A Global Approach to Manipulability Optimisation for a Dual-Arm Manipulator

Marco Faroni[1], Manuel Beschi[2], Antonio Visioli[1], Lorenzo Molinari Tosatti[2]

[1] Dipartimento di Ingegneria Meccanica e Industriale
University of Brescia, Brescia, Italy
Email: {m.faroni003, antonio.visioli}@unibs.it
[2] Istituto di Tecnologie Industriali e Automazione
National Research Council, Milan, Italy
Email: {manuel.beschi, lorenzo.molinaritosatti}@itia.cnr.it

Abstract— **In this paper, we present a new approach to manipulability maximisation for a dual-arm manipulator, which takes into account the manipulability of the overall task. This method tries to overcome the drawbacks given by traditional approaches, which optimise the manipulability of the local configuration of the manipulator, but do not take into account the rest of the task, even though it is known a priori. In this way, it is possible to improve the average manipulability index over the task. The method is applied to a dual-arm system, wherein the task is expressed in terms of relative poses between the end-effectors. For this reason, the kinematic of the system is solved by means of the relative Jacobian.**

I. Introduction

Cooperative manipulators have been deeply investigated due to the advantages they guarantee respect to a single-arm system. In particular, their use allows larger payloads and assembly operations that would be infeasible for a single robot [1]. Furthermore, multi-arm systems usually present intrinsic redundancy, which may be exploited to achieve secondary goals through optimisation [2]. The drawback is a greater complexity in control and planning, which have to deal with extra degrees of freedom and the choice of appropriate cost functions.

Many approaches have been proposed that define a performance index to be optimised in the redundancy resolution.

For cooperative multi-robot systems, resolution criteria have been proposed that properly redistribute the torque over each manipulator [3] or that maximise compliance during the task [4]. Other examples exploit the redundancy to ensure joint-range availability, obstacle avoidance [5], or to generate human-inspired motion [6] [7].

In particular, a great effort has been put in extending the concept of manipulability from single to multi-arm systems. In [8], the author approximates the manipulability ellipsoid of a dual-arm robot with the intersection of the single-arm ellipsoids. The approach proposed in [9] extends the definition of manipulability ellipsoid to the multi-arm case for cooperative manipulation tasks by means of a closed-chain kinematic analysis. Further contributions have been presented in [10] and [11], which analyse manipulability in actuated and non-actuated closed-chain mechanism.

A particular case occurs when the assigned task of a dual-arm manipulator may be expressed in terms of relative poses between the two end-effectors. As shown in Section II-B, this allows the reduction of the dual-arm system to a single-arm-like manipulator. In this way, the manipulability ellipsoids may be directly derived from the so-called relative Jacobian (see [12]). This strategy has been applied, for instance, in [2] and [13].

A peculiarity of the relative-task approach is the expression of the motion with respect to the end-effectors only. As a consequence, since the system is usually redundant, the motion with respect to the world frame is not univocally defined, and the end-effectors assume different poses depending on the redundancy-resolution criteria.

In any case, the typical approach to redundancy resolution consists in a punctual optimisation, in the sense that the resolution tends to minimise the performance index relative to the actual configuration of the system, as in [13] and [14]. This is the case, for instance, of the pseudo-inverse method presented in Section II. The great advantages of these methods are low computational burden and good effectiveness in typical applications. On the other hand, they do not ensure that the mean value of the index is globally optimum. In other words, the solution may be the best for the actual configuration, but the average value of the performance index along the task might not be the optimal one.

A better approach should perform a one-step optimisation to find the configurations that minimise the chosen index throughout the whole task.

This paper, indeed, proposes an algorithm that solves the redundancy issue for a dual-arm system, maximising the manipulability of the whole task, when the desired path between the end-effectors is known a priori. In other words, this method takes into account not only the current configuration of the system, but also the previous and next ones, based on the known path.

A straightforward solution to this issue would be the optimisation of a cost function composed by the sum of the performance indexes relative to every point of the path. Since the resulting computational burden of such approach would be too heavy, an approximated method is proposed.

In particular, the optimisation splits apart into two steps: the first one optimises the sum of the indexes of a certain number of points along the path, whereas the second one optimises the inner points, between the formers.

In this way, the method does not take into account just the punctual configuration of the system, but it tends to maximise the whole task, considering previous and successive configurations as well.

