Mobile Manipulation in Unknown Environments with Differential Inverse Kinematics Control

Adam Heins, Michael Jakob, and Angela P. Schoellig

***Abstract*—Mobile manipulators combine the large workspace of mobile robots with the interactive capabilities of manipulator arms, making them useful in a variety of domains including construction and assistive care. We propose a differential inverse kinematics whole-body control approach for position-controlled industrial mobile manipulators. Our controller is capable of task-space trajectory tracking, force regulation, obstacle and singularity avoidance, and pushing an object toward a goal location, with limited sensing and knowledge of the environment. We evaluate the proposed approach through extensive experiments on a 9 degree-of-freedom omnidirectional mobile manipulator. A video demonstrating many of the experiments can be found at http://tiny.cc/crv21-mm.**

I. Introduction

Having been in widespread use in industrial automation for decades, manipulators are becoming more common in areas like healthcare [1] and assistive robotics [2], where they are expected to perform complex, interactive tasks with incomplete knowledge of the environment. Mobile manipulators are particularly versatile due to the extended workspace provided by a mobile base, allowing them to efficiently perform tasks such as large-scale 3D printing [3]. Such systems are becoming increasingly diverse, with two-legged [4], four-legged [5], and aerial [6] varieties in addition to the more typical wheeled configurations. The redundancy of mobile manipulators allows them to accomplish multiple tasks simultaneously, such as avoiding obstacles with the base while tracking a task-space trajectory with the end effector (EE).

In this work, we present a whole-body control approach for our position-controlled industrial mobile manipulator based on differential inverse kinematics. In contrast to motion planning approaches which often have distinct, sequential planning and execution phases, operating at the control level allows us to react extremely quickly to sensor feedback, such as applied force. In extensive hardware experiments, we demonstrate accurate task-space trajectory tracking, force control, and obstacle and singularity avoidance. We show that our control approach can accomplish various useful tasks, including wiping a surface and pushing an object, based on limited sensor feedback and knowledge of the environment. Figure 1 shows our robot engaging in some of the tasks described herein. The goal of this work is to explore the capabilities of our mobile manipulator and to provide a foundation of basic control behaviours upon which more complex behaviours can be built.

The authors are with the Dynamic Systems Lab (www.dynsyslab.org) at the University of Toronto Institute for Aerospace Studies (UTIAS), Toronto, Canada, and the Vector Institute for Artificial Intelligence, Toronto, Canada. {adam.heins, michael.jakob, angela.schoellig}@robotics.utias.utoronto.ca

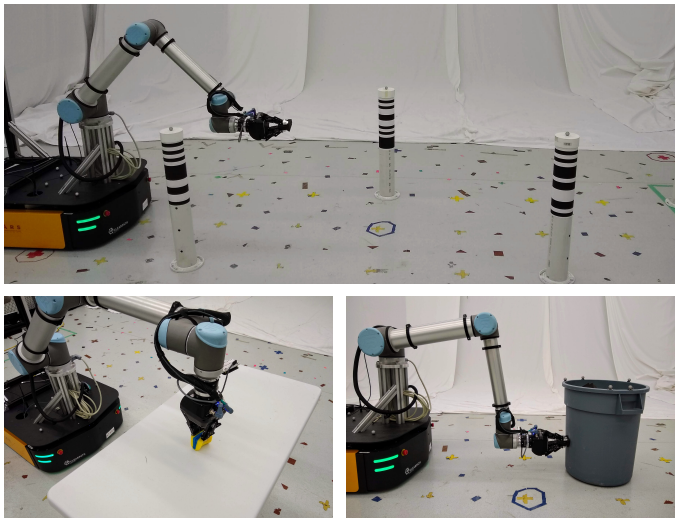

Figure 1: Examples of our mobile manipulator performing tasks using our controller: obstacle avoidance (top), wiping a table (bottom left), and pushing an object (bottom right). A video of the experiments can be found at `http://tiny.cc/crv21-mm`.

A. Related Work

A central goal of our control approach is to solve the inverse kinematics (IK) problem, which is the problem of determining the joint-space configuration of a robot corresponding to a particular task-space pose of the end effector. In control it is typical to solve the IK problem on the differential (i.e. velocity) level, exploiting the linear relationship between the joint-space and task-space velocities. For redundant systems like our mobile manipulator, where there are a greater number of input degrees-of-freedom (DOFs) than the required output DOFs for a given task, the solution can be found analytically using least squares [7]. More sophisticated redundancy resolution methods include the Extended Gradient, Projected Gradient, and Reduced Gradient [8]. However, these methods do not handle inequality constraints, which may include limits on the joint positions and their derivatives [9] or more complex constraints such as obstacle avoidance and field-of-view constraints [10]. Inequality constraints can be introduced by solving the differential IK problem as a general quadratic program (QP) [11], which is the approach we take in this work.

Another method for resolving redundancy is to define redundancy parameters that describe the geometrical relation between the arm and mobile base, the values of which are found by solving an optimization problem online [12], [13]. The advantage of redundancy parameters is that they provide an interpretable notion of the system's redundancy; however,

they may be difficult or impossible to define for certain robot geometries [10].

The redundancy of a system can be exploited to complete multiple tasks simultaneously. In [14] and [15], a task-priority control framework is presented which organizes tasks in a hierarchy. Higher-priority tasks are guaranteed to be satisfied before lower-priority tasks by projecting the latter into the null-space of the former. In [16], this framework is extended to handle inequality constraints. Another approach for handling inequality constraints within a task-priority framework is to use hierarchical quadratic programming [17], [18]. The idea of this approach is that each task is represented as a QP, and the solutions of lower-priority QPs are constrained to lie in the solution set of the higher-priority ones. In this work, we employ the method of [17] to include a singularity-avoidance objective alongside the controller's primary task.

Recent literature on the control of position-controlled mobile manipulators includes [19], which develops a controller for moving to target Cartesian poses with the EE subject to the limits on the joint velocities. Other work has used model predictive control (MPC). The joint-space formulation of [10] results in a quadratic optimization objective, allowing the problem to be solved very efficiently. However, as noted by the authors, accomplishing task-space objectives would require the IK problem to be solved separately. In contrast, [20] handles task-space objectives and constraints directly in a nonlinear MPC problem, which is solved efficiently through the use of sequential linear quadratic (SLQ) MPC [21]. Inspired by the comprehensive nature of the work in [20], our approach is able to solve many similar tasks, including task-space trajectory tracking and admittance control. However, in contrast, we focus on a simpler and less computationally demanding controller formulation based on differential IK rather than MPC. We also demonstrate different objectives, including singularity-avoidance and a pushing task.

