2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
June 28-30, 2023. Seattle, Washington, USA

Cooperative Time-Optimal Trajectory Generation for a Heterogeneous Group of Redundant Mobile Manipulators*

Alice Hierholz[1], Andreas Gienger[1] and Oliver Sawodny[1]

Abstract— **In order to increase productivity and resource efficiency in the construction sector, new approaches are being pursued to automate tasks in the interior fitting of existing buildings. For this purpose, a heterogeneous group of redundant mobile manipulators is used for the time-consuming exact positioning of workpieces. The focus lies hereby on cooperative time-optimal trajectory generation for mobile manipulators. To minimize the effort for modeling and formulating the optimal control problem for the different robots, a unified approach based on the Unified Robotics Description Format is developed. The optimal control problem considers the equations of motion of the robots and existing kinematic and dynamic constraints as well as collisions. To enable cooperative trajectories, additional kinematic constraints are imposed on the poses of the end effectors of each mobile manipulator. Different numerical methods, degrees of model abstraction and cost-functionals are investigated with respect to the generated trajectories and the required computation time towards real-time capability for future adaptive trajectory generation using sensor feedback.**

I. INTRODUCTION

Productivity and efficiency improvements in the construction industry have stagnated for decades. The study in [1] shows, that the productivity of the global economy has increased by an average of 2.8% per year over the last 20 years. By contrast, the construction industry has only been able to increase its productivity by 1.0%. Therefore, the automation of time-consuming tasks, like the exact positioning of workpieces, e. g. studs and panels of a stud wall, is investigated. To meet the diverse requirements on the construction site, a group of heterogeneous mobile manipulators as in Fig. 1 is considered, for which cooperative time-optimal trajectories considering constraints and collisions are generated.

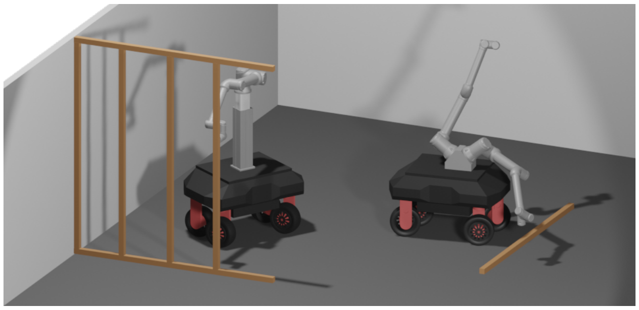


Fig. 1: Two heterogeneous mobile manipulators erecting a stud wall.

In the literature, different approaches for trajectory generation in different fields of applications have been considered. In [2], the cooperative collision-free trajectory generation of a dual arm robot is investigated. The kinematic chain between the two arms is considered, which is generated when they grasp the same object. The overall problem is solved via an optimal control problem (OCP), which minimizes the joint velocities. In [3], the same issue but with two robots is considered. Two possible solutions are presented: First, a leader-follower approach and second, an approach in which the two cooperating robots are considered as a coherent system. Cooperative movement of an object by a number of identical manipulators is considered in [4]. The focus lies on the development of static and dynamic load distribution approaches to determine the optimal gripping points of the different manipulators. In [5], trajectory generation for cooperative mobile manipulators is studied by solving inverse kinematics using a global optimization approach, where the joint position error is minimized. However, only kinematic constraints and no dynamic constraints are considered. Trajectory generation for high-precision positioning of components by multiple manipulators is considered in [6]. The construction of a masonry vault structure, by manipulators working independently, is studied. Thus, no direct cooperation takes place. Trajectory generation for a larger group of manipulators is investigated in [7]. Here, one robot holds the workpiece as long as the other manipulators are machining it. Kinematic chains are not considered. The task is solved by an initial trajectory generation in the workspace, followed by an underlying generation of the joint trajectories for the individual robots. In [8], two manipulators operating independently in a confined workspace are examined. In particular, the generation of collision-free time-optimal trajectories using model predictive control is considered.

The mentioned publications deal with homogeneous groups of robots. The focus remains on the trajectory generation considering collisions. Time-optimality is only taking into account once and the cooperation between robots, and the resulting kinematic chains, only in some publications. However, studies on heterogeneous groups of robots could not be found. For this reason, the cooperative time-optimal trajectory generation of different mobile manipulators, considering constraints and collisions, is addressed in this work.

