



Article Motion Planning of Nonholonomic Mobile Manipulators with Manipulability Maximization Considering Joints Physical Constraints and Self-Collision Avoidance

Josuet Leoro [†] and Tesheng Hsiao ^{*,†}

Department of Electrical and Computer Engineering, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan; jl2005ec.04g@g2.nctu.edu.tw

* Correspondence: tshsiao@cn.nctu.edu.tw

+ These authors contributed equally to this work.

Abstract: The motion of nonholonomic mobile manipulators (NMMs) is inherently constrained by joint limits, joint velocity limits, self-collisions and singularities. Most motion planning algorithms consider some of the aforementioned constraints, however, a unified framework to deal with all of them is lacking. This paper proposes a motion planning solution for the kinematic trajectory tracking of redundant NMMs that include all the constraints needed for practical implementation, which improves the manipulability of both the entire system and the manipulator to prevent singularities. Experiments using a 10-DOF NMM demonstrate the good performance of the proposed method for executing 6-DOF trajectories while satisfying all the constraints and simultaneously maximizing manipulability.

Keywords: motion planning; trajectory tracking; mobile manipulator; joint constraints; self-collision avoidance; manipulability

1. Introduction

A mobile manipulator is a robotic system that consists of a standard robot manipulator mounted on a mobile platform. This system integrates the dexterity provided by the manipulator with the extended workspace provided by the platform. Additionally, the combination of both subsystems usually introduces kinematic redundancy, which increases flexibility and dexterity. Therefore, mobile manipulators are suitable to perform delicate tasks over a large space, such as welding large parts or painting large, curvy surfaces.

The practical applications of robotic systems commonly define tasks by either a pointto-point movement or a continuous path of the end-effector in task space (also known as operational space). This paper aims to solve the latter, and in particular, focuses on the task space trajectory tracking problem. A trajectory is a path on which a timing law is specified, for instance in terms of velocities and/or accelerations [1]. In other words, not only is the end-effector's pose profile defined, but so is its velocity profile. To accomplish this, a motion planning algorithm that exploits the capabilities of both the manipulator and the mobile platform and that coordinates their movements is required. The redundancy of mobile manipulators can be used to perform additional subtasks or satisfy system constraints. These constraints include joint limits, joint velocity limits, joint velocity boundary constraints (i.e., the constraints on the initial and final joint velocities), and self-collision avoidance. Furthermore, for task space trajectory tracking to be achievable, it is important that the system is kept away from singularities. All these requirements make the motion planning for trajectory tracking a challenging problem.

There exist recent efforts in solving the motion planning of mobile manipulators [2,3]. Liao et al. [2] presented an optimization-based solution that not only handles constraints at the position level, but can also set a target joint configuration for the manipulator at



Citation: Leoro, J.; Hsiao, T. Motion Planning of Nonholonomic Mobile Manipulators with Manipulability Maximization Considering Joints Physical Constraints and Self-Collision Avoidance. *Appl. Sci.* 2021, *11*, 6509. https://doi.org/ 10.3390/app11146509

Academic Editors: Pedro Neto, António Paulo Moreira and Félix Vilariño

Received: 27 June 2021 Accepted: 14 July 2021 Published: 15 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). the end of the trajectory. A heuristic approach was proposed by Santos et al. [3] that is simple to implement and can accomplish additional constraints such as joint limits and manipulability improvement. Nonetheless, these methods do not deal with constraints at the velocity level and are only applicable to mobile manipulators with omnidirectional platforms. The present work focuses on the motion planning of mobile manipulators with nonholonomic mobile platforms. The movement of this type of platform is constrained by the rolling without slipping condition, which inhibits the platform from instantly moving in any arbitrary direction [4].

The motion planning of NMMs has also been studied in the literature using different approaches [5–10]. De Luca et al. [7] implemented the reduced gradient method for NMMs. This method finds a permutation matrix that helps reduce the velocity input to the subspace of commands that satisfy the given task. The remaining velocity inputs are used to maximize an objective function. Even though this method is computationally more efficient than the projected gradient approach, it is difficult to find such a permutation matrix for highly redundant robots since the Jacobian must be pre-analyzed by hand. Jia et al. [9] studied an adaptive motion distribution and coordinated control between the manipulator and the mobile platform to minimize the end-effector's positioning error.

In task space trajectory tracking, it is important that the motion planning algorithm moves the system away from singularities. This is because the system is in kinematic singularity, and the dexterity of the structure is reduced because the robot's end-effector cannot be moved in a certain direction. In addition, when the system is in the neighborhood of a singularity, small velocities in the task space may cause large velocities in the joint space [1], which is unacceptable because this would result in the failure of the trajectory tracking task, and even damage the mobile manipulator. For these reasons, the manipulability maximization was been included in multiple motion planning methods for NMMs [5,6,8–10]. Bayle et al. [5,6] maximized the system's manipulability using the projected gradient method. Huang et al. [8] studied the coordination of the platform and the manipulator, simultaneously considering the mobile platform stability and the manipulator's manipulability. Although these techniques can successfully follow the end-effector path while considering additional criteria, none of these consider joint constraints.

Furthermore, the solution of the task space trajectory tracking problem must not only consider joint limits but also joint velocity limits. This is because if a joint reaches its velocity limit, the end-effector might not be able to comply with the desired velocity profile. To the authors' knowledge, the literature of motion planning for trajectory tracking in task space with NMMs that includes joint constraints is limited. Zhang et al. [10] proposed formulating the motion planning problem as an optimization problem where the manipulability is maximized and the joint limits and joint velocity limits are included as constraints. This optimization problem is reformulated as a quadratic programming problem and converted into a linear variational inequality problem, that can be solved by different numerical methods. This approach is effective but does not consider boundary constraints for joint velocities. These constraints are also relevant because, for a given task, zero joint velocities are expected at the start and end of the trajectory. Additionally, NMMs are not only subject to physical limits but also to self-collisions, especially between the manipulator and the mobile platform, which are not included in their work.

An important remark is that maximizing the manipulability of the whole system might result in poor manipulability for the arm alone, even though the manipulability for the whole system is preserved or improved [6,10]. Additionally, it is important that both the robot arm subsystem and the whole mobile manipulator system maintain a certain level of manipulability once a coordinated task is completed. After completing an arbitrary path, a subsequent task might only require the arm to manipulate an object. If the arm is in a configuration with low manipulability, executing such a task might not be feasible. For these reasons, in this work, we propose a new manipulability measure for mobile manipulators that, when maximized, and as demonstrated in our experiments,

intrinsically improves the manipulability of the robot arm as well as the manipulability of the whole system.

The solution to the motion planning of NMMs for trajectory tracking presented in this work includes joint constraints (range and maximum velocities), self-collision avoidance and manipulation capability preservation. Figure 1 summarizes how all these constraints are included in our solution. Both the particular and homogeneous solutions of our proposed scheme are weighted to avoid joint limits and self-collisions while the trajectory is tracked. The homogeneous solution is used to maximize the manipulability by exploiting the redundancy of the system. To satisfy joint velocity boundary constraints, the step size for searching the maximum manipulability is varied. The joint velocity limits are satisfied by restraining the maximum step size based on the allowable self-motion.



Figure 1. Summary of the proposed motion planning approach.

Experimental results demonstrate that our method can successfully solve the motion planning problem of NMMs under all the mentioned constraints. This work focuses on task space trajectory tracking at kinematic level. In other words, the outputs of the motion planning algorithm are the joint positions and velocities that will be fed to a joint space dynamic controller for motion control. Then, the motion controller is responsible for suppressing the model uncertainties and external disturbances to guarantee that the actual joint positions follow the ones output by the motion planning algorithm. In the experiments shown in this paper, we use the built-in motion controller of the commercial manipulator and leave the design of our own motion controller for future research.

The contributions of this work are detailed as follows:

- A motion planning solution for NMMs that allows to include joint physical constraints and the execution of a secondary task is presented.
- Multiple constraints required for the practical implementation of task space trajectory tracking are included in a unified solution. These constraints include joint limits, joint velocity limits, joint velocity boundary constraints and self-collision avoidance.
- A new manipulability measure for mobile manipulators is presented. It is demonstrated that the maximization of this measure simultaneously improves the manipulabilities of the whole system and the robot arm.

This paper is organized as follows. Section 2 describes the kinematic modeling of NMMs. Section 3 describes the motion planning problem for trajectory tracking and presents the proposed solution. In Section 4, the concepts that are employed to satisfy each of the mentioned constraints are described and included in the motion planning algorithm.

Experimental results using 6-DOF tasks are presented in Section 5 to validate the efficacy of our approach. Finally, Section 6 concludes the paper.

2. Kinematic Modeling

The kinematic modeling described here follows the procedure of De Luca et al. [7] and Bayle et al. [6]. For a general procedure of kinematic modeling of wheeled mobile manipulators, please refer to Bayle et al. [5].