B. Contributions

Our hope is that this work provides a stimulating look at the wide range of behaviours a mobile manipulator can achieve, despite limited sensor feedback and knowledge of the environment. The primary contributions of this paper are:

1) a whole-body differential IK control approach for position-controlled industrial mobile manipulators capable of handling task-space trajectory tracking, force control, obstacle and singularity avoidance, and pushing behaviour;
2) extensive hardware experiments on our mobile manipulation platform demonstrating the capabilities of the controller and the system as a whole.

C. Notation

We denote vectors and matrices in boldface. The notation $\boldsymbol{X} > 0$ or $\boldsymbol{X} \geq 0$ indicates that the matrix $\boldsymbol{X}$ is positive definite or positive semidefinite, respectively. Unit vectors are denoted with a caret $\hat{(\cdot)}$, $\boldsymbol{I}_n$ denotes the $n \times n$ identity matrix, and $\|\boldsymbol{x}\|$ denotes the Euclidean norm of $\boldsymbol{x}$, with $\|\boldsymbol{x}\|_{\boldsymbol{W}} = \sqrt{\boldsymbol{x}^T \boldsymbol{W} \boldsymbol{x}}$ denoting the norm weighted by $\boldsymbol{W} \geq 0$.

II. System Model

Our system is a position-controlled industrial mobile manipulator with a 6 DOF robotic arm mounted on a 3 DOF omnidirectional mobile base. We control the robot as single $n = 9$ DOF system with generalized joint configuration $\boldsymbol{q} = [\boldsymbol{q}_b^T, \boldsymbol{q}_a^T]^T$, where $\boldsymbol{q}_b = [x_b, y_b, \theta_b]^T$ represents the position and yaw angle of the base with respect to the world and $\boldsymbol{q}_a = [\theta_1, \theta_2, \ldots, \theta_6]^T$ represents the joint angles of the arm. The input to the system is the vector of joint velocities $\boldsymbol{u} = [\boldsymbol{u}_b^T, \boldsymbol{u}_a^T]^T$. The discretized motion model of the system is

$$\boldsymbol{q}_{t+1} = \boldsymbol{q}_t + \Delta t \boldsymbol{B}(\boldsymbol{q}_t)\boldsymbol{u}_t,$$

where the subscript $(\cdot)_t$ denotes the value at the current time, Δt is the controller timestep, and $\boldsymbol{B}(\boldsymbol{q})$ maps the inputs $\boldsymbol{u}$ to the generalized velocities $\dot{\boldsymbol{q}}$:

$$\dot{\boldsymbol{q}} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{u}. \tag{1}$$

For our system,

$$\boldsymbol{B}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{B}_b(\boldsymbol{q}_b) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_6 \end{bmatrix},$$

where

$$\boldsymbol{B}_b(\boldsymbol{q}_b) = \begin{bmatrix} \cos\theta_b & -\sin\theta_b & 0 \\ \sin\theta_b & \cos\theta_b & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In the remainder of the paper, we will only explicitly use the subscript $(\cdot)_t$ when we are not referring to the current timestep.

The mobile base is equipped with a two-dimensional laser range finder which provides range and bearing measurements to obstacles within an arc in front of the base. The EE is equipped with a force-torque sensor that measures the applied wrench $\boldsymbol{w}_{ee} = [\boldsymbol{f}_{ee}^T, \boldsymbol{\tau}_{ee}^T]^T$, where $\boldsymbol{f}_{ee}, \boldsymbol{\tau}_{ee} \in \mathbb{R}^3$ are the force and torque, respectively.

III. Methodology

We use a differential inverse kinematics approach for our controller, which relies on the linear mapping between joint and Cartesian velocities,

$$\boldsymbol{V}_{ee} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{2}$$

where $\boldsymbol{V}_{ee} = [\boldsymbol{v}_{ee}^T, \boldsymbol{\omega}_{ee}^T]^T$ is the spatial twist of the end effector, with linear velocity $\boldsymbol{v}_{ee} \in \mathbb{R}^3$ and angular velocity $\boldsymbol{\omega}_{ee} \in \mathbb{R}^3$, and $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{6\times 9}$ is the geometric Jacobian of the robot. Substituting (1) into (2), the control inputs $\boldsymbol{u}$ required to achieve a given EE reference twist $\boldsymbol{V}_{\text{ref}}$ must satisfy

$$\boldsymbol{J}\boldsymbol{B}\boldsymbol{u} = \boldsymbol{V}_{\text{ref}}, \tag{3}$$

where we have dropped the dependence of $\boldsymbol{J}$ and $\boldsymbol{B}$ on $\boldsymbol{q}$ for brevity. Since our system is redundant, there are an infinite number of solutions to (3). We follow the standard approach of solving a QP to generate the optimal inputs at each timestep,

$$\begin{aligned} \boldsymbol{u}_t = \underset{\boldsymbol{u}}{\text{argmin}} \quad & \|\boldsymbol{J}\boldsymbol{B}\boldsymbol{u} - \boldsymbol{V}_{\text{ref}}\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}\|_{\boldsymbol{R}}^2 \\ \text{s.t.} \quad & \boldsymbol{u}_{\min} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\max}, \end{aligned} \tag{4}$$

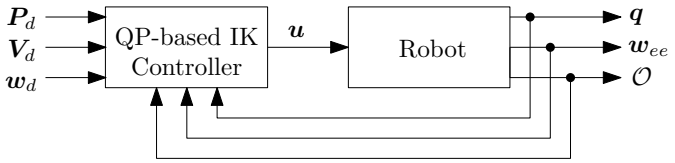

Figure 2: Block diagram of our differential inverse kinematics control approach. Our controller generates joint velocity commands $\boldsymbol{u}$ by solving a QP. Input to the controller is the desired EE pose $\boldsymbol{P}_d$, twist $\boldsymbol{V}_d$, and wrench $\boldsymbol{w}_d$, as well as the actual joint positions $\boldsymbol{q}$, wrench $\boldsymbol{w}_{ee}$, and positions of detected obstacles $\mathcal{O}$.

where $\boldsymbol{Q} \geq \boldsymbol{0}$ and $\boldsymbol{R} > \boldsymbol{0}$ are weight matrices, and $\boldsymbol{u}_{\min}$ and $\boldsymbol{u}_{\max}$ are bounds on the joint velocities. The first term of the objective function of (4) prefers solutions that satisfy (3). The second term serves as a regularizer and prefers solutions with smaller Euclidean norm.