The main contribution of this work consists of two aspects: First, in order to minimize the effort for modeling and formu-

[1] The authors are with the Institute for System Dynamics, University of Stuttgart, 70563 Stuttgart, Germany {alice.hierholz, andreas.gienger, sawodny} @isys.uni-stuttgart.de

*(Partially) Supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2120/1 - 390831618

978-1-6654-7633-1/23/$31.00 ©2023 IEEE

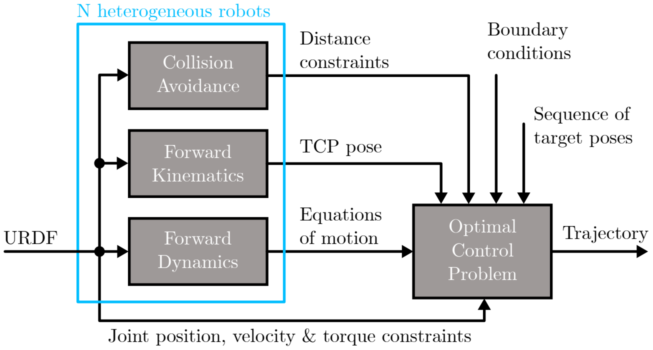


Fig. 2: Standardized approach for modeling heterogeneous robots and incorporating them into an OCP.

lating the OCP for the different robots, a unified approach is developed which automates the task. This is shown schematically in Fig. 2. As a standardized description for each robot the *Unified Robotics Description Format* (URDF) is used. In order to compute the equations of motion, the current tool center point (TCP) pose and the distance constraints for the collision avoidance, recursive algorithms independent of the robot, exploiting the structure of the kinematic tree used in the URDF, are applied. Then, they are embedded into the OCP along with the boundary conditions, the given sequence of target poses, and the joint postion, velocity, and torque constraints extracted directly from the robot description. The trajectory then results from the numerical solution of the OCP. This standardized approach is applicable to any heterogeneous robot group whose individual robots can be described via the URDF. Limitations comprise parallel robots and contact situations, but the approach can be extended to those cases in future works. The second main contribution of this work is the investigation of the required computation times for generating the cooperative time-optimal trajectories for the redundant robots as well as the optimal transition time and the generated trajectories depending on the used numerical method, degree of model abstraction and cost-functional.

Below, the modeling of the mobile manipulators is illustrated in Section II and followed by a system description in Section III. Then the time-optimal trajectory generation is depicted in Section IV and its results are shown in Section V. The paper is closed by a summary and outlook in Section VI.

II. MODELING

In order to generate time-optimal trajectories, kinematic and dynamic modeling of the considered mobile manipulators is necessary. When choosing the algorithms to be used, it is important to ensure that they are independent of the robots, to reduce the additional overhead caused by the heterogeneous group of mobile manipulators.

A. Kinematics

The forward kinematics deals with the computation of the pose, i.e. position and orientation, of the TCP defined as homogeneous transformation matrix

$$\boldsymbol{T}_{\mathrm{TCP}} = \mathrm{FK}\left(\mathcal{M}, \boldsymbol{q}\right) \tag{1}$$

from given joint positions $\boldsymbol{q}$ and multi-body model $\mathcal{M}$ of the rigid robot, which is obtained from the description as a kinematic tree in the URDF. There are several ways to calculate the forward kinematics. One method is based on the Denavit-Hartenberg convention. An alternative is the screw-based method, which is based on the spatial vector notation and uses the product-of-exponentials (PoE) formula in combination with the exponential mapping of screw axis and Rodrigues' formula [9]. In this work, the screw-based method is chosen, due to its flexibility when considering heterogeneous groups of redundant mobile manipulators [10]. Furthermore, the screw-based method is used in the recursive dynamics algorithms described in the next section.

B. Dynamics

The rigid-body dynamics of mobile manipulators can be derived by different approaches. The Lagrangian approach leads to a closed form solution, whereas the Newton-Euler method yields in differentiaton-free recursive algorithms. The resulting equations of motion are identical, but the recursive algorithms are computationally more efficient, especially for robots with a high number of degrees of freedom (DOF) as used in this work [9].

The calculation of the inverse dynamics

$$\boldsymbol{\tau} = \mathrm{ID}\left(\mathcal{M}, \boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}\right) \tag{2}$$

deals with the task of calculating the joint torques $\boldsymbol{\tau}$ from the joint positions $\boldsymbol{q}$ and its derivatives $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$. It can be solved by the *Recursive Newton Euler Algorithm* (RNEA), which is based on the Twist-Wrench-Formulation of the dynamic equations of a single rigid body.

The calculation of the forward dynamics

$$\ddot{\boldsymbol{q}} = \mathrm{FD}\left(\mathcal{M}, \boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\tau}\right) \tag{3}$$

addresses the task of calculating the joint accelerations $\ddot{\boldsymbol{q}}$ from the joint positions $\boldsymbol{q}$ and velocities $\dot{\boldsymbol{q}}$ as well as the joint torques $\boldsymbol{\tau}$. Different recursive algorithms exist to solve this problem, but the computationally most efficient one is the *Articulated Body Algorithm* (ABA), which in consequence is used in this work [11].

III. SYSTEM DESCRIPTION

In this section, two state-space representations of the used mobile manipulators are derived with different degrees of model abstraction. Moreover, the geometry of the mobile manipulators is approximated using different spheres in order to use this approximation for collision avoidance.