Let $r \in \mathbb{R}^m$ be the end-effector's position and/or pose in the task space. The configuration vector q, also known as the generalized coordinates of the mobile manipulator, is given by the combination of the platform configuration vector q_p and the robot arm configuration vector q_a . Figure 2 illustrates these configuration vectors. A frame x'y' is attached to the mobile platform at the center of the wheels' axle (x_p, y_p) , with respect to the world reference frame xy, with its x' axis pointing in the forward direction and the y' axis pointing in the direction parallel to the wheels' axle. The angle between the x axis of the world reference frame and x' attached to the platform is denoted as θ_p . Then, the platform configuration is given by $q_p = \begin{bmatrix} x_p & y_p & \theta_p \end{bmatrix}^T \in \mathbb{R}^3$. The robot arm configuration is given by the position of its joints as $q_a = \begin{bmatrix} q_{a_1} & q_{a_2} & \dots & q_{n_a} \end{bmatrix}^T \in \mathbb{R}^{n_a}$. Finally, the configuration vector of the mobile manipulator is $q = \begin{bmatrix} q_p^T & q_a^T \end{bmatrix}^T \in \mathbb{R}^n$ with $n = 3 + n_a$. The end-effector's position/pose is a function of the configuration vector defined by the kinematic map:

$$\mathbf{r} = f(\mathbf{q}_{\mathbf{p}}, \mathbf{q}_{\mathbf{a}}),\tag{1}$$



Figure 2. Nonholonomic mobile manipulator schematic.

The wheeled platform movement is constrained under the rolling without a slipping assumption on both wheels, which can be expressed as the following nonholonomic constraint:

$$\dot{\boldsymbol{g}}_p = \boldsymbol{G}(\boldsymbol{q}_p)\boldsymbol{u}_p,\tag{2}$$

where $u_p = \begin{bmatrix} v_p & \omega_p \end{bmatrix}^T \in \mathbb{R}^2$ is the velocity input of the platform, consisting of the linear and angular velocities of the platform, which are known as pseudo velocities or quasi-velocities. The columns of matrix $G(q_p)$ span the admissible velocity space for a given platform configuration. The matrix $G(q_p)$ for a differential-drive platform is defined as

$$G(\boldsymbol{q}_{\boldsymbol{p}}) = \begin{bmatrix} \cos \theta_p & 0\\ \sin \theta_p & 0\\ 0 & 1 \end{bmatrix},$$

On the other hand, the robotic arm kinematics at the velocity level is not constrained for any configuration, and therefore, the generalized velocities of the arm are equal to the velocity inputs of the arm:

$$\dot{q}_a = u_a. \tag{3}$$

The velocity input vector for the entire NMM can be constructed as $\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_p^T & \boldsymbol{u}_a^T \end{bmatrix}^T \in \mathbb{R}^{\delta_m}$, with $\delta_m = 2 + n_a$. The dimension δ_m is called the mobility degree of the mobile manipulator and indicates the space dimension of the admissible generalized velocities [6]. Using (2) and (3), the map from the velocity input vector \boldsymbol{u} to the generalized velocities $\boldsymbol{\dot{q}} = \begin{bmatrix} \boldsymbol{\dot{q}}_p^T & \boldsymbol{\dot{q}}_a^T \end{bmatrix}^T$ can be written as

$$\dot{q} = S(q)u, \tag{4}$$

with:

$$S(\boldsymbol{q}) = \begin{bmatrix} G(\boldsymbol{q}_{\boldsymbol{p}}) & 0\\ 0 & I_{n_a} \end{bmatrix},$$

where $G(q_p)$ maps the platform's velocity input vector $u_p = \begin{bmatrix} v_p & \omega_p \end{bmatrix}^T$ to the platform's generalized velocities $\dot{q}_p = \begin{bmatrix} \dot{x}_p & \dot{y}_p & \dot{\theta}_p \end{bmatrix}^T$, and matrix I_{n_a} is an identity matrix of size n_a that sets the arm's generalized velocities equal to its input velocities $\dot{q}_a = u_a$.

The differential kinematics of the NMM is obtained by differentiating the relation (1) with respect to time:

$$\dot{\mathbf{r}} = \begin{bmatrix} J_p(\mathbf{q}_p) & J_a(\mathbf{q}_a) \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_p \\ \dot{\mathbf{q}}_a \end{bmatrix} \\
= \begin{bmatrix} J_p(\mathbf{q}_p) & J_a(\mathbf{q}_a) \end{bmatrix} S(\mathbf{q}) \mathbf{u} \\
= J(\mathbf{q}) S(\mathbf{q}) \mathbf{u} = \bar{J} \mathbf{u},$$
(5)

where the matrices $J_p \in \mathbb{R}^{m \times 3}$ and $J_a \in \mathbb{R}^{m \times n_a}$ are the Jacobians of the platform and the arm, respectively. The matrix $J \in \mathbb{R}^{m \times n}$ is the Jacobian of the mobile manipulator, and $\overline{J} \in \mathbb{R}^{m \times \delta_m}$ is a reduced version of the Jacobian in which the admissible velocities of the platform under the nonholonomic constraint have been included. Equation (5) follows the same form as the differential kinematics of standard manipulators; therefore, the well-known methodologies for the motion planning of redundant manipulators can be extended to NMMs in terms of \overline{J} , including concepts for joint limits avoidance, self-collisions avoidance and manipulability maximization.

3. Motion Planning Method

The motion planning problem for task space trajectory tracking consists of finding the input velocities u, given the desired end-effector's position/pose $r_d(t) \in \mathbb{R}^m$ for $t \in [t_0, t_f]$, such that r(t) follows $r_d(t)$. If the task space velocity of the end-effector is set as

$$\dot{\mathbf{r}} = \dot{\mathbf{r}}_d + K(\mathbf{r}_d - \mathbf{r}),\tag{6}$$

where $K \in \mathbb{R}^{m \times m}$ is a positive definite matrix, then the convergence of r(t) to $r_d(t)$ is guaranteed. Consequently, the motion planning problem is turned into solving the input velocities u from the underdetermined linear equations (5) where \dot{r} is set according to (6).

During the execution of the trajectory, the movement of the joints that get closer to a position constraint (joint limit or self-collision) must be penalized (slowed down). This can be achieved by using a weighting matrix. In this work, we define the weighted input velocity and the reduced weighted Jacobian as follows:

$$\bar{J}_W = \bar{J}W^{\frac{1}{2}}(q) \quad \text{and} \quad u = W^{\frac{1}{2}}(q)u_W, \tag{7}$$

where $W(q) \in \mathbb{R}^{\delta_m \times \delta_m}$ is a configuration-dependent positive semidefinite matrix. It is constructed so that its elements penalize the motion of the joints based upon the system constraints. We choose W(q) as a diagonal matrix in this paper; furthermore, we drop the notational dependency of W on q to simplify the expression in the subsequent derivation.

Using (7), the system (5) can be rewritten as

$$\dot{r} = \bar{J}_W u_W. \tag{8}$$

Because of the kinematic redundancy, i.e., $m < \delta_m$, infinite solutions to u_W exist. The solution to (8) can be decomposed as a sum of a particular solution and a non-zero homogeneous solution. The particular solution satisfies the end-effector's task, while the homogeneous solution changes the manipulator configuration without changing the end-effector's position/pose.

One way to solve the redundant system (8) is to formulate the problem as a constrained optimization problem [1], where the goal is to minimize a cost function and satisfy the constraint (8). We propose the minimization of the quadratic cost function:

$$g(\boldsymbol{u}_{\boldsymbol{W}}) = \frac{1}{2} (\boldsymbol{u}_{\boldsymbol{W}} - \boldsymbol{W}^{\frac{1}{2}} \boldsymbol{u}_{\boldsymbol{0}})^{T} (\boldsymbol{u}_{\boldsymbol{W}} - \boldsymbol{W}^{\frac{1}{2}} \boldsymbol{u}_{\boldsymbol{0}});$$

this choice is aimed to find a vector of velocities u_W that is as close as possible to $W^{\frac{1}{2}}u_0$, where $u_0 \in \mathbb{R}^{\delta_m}$ is a velocity vector that is used to satisfy an additional task such as maximizing the manipulability. The choice of u_0 will be presented shortly. Notice that the physical constraints are also imposed to u_0 by weighting it similarly to how \overline{J}_W is obtained in (7). By using the method of Lagrange multipliers to minimize $g(u_W)$ with the equality constraint (8), the following solution is obtained:

$$u_W^* = \bar{J}_W^{\dagger} \dot{r} + (I - \bar{J}_W^{\dagger} \bar{J}_W) W^{\frac{1}{2}} u_0, \tag{9}$$

where \bar{J}_W^{\dagger} is the Moore–Penrose pseudoinverse of the matrix \bar{J}_W such that $\bar{J}_W \bar{J}_W^{\dagger} = I$, $I - \bar{J}_W^{\dagger} \bar{J}_W$ is the orthogonal projection into the null space of \bar{J}_W . The first term of (9) is the particular solution, and the second term is the homogeneous solution. It is trivial to show that $I - \bar{J}_W^{\dagger} \bar{J}_W$ projects u_0 onto the null space of \bar{J}_W by multiplying both sides of (9) by \bar{J}_W . The velocity input vector is then recovered using the second part of (7):

$$u = W^{\frac{1}{2}} \bar{J}_W^{\dagger} \dot{r} + W^{\frac{1}{2}} (I - \bar{J}_W^{\dagger} \bar{J}_W) W^{\frac{1}{2}} u_0.$$
⁽¹⁰⁾

The degree of kinematic redundancy at the velocity level is relevant for this solution to be feasible, since it defines the number of simultaneous constraints that can be satisfied in the differential kinematics inversion without \overline{J}_W losing its rank. The degree of redundancy for NMMs is calculated as $R = \delta_m - m$. Whilst analyzing the matrix $W^{\frac{1}{2}}$ for three different cases, it is possible to understand the expected behavior of solution (10). Let *z* represent the number of elements that are zero in the diagonal of $W^{\frac{1}{2}}$, i.e., the number of joints that are forced to stop due to *z* simultaneously activated constraints. When z < R, both the particular and homogeneous solutions exist, and therefore, the secondary task will still be considered. When z = R, the system is not redundant anymore and only the particular solution exists. Finally, when z > R, the system (8) has no solution. Therefore, for the solution (10) to exist, the condition $z \le R$ is necessary.

