} closest dynamic obstacles.

C. Mobile Manipulator's Model

Here, we assume a differential drive model for the base and first order dynamics for the robotic manipulator:

$$\dot{\boldsymbol{z}} = \begin{bmatrix} \frac{1}{2}cos(\theta)(u^l + u^r)r_{\text{wheel}} \\ \frac{1}{2}sin(\theta)(u^l + u^r)r_{\text{wheel}} \\ \frac{(u^r - u^l)r_{\text{wheel}}}{L_{\text{wheel}}} \\ \dot{\boldsymbol{q}}_{\text{arm}} \end{bmatrix}, \tag{2}$$

where r_{wheel} and L_{wheel} are the wheel radius and distance between the two controllable wheels, respectively. The discrete transition function $f(\boldsymbol{z}_k, \boldsymbol{u}_k)$ can be found using a discretization scheme (e.g. Backward-Euler or Runge-Kutta). The set of admissible states ($\mathcal{Z}$) and control inputs ($\mathcal{U}$) is defined by the joint position and velocity limits

$$\begin{aligned} \boldsymbol{q}_{\text{min}} \leq \boldsymbol{q}_{\text{arm}} \leq \boldsymbol{q}_{\text{max}} \\ \boldsymbol{u}_{\text{min}} \leq \boldsymbol{u}_{\text{wheels}} \leq \boldsymbol{u}_{\text{max}} \\ \dot{\boldsymbol{q}}_{\text{min}} \leq \dot{\boldsymbol{q}}_{\text{arm}} \leq \dot{\boldsymbol{q}}_{\text{max}}. \end{aligned} \tag{3}$$

where $\mathbf{q}_{\text{min}}$ and $\mathbf{q}_{\text{max}}$, $\mathbf{u}_{\text{min}}$ and $\mathbf{u}_{\text{max}}$ and, $\dot{\mathbf{q}}_{\text{min}}$ and $\dot{\mathbf{q}}_{\text{max}}$ are the minimum and maximum joint position position, wheel velocity limits and joint velocity limits, respectively.

D. Optimization Problem

Consider that a reference path in $\mathbb{R}^2$ is provided for the base, denoted as a sequence of M waypoints, $([x, y]_m)_{m=0}^{M}$. The goal is to generate feasible control commands for the whole mobile manipulator enabling it to track the provided path while avoiding collisions in 3D space with dynamic and static obstacles. Hence, we formulate the trajectory planning problem for the unified system, base plus arm, as an optimization problem. As a result, we can explicitly formulate collision avoidance and kinodynamic constraints and compute control commands to generate feasible and collision-free motions. The optimization problem is formulated as

$$J^\star = \min_{\boldsymbol{z}_{0:N}, \boldsymbol{u}_{0:N}} \sum_{k=0}^{N} J(\boldsymbol{z}_k, \boldsymbol{u}_k), \tag{4a}$$

$$s.t. \quad \boldsymbol{z}_{k+1} = f(\boldsymbol{z}_k, \boldsymbol{u}_k) \quad \forall k < N, \tag{4b}$$

$$\mathcal{B}(\boldsymbol{z}_k) \cap \left(\mathcal{O}^{\text{static}} \cup \mathcal{O}^{\text{dynamic}}\right) = \emptyset, \tag{4c}$$

$$\boldsymbol{u}_k \in \mathcal{U}, \boldsymbol{z}_k \in \mathcal{Z}, \tag{4d}$$

$$\boldsymbol{z}_0 = \boldsymbol{z}(0). \tag{4e}$$

In this formulation, Eq. 4a is the cost function (Section III-E), Eq. 4b represents the kinodynamic constraints of the system (Section III-C), Eq. 4c formalizes collision avoidance with static (Section III-F) and dynamic obstacles (Section III-G), and Eq. 4d defines the set of admissible states and control inputs (Section III-C). Finally, Eq. 4e defines the initial state conditions. Note that we optimize over a prediction horizon N allowing to avoid dynamic obstacles in advance.