A. State-space representations

In this work, a heterogeneous group consisting of two mobile manipulators, illustrated in Fig. 1, is considered. Both robots consist of a mobile platform and manipulators attached to it. The first mobile manipulator has one arm, which is mounted on the mobile platform via a telescopic axis and thus has 10 DOF. The second mobile manipulator is a dual arm robot with 15 DOF, where the two arms are mounted side by side on the mobile platform at an $45\,^{\circ}$ angle. These two robots are suitable for use in interior construction due to their compact size, mobility and flexibility thanks to the three manipulators. The four separately controllable omnidirectional wheels of the mobile platform allow the mobile manipulator to change its position and orientation in the plane at any time. Hence, the movements of the mobile platform can be represented by two translational joints and one rotational joint. This reduces the number of control variables for the mobile platform from eight to three and facilitates the trajectory generation. The mobile platform is therefore described by an integrator chain. The states

$$\boldsymbol{x}_{\mathrm{pf}} = \begin{bmatrix} \boldsymbol{q}_{\mathrm{pf}} \\ \dot{\boldsymbol{q}}_{\mathrm{pf}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_{\boldsymbol{q}_{\mathrm{pf}}} \\ \boldsymbol{x}_{\dot{\boldsymbol{q}}_{\mathrm{pf}}} \end{bmatrix} \tag{4}$$

thus consist of the joint positions $\boldsymbol{q}_{\mathrm{pf}}$ and their velocities $\dot{\boldsymbol{q}}_{\mathrm{pf}}$. The acceleration $\ddot{\boldsymbol{q}}_{\mathrm{pf}}$ is selected as the control variable $\boldsymbol{u}_{\mathrm{pf}}$. This results in the system equations

$$\dot{\boldsymbol{x}}_{\mathrm{pf}} = \begin{bmatrix} \boldsymbol{x}_{\dot{\boldsymbol{q}}_{\mathrm{pf}}} \\ \boldsymbol{u}_{\mathrm{pf}} \end{bmatrix}, \quad \boldsymbol{x}_{\mathrm{pf}}(0) = \boldsymbol{x}_{\mathrm{pf},0}. \tag{5}$$

The manipulators are modeled using the methods described in Section II. The states are

$$\boldsymbol{x}_{\mathrm{arm}} = \begin{bmatrix} \boldsymbol{q}_{\mathrm{arm}} \\ \dot{\boldsymbol{q}}_{\mathrm{arm}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_{\boldsymbol{q}_{\mathrm{arm}}} \\ \boldsymbol{x}_{\dot{\boldsymbol{q}}_{\mathrm{arm}}} \end{bmatrix}. \tag{6}$$

The joint torques $\boldsymbol{\tau}_{\mathrm{arm}}$ are selected as the control variable $\boldsymbol{u}_{\mathrm{arm}}$. This yields the system equations

$$\dot{\boldsymbol{x}}_{\mathrm{arm}} = \begin{bmatrix} \boldsymbol{x}_{\dot{\boldsymbol{q}}_{\mathrm{arm}}} \\ \mathrm{FD}\left(\mathcal{M}, \boldsymbol{x}_{\boldsymbol{q}_{\mathrm{arm}}}, \boldsymbol{x}_{\dot{\boldsymbol{q}}_{\mathrm{arm}}}, \boldsymbol{u}_{\mathrm{arm}}\right) \end{bmatrix} \tag{7}$$

with the initial condition $\boldsymbol{x}_{\mathrm{arm}}(0) = \boldsymbol{x}_{\mathrm{arm},0}$. For a complete mobile manipulator, with the states and control variables

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_{\boldsymbol{q}_{\mathrm{pf}}} \\ \boldsymbol{x}_{\boldsymbol{q}_{\mathrm{arm}}} \\ \boldsymbol{x}_{\dot{\boldsymbol{q}}_{\mathrm{pf}}} \\ \boldsymbol{x}_{\dot{\boldsymbol{q}}_{\mathrm{arm}}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_{\boldsymbol{q}} \\ \boldsymbol{x}_{\dot{\boldsymbol{q}}} \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_{\mathrm{pf}} \\ \boldsymbol{u}_{\mathrm{arm}} \end{bmatrix} \tag{8}$$

the state-space representation is given by

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x}_{\dot{\boldsymbol{q}}} \\ \boldsymbol{u}_{\mathrm{pf}} \\ \mathrm{FD}\left(\mathcal{M}, \boldsymbol{x}_{\boldsymbol{q}_{\mathrm{arm}}}, \boldsymbol{x}_{\dot{\boldsymbol{q}}_{\mathrm{arm}}}, \boldsymbol{u}_{\mathrm{arm}}\right) \end{bmatrix}, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0. \tag{9}$$