Our proposed solution (10) has an advantage over the projected gradient solution used in [5,6], because it includes the physical constraints and penalizes both the particular and homogeneous solutions. Furthermore, in contrast with the weighted least-norm method [11], this solution takes advantage of the redundant joints that have not been penalized, due to physical constraints, in attempt to satisfy the task defined by u_0 .

In this paper, the matrix W is constructed to satisfy joint position constraints (joint limits and self-collision avoidance), which is discussed in Section 4.2. The vector u_0 is designated to locally maximize a proposed new manipulability measure for mobile manipulators, as discussed in Section 4.1. To satisfy the joint velocity constraints, the particular and homogeneous solutions are analyzed. Because the particular solution is in charge of following the end-effector's task, it cannot be modified. On the other hand, the homogeneous solution can be varied to satisfy the joint velocity constraints. Furthermore, the homogeneous solution must also consider the admissible velocity commands with

respect to the nonholonomic constraints [7]. Taking these two points into account, we define vector u_0 as

$$\boldsymbol{u}_{\boldsymbol{0}} = \pm \boldsymbol{\alpha}(t)\boldsymbol{\beta}(t)\boldsymbol{S}^{T}(\boldsymbol{q})\boldsymbol{\nabla}_{\boldsymbol{q}}\boldsymbol{F}(\boldsymbol{q}), \tag{11}$$

where $\alpha(t)$ is a scalar coefficient that is adjusted online to satisfy joint velocity limits (Section 4.3.2), $\beta(t)$ is a positive variation factor used to satisfy joint velocity boundary constraints (Section 4.3.1), S(q) is defined in (4), and $\nabla_q F(q)$ is the gradient of the objective function $F(q) : \mathbb{R}^{3+n_a} \to \mathbb{R}$, which is set to a manipulability measure. The product $\alpha(t)\beta(t)$ is the step size of the gradient step ascent/descent, and the sign of u_0 defines whether the objective function F(q) will be maximized (+) or minimized (-).

Replacing (11) in (10), the final form of the proposed solution for u is:

$$u = u_{p} + \alpha(t)\beta(t)u_{h},$$
with
$$u_{p} = W^{\frac{1}{2}}\bar{J}_{W}^{\dagger}\dot{r},$$

$$u_{h} = W^{\frac{1}{2}}(I - \bar{J}_{W}^{\dagger}\bar{J}_{W})W^{\frac{1}{2}}S^{T}(q)\nabla_{q}F(q),$$
(12)

where the positive sign for the gradient step descent has been used because we are interested in maximizing the manipulability. Notice that when the sign of $\alpha(t)$ is positive, the objective function F(q) is maximized. However, for some cases, $\alpha(t)$ could be negative for short periods of time to respect joint velocity limits, as explained in Section 4.3.

4. Manipulability Maximization and Constraints Satisfaction

In this section, the details for manipulability maximization are described, the weighting matrix *W* is defined so that it penalizes the joint movement to avoid both joint limits and self-collisions, and a scheme to satisfy joint velocity limits as well as joint velocity boundary constraints is presented.

4.1. Manipulability Maximization

The term manipulability, introduced by Yoshikawa [12], describes the ability of a robotic system to provide end-effector's velocities in any direction for a given configuration. The manipulability of wheeled mobile manipulators was studied in detail in [5]. There exist different algebraic measures to characterize the manipulability of a robotic system [5,12–14]. The most widely used measure, known as the manipulability measure and noted here as Ω , is given by $\Omega = \sigma_1 \sigma_2 \dots \sigma_m$, where $\sigma_1, \sigma_2, \dots, \sigma_m$ are the singular values of the system Jacobian J(q). Therefore, Ω is the manipulability measure for system configuration q. The measure Ω is proportional to the volume of the manipulability ellipsoid [12]. Furthermore, it can be shown that Ω has the following property:

$$\Omega = \sqrt{\det(J(\boldsymbol{q})J^{T}(\boldsymbol{q}))}$$
(13)

By definition, $\Omega \ge 0$, and $\Omega = 0$ if and only if rank(J(q)) < m. The elements of $\nabla_q \Omega$ are given mathematically by

$$\frac{\partial\Omega}{\partial q_i} = -\frac{1}{2}\Omega \cdot \operatorname{trace}\left(\left(JJ^T\right)^{-1}\left(\frac{\partial J}{\partial q_i}J^T + J\left(\frac{\partial J}{\partial q_i}\right)^T\right)\right),\tag{14}$$
with $i = 1, 2, \dots, n.$

As mentioned before, we are not only concerned with improving the manipulability of the whole system but also that of the arm alone. In this work, we present a new manipulability measure that better expresses the manipulability of a mobile manipulator, because both the manipulability of the arm and the whole system are intrinsically considered. Let us define the manipulability of the whole system, denoted as Ω_{p+a} , and the manipulability of the robot arm, denoted as Ω_a , to be the measures obtained using Equation (13) with the Jacobians \overline{J} and J_a from Equation (5), respectively. Notice that by using the reduced Jacobian \overline{J} , the platform's nonholonomic constraints are included in the manipulability measure of the whole system. Our proposed manipulability measure for mobile manipulators is defined as

$$\Omega_{MM} = \hat{\Omega}_{p+a} \hat{\Omega}_a, \tag{15}$$

where $\hat{\Omega}_{p+a}$ and $\hat{\Omega}_{a}$ are the normalized manipulabilities computed by dividing them by their respective maximum values over all the possible configurations of the system:

$$\hat{\Omega}_{p+a} = rac{\Omega_{p+a}}{\Omega_{p+a_{max}}} \quad ext{and} \quad \hat{\Omega}_a = rac{\Omega_a}{\Omega_{a_{max}}}.$$

The normalized values are used in our proposed measure to make sure that all its components are in the same scale. This normalization also makes the measures invariant to units and chosen reference frame [14,15].

Notice that $J = [J_p \quad J_a]S(q) \in \mathbb{R}^{m \times \delta_m}$ in (5) indicates the relation between J and J_a . Hence, the singular values of \bar{J} are not necessarily the same as the singular values of J_a . There are cases where J_a is rank-deficient while \bar{J} is not. For these cases, the whole system manipulability $\Omega_{p+a} \neq 0$ even though the arm is in a singular configuration. Therefore, the measure Ω_{p+a} fails to express the ability of the arm alone to provide end-effector's velocities in any direction. As a result, maximizing the measure Ω_{p+a} might result in the poor manipulability of the arm, even though the manipulability of the whole system is preserved or improved. This is an issue that has been discussed in the literature [6,10].

On the other hand, our proposed measure Ω_{MM} is defined by the product of the singular values of \overline{J} and J_a , i.e., the product of the manipulability ellipsoids of \overline{J} and J_a . Therefore, it encapsulates the abilities of the whole system and the arm to provide end-effector's velocities in any direction. If any of the singular values of J_a or \overline{J} is zero, e.g., the arm is in a singularity, the measure $\Omega_{MM} = 0$. Hence, the measure Ω_{MM} is a better representation of the manipulability of mobile manipulators because it intrinsically considers the manipulability of the arm and the whole system.

Using the product rule, the gradient of Ω_{MM} is calculated as

$$\nabla_{q}\Omega_{MM} = \frac{1}{\Omega_{p+a_{max}}\Omega_{a_{max}}} (\Omega_{a}\nabla_{q}\Omega_{p+a} + \Omega_{p+a}\nabla_{q}\Omega_{a}).$$
(16)

Analyzing the right hand side of (16), it is clear that the manipulability of the arm subsystem affects the gradient of the whole system. Likewise, the manipulability of the whole system affects the gradient of the arm subsystem. Therefore, maximizing Ω_{MM} , i.e., setting $\nabla_q F(q) = \nabla_q \Omega_{MM}$ in our solution (12) will simultaneously improve the manipulabilities of the arm and the whole system. In the experiments section (Section 5), a comparison is performed among the different objective functions for manipulability maximization to demonstrate the advantages of the proposed manipulability measure.

The reader may have noticed that because our proposed motion planning solution (12) multiplies $W^{\frac{1}{2}}$ with the gradient $\nabla_q F(q)$, the direction of the original gradient that maximizes the manipulability is changed. However, as our goal is to push the system away from singularities by gradually improving the manipulability rather than finding the shortest path towards its maximum value, it is sufficient to prove that the weighted vector $W^{\frac{1}{2}}u_0$ also points in a direction that increases the manipulability as the unweighted gradient u_0 does. Since $W^{\frac{1}{2}}$ is positive semidefinite, we have $u_0^T W^{\frac{1}{2}} u_0 \ge 0$ for all u_0 . This implies that the weighted gradient also points in a direction that increases the manipulability.

