Payload-aware trajectory optimisation for non-holonomic mobile multi-robot manipulation with tip-over avoidance

Florian Kennel-Maushart[1], Stelian Coros[1]

***Abstract*—Cooperative mobile manipulation is an increasingly important topic in robotics: Just as humans need to collaborate on many tasks, robots need to be able to work together, e.g., to transport heavy or unwieldy objects in unstructured environments. But mobile multi-robot systems pose unique challenges, such as a much larger configuration space for motion planning, stability concerns and, especially for wheeled mobile robots, non-holonomic constraints. To tackle these challenges, we propose a multi-robot, bi-level optimization system, based on a direct transcription formulation for wheeled mobile manipulation. Our formulation uses static forces, calculated for stability objectives at the lower level to inform the wheeled trajectory planning at the higher level. This allows for effective planning, while ensuring safe execution, and improving the open-loop performance on real robots. We demonstrate our model's ability to tackle challenging motion-planning tasks and evaluate its improved real-world performance on the Clearpath Husky mobile platform. Finally, we integrate the system with our previously presented Mixed-Reality interface. *Index Terms*—Multi-Robot Systems, Path Planning for Multiple Mobile Robots or Agents, Cooperating Robots, Robotics and Automation in Construction, Wheeled Robots**

I. Introduction

Industrial automation has long been focused on building specialised facilities around large, fixed-base robots that steadfastly repeat the same task with remarkable precision. Meanwhile, outside of these large, structured facilities, many jobs and tasks require mobility, agility and cooperation. While cooperative mobile manipulation involves numerous challenges, the focus of this work is on trajectory optimization for multiple non-holonomic mobile manipulators transporting an object cooperatively through a restricted environment, while avoiding tip-over. Being able to provide robust and safe solutions to this problem is highly beneficial for a wide variety of jobs and tasks, such as the installation of prefabricated pieces on construction sites (cf. Fig. 1). Wheeled mobile manipulators typically offer some redundancy in their degrees of freedom and theoretically are able to access an infinite workspace. However, for non-holonomic wheeled bases, e.g., skid-steering vehicles, the system is under-actuated, as it cannot directly move sideways. This makes the problem of finding a collision-free path in constrained environments challenging [1]. An interesting additional problem that arises in mobile robots as opposed to fixed-base robots, is the danger for them to tip over when a moment is imposed on the base. Aside from the danger of a robot tipping over, imbalance will also lead to an uneven distribution of the normal forces on the wheels, which in turn highly affects the steering behaviour of the robot [2]. While these problems are already non-trivial to solve in the single-robot case, solving them for a multi-robot system adds additional complexities: Collaborative motion-planning highly constrains the available range of motion for each robot, and can easily lead to the trajectory optimization getting stuck in sub-optimal configurations. This is especially true for shooting-based methods, whereas direct transcription offers more flexibility by only introducing pairwise coupling between control and state variables [3]. We present a novel combination of all of these aspects into a unified method in the context of cooperative multi-robot transport. In detail, the contributions of this work are:

Fig. 1: Render of a multi-robot system cooperatively installing a prefabricated element on a construction site.

- A bi-level optimisation approach, in which we first calculate the normal forces at the wheels according to the robot configuration, and then use these forces to formulate our steering and tip-over avoidance constraints.
- Evaluation of our system in simulation, including difficult obstacle avoidance scenarios with multi-robot setups and an ablation study on optimization factors.
- Evaluation of our system on a real robot, showing how the force-aware model is able to significantly improve the open-loop real-world fidelity over the baseline controller which is normally deployed on the robots.
- Deployment on a real multi-robot system and discussion of the open-loop performance as well as the system's ability to cooperatively carry a payload.

Manuscript received: February 16, 2024; Revised May 15, 2024; Accepted June 15, 2024. This paper was recommended for publication by Editor M. Ani Hsieh upon evaluation of the Associate Editor and Reviewers' comments. This work was partially supported by the HILTI Group and Microsoft.

[1] Florian Kennel-Maushart and Stelian Coros are with the Department of Computer Science, ETH Zurich, 8092 Zurich, Switzerland. florian.maushart@inf.ethz.ch; scoros@inf.ethz.ch

Digital Object Identifier (DOI): see top of this page.

Finally, we discuss our results and summarise our learnings and shortcomings of the presented system. In the accompanying video material, we also show an implementation of this system within a Mixed-Reality (MR) interaction framework.

II. Related Work

Cooperative Fixed-Base Manipulation While we focus on cooperative *mobile* manipulation, looking at cooperative manipulation for fixed-base systems can be instructive. In [4], Adorno et al. introduce a cooperative dual task-space formulation using quaternion algebra, which lets them efficiently represent dual-arm manipulation tasks. They formulate a control law and show applications such as holding a balloon and pouring a glass of water on a small humanoid platform. While not relying on relative pose control, extending our method with quaternion algebra could enable more involved manipulation tasks in future work. Laha et al. [5] present a flexible task planner that uses heuristics to switch between trajectories while executing dual-arm manipulation tasks. This lets them avoid singularities, even with cluttered environments or dynamic obstacles, but requires fast re-planning. Finally, using force regulation feed-back control [6] allows for compensation of deviations between expected and measured forces at the end-effector, which would be interesting for future work.

Skid-steering As skid-steering mobile robots (SSMRs) have low mechanical complexity, they offer advantages such as affordability, ease of maintenance, and ruggedness. Unfortunately, they are complex to model compared to omnidirectional or differential-drive vehicles. As the name implies, the wheels are not in perfect rolling contact with the floor, but are skidding when the vehicle turns. Simplifying assumptions can be made, e.g., reducing the skid-steering formulation to that of a differential-drive vehicle [2]. These formulations, however, neglect the complex wheel-ground interactions and break down as soon as the payload is not evenly distributed on the base [7]. An extensive treatment of kinematic and dynamic steering models for skid-steering is presented by Kozłowski et al. in [8]. They present a tracking controller, and validate it in a Matlab/Simulink simulation. A motion model based on the Instantaneous Center of Rotation (ICR) and its experimental validation is later introduced by Mandow et al. in [9]. The ICR offers a way to reason about the turning behaviour of skid-steering vehicles. The shift of the center of mass (COM) introduced by a manipulator is treated by Liu and Liu in [2]. Their model is parametrised by the ICR offset d_0 as well as wheel-slip parameters s_r, s_l. Our approach, detailed in Sec. III is in-part based on this work. Finally, a lot more recent work has been done on skid-steering, introducing inertia-based [10], MPC [11] or learning-based [12] approaches to improve path tracking, which lie beyond the scope of this paper, as our focus currently lies on improving the open-loop motion planner.