In subsequent sections we develop expressions for $\boldsymbol{V}_{\text{ref}}$ to accomplish EE pose trajectory tracking, force regulation, and an object pushing task. We also discuss the addition of an obstacle avoidance constraint and an additional objective to avoid singular configurations of the robot arm. Figure 2 summarizes our approach in a block diagram.

A. Position and Orientation Tracking

Our first task is to track a desired Cartesian trajectory with the EE. We compute a reference twist using the feedback law

$$\boldsymbol{V}_{\text{ref}}^{\text{pose}} = \boldsymbol{K}_p \Delta \boldsymbol{P} + \boldsymbol{V}_d, \tag{5}$$

where $\boldsymbol{K}_p \geq \boldsymbol{0}$ is a gain matrix, $\Delta \boldsymbol{P}$ is the error between the desired and current EE pose, and $\boldsymbol{V}_d$ is the feedforward desired twist. We express the pose error as $\Delta \boldsymbol{P} = [\Delta \boldsymbol{p}^T, \Delta \boldsymbol{\epsilon}^T]^T$, where $\Delta \boldsymbol{p} = \boldsymbol{p}_d - \boldsymbol{p}_{ee}$ is the position error and $\Delta \boldsymbol{\epsilon}$ is the orientation error, calculated as follows. Let $\mathcal{Q}_d$ and $\mathcal{Q}_{ee}$ represent the desired and actual EE orientation quaternions, respectively. The orientation error $\Delta \boldsymbol{\epsilon}$ is the vector part of the error quaternion $\Delta \mathcal{Q} = \mathcal{Q}_d \star \mathcal{Q}_{ee}^{-1}$, where $\star$ denotes quaternion multiplication.

In contrast to approaches like [11] that formulate a QP with pose tracking as an equality constraint, we prefer the flexible form (4) that allows the controller to deviate from the desired trajectory to avoid violating other constraints. For example, safety constraints such as obstacle avoidance may require the robot to deviate from the desired trajectory to avoid collision.

B. Admittance Control

Admittance control allows the robot to alter its motion based on the force and torque applied at the EE [22]. We wish to make the EE act like the spring-damper

$$\Delta \boldsymbol{V} + \boldsymbol{K}_p \Delta \boldsymbol{P} + \boldsymbol{K}_f \Delta \boldsymbol{w} = \boldsymbol{0}, \tag{6}$$

where $\Delta \boldsymbol{V} = \boldsymbol{V}_d - \boldsymbol{V}_{ee}$, $\boldsymbol{K}_f \geq \boldsymbol{0}$ is a gain matrix, and $\Delta \boldsymbol{w} = \boldsymbol{w}_d - \boldsymbol{w}_{ee}$. In order to achieve (6), the EE must follow the reference twist

$$\boldsymbol{V}_{\text{ref}}^{\text{ad}} = \boldsymbol{V}_d + \boldsymbol{K}_p \Delta \boldsymbol{P} + \boldsymbol{K}_f \Delta \boldsymbol{w}, \tag{7}$$

which is simply the pose tracking reference (5) with the addition of a wrench error term. A pose tracking controller that ignores applied forces is recovered when $\boldsymbol{K}_f = \boldsymbol{0}$. Alternatively, a controller that ignores pose errors and only reacts to force is obtained when $\boldsymbol{K}_p = \boldsymbol{0}$. Finally, force control can be performed in certain Cartesian directions while pose tracking is performed in others by making only one of the corresponding rows in $\boldsymbol{K}_p$ and $\boldsymbol{K}_f$ non-zero.

This admittance law allows the robot to accomplish tasks where force must be regulated in the direction normal to a surface while a trajectory is tracked in the tangential directions, such as drawing or wiping a table. The law also allows the robot to change its motion in response to applied force, so that it can stop in the case of an unexpected collision or be manually moved by a human user.

C. Obstacle Avoidance

The mobile base is equipped with a two-dimensional laser range finder that provides range and bearing measurements to objects in an arc in front of the robot. For safety, we want our controller to avoid collisions between the base and surrounding obstacles. To do so, we model the base as a circle centered at $\boldsymbol{p}_b = [x_b, y_b]^T$ with radius r_b. For each obstacle o in the set of detected obstacles $\mathcal{O}$, we calculate

$$\hat{\boldsymbol{n}}_o = \frac{\boldsymbol{p}_o - \boldsymbol{p}_b}{\|\boldsymbol{p}_o - \boldsymbol{p}_b\|},$$
$$d_o = \|\boldsymbol{p}_o - \boldsymbol{p}_b\| - r_o,$$

where $\boldsymbol{p}_o$ is the position of object o in the world frame. Following the velocity damper approach originally introduced in [23], if the distance to the obstacle d_o is less than a specified influence distance d_i, we apply the constraint

$$\hat{\boldsymbol{n}}_o^T \dot{\boldsymbol{p}}_b \leq \xi \frac{d_o - d_s}{d_i - d_s}, \tag{8}$$

where $d_s > 0$ is a safety distance that cannot be crossed and $\xi > 0$ is a tunable scaling factor. Intuitively, this constraint limits the allowable base velocity in the direction of the obstacle $\hat{\boldsymbol{n}}_o$. If the distance d_o is reduced to the safety distance d_s, then the base velocity is constrained to be less than or equal to zero, so the base cannot move toward the obstacle at all.

Since $\dot{\boldsymbol{p}}_b$ simply consists of the top two elements of $\dot{\boldsymbol{q}}$, we can relate it to the control input $\boldsymbol{u}$ using

$$\dot{\boldsymbol{p}}_b = \begin{bmatrix} \boldsymbol{I}_2 & \boldsymbol{0} \end{bmatrix} \dot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{I}_2 & \boldsymbol{0} \end{bmatrix} \boldsymbol{B} \boldsymbol{u},$$

which facilitates adding the constraint (8) to our QP (4).