4.2. Joint Position Constraints

Joint position constraints are common requirements in the motion planning of robotic systems. In this work, joint limits and self-collision avoidance are considered. These constraints are included in the proposed solution (12) using matrix W, which we define as the product of three terms as follows:

$$W = W_{Ilim} W_{Col} T_a^{-1}$$

where W_{Jlim} is the weighting matrix for joint limits constraints, W_{Col} is the weighting matrix for self-collision avoidance, and the matrix T_q (known as the Jacobian normalization matrix), is used to normalize the velocity commands u as follows [12]:

$$T_q = \operatorname{diag}\left(\frac{1}{u_{1_{max}}}, \frac{1}{u_{2_{max}}}, \dots, \frac{1}{u_{\delta_{mmax}}}\right),$$

where $u_{i_{max}}$ is the *i*th maximum velocity command. The weighting of this matrix handles the differences in units and scales among all the joints.

For mobile manipulators, matrices W_{Jlim} and W_{Col} should be constructed only considering the arm. This is because there is no limit to the movement of the mobile platform along each degree of freedom. Likewise, the movement of the mobile platform cannot be used to avoid self-collisions because the platform moves the base of the arm. Therefore, in this work, the structure of both weighting matrices was designed to take this into account. To simplify the presentation, let W_g represent either W_{Jlim} or W_{Col} . W_g is a diagonal $\delta_m \times \delta_m$ matrix with the following form:

$$W_g = \begin{bmatrix} I_2 & 0 \\ 0 & W_a \end{bmatrix},$$

where I_2 is an identity matrix of size 2 (for the case of the differential drive) and W_a is a diagonal $n_a \times n_a$ matrix given by

$$W_a = egin{bmatrix} w_1 & 0 & 0 & \dots & 0 \ 0 & w_2 & 0 & \dots & 0 \ dots & dots & dots & dots & dots & dots \ 0 & 0 & 0 & \dots & w_{n_a} \end{bmatrix}$$
 ,

where each element of the diagonal matrix W_a is defined using a performance criterion H(q).

In the next two subsections, two separate criterion functions for joint limits and selfcollision avoidance are defined. These two criterion functions have common properties. The values of H(q) and $|\partial H(q)/\partial q_i|$ become very large as the constraints are violated. When constructing matrix W_a using these criterion functions, the joint movement towards or away from a constraint must be contemplated [11,16]. Under this consideration, the elements of W_a are given by

$$w_{i} = \begin{cases} \frac{1}{1+\left|\frac{\partial H(\boldsymbol{q})}{\partial q_{i}}\right|}, & \text{if } \Delta \left|\frac{\partial H(\boldsymbol{q})}{\partial q_{i}}\right| > 0\\ 1, & \text{if } \Delta \left|\frac{\partial H(\boldsymbol{q})}{\partial q_{i}}\right| \le 0, \end{cases}$$

$$\text{with } i = 1, 2, \dots, n_{q_{i}}$$

$$(17)$$

where $\Delta |\partial H(q) / \partial q_i|$ is the change rate of $|\partial H(q) / \partial q_i|$ with respect to time, and is numerically calculated during implementation. With this choice, a value of one is assigned to w_i , indicating that no penalty is imposed on the *i*th joint, if the *i*th joint is not moving $(\Delta |\partial H(q) / \partial q_i| = 0)$, or it moves away from a constraint $(\Delta |\partial H(q) / \partial q_i| < 0)$. On the other hand, w_i tends towards zero if the movement of the *i*th joint gets closer to a constraint.

Hence, the element w_i penalizes the movement of the *i*th joint by means of (12) if it moves towards a constraint and stops the joint if it is too close to it.

4.2.1. Joint Limits Avoidance

To construct the weighting matrix for joint limits avoidance W_{Jlim} , a well-known criterion function [17] is used:

$$H_{Jlim}(\boldsymbol{q}) = \sum_{i=1}^{n_a} \frac{1}{4\gamma} \frac{(q_i^+ - q_i^-)^2}{(q_i^+ - q_i)(q_i - q_i^-)},$$

where q_i is the *i*th joint angle, q_i^- and q_i^+ are its lower and upper limits, respectively, and γ is a scalar constant that adjusts the rate of change of $H_{Jlim}(q)$. This criterion function increases as the joint gets closer to its limits and goes to infinity at the joint bounds. The elements of the gradient of this function are given by

$$\frac{\partial H_{Jlim}(\boldsymbol{q})}{\partial q_i} = \frac{1}{4\gamma} \frac{(q_i^+ - q_i^-)^2 (2q_i - q_i^+ - q_i^-)}{(q_i^+ - q_i)^2 (q_i - q_i^-)^2}.$$
(18)

Each element $\partial H_{Jlim}(q) / \partial q_i$ is equal to zero if q_i is at the middle of its joint range and goes to infinity at either limit.

4.2.2. Self-Collision Avoidance

To construct the weighting matrix for self-collision avoidance W_{Col} , an exponential function of the distance between two collision points is used as the criterion function [16]:

$$H_{Col}(\boldsymbol{q}) = \rho e^{-c_1 d(\boldsymbol{q})} d(\boldsymbol{q})^{-c_2},$$

where $\rho > 0$ controls the amplitude of $H_{Col}(q)$, and $c_1, c_2 > 0$ control the rate of decay. This function has a maximum value when the distance *d* between two links is zero, and exponentially decreases as this distance increases. The distance between two collision points is defined as $d(q) = ||p_{l1} - p_{l2}||_2$, where p_{l1} and p_{l2} represent the position vectors, referred to a common frame, of the collision points on two nonadjacent links. p_{l1} and p_{l2} can be calculated from the configuration vector *q* through forward kinematics.

The elements of the gradient of $H_{Col}(q)$ are given by

$$\frac{\partial H_{Col}(q)}{\partial q} = \frac{\partial H_{Col}(q)}{\partial d} \frac{\partial d}{\partial q},\tag{19}$$

where each of the partial derivatives is given by

$$\frac{\partial H_{Col}(\boldsymbol{q})}{\partial d} = -\rho e^{-c_1 d} d^{-c_2} (c_2 d^{-1} + c_1), \tag{20}$$

$$\frac{\partial d}{\partial q} = \frac{1}{d} [J_1^T (p_{l1} - p_{l2}) + J_2^T (p_{l2} - p_{l1})], \qquad (21)$$

where J_1 and J_2 are the associated Jacobian matrices of p_{l1} and p_{l2} , respectively. The collision points are chosen from the surface of the links for which the collision distance is computed. For the case of potential collisions between the arm and the mobile platform, p_{l2} is picked as a point fixed on the surface of the mobile platform (p_{l2} does not move with respect to the arm). By using a frame attached to the mobile platform as the common frame, and selecting p_{l2} as a fixed point, (21) reduces to:

$$\frac{\partial d}{\partial q} = \frac{1}{d} [J_1^T (\boldsymbol{p_{l1}} - \boldsymbol{p_{l2}})]. \tag{22}$$

When constructing matrix W_a , the partial derivative $\partial d/\partial q$ in (19) was calculated using (21) or (22) depending on whether the collision is evaluated between two moving links or

a moving link and a fixed one, respectively. Each element $\partial H(q)/\partial q_i$ tends toward zero as the distance between two evaluated collision points increases, and tends towards infinity as the distance gets closer to zero. Because d = 0 is not admissible, i.e., the two links are in contact, the chosen points must contemplate a threshold.

Multiple pairs of potential collision points might exist in a robotic system. Let N_c be the number of potential collision point pairs that are evaluated. A self-collision matrix W_{Col_j} is constructed for each distance d_j , with $j = 1, 2, ..., N_c$. The weighting matrix that includes the contribution of each evaluated pair is finally obtained by

$$W_{Col} = \prod_{j=1}^{N_c} W_{Col_j}.$$
(23)

With this combination, the *i*th diagonal element of the matrix W_{Col} penalizes the movement of the *i*th joint. In contrast with the combination of collision weighting matrices proposed in [16], the combination (23) guarantees that the joints are stopped if they attempt to decrease any of the collision distances d_j to a value of zero. Since some potential collisions are taken care of by the joint limits, the number of points considered in (23) is small, and therefore the computation cost can be relieved.

4.3. Joint Velocity Constraints

In a task space trajectory tracking problem, joint constraints at the velocity level are as important as joint limits and self-collision avoidance. These constraints include joint velocity boundary constraints and joint velocity limits. Satisfying joint velocity boundary constraints is a requirement for the end-effector to stop at the beginning and end of the task, i.e., the initial and final joint velocities must be zero. Joint velocity limits must be considered because the motion planning algorithm might generate joint velocity commands that are out of bounds in order to follow the task space velocity profile. These unfeasible velocities cannot be achieved by the real joints. Therefore, the trajectory tracking will fail if the joint velocity limits are not respected.

4.3.1. Joint Velocity Boundary Constraints

To satisfy joint velocity boundary constraints, the initial and final joint velocities must be zero. u_p in (12) is directly associated with the end-effector's task; therefore, it implicitly satisfies the boundary constraints. However, u_h in (12) does not necessarily satisfy them, because the manipulability maximization task is dependent on the mobile manipulator configuration at the initial and final poses. In this work, we propose using a time-varying self-motion magnitude to handle these constraints. The objective is to avoid non-zero values of u_h only at the beginning and end of the trajectory. With this in mind, it is proposed to set the variation factor $\beta(t)$ to increase from zero to one at the beginning of the trajectory, maintain a value of one for most of the trajectory, and then decrease from one to zero at the end of the trajectory.