Tip-over avoidance The risk of tipping over can usually be neglected for systems with favorable payload to base weight ratios. However, smaller systems offer advantages e.g., more agility and typically a smaller footprint, allowing them to move through cluttered environments and be transported between locations more easily. In [13], Ding et al. present an online dynamic tip-over avoidance algorithm. They derive the moments acting on the base due to end-effector and wheel motion. They then present simulation results and verification on an omnidirectional robot. They do not use their method in the trajectory optimisation, but switch to tip-over-avoidance during execution if their criterion is met. We would like to avoid unexpected behaviour from an operator point-of-view by including these considerations already in the motion planning. Pankert et al. present a perceptive model predictive controller for continuous mobile manipulation in [14]. They use a Zero Moment Point objective for stability, while tracking an end-effector target force. Using a penalty method, they can deal with tracking errors of the non-holonomic base, but they don't consider these forces for steering. Finally, Petrovic et al. recently presented an extended model for tip-over stability on uneven terrain [15]. While we do not consider sloped terrain yet due to the lack of sensing capabilities on the robot, we could include a similar objective into our system in the future with slight adjustments of the formulation.

Cooperative Transport A cooperative transport system with mobile robots can take many different forms, depending on the robot systems and transport strategies. Extensive reviews of different approaches have been conducted in [16] and [17]. While these reviews include strategies such as pushing and caging, for our application domain we are most interested in grasping. The above mentioned reviews also differentiate between centralised and decentralised planners. While decentralised planners offer better scalability and robustness, centralised planners are often significantly simpler to implement. A system to cooperatively grasp and lift a payload is presented in [18]. The authors focus on the distribution of the robots around the payload and the interaction forces. The mobile manipulator itself is based on a differential-drive system with a low center of mass. Therefore, tip-over considerations or complex steering are not addressed. Zhao et al. present the stability analysis of a more complex towing system in [19], focusing on payload stability and tip-over avoidance. A limited numerical analysis of the proposed model with pre-determined joint-angles is presented. Two examples of motion-planners for mobile manipulator teams which rely on formation control as a policy for the mobile bases are [20] and [21]. These systems *a priori* try to stay close to a target formation, which limits the range of maneuvers they can exhibit. An MPC formulation for a similar setting has been presented and evaluated in [22] and a similar setup, using dual-quaternion algebra is presented in [23]. While these systems were deployed on omnidirectional robot bases, a more recent extension to this for non-holonomic agents has been proposed in [24]. A system that more closely matches the system we want to examine has been used by Xu et al. in [25]. Although using similar robots, the proposed method only takes care of force tracking, based on the position of the payload following a predetermined trajectory. Finally, a system that most closely matches our setup has been proposed quite recently by Zhang et al. in [26] in an arXiv preprint. Their robot team exhibits some impressive maneuvers, but relies on omnidirectional bases and a leader-follower scheme to transport the payload to a given goal position. They also do not consider the influence of the payload weight on steering.

III. METHOD

A. Force Model

We consider a multi-robot system with the forces depicted in Fig. 2. A detailed model of all the transforms as well as the URDF model used in our simulation can be found in [27].

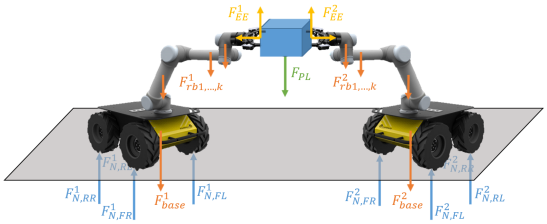

Fig. 2: Force diagram of the multi-robot system, with supporting forces (blue), robot weight (orange), external payload (green) and internal forces at the end-effector (yellow).

We are developing our model under two main *assumptions*:

1) All robots can meet their end-effector targets, i.e., the only forces acting at the end-effectors are due to the weight of the payload and the manipulator self-weight.
2) The system is in quasi-static equilibrium, i.e., inertial effects due to base or arm acceleration are not considered.

This limits how quickly our robots can accelerate to not violate these assumptions. In practice, we achieve this by adding smoothing to our optimisation (cf. $\mathcal{O}_{sm}$ in Eq. 5).

B. Skid-steering

A model that integrates the skidding of the wheels, as well as the ICR axis offset as a function of a manipulator pose is presented by Liu and Liu in [2]. To summarize, the equations of motion are described as follows:

$$\begin{aligned}\dot{x} &= \frac{[r(u_l+u_r)+(s_l+s_r)]\cos(\phi)}{2} \\ &\quad + \frac{d_0\,[r(u_r-u_l)+(s_r-s_l)]\sin(\phi)}{d_m} \\ \dot{y} &= \frac{(r(u_l+u_r)+(s_l+s_r))\sin(\phi)}{2} \\ &\quad - \frac{d_0\,[r(u_r-u_l)+(s_r-s_l)]\cos(\phi)}{d_m} \\ \dot{\phi} &= \frac{r(u_r-u_l)+(s_r-s_l)}{d_m},\end{aligned} \tag{1}$$

with u_r, u_l being the wheel speed commands, d_0 the offset of the ICR axis from the geometric center of the robot in local x-direction, s_r and s_l the right and left slip-velocities, i.e., the speeds at which the wheels slip against the driving direction. Finally, x, y and ϕ are the base position and orientation, d_m the robot base width and r the wheel radius. A representation of these parameters can be seen in Fig. 3, where the offset of the ICR compared to the geometric center is also visualized.

C. Bi-level optimization formulation

We want to optimise the following system:

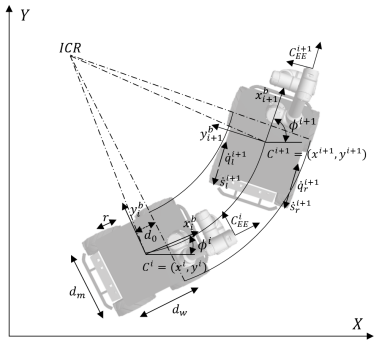

Fig. 3: Steering and model parameters (adapted from [2]).