It can be simplified to an integrator chain by neglecting the nonlinear couplings. The simplified control variable $\tilde{\boldsymbol{u}}_{\mathrm{arm}}$ then corresponds to the joint accelerations $\ddot{\boldsymbol{q}}_{\mathrm{arm}}$, similar to the mobile platform in (5). The alternative state-space representation with $\tilde{\boldsymbol{x}} = \boldsymbol{x}$ and $\tilde{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{u}_{\mathrm{pf}}^T, \tilde{\boldsymbol{u}}_{\mathrm{arm}}^T \end{bmatrix}^T$ results in

$$\dot{\tilde{\boldsymbol{x}}} = \begin{bmatrix} \boldsymbol{x}_{\dot{\boldsymbol{q}}} \\ \tilde{\boldsymbol{u}} \end{bmatrix}, \quad \tilde{\boldsymbol{x}}(0) = \tilde{\boldsymbol{x}}_0. \tag{10}$$

Both degrees of model abstraction are considered in the following sections for further inspections.

B. Approximation of the geometry for collision avoidance

In order to generate reasonable trajectories for cooperating mobile manipulators, it is essential to implement collision avoidance. This includes the prevention of self-collisions, e. g. of the manipulator arm with its platform, and collisions between the mobile manipulator and its environment or other mobile manipulators. To implement the collision avoidance, an approach based on [12] is chosen in which the geometry of the mobile manipulators is approximated by suitably selected spheres. Each of the S spheres has a fixed radius r_i, $i = 1, ..., S$ and a center $\boldsymbol{c}_i(\boldsymbol{x}_{\boldsymbol{q}})$ which can be determined by solving the forward kinematics. For each set of potentionally colliding spheres it must hold that the distance between the centers of those spheres has to be greater or equal the sum of the radii, which can be written as

$$||\boldsymbol{c}_i(\boldsymbol{x}_{\boldsymbol{q}}) - \boldsymbol{c}_j(\boldsymbol{x}_{\boldsymbol{q}})||_2 \geq r_i + r_j. \tag{11}$$

The collection of all distance constraints is rewritten as $\mathrm{col}\left(\mathcal{M}, \boldsymbol{x}_{\boldsymbol{q}}\right) \leq 0$ and then embedded in the OCP as trajectory constraint. Unlike [12], the entire geometry of the robots is not approximated by spheres in order to save computation time. By analyzing the possible collision points, only the required spheres are kept. Qualitatively, the chosen spheres, for the case of self-collisions, are shown in Fig. 3. The base body of the mobile platforms is approximated by two spheres, each wheel by one sphere and each manipulator by a total of six spheres. The telescopic axis is approximated via a total of six spheres distributed over the full length of the extended rod. After defining the necessary spheres, it is important to determine which collisions are possible. Only possible collisions are considered in the OCP in order to keep the number of distance constraints and thus the computation time low. A collision between the wrist joint and the elbow joint can be excluded, just like collisions between the shoulder joint and the elbow joint or the mobile platform.

To prevent collisions between the two mobile manipulators, spheres different from those used in the consideration of self-collisions can be used for approximation. Here, significantly fewer but larger spheres are sufficient for collision avoidance, which is illustrated in Fig. 4. One large sphere is chosen for each mobile platform, in order to ensure enough safety distance between the mobile manipulators.

IV. TIME-OPTIMAL TRAJECTORY GENERATION

The time-optimal trajectories for a group of n mobile manipulators with the states $\hat{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_n^T \end{bmatrix}^T$ and control variables $\hat{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{u}_1^T, \ldots, \boldsymbol{u}_n^T \end{bmatrix}^T$ are generated by solving the following OCP:

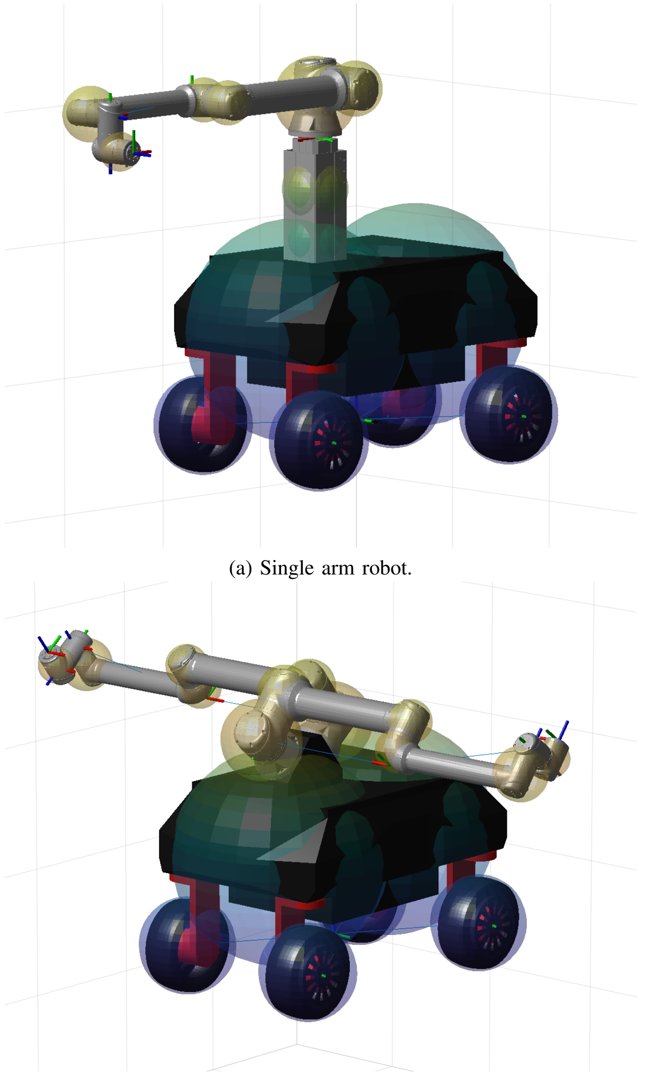
$$\min_{\hat{\boldsymbol{u}}(\cdot)} \quad J(t_{\mathrm{f}}, \hat{\boldsymbol{u}}(t)) \tag{12a}$$