Let t_b be the blending time for $\beta(t)$ to increase from zero to one, and t_f be the trajectory execution time. To achieve a smooth transition, a 5th order polynomial $\beta_1(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5$ is used when $t < t_b$. The decrement in $\beta(t)$ at the end of the trajectory, when $t > t_f - t_b$, is the complement of the polynomial $\beta_1(t) = 0$, $\beta_1(t) = 1 - \beta_1(t - t_f + t_b)$. By imposing the conditions $\beta_1(0) = 0$, $\beta_1(t_b) = 1$, $\dot{\beta}_1(0) = 0$, $\dot{\beta}_1(t_b) = 0$, $\ddot{\beta}_1(t_b) = 0$, $\ddot{\beta}_1(t_b) = 0$, the values of the coefficient of $\beta_1(t)$ are found.

After finding the polynomial coefficients, the self-motion magnitude variation factor is given by

$$\beta(t) = \begin{cases} a_3 t^3 + a_4 t^4 + a_5 t^5, & \text{if } t < t_b \\ 1, & \text{if } t_b \le t \le t_f - t_b \\ 1 - (a_3 (t - t_f + t_b)^3 + a_4 (t - t_f + t_b)^4 + \text{if } t > t_f - t_b, \\ a_5 (t - t_f + t_b)^5) \end{cases}$$

$$(24)$$

with:

$$a_0 = 0, \quad a_1 = 0, \quad a_2 = 0,$$

 $a_3 = \frac{10}{t_b^3}, \quad a_4 = -\frac{15}{t_b^4}, \quad a_5 = \frac{6}{t_b^5}.$

The blending time t_b is chosen by the user. Figure 3 shows the shape of $\beta(t)$ for $t_f = 15(s)$, $t_b = 2.5(s)$.



Figure 3. Example shape of self-motion variation factor β .

4.3.2. Joint Velocity Limits

To satisfy joint velocity limits, the maximum magnitude of self-motion is determined such that the velocity limit for each joint is not violated [18], namely:

$$|u_{i_v}(t) + \alpha \beta u_{i_h}(t)| \le u_i^+, \tag{25}$$

where u_i^+ is the *i*th joint velocity limit with $i \in 1, 2, ..., \delta_m$. Therefore, to avoid exceeding the joint velocity limits, $\alpha(t)$ must be selected such that it satisfies (25). The upper and lower limits of $\alpha(t)$ can be found using this equation. For each joint, it can be shown that the maximum and minimum values of $\alpha(t)$, denoted by $\alpha_{i_{max}}(t)$ and $\alpha_{i_{min}}(t)$, respectively, are given by

$$\alpha_{i_{max}}(t) = \max\left(\frac{u_i^+ - u_{i_p}(t)}{\beta(t)u_{i_h}(t)}, \frac{-u_i^+ - u_{i_p}(t)}{\beta(t)u_{i_h}(t)}\right),\tag{26}$$

$$\alpha_{i_{min}}(t) = \min\left(\frac{u_i^+ - u_{i_p}(t)}{\beta(t)u_{i_h}(t)}, \frac{-u_i^+ - u_{i_p}(t)}{\beta(t)u_{i_h}(t)}\right).$$
(27)

Then, $\alpha_{max}(t)$ and $\alpha_{min}(t)$ for all the joints are:

$$\alpha_{max}(t) = \min\left(\alpha_{1_{max}}(t), \alpha_{2_{max}}(t), \dots, \alpha_{\delta_{mmax}}(t)\right),\tag{28}$$

$$\alpha_{\min}(t) = \max\left(\alpha_{1_{\min}}(t), \alpha_{2_{\min}}(t), \dots, \alpha_{\delta_{m_{\min}}}(t)\right),\tag{29}$$

where $\alpha_{max}(t)$ is the self-motion magnitude limit. In [18], the maximum value of $\alpha_{max}(t)$ and minimum value of $\alpha_{min}(t)$ are calculated for the whole trajectory, and the upper bound of $\alpha_{max}(t)$ or lower bound of $\alpha_{min}(t)$ is used as the step size to take advantage of the maximum admissible velocities. In our approach, a suitable initial value α_s is selected

through experimentation and the upper and lower limits of α are calculated for each time *t*. The value of α at time *t* is then given by

$$\alpha(t) = \begin{cases} \alpha_{max}(t), & \text{if } \alpha_s > \alpha_{max}(t), \\ \alpha_{min}(t), & \text{if } \alpha_s < \alpha_{min}(t), \\ \alpha_s, & \text{otherwise.} \end{cases}$$
(30)

This technique, in contrast with using the maximum/minimum self-motion for the entire trajectory, prevents sudden joint accelerations due to the use of maximum velocities in every step. Note that for some cases $\alpha(t)$ may be negative if $\alpha_{max}(t)$ is negative, which means that the joint velocity limits cannot be satisfied without decreasing the manipulability of the system. The task can be executed as long as the system is still away from any singularity. The reader may have noticed that $\beta(t)$ is in the denominator of (26) and (27), and as shown in Section 4.3.1, $\beta(t)$ is zero at the beginning and the end of the trajectory. For these two cases, $\alpha_{min} = -\infty$ and $\alpha_{max} = \infty$, but this does not cause instability in the system because by following (30), the value of $\alpha(t)$ is set to α_s .

It is also possible to detect whether a task can be accomplished or not by using the limits of $\alpha(t)$. If the inequality $\alpha_{max}(t) < \alpha_{min}(t)$ is true, then a suitable value of $\alpha(t)$ which avoids the joint velocity limits does not exist, because even the particular solution will violate them. In other words, the given task space trajectory cannot be accomplished without violating at least one of the joint velocity limits. For these situations, the task space trajectory must be replanned with a longer execution time t_f or with lower end-effector's maximum velocities.

5. Experiments

Experiments were carried out to verify the efficacy of our scheme to solve the motion planning problem for trajectory tracking at the kinematic level. In this section, the results for the tracking of two trajectories, a Lissajous trajectory (see Section 5.4), and an elliptic trajectory (see Section 5.5), are analyzed. These trajectories were picked to demonstrate the ability of our approach to comply with the different constraints introduced in this manuscript while improving the manipulabilities of the whole system and the robot arm. For the Lissajous trajectory, a comparison of different objective functions for manipulability maximization is made to highlight the advantages of the proposed manipulability measure for mobile manipulators.

The experiments were performed using a 10-DOF NMM developed by the Industrial Technology Research Institute (ITRI), as shown in Figure 4. This NMM is composed of a differential-drive wheeled mobile platform, a prismatic joint mounted on top of the platform, and a Universal Robots UR5 6-DOF robotic arm attached to the prismatic joint. From this point forward, the UR5 arm's joints are denoted as q_{ai} with i = 1, 2, ..., 6 and the prismatic joint as z_{pj} . The respective Denavit–Hartenberg (DH) parameters are shown in Table 1. The joint limits and joint velocity limits are shown in Table 2.



Figure 4. Nonholonomic mobile manipulator used for the experiments.

Joint	<i>a</i> (m)	α (rad)	<i>d</i> (m)	θ (rad)
mob. plat.	0	0	0	θ_p
z_{pj}	-0.049	0	z_{pj} +0.5562	Ó
q_{a1}	0	$\pi/2$	0.08916	$q_{a1} + \pi$
9a2	-0.425	0	0	q _{a2}
9a3	-0.39225	0	0	q _a 3
q_{a4}	0	$\pi/2$	0.1093	q_{a4}
q_{a5}	0	$-\pi/2$	0.09465	q_{a5}

0.0823

0

Table 1. D-H parameters.

 q_{a6}

Table 2. Joints physical constraints.

0

Joint	q _{min}	q _{max}	<i>u_{max}</i>
v_p	$-\infty$	∞	0.3 (m/s)
ω_p	$-\infty$	∞	$\pi/2$ (rad/s)
z_{pi}	0 (m)	0.25 (m)	0.025 (m/s)
q_{a1}	-1.7453 (rad)	0.0175 (rad)	π (rad/s)
q _{a2}	$-\pi/2$ (rad)	0.4363 (rad)	π (rad/s)
q _a 3	0 (rad)	π (rad)	π (rad/s)
q_{a4}	-2π (rad)	2π (rad)	π (rad/s)
q_{a5}	-2π (rad)	2π (rad)	π (rad/s)
<i>q</i> _{a6}	-2π (rad)	2π (rad)	π (rad/s)

The Jacobian of the arm J_a used for calculation of $\hat{\Omega}_a$ in (15) was constructed only using the UR5 arm without the prismatic joint. If the prismatic joint were included, the arm would be allowed to stretch for most tasks since it would always be able to move the end-effector vertically using the prismatic joint, i.e., the manipulability is not affected when the arm is horizontally stretched. Stretching the arm for most tasks is an undesirable behavior, and therefore, the manipulability maximization of the UR5 arm without the prismatic joint is a suitable selection. The Jacobian J_a is calculated with respect to the frame $X_a Y_a Z_a$ shown in Figure 4.