$$\begin{aligned}\min_{\mathbf{q}^*} \quad & \mathcal{O}_U(\mathbf{q}^*, \mathbf{f}) \\ \text{s.t.} \quad & \mathbf{f}_{i,\perp}(\mathbf{q}) \geq 0 && \forall \mathbf{f}_i \in \mathbf{f} \\ & -\mathbf{f}_{i,\perp}(\mathbf{q})\cdot\mu \;\leq\; \mathbf{f}_{i,\parallel}(\mathbf{q}) \;\leq\; \mathbf{f}_{i,\perp}(\mathbf{q})\cdot\mu && \forall \mathbf{f}_i \in \mathbf{f} \\ \text{where} \quad & \mathbf{f}(\mathbf{q}) = \operatorname*{argmin}_{\mathbf{f}^*} \mathcal{O}_L(\mathbf{q}, \mathbf{f}^*)\end{aligned} \tag{2}$$

with lower- and upper-level objectives $\mathcal{O}_L$ and $\mathcal{O}_U$ respectively. We discretize the problem into t timesteps of length dt, but in the following we will only indicate the discretization where useful, for conciseness. The first decision variable is $\mathbf{q} := (\mathbf{q}_0, \ldots, \mathbf{q}_n)$, where $\mathbf{q}_0$ represents the payload. The n robotic agents follow as $\mathbf{q}_1$ to $\mathbf{q}_n$. We simultaneously optimize over the state and input, using a direct transcription scheme [28]. For this, we assemble the full vector $\mathbf{q}_i := [\boldsymbol{\theta}_i, \mathbf{u}_i]^T$ for each robot, where $\boldsymbol{\theta}_i$ are the state variables, namely the base pose and arm joint angles, and $\mathbf{u}_i := [u_l, u_r]$ are the wheel velocity inputs for the left and right wheel-pairs. For $\mathbf{q}_0$ there are no velocity inputs. The second decision variable, $\mathbf{f} := (\mathbf{f}_1, \ldots, \mathbf{f}_n)$, contains the supporting forces for each robot as $\mathbf{f}_i := (f_{FL}, f_{FR}, f_{RL}, f_{RR})$, with front-left to rear-right wheel forces as indicated in Fig. 2. In our concrete example, the lower level $\mathcal{O}_L$ solves the quasi-static force and torque equilibrium, which gives us the normal forces at the wheel contact points $\mathbf{r}_i$ in a differentiable formulation:

$$\begin{aligned}\mathcal{O}_L(\mathbf{q}, \mathbf{f}) = \Big|\Big| \sum_i \mathbf{f}_i - \bar{\mathbf{f}} \Big|\Big|_2^2 + \Big|\Big| \sum_i (\mathbf{r}_i \times \mathbf{f}_i) - \bar{\tau}(\mathbf{q}) \Big|\Big|_2^2 \\ + \omega_R \|\mathbf{f}\|,\end{aligned} \tag{3}$$

where $\bar{\mathbf{f}}$ and $\bar{\tau}$ are the total forces and moments generated by the weight of the robot and external payload, and the last term is used to regularize the forces towards a smaller norm with regularisation weight ω_R. The upper level, $\mathcal{O}_U$, can be split into three terms, which either depend on $\mathbf{q}$, $\mathbf{f}$ or both.

$$\mathcal{O}_U(\mathbf{q}, \mathbf{f}) = \mathcal{O}_1(\mathbf{q}) + \mathcal{O}_2(\mathbf{q}, \mathbf{f}) + \mathcal{O}_3(\mathbf{f}). \tag{4}$$

The first term,

$$\begin{aligned}\mathcal{O}_1(\mathbf{q}) = \mathcal{O}_{EE}(\mathbf{q}) + \omega_{sm}\mathcal{O}_{sm}(\mathbf{q}) \\ + \;\omega_{lim}\mathcal{O}_{lim}(\mathbf{q}) + \mathcal{O}_{DCA}(\mathbf{q})\end{aligned} \tag{5}$$

contains sub-objectives that purely depend on $\mathbf{q}$. In detail, the end-effector sub-objective $\mathcal{O}_{EE}(\mathbf{q}_i, \mathbf{q}_0)$

$$\begin{aligned}\mathcal{O}_{EE,t}(\mathbf{q}_i, \mathbf{q}_0) = \|p(\mathcal{F}_{ee}(\mathbf{q}_{i,t})) - p(\mathcal{F}_{a,i}(\mathbf{q}_{0,t}))\|^2 + \\ \|R(\mathcal{F}_{ee}(\mathbf{q}_{i,t})) - R(\mathcal{F}_{a,i}(\mathbf{q}_{0,t}))\|^2,\end{aligned} \tag{6}$$

minimizes the error in position $p(\cdot)$ and rotation $R(\cdot)$ between the forward-kinematic solution $\mathcal{F}$ for the end-effector of each robot and the respective attachment point on the payload, indicated as $\mathcal{F}_{a,i}$. For the full $\mathcal{O}_{EE}(\mathbf{q}) = \omega_{EE} \sum_{i=1}^{n} \sum_{t=0}^{T} \mathcal{O}_{EE,t}(\mathbf{q}_i, \mathbf{q}_0) + \omega_{tar}\mathcal{O}_{EE,T}(\mathbf{q}_0, p_{tar})$, we add the terms for all robots over all time steps as well as an additional term to drive the payload pose $\mathbf{q}_0$ at the final step T towards the user-defined target pose p_{tar}. The smooth acceleration sub-objective

$$\mathcal{O}_{sm}(\mathbf{q}) = \sum_{t=0,t>2}^{T} \frac{1}{dt^2} \|\mathbf{q}_t - 2\mathbf{q}_{t-1} + \mathbf{q}_{t-2}\|^2 \tag{7}$$

minimizes the acceleration, i.e. the rate of change for each entry in $\mathbf{q}$, over all time steps. As in [29], the joint-limits objective $\mathcal{O}_{lim}(\mathbf{q})$ is implemented as a soft barrier function. We define the unilateral quadratic functions:

$$B_b^+(a) = \begin{cases} 0 & a \leq b + \varepsilon \\ \sigma(a-b)^2 & a > b + \varepsilon \end{cases} \tag{8}$$

$$B_b^-(a) = \begin{cases} 0 & a \geq b + \varepsilon \\ \sigma(a-b)^2 & a < b + \varepsilon \end{cases} \tag{9}$$

such that for each joint angle $q_{i,j}$ at timestep t we get term:

$$\mathcal{O}_{lim,i}(q_{i,j}) = B^+_{q_{\max,i,j}}(q_{i,j}) + B^-_{q_{\min,i,j}}(q_{i,j}).$$

with maximum and minimum joint angles q_{max} and q_{min}, stiffness σ and limit ε. We then sum up all joint angle objectives $\mathcal{O}_{lim,i}$ over all timesteps t for the full objective $\mathcal{O}_{lim}$. Finally, for the differentiable collision-avoidance sub-objective $\mathcal{O}_{DCA}(\mathbf{q})$, we formulate self-collision, world-collision and inter-agent collision terms for each robot, with their respective weights $\omega_{DCA,self}$, $\omega_{DCA,world}$, $\omega_{DCA,inter-agent}$. These terms are based on differentiable distance formulations between pairs of collision primitives. For conciseness we refer the reader to [29] for more detail. The second term,