E. Cost Function

To track the reference path, we first create a continuous path representation by approximating the provided reference path using a cubic *Bézier Curve* and using a normalized time parametrization, $\phi_k \in [0, 1]$. We denote $\boldsymbol{p}_k$ and θ_k as the predicted base position and orientation at time step k and $\boldsymbol{p}_k^r(\phi_k) = [x, y]_k^r$ and $\theta_k^r(\phi_k)$ as the reference base position and orientation at future time-step k, respectively. Then, we define a tracking error vector $\boldsymbol{e}_k := [e^c(\boldsymbol{z}_k, \phi_k), e^l(\boldsymbol{z}_k, \phi_k)]^{\text{T}}$ composed by the contour $e^c(\boldsymbol{z}_k, \phi_k)$ and a lag-error $e^l(\boldsymbol{z}_k, \phi_k)$, and computed as it follows

$$\boldsymbol{e}_k = \begin{bmatrix} cos(\theta_k^r) & sin(\theta_k^r) \\ -sin(\theta_k^r) & cos(\theta_k^r) \end{bmatrix} (\boldsymbol{p}_k^{\text{r}} - \boldsymbol{p}_k). \tag{5}$$

The cost function $J(\boldsymbol{z}_k, \boldsymbol{u}_k)$ is composed of the weighted ($W_{\mathbf{e}}$) quadratic tracking error, the weighted ($W_{\mathbf{q}}$) arm configuration distance-to-goal and the weighted ($W_{\mathbf{u}}$) quadratic inputs to penalize high control commands . The difference between the current and desired orientation is quadratically weighted to ensure that the robot is moving forward (w_θ). In addition, to relax the problem, we introduce the slack variable s and penalize its weighted norm

$$\begin{aligned} J(\boldsymbol{z}_k, \boldsymbol{u}_k) = \|\mathbf{e}_k\|_{W_{\mathbf{e}}} + \|\boldsymbol{u}_k\|_{W_{\mathbf{u}}} + \|(\theta_k^r - \theta_k)\|_{w_\theta} \\ + \|\boldsymbol{q}_k - \boldsymbol{q}_{\text{des,k}}\|_{W_q} + \|s_k\|_{w_{\text{slack}}}. \end{aligned} \tag{6}$$

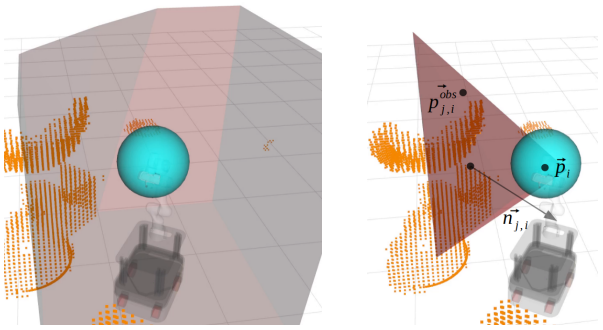
F. Collision Avoidance For Static Obstacles

In this paper, we tackle the problem of avoiding static obstacles in 3D space. Hence, we employ an octree representation of the static obstacles fed directly from 3D sensor data (e.g., depth camera). Given this information, we propose to model the free space as a set of convex polyhedrons around the robot's links instead of explicitly describing individual obstacles. This representation allows us to limit the number of collision constraints regardless of the number of obstacles, and depending only on the number of robot's links and the number of planes used for the convex regions. To compute the convex regions, we employ the method proposed in [5] using an ellipsoid based regional inflation. Fig. 2 depicts an example of one of these convex regions computed for one robot's link. For each $i \in [1, n_{\text{links}}]$, we compute a polyhedron with n_{planes} planes representing the free-space around the i-th link.