$$\text{s.t.} \quad \dot{\hat{\boldsymbol{x}}}(t) = \boldsymbol{f}(\hat{\boldsymbol{x}}(t), \hat{\boldsymbol{u}}(t)), \tag{12b}$$

$$\boldsymbol{\Psi}(\hat{\boldsymbol{x}}(t_0), \hat{\boldsymbol{x}}(t_{\mathrm{f}})) = \boldsymbol{0}, \tag{12c}$$

$$\boldsymbol{g}(\hat{\boldsymbol{x}}(t), \hat{\boldsymbol{u}}(t)) \leq \boldsymbol{0} \quad \forall t \in [t_0, t_{\mathrm{f}}]\,. \tag{12d}$$

Since time-optimal trajectories are required, the cost-functional (12a) is chosen to be $J_1(t_{\mathrm{f}}) = t_{\mathrm{f}}$. As the results



(a) Single arm robot.

(b) Dual arm robot.

Fig. 3: Approximation of the mobile manipulators geometry using suitable spheres to avoid self-collisions.

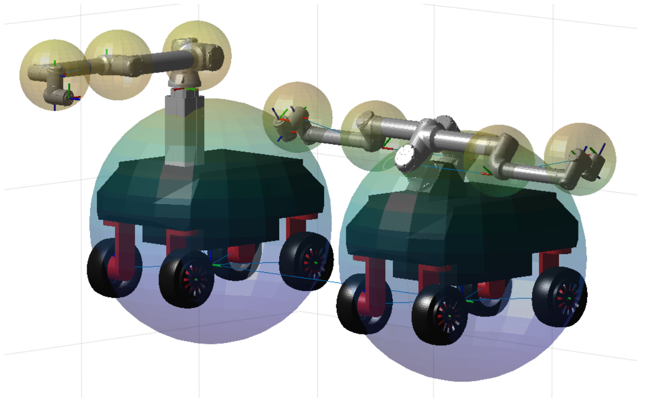


Fig. 4: Approximation of the mobile manipulators geometry using suitable spheres to avoid collisions between the robots.

will show in the next section, it is reasonable to extend the cost-functional

$$J_2(t_\mathrm{f}, \hat{\boldsymbol{u}}(t)) = t_\mathrm{f} + \int_{t_0}^{t_\mathrm{f}} \hat{\boldsymbol{u}}^T(t)\boldsymbol{Q}\hat{\boldsymbol{u}}(t)\mathrm{d}t \tag{13}$$

by penalizing the control variables in order to deal with the redundancy of the system. The system equations (12b) correspond to the selected state-space representation (9) or (10) of the mobile manipulators. The equality constraint (12c)

$$\boldsymbol{\Psi}(\hat{\boldsymbol{x}}(t_0), \hat{\boldsymbol{x}}(t_\mathrm{f})) = \begin{bmatrix} \hat{\boldsymbol{x}}(t_0) - \hat{\boldsymbol{x}}_{t_0} \\ \mathrm{FK}\left(\mathcal{M}, \hat{\boldsymbol{x}}_{\boldsymbol{q}}(t_\mathrm{f})\right) - \boldsymbol{T}_{\mathrm{TCP,f}} \\ \hat{\boldsymbol{x}}_{\dot{\boldsymbol{q}}}(t_\mathrm{f}) \end{bmatrix} \tag{14}$$

enforces the initial condition $\hat{\boldsymbol{x}}_{t_0}$ and the terminal condition, which consists of the desired target poses $\boldsymbol{T}_{\mathrm{TCP,f}}$ as well as the target joint velocity which is zero. The trajectory constraints (12d)

$$\boldsymbol{g}(\hat{\boldsymbol{x}}(t), \hat{\boldsymbol{u}}(t)) = \begin{bmatrix} \hat{\boldsymbol{x}}(t) - \hat{\boldsymbol{x}}_{\max} \\ \hat{\boldsymbol{x}}_{\min} - \hat{\boldsymbol{x}}(t) \\ \hat{\boldsymbol{u}}(t) - \hat{\boldsymbol{u}}_{\max} \\ \hat{\boldsymbol{u}}_{\min} - \hat{\boldsymbol{u}}(t) \\ \mathrm{col}\left(\mathcal{M}, \hat{\boldsymbol{x}}_{\boldsymbol{q}}(t)\right) \end{bmatrix} \tag{15}$$

enforce the kinematic and dynamic constraints of the joints as well as the distance constraints for collision avoidance.