$$\mathcal{O}_2(\mathbf{q}, \mathbf{f}) = \omega_{steer}\mathcal{O}_{steer}(\mathbf{q}, \mathbf{f}) + \omega_{tip}\mathcal{O}_{tip}(\mathbf{q}, \mathbf{f}) \tag{10}$$

contains sub-objectives, which depend both on $\mathbf{q}$ and $\mathbf{f}$. We formulate $\mathcal{O}_{tip}(\mathbf{q}, \mathbf{f})$ as a soft-barrier function, with a predetermined threshold at the minimal supporting force we want to ensure, similarly to $\mathcal{O}_{lim}$ above. In theory, as long as the force remains positive, the vehicle will not tip over, although in practice we typically establish a positive threshold to account for dynamic effects that our model does not cover. It is worth noting, that this formulation is conceptually very similar to Zero-Moment Point formulations as used for example in [14], where the center of gravity of the robot is constrained to stay within a convex geometric boundary. The steering objective $\mathcal{O}_{steer}(\mathbf{q}, \mathbf{f})$ ensures that the change of robot state variables x, y and ϕ (see Fig. 3) matches the dynamics from Eq. 1 as:

$$\begin{aligned}\mathcal{O}_{t,steer}(\mathbf{q}, \mathbf{f}) = ||\dot{x_t} \cdot dt - (x_{t+1} - x_t)||^2 \\ + ||\dot{y_t} \cdot dt - (y_{t+1} - y_t)||^2 \\ + ||\dot{h(\phi_t)} \cdot dt - (h(\phi_{t+1}) - h(\phi_t))||^2\end{aligned} \tag{11}$$

at each timestep t of duration dt. For the heading angle ϕ we use a vector formulation, indicated by $h(\phi)$, which ensures that we do not encounter problems when changing from negative to positive heading angles. Finally, we formulate the constraints on the forces from Eq. 2 as soft barrier functions,

$$\mathcal{O}_3(\mathbf{f}) = \left(\mathcal{S}_1^+(\mathbf{f}) + \mathcal{S}_1^-(\mathbf{f})\right) + \mathcal{S}_2(\mathbf{f}). \tag{12}$$

Specifically, this constrains the normal forces $\mathbf{f}_{i,\perp}(\mathbf{q})$ to remain positive (supporting the vehicle) and the tangential forces $\mathbf{f}_{i,\parallel}(\mathbf{q})$ to remain within a very small friction cone with friction coefficient μ, which we therefore set to $\mu = 10^{-2}$:

$$\mathcal{S}_1^+ = \sum_i \begin{cases} (\mathbf{f}_{i,\parallel} - \mu \cdot \mathbf{f}_{i,\perp})^2, & \text{if } \mathbf{f}_{i,\parallel} \geq \mu \cdot \mathbf{f}_{i,\perp}, \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

$$\mathcal{S}_1^- = \sum_i \begin{cases} (\mathbf{f}_{i,\parallel} - \mu \cdot \mathbf{f}_{i,\perp})^2, & \text{if } \mathbf{f}_{i,\parallel} \leq -\mu \cdot \mathbf{f}_{i,\perp}, \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

$$\mathcal{S}_2 = \sum_i \begin{cases} (\mathbf{f}_{i,\perp})^2, & \text{if } \mathbf{f}_{i,\perp} \leq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

D. Solving the optimization problem

Having formulated Eq. 2 as an unconstrained bi-level optimisation problem, we can use Newton's method, for which we require the first and second-order derivatives $\frac{d\mathcal{O}_U}{d\mathbf{q}}$ and $\frac{d^2\mathcal{O}_U}{d\mathbf{q}^2}$. We can use sensitivity analysis [30] to find $\frac{d\mathbf{f}}{d\mathbf{q}}$, by defining (omitting dependency of $\mathbf{f}$ on $\mathbf{q}$ for conciseness)

$$g(\mathbf{q}, \mathbf{f}) := \frac{d}{d\mathbf{f}}\mathcal{O}_L(\mathbf{q}, \mathbf{f}) = 0 \tag{16}$$

and then take the derivative

$$\frac{dg}{d\mathbf{q}} = \frac{\partial g}{\partial \mathbf{f}}\frac{d\mathbf{f}}{d\mathbf{q}} + \frac{\partial g}{\partial \mathbf{q}} = 0 \tag{17}$$

which gives us

$$\frac{d\mathbf{f}}{d\mathbf{q}} = -\left(\frac{\partial g}{\partial \mathbf{f}}\right)^{-1}\frac{\partial g}{\partial \mathbf{q}}. \tag{18}$$

We can ensure that $\frac{\partial g}{\partial \mathbf{f}}$ can be inverted by adjusting ω_R in Equation 3. Finally,

$$\frac{d\mathcal{O}_U}{d\mathbf{q}} = \frac{\partial \mathcal{O}_U}{\partial \mathbf{f}}\frac{d\mathbf{f}}{d\mathbf{q}} + \frac{\partial \mathcal{O}_U}{\partial \mathbf{q}} = -\frac{\partial \mathcal{O}_U}{\partial \mathbf{f}}\left(\frac{\partial g}{\partial \mathbf{f}}\right)^{-1}\frac{\partial g}{\partial \mathbf{q}} + \frac{\partial \mathcal{O}_U}{\partial \mathbf{q}}, \tag{19}$$

For conciseness, we omit the details of the derivation here, but using the same method as presented in [29] we get the second order derivative approximation

$$\begin{aligned}\frac{d^2\mathcal{O}_U}{d\mathbf{q}^2} \approx \left(\frac{d\mathbf{f}}{d\mathbf{q}}\right)^T \frac{\partial \mathcal{O}_U}{\partial \mathbf{f}^2}\left(\frac{d\mathbf{f}}{d\mathbf{q}}\right) \\ + \left(\frac{d\mathbf{f}}{d\mathbf{q}}\right)^T \frac{\partial \mathcal{O}_U}{\partial \mathbf{f}\partial \mathbf{q}} + \left(\frac{\partial \mathcal{O}_U}{\partial \mathbf{f}\partial \mathbf{q}}\right)^T \frac{d\mathbf{f}}{d\mathbf{q}} + \frac{\partial^2 \mathcal{O}_U}{\partial \mathbf{q}^2}.\end{aligned} \tag{20}$$