Then, we impose a linear inequality constraint between each j-th polyhedron plane and i-th link to ensure $\mathcal{B}(\boldsymbol{z}_k) \cap \mathcal{O}^{\text{static}} = \emptyset$ as

$$\begin{aligned} \mathbf{a}^T \mathbf{p}_i \leq b - (r_i + d_{\text{safety}}) \\ \forall i \in \{1, \dots, n_{\text{links}}\}, \quad \forall j \in \{1, \dots, n_{\text{planes}}\}, \end{aligned} \tag{7}$$

with $\mathbf{a} = \mathbf{n}_{j,i}$ and $b = -\mathbf{n}_{j,i}^T \mathbf{p}_{j,i}^{\text{obs}}$, where $\mathbf{n}_{j,i}$ is the normal vector and $\mathbf{p}_{j,i}^{\text{obs}}$ a point on the j-th polyhedron's plane



(a) Polyhedron (b) Single Inequality

Fig. 2: Generated polyhedron representing the free space in the presence of sensed pointcloud (orange) for last link and the corresponding sphere on the robot (blue) and visualization sphere-plane inequality constraint (right).

enclosing the i-th robot link, $\mathbf{p}_i$ is the i-th link position, and d_{safety} an hyper-parameter that acts as a safety margin. The proposed collision constraint ensures that each link's space is inside the convex region and thus free of static obstacles, as depicted in Fig.2.

G. Collision Avoidance For Moving Obstacles

To avoid moving obstacles, it is necessary to propagate the states of the dynamic obstacles over the planning horizon N. Using the previous constraint for dynamic collision avoidance requires the propagation of the 3D octree and the computation of the convex polyhedra for every stage, which is highly computationally expensive.

Hence, we propose to model dynamic obstacles as spheres. For each dynamic obstacle d we assume to know the position $\boldsymbol{p}_d^{\text{dyn}}$, velocity $\mathbf{v}_d^{\text{dyn}}$, and radius r_d^{dyn}, with $d = \{1, \dots, n_{\text{dyn}}\}$. Then, we employ a constant velocity model to obtain predictions on the dynamic obstacle's future positions, $\bar{\boldsymbol{p}}_{d,k}^{\text{dyn}} = \boldsymbol{p}_d^{\text{dyn}} + k\Delta t \boldsymbol{v}_d^{\text{dyn}}$. Finally, we define a non-linear collision avoidance constraint ensuring that the obstacle's space does not intersect with any link's space, $\mathcal{B}_i(\boldsymbol{z}_k) \cap \mathcal{O}^{\text{dynamic}} = \emptyset \; \forall i \in [1, n_{\text{links}}]$, imposing that the distance between both bounding spaces is larger than the sum of their radius and the previously introduced safety margin

$$\begin{aligned} &\left\| \bar{\boldsymbol{p}}_{d,k}^{\text{dyn}} - \mathbf{p}_{i,k} \right\| \geq r_d^{\text{dyn}} + r_i + d_{\text{safety}} \\ &\forall i \in \{1, \dots, n_{\text{links}}\},\ d \in \{1, ..., n_{dyn}\},\ k \in 0, ..., N, \end{aligned} \tag{8}$$

where $\mathbf{p}_i$ is the position of the i-th robot link and r_d^{dyn} the d-th dynamic obstacle radius.

IV. Experimental Results

The presented method is evaluated in simulation and with the real hardware. After a short introduction to the experimental setup, three simulation scenarios and one real-world scenario are introduced. We compare the presented method with a sequenced MPC formulation in which arm motion and locomotion are sequenced and a conventional MPC formulation in which inequalities are formulated for individual obstacles. Scenarios are considered infeasible when following the trajectory without global replanning is not possible without violating the collision constraints.

TABLE I: Positions of spheres for the volumetric representation.