D. Manipulability

The manipulability index (MI) [24], defined as

$$m(\boldsymbol{q}) = \sqrt{\det(\boldsymbol{J}\boldsymbol{J}^T)}, \tag{9}$$

quantifies the distance of the manipulator to a singular configuration. Like [25], we seek to incorporate an objective to maximize the manipulability into our controller to avoid singular configurations of the arm. Configurations farther from

singularity also require lower joint velocities to achieve a given EE twist. A first-order Taylor series expansion of $m(\boldsymbol{q})$ yields

$$m(\boldsymbol{q} + \Delta\boldsymbol{q}) \approx m(\boldsymbol{q}) + \nabla m^T \Delta\boldsymbol{q}, \tag{10}$$

where ∇m is the gradient of $m(\boldsymbol{q})$. In contrast to [25], in which

$$\nabla m = \left[\frac{\partial m}{\partial q_1}, \ldots, \frac{\partial m}{\partial q_n}\right]^T,$$

is calculated using finite differences, we follow a similar approach to [26] and compute it analytically by evaluating

$$\frac{\partial m}{\partial q_i} = m(\boldsymbol{q}) \operatorname{tr}\left((\boldsymbol{J}\boldsymbol{J}^T)^{-1}\boldsymbol{J}\left(\frac{\partial \boldsymbol{J}}{\partial q_i}\right)^T\right),$$

where $\operatorname{tr}(\cdot)$ denotes the matrix trace, which avoids approximation and which we found to be computationally faster. Note that we only use the 6×6 block of the Jacobian matrix corresponding to the arm joints in the above calculation. Substituting $\Delta\boldsymbol{q} = \Delta t \boldsymbol{B}\boldsymbol{u}$ into (10), we can maximize (9) by adding the term

$$-w\Delta t \nabla m^T \boldsymbol{B}\boldsymbol{u} \tag{11}$$

to the objective of the QP (4), where $w \geq 0$ is a scalar weight and the negative sign results in the value being maximized.

However, this approach causes a conflict with our primary objective to achieve $\boldsymbol{V}_{\text{ref}}$. Alternatively, instead of adding (11) to the objective of (4), we can employ the task-priority approach of [17] and solve a second QP at each timestep, which is constrained to lie in the solution set of the first:

$$\begin{aligned} \boldsymbol{u}_t = \underset{\boldsymbol{u}}{\operatorname{argmin}} \quad & -w\Delta t \nabla m^T \boldsymbol{B}\boldsymbol{u} + \|\boldsymbol{u}\|^2_{\boldsymbol{R}} \\ \text{s.t.} \quad & \boldsymbol{J}\boldsymbol{B}\boldsymbol{u} = \boldsymbol{J}\boldsymbol{B}\boldsymbol{u}^\star \\ & \boldsymbol{u}_{\min} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\max}, \end{aligned} \tag{12}$$

where $\boldsymbol{u}^\star$ is the solution to the first QP (4). The constraints from (4) are included, in addition to the new equality constraint $\boldsymbol{J}\boldsymbol{B}\boldsymbol{u} = \boldsymbol{J}\boldsymbol{B}\boldsymbol{u}^\star$, which ensures that the primary task (3) is still optimally satisfied. For simplicity we include the same regularization weight $\boldsymbol{R}$ in (12) as in (4), but in general these need not be the same.

Subsequently, we will refer to the first approach as the Objective approach, because (11) is added directly to the objective function of (4). We will refer to the second approach as the Constraint approach, as the optimal solution of (11) is constrained to lie in the solution space of (4). The Constraint approach is intuitively attractive because it does not interfere with the primary task (3). In experiments presented below, we confirm that the Constraint approach improves the MI without reducing the pose tracking performance, while the Objective approach reduces tracking performance as w increases.

E. Pushing

There is a rich literature on robot pushing (see [27] for a recent survey), much of which is focused on modelling the object being pushed. The model can then be used to plan motions for the robot that push the object to a desired position. Instead, we propose a simple heuristic pushing control law that is easily incorporated into our differential IK QP formulation (4). We do not model or even localize the object being pushed, but rather generate motions based only on the sensed force at the end effector and the target position. We assume a single point of contact at the EE. Working in the x-y plane, let $\Delta\boldsymbol{p}_{xy} = \boldsymbol{S}_{xy}\Delta\boldsymbol{p}$ and $\boldsymbol{f}_{xy} = \boldsymbol{S}_{xy}\boldsymbol{f}_{ee}$, where

$$\boldsymbol{S}_{xy} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

We define

$$\hat{\boldsymbol{n}}_p = \frac{\Delta\boldsymbol{p}_{xy}}{\|\Delta\boldsymbol{p}_{xy}\|}$$

and

$$\hat{\boldsymbol{n}}_f = \begin{cases} \boldsymbol{f}_{xy}/\|\boldsymbol{f}_{xy}\| & \text{if } \|\boldsymbol{f}_{xy}\| \geq f_T, \\ \hat{\boldsymbol{n}}_{f,t-1} & \text{else,} \end{cases} \tag{13}$$

where $f_T \geq 0$ is a force threshold that prevents $\hat{\boldsymbol{n}}_f$ from being updated unless a sufficiently large force is present, which serves to reject noise.

In a pushing scenario, force is the result of contact with the pushed object, so $\hat{\boldsymbol{n}}_f$ represents a contact direction. The controller must balance between moving in the direction $\hat{\boldsymbol{n}}_f$ to stay in contact with the object and moving in the direction $\hat{\boldsymbol{n}}_p$ to reach the goal location. With reference to Figure 3, if the EE and object are aligned so that $\hat{\boldsymbol{n}}_f = \hat{\boldsymbol{n}}_p$, then moving in that direction pushes the object toward the goal. If they are not aligned, then the EE should actually move in a direction away from $\hat{\boldsymbol{n}}_p$ to try to turn the object toward the target position. We calculate the EE direction of motion using

$$\hat{\boldsymbol{n}}_{\text{push}} = \operatorname{slerp}(\hat{\boldsymbol{n}}_f, \hat{\boldsymbol{n}}_p, \alpha_{\text{push}}), \tag{14}$$

where slerp is the spherical linear interpolation function [28], defined as

$$\operatorname{slerp}(\hat{\boldsymbol{a}}, \hat{\boldsymbol{b}}, \alpha) = \frac{\sin((1-\alpha)\theta)}{\sin\theta}\hat{\boldsymbol{a}} + \frac{\sin(\alpha\theta)}{\sin\theta}\hat{\boldsymbol{b}},$$

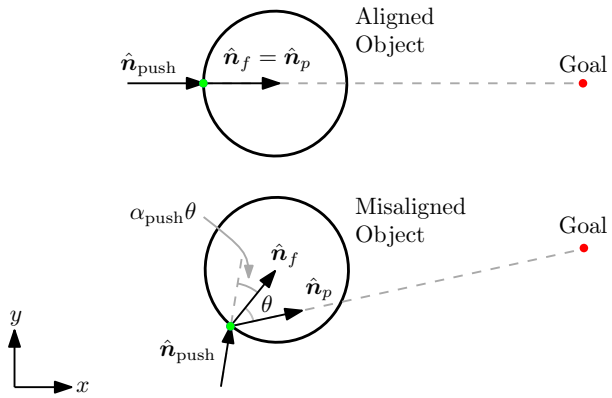

Figure 3: Top view of two scenarios in which the EE is in contact with a round object (at the green point) and trying to move to the goal position (red point). On the top, the object and EE are aligned so that $\hat{\boldsymbol{n}}_f = \hat{\boldsymbol{n}}_p$, and pushing in that direction will successfully move toward the goal. On the bottom, the object is misaligned. If the EE just pushes along $\hat{\boldsymbol{n}}_p$ toward the goal location, the object would slip aside and contact would be lost. Instead, the EE pushes in direction $\hat{\boldsymbol{n}}_{\text{push}}$ in order to turn the object toward the goal without losing contact.