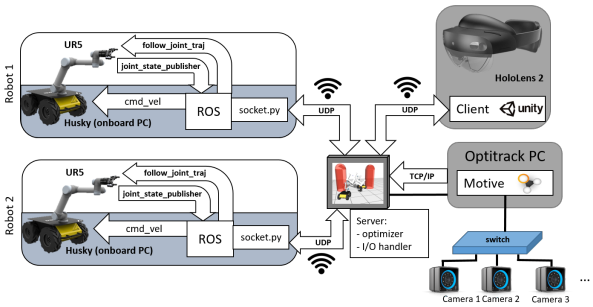

Fig. 4: System Architecture

With the first- and second-order derivatives established, we use Newton's method to find a solution to Eq. 2. To do this, we also need the derivative

$$\frac{d\mathcal{O}_{steer}}{d\mathbf{q}} = \frac{\partial \mathcal{O}_{steer}}{\partial \mathbf{q}} + \frac{\partial \mathcal{O}_{steer}}{\partial \mathbf{s}} \left(\frac{\partial \mathbf{s}}{\partial \mathbf{f}} \frac{d\mathbf{f}}{d\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial \mathbf{q}} \right) \tag{21}$$

where

$$\mathbf{s} = \mathbf{s}(\mathbf{f}(\mathbf{q})) = (d_0, s_l, s_r), \tag{22}$$

considers the dependence of the steering parameters on $\mathbf{f}(\mathbf{q})$. As shown above, the steering parameters consist of d_0, s_l and s_r, where d_0 is the axis offset of the ICR and s_l and s_r are the left and right skidding velocities, which we model as

$$d_0 = \frac{d_w(f_{FL} + f_{FR} - f_{RL} - f_{RR})}{2\sum \mathbf{f}} \tag{23}$$

with wheelbase d_w and, using slip factor c,

$$s_l = -c \cdot u_l, \quad s_r = -c \cdot u_r. \tag{24}$$

The same can be done for the derivatives of the tip-over objective $\frac{d\mathcal{O}_{tip}}{d\mathbf{q}}$ and (without having to go through the nested partial derivatives) for Eq. 12. Similarly, following Eq. 20 the Hessians can be obtained.

Remark. *Eq. 18 requires us to take the derivatives of the cross product in Eq. 3. Excluding trivial factors for brevity, we examine only* $c(\mathbf{q}, \mathbf{f}) := \frac{d}{d\mathbf{f}} \left|\left| \sum_i (\mathbf{r}_i \times \mathbf{f}_i) - \bar{\tau}(\mathbf{q}) \right|\right|_2^2$ *in detail. We get*

$$\frac{\partial c}{\partial \mathbf{f}} = 2 \begin{bmatrix} [r_1]_\times^T [r_1]_\times & \dots & [r_1]_\times^T [r_4]_\times \\ \vdots & \ddots & \vdots \\ [r_4]_\times^T [r_1]_\times & \dots & [r_4]_\times^T [r_4]_\times \end{bmatrix}$$

$$\frac{\partial c}{\partial \mathbf{q}} = 2 \begin{bmatrix} \sum_{rb} ([r_{rb} \times f_{rb}]_\times \mathbb{J}_1 - [r_1]_\times^T [f_{rb}]_\times^T \mathbb{J}_{rb}) \\ \vdots \\ \sum_{rb} ([r_{rb} \times f_{rb}]_\times \mathbb{J}_4 - [r_4]_\times^T [f_{rb}]_\times^T \mathbb{J}_{rb}) \end{bmatrix},$$

where $\mathbb{J}_{rb}$ *is the Jacobian for the centre of gravity of each rigid body, whose weight contributes to* $\bar{\tau}$ *and* $[v]_\times$ *is the skew-symmetric matrix for the cross-product with vector* v.

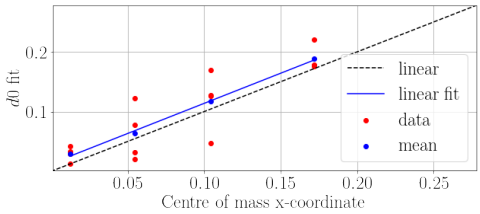

Fig. 5: Results of fitting d_0.

E. System

Our system is depicted in Fig. 4. We are using two Clearpath Husky skid-steering robots with UR5e robot arms, equipped with a Robotiq 2f-85 and 3-finger gripper, respectively. A centralised server communicates with the robots via websocket and receives back information about the current joint state from the joint state publisher of each robot, as well as their global position from an Optitrack System, which we use to track the trajectories for evaluation. Finally, we can connect to the server with our HoloLens app, which lets us visualise trajectories and input a payload goal via hand gestures. A depiction of the HoloLens system can be found in the video.

IV. Results

A. Simulation

In order to evaluate the validity of the parameter fits (Equations 23 - 24) we run multiple trajectories on a real Husky, with different arm configurations. We record the center of mass by measuring the normal forces on the wheel before each run, using four scales. We fit our simulation results using the Python symfit module (v. 0.5.6) and find a good agreement with a linear fit for d_0 between -0.275 and 0 (cf. Fig. 5) as well as a constant negative slip factor $c = -0.04$ for s_r, s_l. The limits for d_0 correspond to half of the wheelbase, as expected.

B. Tip-over avoidance

Our tip-over avoidance objective, which effectively enforces that no supporting forces drop below a pre-defined threshold, allows the robot to plan paths that keep the robot in (static) balance throughout the trajectory. A result with a highly restrictive lower threshold of 170N per wheel can be seen in Fig. 6. This keeps the robots from assuming unfavourable configurations, like the one that would be a straight-forward solution for the system as the one in the top part of Fig. 6.

Type	Weight	Value	Stiffness (σ)	limit (ε)
System	$\omega_{DCA,inter-agent}$	5	25	0.25
	ω_{EE}	1e5	-	-
Payload	$\omega_{DCA,world}$	200	5	0.05
	ω_{sm}	0.05	-	-
	ω_{tar}	1e4	-	-
Husky	$\omega_{DCA,world}$	200	5	0.25
	$\omega_{DCA,self}$	200	25	0.15
	ω_{steer}	2e5	-	-
	ω_{tip}	10	5	10
	ω_{sm}	0.1	-	-
	ω_{lim}	1e4	25	0.005
$\mathbf{O_L}(\mathbf{q}, \mathbf{f})$	$\omega_{\mathbf{R}}$	0.01	-	-

TABLE I: Objective Weights and Parameters

Trajectory	$\mu_{err,pos}$	$\sigma_{err,pos}$	$\mu_{err,yaw}$	$\sigma_{err,yaw}$
90° rotation off	0.185m	± 0.147m	4.73°	± 1.23°
90° rotation on	**0.1m**	**± 0.071m**	**2.77°**	**± 0.68°**
forward-right off	0.146m	± 0.117m	9.75°	± 24.18°
forward-right on	**0.021m**	**± 0.005m**	**5.36°**	**± 2.9°**
forward 180° off	0.172m	± 0.116m	11.97°	± 10.48°
forward 180° on	**0.048m**	**± 0.039m**	**11.54°**	**± 9.1°**

TABLE II: Results of the force-aware trajectory optimisation (*on*) over planning without load considerations (*off*).