Parent Link	Offset	Radius
base link	[0, 0, 0.25]	0.25
base link	[0.3, 0, 0.25]	0.25
link 2	[0, -0.1896, 0]	0.2275
link 7	[0, 0, 0]	0.3

A. Experimental Setup

a) Hardware Setup: The mobile manipulator used to validate this approach consists of the mobile base *Clearpath™ Boxer* and the robotic manipulator *Franka Emika Panda*, see Fig. 1. The resulting system has 10 degrees of freedom (DoF's) for which the dynamics are approximated using a Runge-Kutta scheme to obtain the discrete transition function $f(\boldsymbol{z}_t, \boldsymbol{u}_t)$. The presented robot is equipped with low level controllers that accept commanded velocities for all joints, one LiDAR sensor and one depth camera pointing forward. Laser data and camera depth images are fused into one pointcloud using the octomap framework [30]. Polyhedrons to describe the convex space around the links are computed using the *DecompUtil* presented in [5], into which the pointcloud generated from the vertex centers of the occupied cells of the octomap are fed. Joint positions for the arm are known at every time step using the encoders and the pose of the base is estimated using SLAM. The implementation is realized in the *Robotics Operating System* (ROS) framework, as it allows simple integration of the different components. The underlying MPC problem is solved using *FORCES-Pro* solver [31] and employing an interior-point method [32]. We used a laptop with an Intel Core i7 and 32GB of RAM to run the simulations and an Intel NUC with an Intel Core i7 and 8GB to run the real-world experiments.

b) Parameter Details: The presented robot has nine links which are represented by four spheres for collision avoidance. The positions of the spheres on the kinematic chain are explicitly given in Table I. Note that the robot is not fully contained in the union of spheres. For $d_{\text{safety}} = 0$, this could potentially result in collision. However, much larger spheres would result in the inability to move close to obstacles when manipulation is requested. The parametrization with d_{safety} allows to flexibly change between different motion types, e.g. manipulation (low safety margin) and navigation (large safety margin). The centers of the given spheres are also used as seed points for the convex region generation. Two different step size were used over the time horizon, $\Delta t_1, \Delta t_2$. All remaining parameter settings are summarized in Table II.

B. Manipulation Scenarios

We compare our proposed method against two baseline methods: a decoupled MPC baseline (i.e., locomotion and arm motion are performed sequentially) and a coupled MPC

TABLE II: Parameter Settings

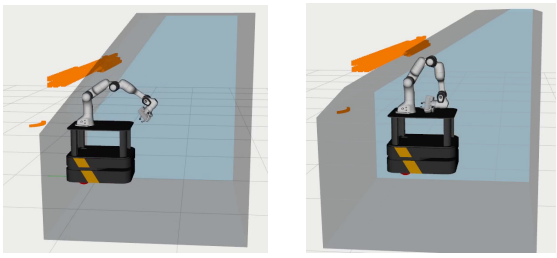
Parameter	Static Scenarios	Dynamic Scenario
Prediction Horizon	11 sec	$11s$ sec
$\Delta t_1/\Delta t_2$	0.2/1.0 sec	0.2/1.0 sec
#Planes/Link	15	15
d_{safety}	0.15 m	0.25 m
$W_{\mathbf{e}}$	$5.0\boldsymbol{I}_2$	$5.0\boldsymbol{I}_2$
w_θ	2.0	2.0
$W_{\mathbf{q}}$	$0.7\boldsymbol{I}_7$	$0.7\boldsymbol{I}_7$
$W_{\mathbf{u}}$	$\begin{bmatrix} 0.05\boldsymbol{I}_2 & 0 \\ 0 & 5\boldsymbol{I}_7 \end{bmatrix}$	$\begin{bmatrix} 0.05\boldsymbol{I}_2 & 0 \\ 0 & 5\boldsymbol{I}_7 \end{bmatrix}$
w_{slack}	10^5	10^5

formulation based on sphere-sphere inequality constraints, as proposed in [9]. Moreover, as this work presents a method for local trajectory optimization, the effect of global replanning during the execution is not considered. To access our method's performance, we present simulation results for:

- Static collision avoidance with a horizontal bar [9] (Sub-section IV-B.0.a);
- 2D trajectory tracking while avoiding collisions with randomly placed static obstacles (Sub-section IV-B.0.b);
- Dynamic collision avoidance with a moving obstacle (Sub-section IV-B.0.c).