Figure 4: The mobile manipulator used in the experiments, shown in the configuration used at the start of trajectory executions. For reference, the base is approximately $1.0\,\mathrm{m} \times 0.8\,\mathrm{m}$ in size.

for unit vectors $\hat{\boldsymbol{a}}$ and $\hat{\boldsymbol{b}}$, where θ is the angle between them and α is the interpolation parameter.

Somewhat unusually, we use a negative interpolation parameter, $-1 \leq \alpha_{\text{push}} \leq 0$, because we want to move *away* from the direction $\hat{\boldsymbol{n}}_p$ to turn the object toward the goal position. The angle between $\hat{\boldsymbol{n}}_{\text{push}}$ and $\hat{\boldsymbol{n}}_f$ is proportional to the angle between $\hat{\boldsymbol{n}}_f$ and $\hat{\boldsymbol{n}}_p$, achieving the desired behaviour.

We found that the controller was more reliable if (13) was modified to

$$\hat{\boldsymbol{n}}_f = \begin{cases} \boldsymbol{f}_{xy}/\|\boldsymbol{f}_{xy}\| & \text{if } \|\boldsymbol{f}_{xy}\| \geq f_T, \\ \text{slerp}(\hat{\boldsymbol{n}}_{f,t-1}, \hat{\boldsymbol{n}}_p, \alpha_f) & \text{else,} \end{cases}$$

so that in the absence of meaningful force measurements, $\hat{\boldsymbol{n}}_f$ rotates toward the direction of the target position $\hat{\boldsymbol{n}}_p$, where $0 \leq \alpha_f \leq 1$ controls the rate of convergence. This way, if contact with the object is lost, the EE will start moving back toward the target position until contact is made again.

The twist reference supplied to (4) for the EE to push an object is

$$\boldsymbol{V}_{\text{ref}}^{\text{push}} = \begin{bmatrix} \boldsymbol{I}_2 \\ \boldsymbol{0} \end{bmatrix} v_{\text{push}} \hat{\boldsymbol{n}}_{\text{push}} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_4 \end{bmatrix} \boldsymbol{K}_p \Delta \boldsymbol{P}, \qquad (15)$$

where the direction $\hat{\boldsymbol{n}}_{\text{push}}$ is followed in the x-y plane, and a simple proportional law is used in the other Cartesian directions. The speed in the pushing direction is controlled by v_{push}. In our experiments we simply set this value to a constant, but more sophisticated methods may scale the value based on the distance to the goal location or the angle between $\hat{\boldsymbol{n}}_f$ and $\hat{\boldsymbol{n}}_p$.

IV. Experiments

We perform numerous experiments to validate our approach on our physical mobile manipulator, shown in Figure 4. We begin by describing our experimental setup, then present results for pose tracking, admittance control, obstacle avoidance, manipulability maximization, and pushing. A video demonstrating many of the experiments can be found at `http://tiny.cc/crv21-mm`.

A. Experimental Setup

The mobile manipulator used for experiments consists of a 6 DOF UR10 industrial manipulator mounted on a 3 DOF omnidirectional Ridgeback mobile base. The pose of the mobile base is measured using a Vicon indoor motion capture system and the arm's joint angles are measured using its internal joint encoders. A Robotiq FT 300 force torque sensor at the EE wrist measures the applied wrench. A Hokuyo UST-10LX laser range finder is mounted at the front of the base, which we use to detect obstacles in a 180° arc in front of the robot, with a nominal detection range between 0.06 and $10\,\mathrm{m}$.

Our proposed controller is implemented using ROS and the QP solver qpOASES [29]. It is run on a computer with four Intel Core i7 CPUs at $2.5\,\mathrm{GHz}$ and $8\,\mathrm{GB}$ of RAM. The control rate is $125\,\mathrm{Hz}$, the rate at which robot accepts input commands. In all experiments, we set the regularization weight to $\boldsymbol{R} = \boldsymbol{I}_9$.

Table I: RMS error of end effector position (in mm) and orientation (in rad) for the seven trajectories with and without orientation tracking. Each value is the average of three runs.

	Position-only		Full pose	
Trajectory	Position	Orientation	Position	Orientation
Line	1.91	0.16	1.84	0.0010
Sine	5.58	0.35	5.60	0.0036
Figure 8	3.76	0.23	4.51	0.0030
Square	4.81	0.41	5.24	0.0023
Ellipse	3.07	0.27	3.07	0.0020
Spiral	4.59	0.22	5.56	0.0032
Rotate	—	—	3.27	0.0074

B. Position and Orientation Tracking

We begin by demonstrating our controller's accuracy when tracking a desired EE position and orientation in free space, using the pose tracking reference (5). We test on seven different trajectories, each with a duration of $30\,\mathrm{s}$. The six shown in Figure 5 specify a constant orientation, with maximum linear velocities between $0.13\,\mathrm{m/s}$ and $0.44\,\mathrm{m/s}$. The seventh trajectory (Rotate) maintains a constant position while rotating the EE 90° about the z-axis, followed by 90° about the x-axis (in the world frame).

We run two sets of experiments: once with only position tracking, and once with both position and orientation tracking. The exception is the Rotate trajectory, which only produces movement if orientation tracking is performed. To track position only and ignore orientation we set the

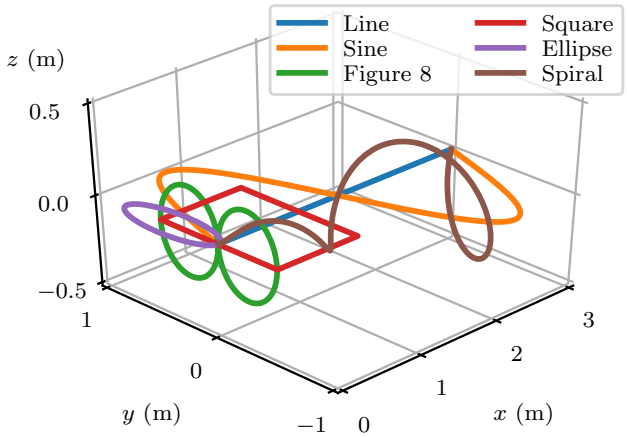

Figure 5: Six end effector paths used to test pose tracking. Each trajectory has a duration of 30 seconds and a maximum linear speed between $0.13\,\mathrm{m/s}$ and $0.44\,\mathrm{m/s}$.