C. Force-aware steering

In order to show the effects of the improved steering objective in isolation, we run multiple trajectories on a single robot, and summarize tracking performance in Table II. An example of the corresponding tracking of the real robot motion compared to the predicted trajectory can be seen in Fig. 7. The *90° rotation* objective defines a target for the base only, while the *forward-right* objective defines only an end-effector target, and *forward 180°* defines separate base and arm goal poses. Especially in scenarios with an extended arm and large rotations, we have a significantly improved open-loop performance compared to the base model. We want to point out that the optimization takes advantage of the coupling between the steering and the arm pose, even when there is no explicit target given for the arm, which can be seen in the accompanying video. It is also visible that, while our system shows a clear improvement, offsets still do occur, e.g., due to uncaptured dynamic effects or small angular offsets in the initial configuration.

D. Multi-robot trajectory optimization

We run multiple optimizations for two and three robot teams, cooperatively carrying a payload from an initial to a goal location, while avoiding obstacles. For each of these experiments, we enable the force-aware steering objective as well as the tip-over limits. A planned trajectory for two robots can be seen in Fig. 8. The trajectory requires the robots to lift a payload over an obstacle, indicated by a red sphere. The starting position of the base and the planned base path is indicated by bright red cuboids, whereas the end-effector path is indicated by small spheres. The graphs show the calculated forces per wheel as well as the change of the d_0 parameter over the trajectory. It can be seen that changes in the parameter d_0 are clearly reflected in the force distribution across the wheels.

A planned trajectory for three robots carrying a payload through a door can be seen in Fig. 9. The collision primitives for the door, robot and payload collision primitives are drawn as transparent red cuboids and capsules. For clarity, the base path is only visualised with small spheres, the same as for the end-effector paths. More real-world trajectories such as the one in Fig. 10 as well as an MR interface visualisation can be found in the accompanying video material.

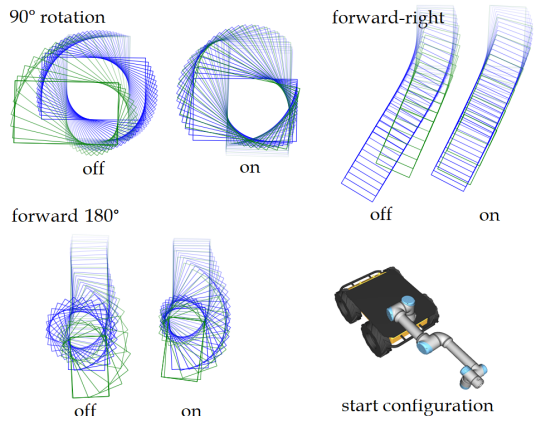

Fig. 7: Trajectory optimization without (left) and with payload-aware steering (right). The planned trajectory is depicted in blue, the motion capture of the real robot in green. Here, the start config. is front-heavy, as shown on the bottom right.

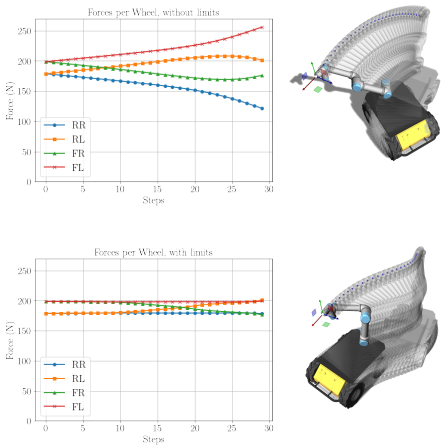

Fig. 6: Comparison of the calculated forces over planned trajectories without (top) and with a 170N lower threshold on the force per wheel (bottom).

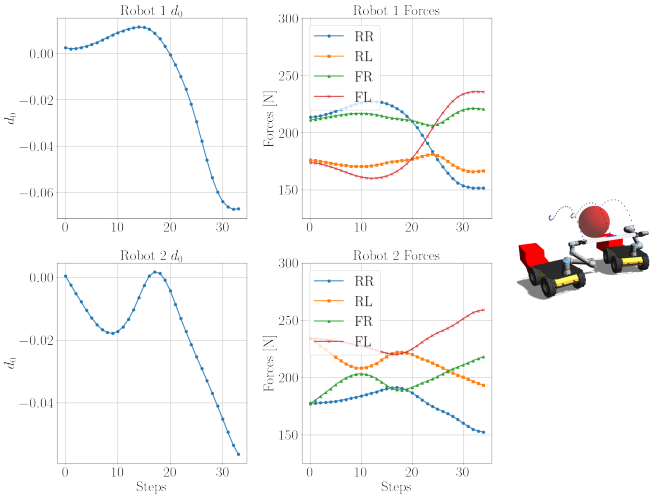

Fig. 8: Values for d_0 and calculated forces per wheel for a two-robot trajectory with obstacle avoidance and a per-wheel threshold of 100N and payload-aware steering enabled.

Objective	ω factor	Min F [N]	SC [mm]	WC [mm]	Max EE [mm]	Avg EE [m]	Target [mm]	Steer [-]
Baseline	-	109.82 -	25 -	179 -	0.31 -	3.35e-5 -	4.4 -	6.1e-3 -
$\mathcal{O}_{tip}$	x10	109.93 ↑	25 -	180 -	0.32 -	3.38e-5 -	4.3 -	6.3e-3 -
	x0.1	109.59 ↓	25 -	178 -	0.34 -	3.27e-5 -	4.2 -	5.6e-3 ↓
$\mathcal{O}_{steer}$	x10	109.52 ↓	25 -	197 ↑	0.38 -	4.44e-5 ↑	4.4 -	6.4e-3 -
	x0.1	109.83 -	25 -	179 -	0.32 -	3.02e-5 ↓	4.2 -	5.2e-2 ↑
$\mathcal{O}_{EE}(\mathbf{q}_i, \mathbf{q}_0)$	x10	109.82 -	25 -	179 -	0.035 ↓	3.3e-6 ↓	4.4 -	6.4e-3 -
	x0.1	109.83 -	25 -	178 -	2.9 ↑	2.5e-4 ↑	3.8 ↓	4.7e-3 ↓
$\mathcal{O}_{EE}(\mathbf{q}_0, p_{tar})$	x10	109.79 -	25 -	182 ↑	0.6 ↑	5.3e-5 ↑	0.7 ↓	9.9e-3 ↑
	x0.1	109.91 ↑	25 -	170 ↓	0.14 ↓	1.5e-5 ↓	21 ↑	1.6e-3 ↓
$\mathcal{O}_{DCA,self}$	x10	109.83 -	25 -	174 ↓	0.35 -	4.5e-5 ↑	4.4 -	6.3e-3 -
	x0.1	109.83 -	23 ↓	183 ↑	0.35 -	3.32e-5 -	4.3 -	6.0e-3 -