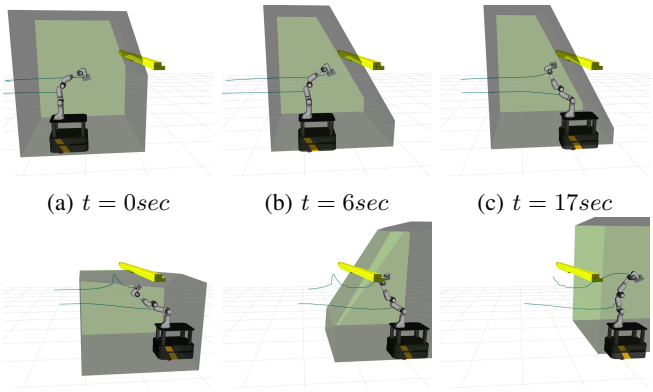
Finally, we present experimental results on a mobile manipulator performing a real manipulation task (Sub-section IV-B.0.d).

a) Horizontal Bar: A horizontal bar is placed between the start and goal configuration. In contrast to the work in [9], no object detection is required in our approach, as the sensed point cloud is fed directly into the convex region generator. The base's global path consists of a simple straight-line motion to a pose behind the bar. The motion of the manipulator and the generated convex regions for the last link of the kinematic chain are depicted in Fig. 4. The advantage of whole-body optimization can be extracted when the horizontal bar is placed at a lower z-positions. In Fig. 3, such a situation is visualized for $z = 1.3$. In the visualized case, it is not possible to move underneath the bar with the sequenced approach. On the other hand, the coupled method can navigate safely avoiding collision by moving the arm into an extended position in which the absolute height is smaller than when having it folded.



(a) coupled (b) decoupled

Fig. 3: Advantage of coupled MPC when moving underneath a horizontal bar (orange).



(a) $t = 0sec$ (b) $t = 6sec$ (c) $t = 17sec$

(d) $t = 25sec$ (e) $t = 32sec$ (f) $t = 42sec$

Fig. 4: Avoiding an horizontal bar. The convex region for the last link of the kinematic chain.

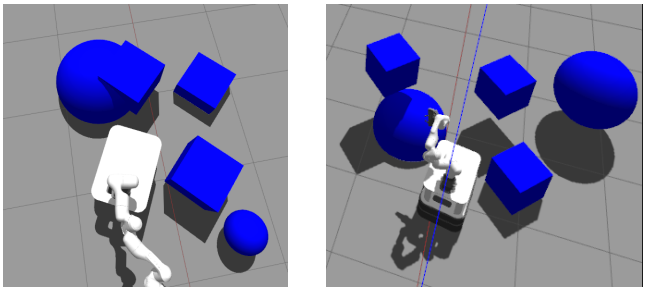


Fig. 5: Example for infeasible cases, that were excluded from the test set in randomized scenario.

b) Randomized Obstacles: In this scenario, the robot is placed in an unstructured environment with several static obstacles. A global path for the base to reach the goal is computed, but the path is blocked with randomly generated obstacles, uniformly distributed on the intervals $x \in [2m, 5m], y \in [-2m, 2m], z \in [0m, 2m]$. Only those obstacles visible to the LiDAR sensors are available for the global planner which generates a path in the 2D plane for the base's motion. Among the randomly generated cases, only those that are feasible for an MPC trajectory optimizer are considered. Two examples for infeasible cases are depicted in Fig. 5.