Due to the lack of a reliable positioning system, the odometry of the wheels was used in the experiments to compute the position and orientation of the mobile platform, and the forward kinematics were used to compute the end-effector's pose, which in turn was fed back to the motion planning algorithm. Therefore, the errors presented in the experiments are not the real-world errors, but the errors in the trajectory in which it is assumed that a reliable positioning system exists. Furthermore, as mentioned in the introduction section, the scope of this work is the motion planning of NMM for trajectory tracking at the kinematic level; therefore, the objective of the presented experiments is to validate the proposed algorithm at the kinematic level. Problems inherent to the dynamic behavior of the system, including vibrations of the mechanical structure, friction from the ground, etc., have an impact on the real end-effector tracking error, but they are out of scope of this paper and not considered in the algorithm. For these reasons, the simulation results of the position and orientation errors along the trajectory are also shown in this manuscript to highlight the performance of our motion planning algorithm.

An important remark is that the system vibrations due to the NMM mechanical structure greatly affected the performance of the motion planning algorithm when the system was close to its joint limits or self-collision constraints. This behavior was not observed when performing the simulations. To address this issue, a moving average filter with a window size of five was used to filter the gradients of the criterion functions for joint limits avoidance and self-collision avoidance. This filter diminished the impact of such vibrations on the motion planning algorithm.

 q_{a6}

5.1. Orientation Error for 6-DOF Tasks

In all the experiments, 6-DOF tasks were performed. For a 6-DOF task (m = 6), where the position and orientation of the end-effector are considered, the expression $r_d - r$ (mentioned in Section 3) has a specific definition that depends on the orientation representation, i.e., $\tilde{r} = r_d - r$ does not hold for all orientation representations [1]. In this work, unit quaternions are used to describe the end-effector's orientation because of their efficiency and nonsingular representation for all orientations [1,19–21].

A unit quaternion Q = [w + xi + yj + zk] is represented in scalar-vector form by $Q = \{s, v\}$ with $s \in \mathbb{R}$ and $v \in \mathbb{R}^3$, where *s* and *v* are called the scalar and vector elements of *Q*, respectively. The desired and current pose are defined using unit quaternions for orientation as $\mathbf{r}_d = [\mathbf{P}_d \quad Q_d]^T$ and $\mathbf{r}_c = [\mathbf{P}_c \quad Q_c]^T$, where $\mathbf{P}_d = [x_d, y_d, z_d]$ and $\mathbf{P}_c = [x_c, y_c, z_c]$ are the desired position and current position, respectively, and $Q_d = \{s_d, v_d\}$ and $Q_c = \{s_c, v_c\}$ are the desired orientation and current orientation, respectively. The position error \mathbf{e}_P is defined as $\mathbf{e}_P = \mathbf{P}_d - \mathbf{P}_c$. The orientation error can be described in terms of the quaternion $\Delta Q = \{\Delta s, \Delta v\}$, where [1]:

$$\Delta s = s_c s_d - v_d^T v_c,$$

$$\Delta v = s_c v_d - s_d v_c - v_d \times v_c.$$
(31)

If the desired orientation and current orientation are aligned, i.e., with zero orientation error, then $\Delta Q = \{1, 0\}$. Consequently, it would be sufficient to define the orientation error to be Δv . It is also important to follow a convention for the sign of the quaternion because $Q = \{s, v\}$ and $-Q = \{-s, -v\}$ represent the same orientation. A common convention is to keep the scalar quaternion element positive. Take this into account and the orientation error is defined as follows:

$$\boldsymbol{e}_{O} = \begin{cases} \boldsymbol{\Delta}\boldsymbol{v}, & \text{if } \Delta s \ge 0\\ -\boldsymbol{\Delta}\boldsymbol{v}, & \text{if } \Delta s < 0. \end{cases}$$
(32)

Separating the position and orientation errors, the motion planning control law (6) is rewritten as

$$\dot{\boldsymbol{r}} = \dot{\boldsymbol{r}}_d + \begin{bmatrix} K_P & 0\\ 0 & K_O \end{bmatrix} \begin{bmatrix} e_P\\ e_O \end{bmatrix},\tag{33}$$

where K_P and K_O are positive definite diagonal 3 × 3 matrices. Note that e_O in (32) is not an angle difference but its size represents the size of the orientation error that, as shown in [1,22], can achieve the convergence of the orientation error.

5.2. Evaluated Self-Collision Pairs

The types of self-collision can be significantly reduced by setting the joint limits properly. In addition, collisions among the first three links and the last three links can be taken care of by maximizing the arm manipulability, because the 6-DOF manipulator has poor manipulability if the arm is retracted to the point where the wrist is close to the base of the arm. Therefore, only the self-collision between the arm and the mobile platform needs consideration. Such a type of collision takes place when the elbow of the manipulator collides with the top of the platform, or when the wrist collides with the front of the platform. The distances associated with these potential self-collisions are depicted in Figure 5. Given that the platform is fixed with respect to the arm, setting frame $X_p Y_p Z_p$ (located on the center of the wheels' axle, as shown in Figure 4) as the reference frame allows to use (22), where p_{l1} is a point in the arm's elbow or wrist, correspondingly, and p_{l2} is a fixed point with respect to $X_p Y_p Z_p$:



Figure 5. NMM's self-collision distances.

As illustrated in Figure 5, to prevent the elbow collision, a collision pair between the *z* component of p_{elbow} with respect to frame $X_pY_pZ_p$ and a point located at 0.5 (m) from the origin of frame $X_pY_pZ_p$ in the Z_p direction is selected. The distance for this pair is named d_{elbow} . To prevent wrist collision, a collision pair between the *x* component of p_{wrist} with respect to frame $X_pY_pZ_p$ and a point located at 0.37 (m) from the origin of frame $X_pY_pZ_p$ in the X_p direction is selected. The distance for this pair is named d_{wrist} . The wrist collision is only evaluated if the wrist height h_{wrist} , the *z* component of p_{wrist} with respect to frame $X_pY_pZ_p$, is lower than 0.5 (m), otherwise, its associated weighting collision matrix is assigned to identity. This was done to avoid restricting the movement of the joints that reduce d_{wrist} when the wrist is above the top of the mobile platform. All these parameters were chosen based on the physical dimensions of the NMM, and the second point in each of the collision pairs was selected so that when d_{elbow} and d_{wrist} are zero, a gap still exists between the potentially colliding links.

5.3. Common Parameters of the Simulations and Experiments

For all the experiments, the selected feedback gain matrices in (33) are $K_P = 10I_{3\times3}$ and $K_O = 20I_{3\times3}$, where *I* is an identity matrix. The initial value of $\alpha(t)$ in (30) is set to $\alpha_s = 3$, and the blending time is set to $t_b = 0.2t_f$ for the variation factor $\beta(t)$ in (24). The parameters for self-collision avoidance in (20) are $\rho = 1 \times 10^{-3}$, $c_1 = 50$ and $c_2 = 1$. The starting configuration of the robot arm is $q_a = \begin{bmatrix} 0 & -80 & 110 & -120 & -90 & 0 \end{bmatrix}$ (°). Furthermore, a sampling time $t_s = 0.02(s)$ was used for the simulations and experiments.

5.4. Lissajous Trajectory

The Lissajous trajectory was picked to demonstrate the ability of the proposed approach to track a trajectory for which the nonholonomic constraints greatly affect the movement of the system. The proposed Lissajous trajectory is defined by

$$\mathbf{r}_{d}(t) = \begin{bmatrix} \mathbf{P}_{d}(t) \\ Q_{d}(t) \end{bmatrix} = \begin{bmatrix} x_{0} \\ y_{0} \\ z_{0} \\ Q_{0} \end{bmatrix} + \begin{bmatrix} A\cos(s(t) + \pi/2) \\ B\cos(2(s(t) + \pi/2) + \pi/2) \\ C\cos(2s(t)) - C \\ 0 \end{bmatrix},$$

where $[x_0, y_0, z_0]^T$ is the end-effector's starting position, s(t) is the trajectory timing variable, and *A*, *B* and *C* define the size of the path. The orientation is set to be the same for the entire

trajectory, i.e., $Q(t) = Q_0$. Notice that this is a 6-DOF trajectory because the orientation is constrained for the entire trajectory.

The starting configurations of the mobile platform and prismatic joint for this trajectory were set to $x_p = -0.1$ (m), $y_p = -0.13$ (m), $\theta_p = -90(^{\circ})$ and $z_{pj} = 0.2$ (m). With this configuration, the end-effector's initial pose is given by $P_0 = \begin{bmatrix} -0.009 & -0.6491 & 0.9884 \end{bmatrix}$ (m) and $Q_0 = \{0, 0i + 1j + 0k\}$. A trapezoidal velocity profile [1] was used for the derivative of the timing variable $\dot{s}(t)$. The parameters for the path size were set to A = 1.3 (m), B = 1.3 (m) and C = 0.27 (m), and an execution time $t_f = 64(s)$ was chosen. Figure 6a,b show snapshots of the NMM's movement along the Lissajous trajectory, in simulations and experiments, respectively.

Figure 7 compares the trajectory tracking results between the simulation and the experiment. In the simulation, the position and orientation errors are kept small during the whole trajectory, as shown in Figure 7b,c. This demonstrates the good tracking performance at the kinematic level of our proposed motion planning algorithm. The observed larger errors in the experiments, as exhibited in Figure 7e,f, are due to the vibrations of the system during the experiments. Furthermore, as mentioned before, the control of the dynamic behavior of the system is beyond the scope of this work. Nevertheless, the position errors in the experiments are kept below 2×10^{-3} (m) and the orientation errors below 1.5×10^{-3} . Likewise, we observe that the obtained trajectories in the simulation (Figure 7a) and the experiment (Figure 7d) are fairly similar.