TABLE III: Parameter ablation study. Comparison after 100 optimisation steps of minimum supporting force (Min F), minimum self-collision (SC) and world-collision (WC) distances, maximum and average end-effector distance to payload (Max/Avg EE), final distance of payload to target (Target) and steering objective residual (Steer). The baseline uses the parameters described in Tab. I and a 100N lower threshold on $\mathcal{O}_{tip}$, and computes 35 time steps with $dt = 0.2s$.

E. Ablation Study and Computational Efficiency

For the experiments in Section IV-D we tune the weights to the values in Table I. These weights ensure that all constraints are solved to a small residual and collisions are avoided. As is well-known for optimisation-based methods, changes to these weights will influence the overall convergence point of the system. To illustrate some of the trade-offs, we conduct an ablation study for a configuration similar to the one shown in Fig. 8. The threshold force is set to 100N and we calculate 35 steps with $dt = 0.2s$. We change the tip-over, steering, end-effector ($\mathcal{O}_{EE}(\mathbf{q}_i, \mathbf{q}_0)$), payload target ($\mathcal{O}_{EE}(\mathbf{q}_0, p_{tar})$) and self-collision weights by a factor of 10 and 0.1. The results are shown in Tab. III.: The minimal force over all wheels of all robots (Min F) as well as self-collision distance (SC) and world-collision distance of the payload (WC) are always respected. A collision would be indicated by a negative value for SC or WC. With the exception of lowering the weight for $\mathcal{O}_{EE}(\mathbf{q}_0, p_{tar})$, the maximum distance of the end-effectors to their attachment points (Max EE) stays below 1mm, on average (Avg EE) even below 0.1mm. The distance of the payload to the target on the final step (Target) can be lowered by increasing the weight of $\mathcal{O}_{EE}(\mathbf{q}_0, P_{tar})$, however, at the cost of slightly worse convergence of the end-effector objective. Finally, the steering objective residual (Steer) could generally be improved at the cost of a slightly worse $\mathcal{O}_{tip}$, which would however still stay above the defined threshold. On the computational cost: For this scenario, one solver iteration took on average 1470ms $\pm$ 150ms. Computing the gradient and Hessian for one robot takes between 39%-41% of that time. When lowering the number of time steps to 20, the average optimizer step takes 701ms, while increasing to 50 time steps increases that time to 1860ms.

V. Discussion

As shown in Sec. IV-C, especially Fig. 7, the payload-aware steering allows us to achieve an open-loop precision which the implementation without awareness does not. The inclusion of payload-aware steering also allows the optimizer to make use of the arm to actively influence the steering, e.g., to make tighter turns possible, as seen in the *forward 180°* scenario. Our system also allows for the optimisation in more complex scenarios, with obstacles (cf. Fig. 8) and any number of robots cooperating (cf. Fig. 9). One current limitation of the system is that we do not include dynamic effects like wheel acceleration and the associated torque. As outlined in Sec. III-A, we assume that end-effector targets can perfectly be met (which is verified through the results in Tab. III), and therefore don't currently model the force that a robot exerts on the payload, and thereby on other robots. This would certainly be an interesting avenue for future evaluation, as it is clearly visible from experiments on the real robots (cf. Fig. 10) that they do not follow the planned trajectory perfectly, which puts stress on the payload. Furthermore, considering these forces might stabilize the system even when individual robots are in an unstable state. Another future avenue is to now combine

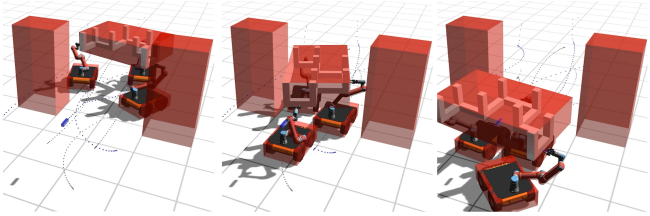

Fig. 9: Results of the trajectory optimization for a system of three Husky robots carrying a frame piece through a narrow passage. The collision primitives are highlighted in red.

Fig. 10: Trajectories on two real Husky robots

our trajectory planner with a feedback-controller, e.g., a force-tracking solution such as presented in [6]. While we evaluated our parameter mapping in simulation (cf. Sec. IV-A), the data does seem to be quite noisy and there are potentially strong dynamic and nonlinear effects which could influence d_0 as well as s_r and s_l (cf. [2]). One way to improve this would be to learn the parameters, e.g., as proposed by Sukhija et al. in [31]. Finally, another interesting future avenue would be the evaluation of the MR user interface, as our goal is to eventually deploy such systems on construction sites to help with installation and transportation tasks.

VI. Conclusion

We have presented a novel method for trajectory optimisation of multiple skid-steering mobile robots, enabling non-trivial trajectory planning for cooperative transport. Our system achieves good open-loop performance by considering forces at the wheels induced by the arm state. We have shown the derivations for our bi-level optimization scheme, and how we can use it to optimize for complex multi-robot systems with a common payload and obstacle avoidance scenarios in simulation. Finally, we presented a real-world implementation complete with a MR user interface, and have conducted multiple experiments on real hardware, as well as an ablation study on our simulation results. For future work, we would like to additionally integrate our method with a closed-loop controller for longer-horizon problems and bring our system closer to real applications in construction and warehousing by studying the MR user interface in more detail.

References

[1] S. Thakar, S. Srinivasan, S. Al-Hussaini, P. M. Bhatt, P. Rajendran, Y. Jung Yoon, N. Dhanaraj, R. K. Malhan, M. Schmid, V. N. Krovi *et al.*, "A survey of wheeled mobile manipulation: A decision-making perspective," *Journal of Mechanisms and Robotics*, vol. 15, no. 2, p. 020801, 2023.

[2] Y. Liu and G. Liu, "Modeling of tracked mobile manipulators with consideration of track–terrain and vehicle–manipulator interactions," *Robotics and Autonomous Systems*, vol. 57, no. 11, pp. 1065–1074, 2009.