A successful trajectory of the coupled MPC planner is depicted in Fig. 6. A key advantage of our method is that when the environment is densely populated with obstacles, the solving times are not affected when using convex regions to represent the free space, see Fig. 7. Explicitly formulating sphere-sphere inequality constraints results in an increase

	decoupled	coupled
mean	219.65s	113.822s
std. deviation	24.21s	8.35s
min	199.27s	106.83s
max	270.25s	131.24s

TABLE III: Compared execution times coupled and decoupled approach for cases that were feasible for both methods.

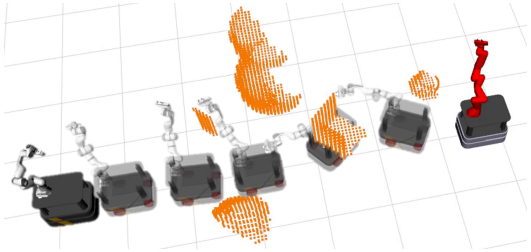
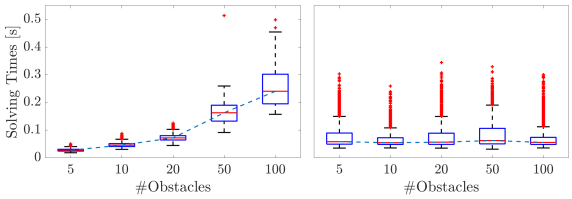


Fig. 6: Five obstacles avoidance, final configuration in red, occupied voxels in the octomap are represented in orange.



(a) Obstacles as spheres (b) Convex regions

Fig. 7: Comparison solver performance for an increasing number of obstacles.

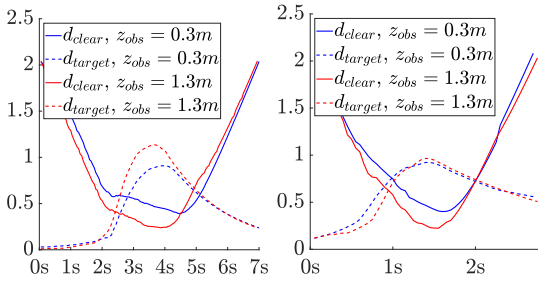
of solving time as the environment becomes more densely populated with obstacles. Note that convex region generation becomes only beneficial as the number of obstacles exceeds a critical value, in this case, for 50 obstacles. Furthermore, parallelizing the locomotion and arm motion allows to reduce the mean overall operational time by 48%, see Table III.

c) Dynamic Obstacle: Here, we evaluate dynamic collision avoidance with a single moving obstacle for different obstacle's velocities. As dynamic object detection and velocity estimation are out of the scope of this work, the state of the obstacle is assumed to be known to the robot during the entire process. In Fig. 9, the experiment is visualized. The goal pose (light grey) is to be reached but a moving sphere, conflicting the goal, must be avoided at all time. The proposed MPC formulation's reactivity is investigated using the clearing distance $d_{\text{clear}} = \min_i \left\|\mathbf{p}^{\text{dyn}} - \mathbf{p}_i\right\| - r_i - r_d^{\text{dyn}}$ and the distance to target, $d_{\text{target}} = \|\boldsymbol{z}_{\text{des}} - \boldsymbol{z}\|$, where $\boldsymbol{z}_{\text{des}}$ is composed of the desired base and arm configuration. Different velocities ($\boldsymbol{v}_d^{\text{dyn}}$) and different heights (z_{obs}) of the moving obstacles were investigated. Fig. 8 shows that our approach successfully avoids collision with dynamic obstacles when moving towards the goal.

d) Real-World Experiment: We evaluated the presented method in real-world scenarios in a simple pick & place setup. Fig. 1 depicts the experimental scenario, where the robot picks up an object on the left (pose in light green) and moves to the target pose (pose in light red) without colliding with the obstacle visualized in light blue. Intermediate poses of the successful trajectory are visualized in Fig. 10. A video of the experiment is attached to this work.