If a workpiece needs to be moved cooperatively by n mobile manipulators, additional trajectory constraints need to be introduced. When gripping the workpiece, a closed kinematic chain is created between the manipulators, meaning that the movements of the mobile manipulators are coupled. From a kinematic point of view, the distance $d_{\mathrm{TCP},ij}$ between the positions $\boldsymbol{p}_{\mathrm{TCP},i/j}(\boldsymbol{x}_{i/j,\boldsymbol{q}}(t))$ of the TCPs of the mobile manipulators

$$||\boldsymbol{p}_{\mathrm{TCP},i}(\boldsymbol{x}_{i,\boldsymbol{q}}(t)) - \boldsymbol{p}_{\mathrm{TCP},j}(\boldsymbol{x}_{j,\boldsymbol{q}}(t))||_2 = d_{\mathrm{TCP},ij} \quad \forall t, \tag{16}$$

must remain constant throughout the trajectory, since no sliding at the gripping points is assumed. In addition, the orientations $\boldsymbol{R}_{\mathrm{TCP},i}(\boldsymbol{x}_{i,\boldsymbol{q}}(t))$ of the TCPs

$$\boldsymbol{R}_{\mathrm{TCP},1}(\boldsymbol{x}_{1,\boldsymbol{q}}(t)) = ... = \boldsymbol{R}_{\mathrm{TCP},n}(\boldsymbol{x}_{n,\boldsymbol{q}}(t)) \quad \forall t, \tag{17}$$

must correspond, when assuming an identical gripping orientation. By adding those additional kinematic conditions in (15), constraint forces in the workpiece are prevented.

To solve the OCP, direct numerical methods are chosen due to the complexity of the problem. The simultaneous approach (SA) as well as the direct collocation (DC) are investigated with respect to the generated trajectories and the required computation time.

V. RESULTS

The methodical approaches presented so far are examined in the following. First point-to-point trajectories of a single mobile manipulator, then a cooperative point-to-point trajectory of two mobile manipulators are evaluated considering the optimal transition time, the computation time and the generated trajectories of the TCPs. For the implementation, the open source software *CasADi* is used, which applies automatic differentiation to calculate the required gradients [13].

A. Point-to-point trajectory

The OCP set up in (12) is solved for 10 randomly generated target poses for a single mobile manipulator with both numerical methods using different degrees of model abstraction and different cost-functionals. The desired target poses represent different pickup points for workpieces in order to erect a stud wall. The mean results and the results of an exemplary trajectory are summarized in Table I. It is mentioned that the optimal transition time t_{opt} does not change for the different approaches. In terms of the mean computation time $\bar{t}_{\text{comp}}$, large differences between the numerical methods are apparent. When using the forward dynamics, SA takes in average 285 times longer than the DC to solve the OCP. In the case of the integrator chain, SA takes 146 times as long. This reflects in the mean number of iterations $\bar{n}_{\text{iter}}$, which is higher for SA in each case. This difference can be explained by the characteristics of the different numerical methods. The computation of the required gradients for SA is generally very time-consuming, since the considered matrices are fully occupied. In DC, the gradients are generally easier to compute since the matrices under consideration are sparse. This is exploited by the solver to save computation time. Therefore, only the DC approach is pursued for further evaluations. Comparing the computation time considering the different degrees of model abstraction, it can be seen that using the integrator chain is significantly faster regardless of the numerical method. When using the integrator chain, the torque constraints can not directly be considered in the OCP. Hence the inverse dynamics are used in retrospect to calculate the required torques for the generated trajectory, in order to check if they comply with the torque constraints, which is the case here.

The TCP paths are calculated from the generated joint trajectories using the forward kinematics and are shown for an exemplary point-to-point trajectory in Fig. 5 together with the mobile manipulator in its initial position. It can be seen that the time-optimal trajectories computed with the cost-functional J_1, especially the case using DC and the forward dynamics, do not take a direct path to the desired target pose and thus have an increased positioning effort, which is not desired from a user's perspective. This effect is particularly clear for joint q_6 which is the shoulder joint of the manipulator, as illustrated in Fig. 6, and may be due to the redundancy of the mobile manipulator, whereby the movements of this joint have no influence on the time-optimality of the trajectory. In order to deal with this effect, the cost-functional is extended as in (13). The optimal transition time is not significantly affected by the change of the cost-functional since the weighting matrices are chosen very small in order to deal with the redundancy but not lose time-optimality. Since the control variables are different for the two model abstractions, different weighting matrices $\boldsymbol{Q}_{\text{FD}} = 10^{-2}\boldsymbol{I}$ and $\boldsymbol{Q}_{\text{INT}} = 10^{-8}\boldsymbol{I}$ are chosen. Regarding the mean computation time, there are only minimal changes as can be seen in Table I. The exemplary results are shown in Fig. 5 and 6. The difference in the generated trajectories between the different degrees of model abstraction is clearly reduced when using J_2. Thus, when using J_2, the integrator chain should be used instead of the forward dynamics in order to save computation time.