[3] R. Tedrake, *Underactuated Robotics*. MIT, 2023. [Online]. Available: https://underactuated.csail.mit.edu

[4] B. V. Adorno, P. Fraisse, and S. Druon, "Dual position control strategies using the cooperative dual task-space framework," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 3955–3960.

[5] R. Laha, M. Becker, J. Vorndamme, J. Vrabel, L. F. Figueredo, M. A. Müller, and S. Haddadin, "Predictive multi-agent based planning and landing controller for reactive dual-arm manipulation," *IEEE Transactions on Robotics*, 2023.

[6] D. Sun, Q. Liao, and A. Loutfi, "Single master bimanual teleoperation system with efficient regulation," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1022–1037, 2020.

[7] D. Baril, V. Grondin, S.-P. Deschênes, J. Laconte, M. Vaidis, V. Kubelka, A. Gallant, P. Giguere, and F. Pomerleau, "Evaluation of skid-steering kinematic models for subarctic environments," in *2020 17th Conference on Computer and Robot Vision (CRV)*. IEEE, 2020, pp. 198–205.

[8] K. Kozłowski and D. Pazderski, "Modeling and control of a 4-wheel skid-steering mobile robot," *International journal of applied mathematics and computer science*, vol. 14, no. 4, pp. 477–496, 2004.

[9] A. Mandow, J. L. Martinez, J. Morales, J. L. Blanco, A. Garcia-Cerezo, and J. Gonzalez, "Experimental kinematics for wheeled skid-steer mobile robots," in *2007 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2007, pp. 1222–1227.

[10] J. L. Martínez, J. Morales, A. Mandow, S. Pedraza, and A. García-Cerezo, "Inertia-based icr kinematic model for tracked skid-steer robots," in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, 2017, pp. 166–171.

[11] M. Giftthaler, F. Farshidian, T. Sandy, L. Stadelmann, and J. Buchli, "Efficient kinematic planning for mobile manipulators with non-holonomic constraints using optimal control," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3411–3417.

[12] C. Ordonez, N. Gupta, B. Reese, N. Seegmiller, A. Kelly, and E. G. Collins Jr, "Learning of skid-steered kinematic and dynamic models for motion planning," *Robotics and Autonomous Systems*, vol. 95, pp. 207–221, 2017.

[13] X. Ding, Y. Liu, J. Hou, and Q. Ma, "Online dynamic tip-over avoidance for a wheeled mobile manipulator with an improved tip-over moment stability criterion," *IEEE Access*, vol. 7, pp. 67 632–67 645, 2019.

[14] J. Pankert and M. Hutter, "Perceptive model predictive control for continuous mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6177–6184, 2020.

[15] G. R. Petrović and J. Mattila, "Analytic solutions for wheeled mobile manipulator supporting forces," *IEEE Access*, vol. 10, pp. 43 235–43 255, 2022.

[16] E. Tuci, M. H. Alkilabi, and O. Akanyeti, "Cooperative object transport in multi-robot systems: A review of the state-of-the-art," *Frontiers in Robotics and AI*, vol. 5, p. 59, 2018.

[17] H. Farivarnejad and S. Berman, "Multirobot control strategies for collective transport," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 205–219, 2022.

[18] B. Hichri, L. Adouane, J.-C. Fauroux, Y. Mezouar, and I. Doroftei, "Cooperative mobile robot control architecture for lifting and transportation of any shape payload," in *Distributed Autonomous Robotic Systems: The 12th International Symposium*. Springer, 2016, pp. 177–191.

[19] X. Zhao, Z. Zhao, S. Zhang, and C. Su, "Stability analysis of wheeled mobile multi-robot coordinated towing system," *Journal of Mechanical Science and Technology*, vol. 36, no. 1, pp. 407–416, 2022.

[20] M. Dogar, R. A. Knepper, A. Spielberg, C. Choi, H. I. Christensen, and D. Rus, "Multi-scale assembly with robot teams," *The International Journal of Robotics Research*, vol. 34, no. 13, pp. 1645–1659, 2015.

[21] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control and object transport in dynamic environments via constrained optimization," *The International Journal of Robotics Research*, vol. 36, no. 9, pp. 1000–1021, 2017.

[22] A. Nikou, C. Verginis, S. Heshmati-Alamdari, and D. V. Dimarogonas, "A nonlinear model predictive control scheme for cooperative manipulation with singularity and collision avoidance," in *2017 25th Mediterranean Conference on Control and Automation (MED)*. IEEE, 2017, pp. 707–712.

[23] H. J. Savino, L. C. Pimenta, J. A. Shah, and B. V. Adorno, "Pose consensus based on dual quaternion algebra with application to decentralized formation control of mobile manipulators," *Journal of the Franklin Institute*, vol. 357, no. 1, pp. 142–178, 2020.

[24] T. Recker, M. Heinrich, and A. Raatz, "A hybrid control approach on handling orientation constraints and tracking errors in formation control for multiple nonholonomic mobile manipulators," in *2021 20th International Conference on Advanced Robotics (ICAR)*. IEEE, 2021, pp. 891–896.

[25] P. Xu, J. Zheng, J. Zhang, K. Zhang, Y. Cui, and Q. Tang, "Distributed position-force control for cooperative transportation with multiple mobile manipulators," in *Advances in Swarm Intelligence: 12th International Conference, ICSI 2021, Qingdao, China, July 17–21, 2021, Proceedings, Part II 12*. Springer, 2021, pp. 111–118.

[26] H. Zhang, H. Song, W. Liu, X. Sheng, Z. Xiong, and X. Zhu, "Hierarchical motion planning framework for cooperative transportation of multiple mobile manipulators," *arXiv preprint arXiv:2208.08054*, 2022.

[27] Clearpath, "husky GitHub repository," GitHub, accessed: 2024-02-07. [Online]. Available: https://github.com/husky/husky

[28] D. Pardo, L. Möller, M. Neunert, A. W. Winkler, and J. Buchli, "Evaluating direct transcription and nonlinear optimization methods for robot motion planning," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 946–953, 2016.

[29] S. Zimmermann, G. Hakimifard, M. Zamora, R. Poranne, and S. Coros, "A multi-level optimization framework for simultaneous grasping and motion planning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2966–2972, 2020.

[30] R. H. Jackson and G. P. Mccormick, "Second-order sensitivity analysis in factorable programming: Theory and applications," *Mathematical Programming*, vol. 41, pp. 1–27, 1988.

[31] B. Sukhija, L. Treven, C. Sancaktar, S. Blaes, S. Coros, and A. Krause, "Optimistic active exploration of dynamical systems," in *Proc. Neural Information Processing Systems (NeurIPS)*, December 2023.