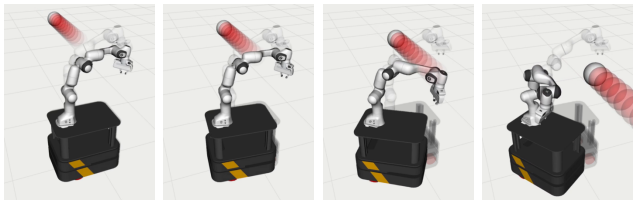
V. Conclusion

This work proposes a whole-body trajectory optimization to navigate in unstructured and simplified dynamic environment safely. Evolution of the environment, i.e., dynamic obstacles, were incorporated, and static collision avoidance is realized by a union of convex regions describing the free space. By representing the free space, rather than individual obstacles, the number of inequality constraints is limited. The scaling of the method for an increasing number of obstacles and its ability to avoid collision with moving obstacles were shown in randomized scenarios. Real-time applicability was demonstrated on a 10-DoF's mobile manipulator in a Pick & Place test case. The proposed approach overcomes the limitations of previous works and allows whole-body trajectory optimization in dynamic environments.



(a) $\|\boldsymbol{v}_d\| = 0.2m/s$ (b) $\|\boldsymbol{v}_d\| = 0.5m/s$

Fig. 8: Clearance from moving obstacle and distance to target position for different obstacle velocities.



(a) $t = 0sec$ (b) $t = 1sec$ (c) $t = 2sec$ (d) $t = 3sec$

Fig. 9: Motion avoiding moving obstacle (red) while attempting to reach target (light grey).

Fig. 10: Trajectory in mock-up store avoiding obstacles, goal configuration in light grey.

REFERENCES

[1] S. Thakar, L. Fang, B. Shah, and S. Gupta, "Towards Time-Optimal Trajectory Planning for Pick-and-Transport Operation with a Mobile Manipulator," in *IEEE International Conference on Automation Science and Engineering*, vol. 2018-Augus. IEEE Computer Society, dec 2018, pp. 981–987.

[2] S. Thakar, P. Rajendran, V. Annem, A. Kabir, and S. Gupta, "Accounting for part pose estimation uncertainties during trajectory generation for part pick-up using mobile manipulators," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May. Institute of Electrical and Electronics Engineers Inc., may 2019, pp. 1329–1336.

[3] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 2010, pp. 797–803.

[4] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May. Institute of Electrical and Electronics Engineers Inc., may 2019, pp. 6015–6022.

[5] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-D complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, jul 2017.

[6] J. Tordesillas, B. T. Lopez, and J. P. How, "FASTER: Fast and Safe Trajectory Planner for Flights in Unknown Environments," in *IEEE International Conference on Intelligent Robots and Systems*, 2019, pp. 1934–1940. [Online]. Available: https://github.com/jtorde

[7] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model Predictive Contouring Control for Collision Avoidance in Unstructured Dynamic Environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4459–4466, 2019.

[8] G. B. Avanzini, A. M. Zanchettin, and P. Rocco, "Constraint-based Model Predictive Control for holonomic mobile manipulators," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem. Institute of Electrical and Electronics Engineers Inc., dec 2015, pp. 1473–1479.

[9] G. Buizza Avanzini, A. M. Zanchettin, and P. Rocco, "Constrained model predictive control for mobile robotic manipulators," pp. 19–38, 2018. [Online]. Available: https://doi.org/10.1017/S0263574717000133

[10] S. M. LaValle, "Planning algorithms," *Planning Algorithms*, vol. 9780521862, pp. 1–826, 2006.

[11] M. Mukadam, J. Dong, X. Yan, F. Dellaert, and B. Boots, "Continuous-time Gaussian process motion planning via probabilistic inference," *International Journal of Robotics Research*, vol. 37, no. 11, pp. 1319–1340, jul 2018. [Online]. Available: http://arxiv.org/abs/1707.07383 http://dx.doi.org/10.1177/0278364918790369

[12] D. J. Webb and J. Van Den Berg, "Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2013, pp. 5054–5061.