Figure 8 illustrates the experiment results. Note that the negative joint velocity limit $-u_i^+$ is denoted as u_i^- in the pertinent figures. The manipulabilities of both the arm and the complete system are kept high during the execution of the trajectory, and their final values are higher than at the start of the trajectory, as shown in Figure 8b. It is important to remark that there are no potential self-collisions during the execution of this trajectory. Even though d_{wrist} is negative at the beginning of the trajectory, as shown in Figure 8c, the wrist is above the top of the mobile platform and therefore the joint movements were not restricted, as explained in Section 5.2.



Figure 6. Snapshots of the NMM's motion for the Lissajous trajectory tracking: (a) simulations; and (b) experiments.



Figure 7. Tracking performance comparison between simulations and experiments for the Lissajous trajectory. End-effector's trajectory in simulations (**a**) and experiments (**d**). Position errors in simulations (**b**) and experiments (**e**). Orientation errors in simulations (**c**) and experiments (**f**). Notice that the position and orientation errors, panels (**b**,**c**), respectively, are small in the simulations. In addition, notice that the trajectories in simulations (**a**) and experiments (**b**) are quite similar. The reason for the larger errors obtained in the experiments (panel (**e**,**f**)) are discussed in Section 5.4.

Figure 8d–i demonstrate the fulfillment of the joint limits and joint velocity boundary constraints. All the joints are kept within their corresponding limits. Notice that the movement of q_{a1} , Figure 8f, is restricted in the time interval t = (29, 32)(s) to respect its lower limit. This time interval corresponds to the trajectory section between snapshots number three and four in both Figure 6a,b. For this section of the trajectory, the movement of the end-effector in the XY plane is taken care of by the platform due to the restriction imposed to q_{a1} . We observe in Figure 8e that the velocity limits of the prismatic joint are respected. The remaining joints do not reach their respective limits as shown in Figure 8f–h. Furthermore, as seen in Figure 8d,e,i, the velocity profiles for all the joints are smooth and satisfy the boundary constraints, i.e., the initial and final joint velocities are equal to zero.

To demonstrate the advantages of the proposed manipulability measure Ω_{MM} , a comparison of the manipulability maximization results using different objective functions in simulations of the Lissajous trajectory tracking is shown in Figure 9. Here, a task is considered as failed when none of the constraints are satisfied, and thus the simulation is stopped. These results were obtained by using the same parameters shown in Section 5.3 with the only change being the objective function.

The Lissajous trajectory tracking fails when the objective function is set to maximize the manipulability of the arm, i.e., $F(q) = \hat{\Omega}_a$, as shown in Figure 9a. The manipulabilities of both the arm and the whole system start a fast decay after t = 44s. This decrement in the manipulabilities is the result of restricting the homogeneous solution to respect the joint velocity limits. Consequently, the arm manipulability is not maximized anymore and the system moves towards singularity, ultimately failing the task. The results of setting the objective function to the manipulability of the whole system, i.e., $F(q) = \hat{\Omega}_{p+a}$, are shown in Figure 9b. In this case, the task succeeds, however, the manipulability of the arm



deteriorates, and at the end of the trajectory has a value close to zero. This is the behavior discussed in previous studies [6,10].

Figure 8. Experiment results for the Lissajous trajectory tracking: (**a**) mobile platform's trajectory; (**b**) normalized manipulability measures; (**c**) self-collision distances; (**d**) mobile platform's velocity commands; (**e**) prismatic joint's position and velocity; (**f**–**h**) arm's joint positions; and (**i**) arm's joint velocities. Notice that all the joint limits and velocity limits are respected (panels (**d**–**i**)). Furthermore, the manipulabilities of both the arm and the whole system are improved (panel (**b**)). See Section 5.4 for a detailed discussion of this figure.

Figure 9c shows the results of maximizing a linear combination of the manipulabilities of the arm and the whole system, i.e., $F(q) = 0.5\hat{\Omega}_{p+a} + 0.5\hat{\Omega}_a$. Figure 9d shows the results of maximizing the proposed manipulability measure for mobile manipulators, i.e., $F(q) = \Omega_{MM}$ from Equation (15). Both objective functions preserve the manipulability of the arm. However, it is clear that maximizing the proposed measure has better results overall. First, the manipulability of the arm is higher along the trajectory in Figure 9d compared to the arm manipulability obtained with the linear combination in Figure 9c. Secondly, the obtained final manipulabilities have improved with respect to the initial ones in Figure 9d. On the other hand, the linear combination improves the manipulability of the whole system, but not that of the arm with respect to the values at the start of the trajectory, as shown in Figure 9c.



Figure 9. Lissajous trajectory's comparison of objective functions for manipulability maximization in simulations: (**a**) maximization of the manipulability of the arm; (**b**) maximization of the manipulability of the whole system; (**c**) maximization of a linear combination of the manipulability of the arm and the whole system; and (**d**) maximization of the proposed mobile manipulability measure. This figure demonstrates the advantages of the manipulability measure for mobile manipulators (panel (**d**)) presented in Section 4.1. See Section 5.4 for a detailed discussion of this figure.

5.5. Elliptic Trajectory

The elliptic trajectory was picked to demonstrate the ability of the proposed approach to comply with joint velocity limits and to prevent self-collisions. This trajectory consists of moving the end-effector from an initial pose $r_0 = \begin{bmatrix} P_0 & Q_0 \end{bmatrix}^T$ to a desired final pose $r_d = \begin{bmatrix} P_d & Q_d \end{bmatrix}^T$ using an elliptic path. The trajectory position is defined by

$$\mathbf{P}_{\mathbf{d}}(t) = \begin{bmatrix} A\cos(s(t)) + c_x \\ B\sin(s(t)) + c_y \\ m(s(t) - s_0) + z_0 \end{bmatrix},$$

where A, B, c_x , c_y and s_0 are calculated using the XY coordinates of P_0 and P_d , such that the elliptic path is centered at the closest point to the origin, while it covers a 90° angle between the start and end points. The trajectory's Z coordinate follows a straight line between the points (z_0, s_0) and (z_d, s_d) , where z_0 and z_d are the Z coordinates of P_0 and P_d , respectively; s_0 and s_d are the starting and ending angles of the elliptic path, respectively; and m in the equation above is the slope of this line. A fifth-order polynomial profile [1] was used for the timing variable s(t). The orientations and rotational velocities along the trajectory were computed using quaternion polynomials [23]. This technique has two main advantages: a smooth velocity profile is obtained, and the rotational velocities and accelerations in the task space are included as boundary constraints.

The starting configuration of the mobile platform and prismatic joint for this trajectory was set to $x_p = -1.3$ (m), $y_p = 0.56$ (m), $\theta_p = 0(^{\circ})$, and $z_{pj} = 0.24$ (m). With this configuration, the end-effector's initial pose is given by $P_0 = \begin{bmatrix} -0.781 & 0.669 & 1.028 \end{bmatrix}$ (m) and $Q_0 = \{0, 0.7071i - 0.7071j + 0k\}$. The final pose was selected to have position $P_d = \begin{bmatrix} 1.55 & -1.0 & 0.26 \end{bmatrix}$ (m) and orientation $Q_d = \{0.2706, 0.6533i + 0.6533j - 0.2706k\}$, and an execution time $t_f = 20(s)$ was chosen. Figure 10a,b show snapshots of the NMM's movement along the elliptic trajectory, in simulations and experiments, respectively.

Figure 11 compares the trajectory tracking results between the simulation and the experiment. The small position and orientation errors obtained in the simulation (Figure 11b,c) again demonstrate the good tracking performance of the proposed approach. In the experiment, the position errors are kept below 1.5×10^{-3} (m) and the orientation errors are kept below 1×10^{-3} as depicted in Figure 11e,f. Once again, the vibrations of the system played a role in the larger errors seen in the experiment.







(b)

Figure 10. Snapshots of the NMM's motion for the elliptic trajectory tracking: (a) simulations; and (b) experiments.





Figure 11. Tracking performance comparison between simulations and experiments for elliptic trajectory. End-effector's trajectory in simulations (**a**) and experiments (**d**). Position errors in simulations (**b**) and experiments (**e**). Orientation errors in simulations (**c**) and experiments (**f**). Notice that the position and orientation errors, panels (**b**,**c**), respectively, are small in the simulations. In addition, notice that the trajectories in simulations (**a**) and experiments (**b**) are quite similar. The reason for the larger errors obtained in the experiments (panel (**e**,**f**)) are discussed in Section 5.5.

Figure 12 illustrates the experiment results. The manipulabilities of both the arm and the complete system are again improved at the end of the trajectory compared to the initial configuration, as shown in Figure 12b. However, we notice how both manipulabilities decreased during the time interval t = (8, 15)(s). This behavior is due to the value of $\alpha(t)$ (manipulability maximization step size) being negative for this span of time in order to respect the joint velocity limits, as discussed in Section 4.3.2. Notice that v_p (Figure 12d) and \dot{z}_{pj} (Figure 12e) are kept at their maximum values during this span of time. It would not be possible to track this particular trajectory while respecting the joint velocity limits without stretching the arm; hence, the manipulabilities of the complete system and the arm decreased.