Table II: Control effort for six trajectories when only position is tracked. Trajectories are executed using the full combined system, only the arm, and only the base. Each value is the average of three runs.

Trajectory	Combined	Arm-only	Base-only
Line	0.095	—	0.099
Sine	0.118	—	0.132
Figure 8	0.063	0.097	—
Square	0.091	—	0.103
Ellipse	0.039	0.095	0.045
Spiral	0.115	—	—

weight $\boldsymbol{Q} = 100\,\mathrm{diag}(1,1,1,0,0,0)$; to track the full pose, we use $\boldsymbol{Q} = 100\boldsymbol{I}_6$. In both cases, we set $\boldsymbol{K}_p = \boldsymbol{I}_6$.

The root-mean-square (RMS) pose tracking errors are shown in Table I. When only the position is tracked, the controller deviates significantly from the desired orientation. When orientation tracking is added, the orientation error is reduced by about two orders of magnitude for each trajectory. There is a small trade-off between orientation and position error for the Figure 8, Square, and Spiral trajectories, where the position error increases when orientation tracking is added. Overall, the controller is found to produce accurate tracking results across a variety of trajectories.

C. Control Effort

In our approach, we treat the robot as a single system with $n = 9$ inputs. In this section, we examine how much control effort is required when controlling the robot as a single system versus using either the base or arm only. Table II shows the average control effort $\|\boldsymbol{u}\|$ for six trajectories, where only position is tracked. We use the same controller parameters as the previous section. Trajectories are performed with only the arm if they lie within the arm's workspace. Likewise, they are performed with only the base if they are confined to the x-y plane, since the base cannot produce vertical motion by itself. For all trajectories, the control effort is increased when only one subsystem is used. Control effort is particularly increased when only the arm is used, which suggests that these trajectories cause the arm to move toward singular configurations and thus require larger joint velocities to achieve a particular Cartesian twist. Paired with the fact that some trajectories, such as Spiral, are not feasible for either system alone, this result offers a justification for the use of redundant mobile manipulators controlled as a single system.

D. Admittance Control

To test our admittance control law (7), we perform two experiments. In the first experiment, the robot is commanded to apply increasingly large forces vertically against a table. The actual and desired forces in the z-direction are shown in Figure 6. As seen in the figure, the robot is able to successfully apply forces from $5\,\mathrm{N}$ to $40\,\mathrm{N}$ with high accuracy.

The second experiment again requires the robot to regulate to a desired force in the z-direction, but it is now also commanded to simultaneously track a trajectory in the x-y plane. Such a behaviour is required for tasks like wiping a surface, which we demonstrate in the experimental video. Figure 7 shows the results with an elliptical path. The path is traversed in $60\,\mathrm{s}$ and

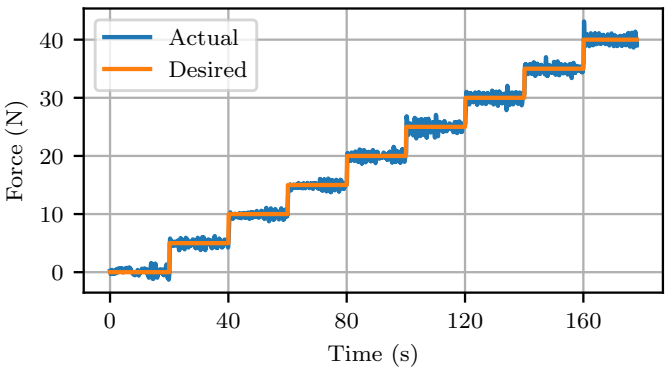

Figure 6: Actual and desired force in the z-direction for a force regulation task.

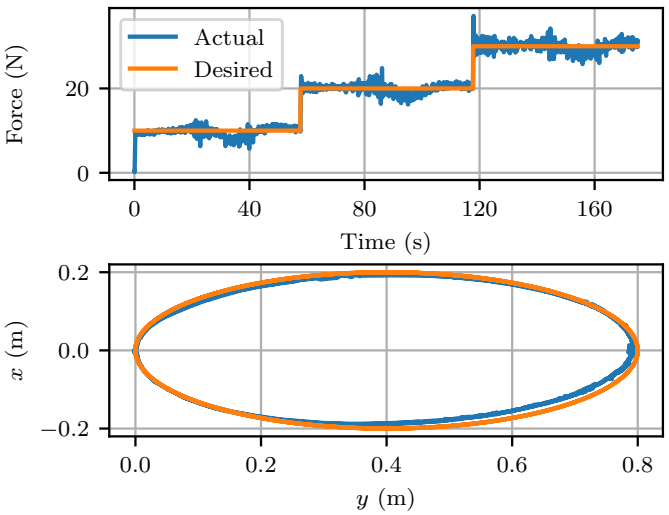

Figure 7: Top: force in the z-direction during a surface tracking task. Bottom: the path followed by the EE in the x-y plane. The elliptical path is traversed in $60\,\mathrm{s}$ and repeated three times with increasing desired force.

repeated three times with increasing desired force. There is a slight deviation from the desired force each time the EE reaches the far side of the elliptical path, but it is still able to track the desired force and the desired trajectory with reasonable accuracy.

In both experiments, we use $\boldsymbol{K}_p = \mathrm{diag}(1,1,0,1,1,1)$, $\boldsymbol{K}_f = \mathrm{diag}(0,0,0.005,0,0,0)$, and $\boldsymbol{Q} = \boldsymbol{I}_6$. We attach a piece of foam to the end effector to reduce the stiffness of the system, allowing the robot to move more quickly while remaining stable.