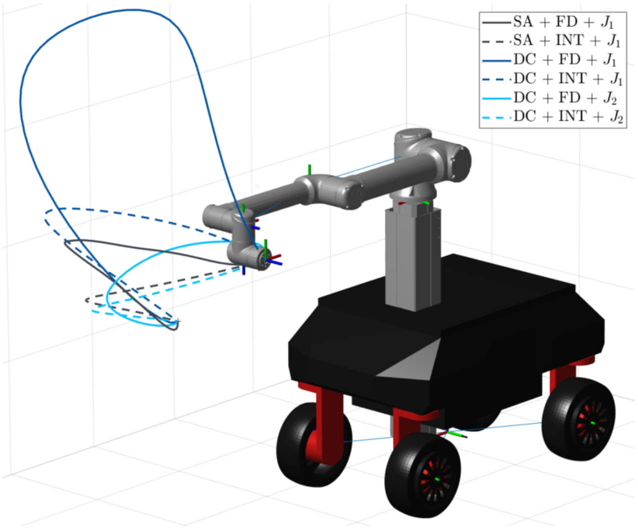


Fig. 5: Comparison of the TCP paths for an exemplary point-to-point trajectory for both numerical methods using different degrees of model abstraction and different cost-functionals.

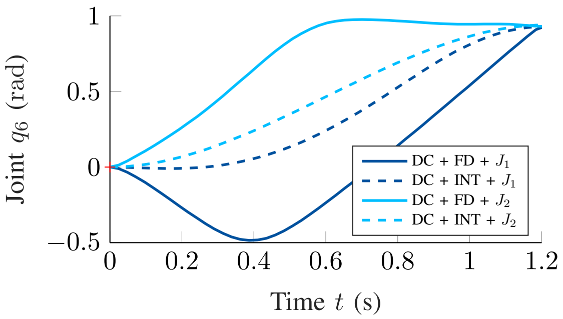


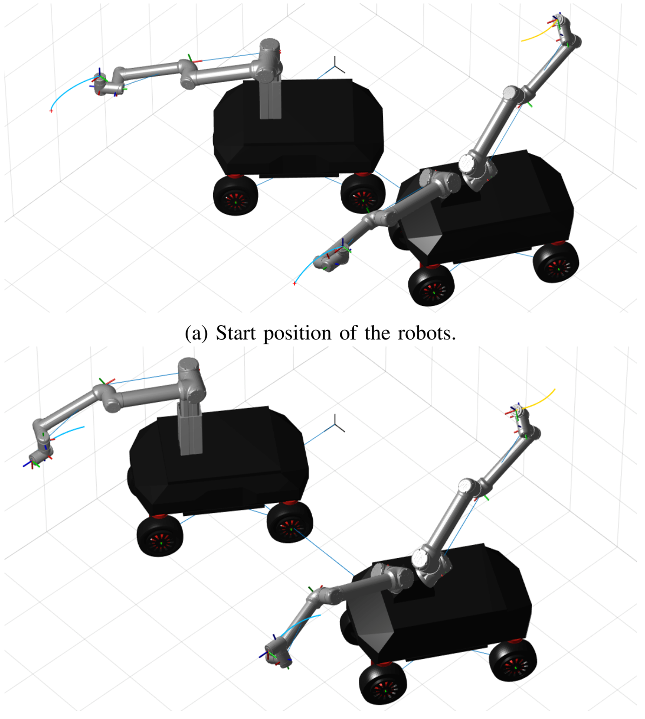
Fig. 6: Generated position trajectories of joint q_6 for an exemplary point-to-point trajectory.

B. Cooperative point-to-point trajectory

The findings from the last section are taken into account for the investigation of a cooperative point-to-point trajectory of the two mobile manipulators presented in chapter III. Thus, only DC, cost-functional J_2 and the integrator chain as state-space representation are used due to the computational efficiency and the advantages regarding the redundancy of the system. The exemplary trajectory represents the motion required to position a panel on a stud wall. The panel is hold by both robots horizontally and is then rotated and positioned vertically on the wall. The OCP is solved in $t_{\text{comp}} = 219\,\text{s}$ and the optimal transition time is $t_{\text{opt}} = 2.02\,\text{s}$. The TCP paths of the two mobile manipulators are shown in Fig. 7, without the panel due to better visbility. In Fig. 7a, the start position and in Fig. 7b, the target position of the mobile manipulators are shown together with the TCP paths.