We observe in Figure 12c how the self-collision distance of the elbow gets close to zero. In this trajectory, the elbow needs to get close to the platform in order for the end-effector to track the desired trajectory. The motion planning algorithm gradually stopped the elbow to prevent the collision with the platform, as shown by the slow decay of d_{elbow} . To achieve this, the movement of the prismatic joint was restricted, as shown in Figure 12e. The wrist also moves towards the front of the platform, as shown by the decrement in d_{wrist} in Figure 12c. This potential self-collision is also prevented by limiting the movement of q_{a2} and q_{a3} , as shown in Figure 12f,g.

Figure 12d–i show that the joint limits, joint velocity limits and joint velocity boundary constraints are also satisfied. All the joints were kept within their respective limits, as shown in Figure 12e–h. As depicted in Figure 12d,e, the platform's linear velocity and the prismatic joint's velocity are kept within their limits. Once more, the velocity profiles are smooth and the initial and final velocities for all the joints are zero.



Figure 12. Experiment results for the elliptic trajectory tracking: (**a**) mobile platform's trajectory; (**b**) normalized manipulability measures; (**c**) self-collision distance; (**d**) mobile platform's velocity commands; (**e**) prismatic joint's position and velocity; (**f**–**h**) arm's joint positions; (**i**) arm's joint velocities. Notice that all the joint limits and velocity limits are respected (panels (**d**–**i**)). The potential self-collisions described in Section 5.2 are prevented (panel (**c**)). Furthermore, the manipulabilities of both the arm and the whole system are improved (panel (**b**)). See Section 5.5 for a detailed discussion of this figure.

6. Discussion

A scheme was proposed to solve the motion planning of NMMs considering joint limits, joint velocity limits, joint velocity boundary constraints, and self-collision avoidance while maximizing the manipulabilities of both the robot arm and the whole system.

The proposed solution uses a weighted input velocity vector and a weighted Jacobian to penalize the movement of joints that get close to a position constraint. A proposed quadratic cost function is minimized when solving the motion planning problem for redundant NMMs. This cost function includes a secondary task to be satisfied that is also weighted to comply with the position constraints. The maximization of an introduced manipulability measure for mobile manipulators is used as the secondary task to push the system away from singularities. In the experiments section, it is demonstrated that maximizing this new measure simultaneously improves the manipulability of the whole system and that of the manipulator alone.

This work focuses on the motion planning for trajectory tracking at the kinematic level, which must not only comply with joint position constraints, but must also respect joint velocity constraints and joint velocity boundary constraints. Joint velocity boundary constraints are satisfied by varying the magnitude of the homogeneous solution at the start and end of the trajectory. The manipulability maximization at these points is not necessarily zero; hence, using an adequate variation in the magnitude of self-motion is needed. Joint velocity limits are satisfied by evaluating the maximum allowable self-motion for each joint. Using this information, the step size of the gradient ascent/descent is limited when required, and consequently, the joint velocity limits are not exceeded.

The experiments for 6-DOF trajectories were conducted to verify the efficacy of the proposed scheme. The results demonstrate that the proposed approach can solve the motion planning problem for NMMs to perform trajectory tracking at the kinematic level while considering the constraints required for real implementation including manipulation capability preservation or improvement.

The experiments designed in Section 5 consider an open environment without obstacles, because this is the scope of our manuscript. However, the proposed solution can be extended to prevent collisions with obstacles by including collision pairs between the robot arm and these obstacles, using the same definitions as in Section 4.2.2. This will penalize the movement of the arm's joints that get closer to an obstacle in the environment. Nevertheless, in the case of the platform, stopping it is not an efficient approach. In this case, an additional task can be added to the solution to push the platform away from the obstacles. One way to achieve this is by using a task priority scheme [24] using the nullspace of the motion planning algorithm.

Even though our work focuses on NMMs, our approach can be effortlessly adapted for use with holonomic mobile manipulators. Future work will focus on dynamic modeling and controller design to deal with system vibrations and tire slip such that the tracking errors of the system can be reduced.

Author Contributions: Conceptualization, methodology, J.L. and T.H.; software, validation, visualization, writing—original draft preparation, J.L.; supervision, project administration, funding acquisition, resources, writing—review and editing, T.H. Both authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Ministry of Science and Technology, Taiwan (Project code MOST 107-2923-E-009-004-MY3).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors wish to thank the Mechanical and Mechatronics Systems Research Laboratories from the Industrial Technology Research Institute (ITRI), Taiwan, for the support and resources provided for this work.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

- 1. Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. *Robotics: Modelling, Planning and Control*, 2009th ed.; Springer: London, UK, 2009; p. 632.
- Liao, J.; Huang, F.; Chen, Z.; Yao, B. Optimization-based motion planning of mobile manipulator with high degree of kinematic redundancy. *Int. J. Intell. Robot. Appl.* 2019, 3, 115–130. [CrossRef]
- Santos, P.C.; Freire, R.C.S.; Carvalho, E.A.N.; Molina, L.; Freire, E.O. M-FABRIK: A New Inverse Kinematics Approach to Mobile Manipulator Robots Based on FABRIK. *IEEE Access* 2020, *8*, 208836–208849. [CrossRef]
- 4. Campion, G.; Bastin, G.; Dandrea-Novel, B. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Trans. Robot. Autom.* **1996**, *12*, 47–62. [CrossRef]
- Bayle, B.; Fourquet, J.Y.; Renaud, M. Manipulability of Wheeled Mobile Manipulators: Application to Motion Generation. Int. J. Robot. Res. 2003, 22, 565–581. [CrossRef]
- Bayle, B.; Renaud, M.; Fourquet, J.Y. Nonholonomic mobile manipulators: Kinematics, velocities and redundancies. J. Intell. Robot. Syst. Theory Appl. 2003, 36, 45–63. [CrossRef]

- De Luca, A.; Oriolo, G.; Giordano, P.R. Kinematic modeling and redundancy resolution for nonholonomic mobile manipulators. Proc. IEEE Int. Conf. Robot. Autom. 2006, 2006, 1867–1873.
- 8. Huang, Q.; Tanie, K.; Sugano, S. Coordinated Motion Planning for a Mobile Manipulator considering Stability and Manipulation. *Int. J. Robot. Res.* **2000**, *19*, 732–742. [CrossRef]
- Jia, Y.; Xi, N.; Cheng, Y.; Liang, S. Coordinated motion control of a nonholonomic mobile manipulator for accurate motion tracking. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 1635–1640.
- Zhang, Y.; Yan, X.; Chen, D.; Guo, D.; Li, W. QP-based refined manipulability-maximizing scheme for coordinated motion planning and control of physically constrained wheeled mobile redundant manipulators. *Nonlinear Dyn.* 2016, *85*, 245–261. [CrossRef]
- Chan, T.F.; Dubey, R.V. A Weighted Least-Norm Solution Based Scheme for Avoiding Joint Limits for Redundant Joint Manipulators. *IEEE Trans. Robot. Autom.* 1995, 11, 286–292. [CrossRef]
- 12. Yoshikawa, T. Foundations of Robotics: Analysis and Control; MIT Press: Cambridge, MA, USA , 1990.
- Lee, J. A study on the manipulability measures for robot manipulators. In Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97, Grenoble, France, 11 September 1997; Volume 3, pp. 1458–1465.
- 14. Patel, S.; Sobh, T. Manipulator Performance Measures—A Comprehensive Literature Survey. J. Intell. Robot. Syst. 2015, 77, 547–570. [CrossRef]
- 15. Staffetti, E.; Bruyninckx, H.; De Schutter, J. *Advances in Robot Kinematics: Theory and Applications*; Lenarčič, J., Thomas, F., Eds.; Spring: Dordrecht, The Netherlands, 2002; p. 536.
- Dariush, B.; Zhu, Y.; Arumbakkam, A.; Fujimura, K. Constrained closed loop inverse kinematics. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 2499–2506.
- Zghal, H.; Dubey, R.V.; Euler, J.A. Efficient gradient projection optimization for manipulators with multiple degrees of redundancy. In Proceedings of the IEEE International Conference on Robotics and Automation, Cincinnati, OH, USA, 13–18 May 1990; pp. 1006–1011.
- 18. Euler, J.A.; Dubey, R.V.; Babcock, S.M. Self motion determination based on actuator velocity bounds for redundant manipulators. *J. Robot. Syst.* **1989**, *6*, 417–425. [CrossRef]
- 19. Chiaverini, S.; Siciliano, B. The unit quaternion: A useful tool for inverse kinematics of robot manipulators. *Syst. Anal. Model. Simul.* **1999**, *35*, 45–60.
- Dam, E.B.; Koch, M.; Lillholm, M. Quaternions, Interpolation and Animation; Technical Report DIKU-TR-98/5; Department of Computer Science, University of Copenhagen: Copenhagen, Denmark, 1998; p. 103.
- Funda, J.; Taylor, R.; Paul, R. On homogeneous transforms, quaternions, and computational efficiency. *IEEE Trans. Robot. Autom.* 1990, *6*, 382–388. [CrossRef]
- 22. Yuan, J. Closed-loop manipulator control using quaternion feedback. IEEE J. Robot. Autom. 1988, 4, 434–440. [CrossRef]
- Shahbazi, M.; Kashiri, N.; Caldwell, D.; Tsagarakis, N. Orientation planning in task space using quaternion polynomials. In Proceedings of the 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), Macau, Macao, 5–8 December 2017; pp. 2343–2348.
- 24. Flacco, F. The tasks priority matrix: A new tool for hierarchical redundancy resolution. In Proceedings of the 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), Cancun, Mexico, 15–17 November 2016; pp. 1–7.