[13] M. Kleinbort, K. Solovey, Z. Littlefield, K. E. Bekris, and D. Halperin, "Probabilistic Completeness of RRT for Geometric and Kinodynamic Planning with Forward Propagation," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 277–283, apr 2019.

[14] J. J. Kuffner and S. M. La Valle, "RRT-connect: an efficient approach to single-query path planning," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2, 2000, pp. 995–1001.

[15] D. Hsu, R. Kindel, J. C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *International Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, mar 2002.

[16] A. Faust, K. Oslund, O. Ramirez, A. Francis, L. Tapia, M. Fiser, and J. Davidson, "PRM-RL: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning," in *Proceedings - IEEE International Conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers Inc., oct 2018, pp. 5113–5120. [Online]. Available: http://arxiv.org/abs/1710.03937

[17] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings - IEEE International Conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers Inc., 1985, pp. 500–505.

[18] S. Haddadin, R. Belder, and A. Albu-Schäffer, "Dynamic motion planning for robots in partially unknown environments," in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 44, no. 1 PART 1. IFAC Secretariat, 2011, pp. 6842–6850.

[19] S. Du, W. Shang, S. Cong, C. Zhang, and K. Liu, "Moving obstacle avoidance of a 5-DOF robot manipulator by using repulsive vector," in *2017 IEEE International Conference on Robotics and Biomimetics, ROBIO 2017*, vol. 2018-Janua. Institute of Electrical and Electronics Engineers Inc., mar 2018, pp. 1–6.

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

[21] F. Zhang, N. Li, T. Xue, Y. Zhu, R. Yuan, and Y. Fu, "An Improved Dynamic Window Approach Integrated Global Path Planning," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, dec 2019, pp. 2873–2878. [Online]. Available: https://ieeexplore.ieee.org/document/8961684/

[22] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 1688–1694.

[23] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Springer Tracts in Advanced Robotics*, vol. 70, no. STAR, 2011, pp. 3–19.

[24] J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, "Optimal Reciprocal Collision Avoidance for Multiple Non-Holonomic Robots," in *Springer Tracts in Advanced Robotics*, vol. 83 STAR, 2012, pp. 203–216. [Online]. Available: http://link.springer.com/10.1007/978-3-642-32723-0%7B%5C_%7D15

[25] W. Schwarting, J. Alonso-Mora, L. Paull, S. Karaman, and D. Rus, "Safe Nonlinear Trajectory Generation for Parallel Autonomy with a Dynamic Vehicle Model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2994–3008, sep 2018.

[26] A. Jain and C. C. Kemp, "Behavior-Based Door Opening with Equilibrium Point Control," *RSS Workshop Mobile Manipulation in Human Environments*, 2009.

[27] S. Chitta, B. Cohen, and M. Likhachev, "Planning for autonomous door opening with a mobile manipulator," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2010, pp. 1799–1806.

[28] C. Li, F. Xia, R. Martin-Martin, S. Savarese, R. Martín-Martín, and S. Savarese, "HRL4IN: Hierarchical Reinforcement Learning for Interactive Navigation with Mobile Manipulators," oct 2019. [Online]. Available: https://sites.google.com/view/hrl4in. http://arxiv.org/abs/1910.11432

[29] S. Ide, T. Takubo, K. Ohara, Y. Mae, and T. Arai, "Real-time trajectory planning for mobile manipulator using model predictive control with constraints," in *URAI 2011 - 2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence*, 2011, pp. 244–249.

[30] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, apr 2013.

[31] A. Domahidi and J. Jerez, "Forces professional," Embotech AG, url=https://embotech.com/FORCES-Pro, 2014–2019.

[32] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari, "Forces nlp: an efficient implementation of interior-point... methods for multistage nonlinear nonconvex programs," *International Journal of Control*, pp. 1–17, 2017.