TABLE I: Comparison of the mean computation time $\bar{t}_{\mathrm{comp}}$ and mean number of iterations $\bar{n}_{\mathrm{iter}}$ for 10 randomly generated target poses as well as the computation time $t_{\mathrm{comp,ex}}$ and number of iterations $n_{\mathrm{iter,ex}}$ for an exemplary trajectory for both numerical methods using different degrees of model abstraction and different cost-functionals.

	Numerical method		Model abstraction		Cost-functional		$t_{\mathrm{comp,ex}}$ (s)	$\bar{t}_{\mathrm{comp}}$ (s)	$n_{\mathrm{iter,ex}}$ (-)	$\bar{n}_{\mathrm{iter}}$ (-)
	Simultaneous approach	Direct collocation	Forward dynamics	Integrator chain	J_1	J_2				
SA + FD + J_1	×		×		×		35916	20090	517	294
SA + INT + J_1	×			×	×		392	277	2308	1560
DC + FD + J_1		×	×		×		58.1	70.4	69	84
DC + INT + J_1		×		×	×		1.24	1.9	28	48
DC + FD + J_2		×	×			×	760.8	87.0	769	104
DC + INT + J_2		×		×		×	1.36	4.4	33	99



(a) Start position of the robots.

(b) Target position of the robots.

Fig. 7: TCP paths for an exemplary cooperative point-to-point trajectory.

Since the integrator chain is assumed, it must be checked if the actuators are able to realize the generated trajectories. Since the panel is transported horizontally in one plane, it is assumed that each manipulator carries half the weight of the workpiece. Hence, an external wrench $\mathcal{F}^{\mathrm{x}} = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & -\frac{1}{2}mg\end{bmatrix}^T$ acts on both TCPs. The inverse dynamics can then be used to calculate the required torques for this trajectory. In this case, only workpieces up to a maximum weight of $m = 9\,\mathrm{kg}$ are possible, which is enough since typical panels do not weigh more.

VI. CONCLUSION

This paper introduced a standardized approach for modeling heterogeneous robots and incorporating them into an OCP, which considers collisions, cooperation, kinematic and dynamic constraints. The investigation of the time-optimal trajectories showed 1.) DC is preferable over SA, 2.) the extended cost-functional J_2 should be used in order to deal with the redundancy of the system, and 3.) when using J_2, the integrator chain should be used in order to save computation time. In future works, further reduction of the computation time to achieve real-time capability and to enable adaptive trajectory generation, that responds to changing environmental conditions using sensor data, will be investigated.

REFERENCES

[1] F. Barbosa *et al.*, “Reinventing construction: A route to higher productivity,” McKinsey Global Institute, Tech. Rep., 2017.

[2] A. Voelz and K. Graichen, “An Optimization-based Approach to Dual-Arm Motion Planning with Closed Kinematics,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[3] I. Pajak, “Real-Time Trajectory Generation Methods for Cooperating Mobile Manipulators Subject to State and Control Constraints,” *Journal of Intelligent and Robotic Systems*, vol. 93, no. 3-4, pp. 649–668, June 2018.

[4] A. Z. Bais, S. Erhart, L. Zaccarian, and S. Hirche, “Dynamic Load Distribution in Cooperative Manipulation Tasks,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[5] J. Hernandez-Barragan, C. Lopez-Franco, N. Arana-Daniel, and A. Y. Alanis, “Inverse kinematics for cooperative mobile manipulators based on self-adaptive differential evolution,” *PeerJ Computer Science*, vol. 7, 2021.

[6] S. Parascho *et al.*, “Robotic vault: a cooperative robotic assembly method for brick vault construction,” *Construction Robotics*, vol. 4, no. 3-4, pp. 117–126, 2020.

[7] F. Basile, F. Caccavale, P. Chiacchio, J. Coppola, and C. Curatella, “Task-oriented motion planning for multi-arm robotic systems,” *Robotics and Computer Integrated Manufacturing*, vol. 28, no. 5, pp. 569–582, Oct. 2012.

[8] A. Tika, N. Gafur, V. Yfantis, and N. Bajcinca, “Optimal Scheduling and Model Predictive Control for Trajectory Planning of Cooperative Robot Manipulators,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9080–9086, 2020.

[9] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, May 2017.

[10] C. R. Rocha, C. P. Tonetto, and A. Dias, “A comparison between the Denavit-Hartenberg and the screw-based methods used in kinematic modeling of robot manipulators,” *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 4, pp. 723–728, Aug. 2011.

[11] R. Featherstone, *Rigid Body Dynamics Algorithms*. Springer, 2008.

[12] Y. Zhao, H.-C. Lin, and M. Tomizuka, “Efficient Trajectory Optimization for Robot Motion Planning,” in *15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2018.

[13] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi - A software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, July 2018.