E. Obstacle Avoidance

To test the obstacle avoidance constraint (8), we have the robot navigate a slalom course composed of four obstacles. In a first experiment, we have the robot follow a line with the EE while avoiding the obstacles with the base. The trajectory is a $4\,\mathrm{m}$ line in the x-direction with a $15\,\mathrm{s}$ duration. In this experiment we only track position, using $\boldsymbol{Q} = 10\,\mathrm{diag}(1,1,1,0,0,0)$ and $\boldsymbol{K}_p = \boldsymbol{I}_6$. Compared to the pose tracking experiments, we have lowered the weight $\boldsymbol{Q}$ to ensure the system remains stable at the increased speed. The velocity damper parameters are $d_i = 1\,\mathrm{m}$, $d_s = 0.1\,\mathrm{m}$, and $\xi = 1$. The circle approximating

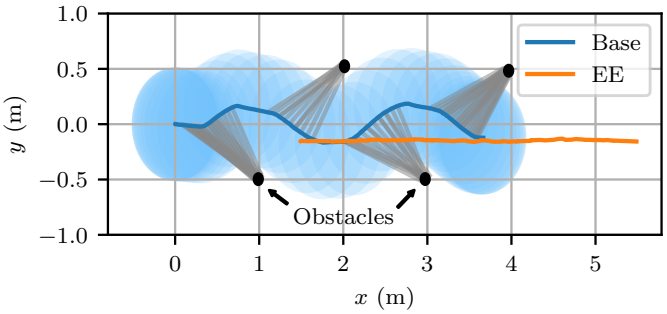

Figure 8: Robot traversing the slalom obstacle course. The EE follows a straight line (orange). The path of the base (blue line), moves to avoid the obstacles (black circles) while facilitating the EE motion. The swept-out area of the circle approximating the base is shown in light blue along the base's path. Obstacle detections within the influence distance d_i of the base are shown as gray lines.

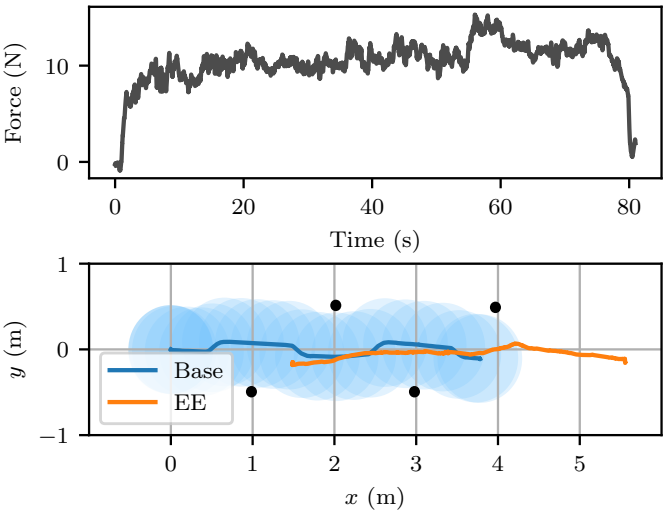

Figure 9: Robot traversing the slalom obstacle course, driven only by force applied by a human user at the EE. Top: the applied force in the x-direction. Bottom: the path of the base and EE through the obstacle course.

the base has radius $r_b = 0.5\,\mathrm{m}$. The robot is successfully able to navigate the course, as shown in Figure 8. The EE is able to follow a fairly straight path with an RMS position error of $4.22\,\mathrm{cm}$ despite the high speed and lateral motion of the base to avoid obstacles.

We perform a second experiment combining obstacle avoidance with admittance control. The robot navigates the same slalom obstacle course, but instead of tracking a commanded task-space trajectory, it is driven purely by force applied at the EE by a human user. We use parameters $\boldsymbol{K}_p = \boldsymbol{0}$, $\boldsymbol{K}_f = 0.1\boldsymbol{I}_6$, and $\boldsymbol{Q} = \boldsymbol{I}_6$ for admittance control, and the same velocity damper parameters as above. The results are shown in Figure 9. The robot is successfully pulled through the course while avoiding obstacles, but the EE path varies due to inconsistencies in the applied force.

F. Manipulability

Next, we compare the Objective and Constraint formulations for manipulability maximization. The robot is commanded to track the Figure 8 trajectory, which requires significant

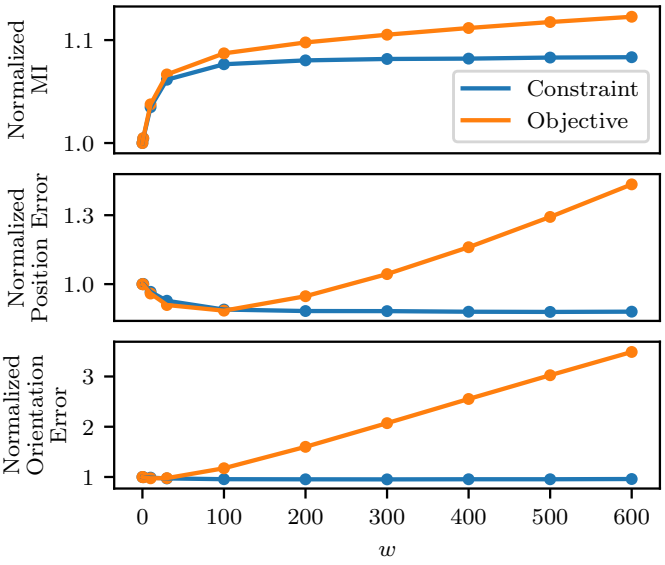

Figure 10: Results of a MI maximization experiment while tracking the Figure 8 trajectory with various weights w on the MI maximization objective. The MI, RMS position error, and RMS orientation error, each normalized such that the value at $w = 0$ is 1, are compared between the two approaches.

arm movement, using (5) with $\boldsymbol{K}_p = \boldsymbol{Q} = \boldsymbol{I}_6$ and varying values of the weight w for each approach. The average MI, RMS position error, and RMS orientation error are shown in Figure 10, where each is normalized so that the value obtained with $w = 0$ is 1. While the Objective formulation produces a slightly higher MI at large values of w, it comes at the cost of significant position and orientation error. In contrast, the Constraint formulation actually decreases tracking error slightly at higher values of w. We hypothesize that increasing the MI using the Constraint formulation actually results in joint configurations that are able to respond to the given tracking commands more effectively. Since our controller does not look ahead over a prediction horizon, maximizing the MI appears to endow it with an alternative, limited form of foresight.

G. Pushing

Finally, we demonstrate a pushing experiment using (15). The robot is commanded to push a barrel (depicted in Figure 1) toward the point $(x, y) = (3, 0)$. The controller parameters are $v_{\text{push}} = 0.2\,\mathrm{m/s}$, $\alpha_{\text{push}} = 0.5$, $\alpha_f = 0.01$, and $f_T = 5\,\mathrm{N}$. The trajectory of the EE and barrel are shown in Figure 11. Around $t = 8\,\mathrm{s}$, the barrel is manually perturbed in the y-direction to test the controller's ability to maintain contact with the object. Immediately following the perturbation, the EE slips before moving back toward the barrel and regaining contact. The barrel is now offset in the y-direction, but by $t = 25\,\mathrm{s}$ the EE has moved across the barrel to push it back toward $y = 0\,\mathrm{m}$. Meanwhile, the EE consistently pushes the barrel forward in the x-direction for the entire duration of the experiment.

Our approach is not a perfect pushing solution: the barrel is not pushed to the exact target point. However, it is a reasonable heuristic for balancing between moving toward a goal position while maintaining contact with an unknown object using only

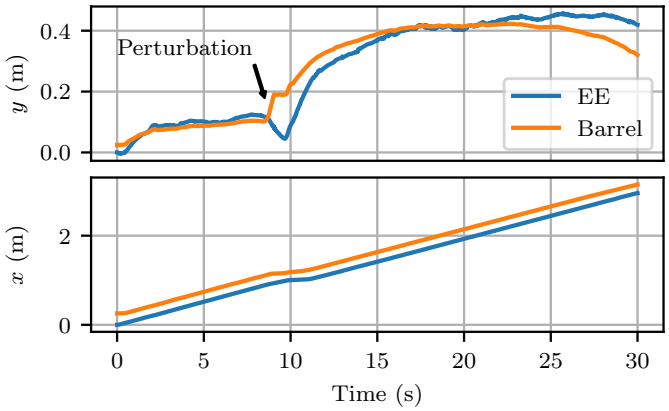

Figure 11: Results of a pushing experiment, where the object is manually perturbed. The controller recovers contact with the object and continues to push it toward the goal.

force measurements. Indeed, unless contact is lost and the object is entirely removed from the path between the EE and the target location, our approach is quite good at ensuring stable contact between the object and EE.

V. Conclusion

We have presented a differential inverse kinematics control strategy for position-controlled industrial mobile manipulators. In extensive hardware experiments, we show that our controller is capable of task-space trajectory tracking, force regulation, obstacle and singularity avoidance, and pushing an object toward a desired goal position, based on limited sensor feedback and knowledge of the environment.

In future work, we are interested in exploring receding horizon control approaches to improve performance and avoid local minima. We would also like to focus more specifically on particular mobile manipulation tasks. For example, we are interested in using MPC for pushing in situations where the properties of the object and environment may not be known.

References

[1] A. Shademan, R. S. Decker, J. D. Opfermann, S. Leonard, A. Krieger, and P. C. W. Kim, "Supervised autonomous robotic soft tissue surgery," *Science Translational Medicine*, vol. 8, no. 337, pp. 337ra64–337ra64, 2016.

[2] T. L. Chen, M. Ciocarlie, S. Cousins, P. M. Grice, K. Hawkins, K. Hsiao, C. C. Kemp, C.-H. King, D. A. Lazewatsky, A. E. Leeper, H. Nguyen, A. Paepcke, C. Pantofaru, W. D. Smart, and L. Takayama, "Robots for humanity: Using assistive robotics to empower people with disabilities," *IEEE Robotics Automation Magazine*, vol. 20, no. 1, pp. 30–39, 2013.

[3] M. E. Tiryaki, X. Zhang, and Q.-C. Pham, "Printing-while-moving: A new paradigm for large-scale robotic 3D printing," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 2286–2291.

[4] K. Kaneko, K. Harada, F. Kanehiro, G. Miyamori, and K. Akachi, "Humanoid robot HRP-3," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 2471–2478.

[5] Boston Dynamics, "Spot arm," https://www.bostondynamics.com/spot-arm, 2021, accessed: 2021-02-18.

[6] F. Ruggiero, V. Lippiello, and A. Ollero, "Aerial manipulation: A literature review," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1957–1964, 2018.

[7] S. Chiaverini, B. Siciliano, and O. Egeland, "Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator," *IEEE Transactions on Control Systems Technology*, vol. 2, no. 2, pp. 123–134, 1994.

[8] A. D. Luca, G. Oriolo, and P. R. Giordano, "Kinematic modeling and redundancy resolution for nonholonomic mobile manipulators," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2006, pp. 1867–1873.

[9] W. Suleiman, "On inverse kinematics with inequality constraints: New insights into minimum jerk trajectory generation," *Advanced Robotics*, vol. 30, no. 17-18, pp. 1164–1172, 2016.

[10] G. B. Avanzini, A. M. Zanchettin, and P. Rocco, "Constrained model predictive control for mobile robotic manipulators," *Robotica*, vol. 36, no. 1, pp. 19–38, 2018.

[11] W. Suleiman, F. Kanehiro, and E. Yoshida, "Infeasibility-free inverse kinematics method," in *Proc. of the IEEE/SICE International Symposium on System Integration*, 2015, pp. 307–312.

[12] G. B. Avanzini, A. M. Zanchettin, and P. Rocco, "Reactive constrained model predictive control for redundant mobile manipulators," in *Proc. of the International Conference on Intelligent Autonomous Systems (IAS)*, 2016, pp. 1301–1314.

[13] R. Ancona, "Redundancy modelling and resolution for robotic mobile manipulators: A general approach," *Advanced Robotics*, vol. 31, no. 13, pp. 706–715, 2017.

[14] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Proc. of the International Conference on Advanced Robotics*, 1991, pp. 1211–1216 vol.2.

[15] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, 1997.

[16] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen, and J. Schrimpf, "Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results," *Frontiers in Robotics and AI*, vol. 3, 2016.

[17] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.

[18] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[19] E. Brzozowska, O. Lima, and R. Ventura, "A generic optimization based cartesian controller for robotic mobile manipulation," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2019, pp. 2054–2060.

[20] 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.

[21] M. Neunert, C. de Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli, "Fast nonlinear model predictive control for unified trajectory optimization and tracking," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2016, pp. 1398–1404.

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

[23] B. Faverjon and P. Tournassoud, "A local based approach for path planning of manipulators with a high number of degrees of freedom," in *Proc. of the IEEE International Conference on Robotics and Automation*, 1987, pp. 1152–1159.

[24] T. Yoshikawa, "Manipulability of robotic mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[25] K. Dufour and W. Suleiman, "On integrating manipulability index into inverse kinematics solver," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 6967–6972.

[26] Y. Zhang, X. Yan, D. Chen, D. Guo, and W. Li, "QP-based refined manipulability-maximizing scheme for coordinated motion planning and control of physically constrained wheeled mobile redundant manipulators," *Nonlinear Dynamics*, vol. 85, no. 1, pp. 245–261, 2016.

[27] J. Stüber, C. Zito, and R. Stolkin, "Let's push things forward: A survey on robot pushing," *Frontiers in Robotics and AI*, vol. 7, 2020.

[28] K. Shoemake, "Animating rotation with quaternion curves," in *Proc. of the Annual Conference on Computer Graphics and Interactive Techniques*, 1985, pp. 245–254.

[29] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.