The paper is organised as follows. In Section II, the theoretical background of manipulability and its extension to dual-arm manipulators is presented. In particular, the case of relative tasks and the use of the relative Jacobian is illustrated. Section III discusses the proposed method, illustrating the performance index and the optimisation problem to solve. A simulated implementation of the method is proposed in Section IV, whereas Section V compares the results with the ones given by the typical pseudo-inverse-based method. Finally, conclusions are exposed in Section VI.

II. Background

A. Manipulability measure

As mentioned in the previous section, manipulability has been extended to dual arm manipulators in different ways (e.g. [8], [9]). Many contributions recall the concept of manipulability ellipsoid, developed for a single arm in [15].

Briefly, given an n-degree-of-freedom manipulator and an m-dimensional task, the differential kinematics gives:

$$\mathbf{v} = \mathbf{J}\,\dot{\mathbf{q}} \tag{1}$$

where $\mathbf{v} \in \mathbb{R}^m$ is the task velocity vector, $\dot{\mathbf{q}} \in \mathbb{R}^n$ is the joint velocity vector, and $\mathbf{J}$ is the $m \times n$ Jacobian matrix. Thus, the set of joint velocities with norm equal to one is represented by the unit sphere:

$$\dot{\mathbf{q}}^T \dot{\mathbf{q}} = 1 \tag{2}$$

which is mapped into an ellipsoid in the task space:

$$\mathbf{v}^T (\mathbf{J}\mathbf{J}^T)^{-1} \mathbf{v} = 1 \tag{3}$$

Exploiting the kineto-static duality, the set of joint torques with unitary norm is given by:

$$\tau^T \tau = 1 \tag{4}$$

from which:

$$\gamma^T (\mathbf{J}\mathbf{J}^T) \gamma = 1 \tag{5}$$

where τ and γ are the joint and task forces respectively.

The former ellipsoid is usually referred to as *velocity ellipsoid*, whereas the latter as *force ellipsoid*.

In the case of a 6-dof manipulator, it is easy to notice that the first three rows of the Jacobian refer to the linear velocity transformation, whereas the last three rows refer to the angular one. This means that a manipulability ellipsoid may be decomposed into two further ellipsoids, referring to the linear and the angular velocity respectively.

Moreover, given a unit vector $\mathbf{u}$ representing a Cartesian direction, the intersection of the force/velocity ellipsoids with $\mathbf{u}$ gives respectively:

$$\alpha = \left(\mathbf{u}^T (\mathbf{J}\mathbf{J}^T) \mathbf{u}\right)^{-1/2} \tag{6}$$

$$\beta = \left(\mathbf{u}^T (\mathbf{J}\mathbf{J}^T)^{-1} \mathbf{u}\right)^{-1/2} \tag{7}$$

A meaningful interpretation of (6) and (7) is the force/velocity transmission ratio of the manipulator. This means that, given a Cartesian task, (6) and (7) may be exploited to maximise the manipulability of the robot along a desired direction.

Unfortunately, this definition of α and β is not physically consistent, since it involves both linear and angular velocities with different units of measurement (see [16]). On the other hand, its use as cost function may be considered as a multi-objective optimisation problem composed of the linear and angular terms, weighted by a suitable diagonal matrix $\mathbf{W}$:

$$\bar{\alpha} = \left(\mathbf{u}^T (\mathbf{J}\mathbf{W}\mathbf{J}^T) \mathbf{u}\right)^{-1/2} \tag{8}$$

$$\bar{\beta} = \left(\mathbf{u}^T (\mathbf{J}\mathbf{W}\mathbf{J}^T)^{-1} \mathbf{u}\right)^{-1/2} \tag{9}$$

B. Relative Jacobian and manipulability

When the desired task of a dual-arm system may be described in terms of relative poses between the two end-effectors, the extension of manipulability ellipsoids comes straightforwardly.

As a matter of fact, in this case a transformation matrix $\mathbf{T}_{e_1e_2} \in \mathbb{R}^{4\times4}$ may be defined, which expresses the pose of the second end-effector frame with respect to the first one. Let $\mathbf{T}_{b_1e_1}, \mathbf{T}_{b_2e_2} \in \mathbb{R}^{4\times4}$ be the transformation matrices of the first and second end-effectors with respect to their base and $\mathbf{T}_{b_1b_2} \in \mathbb{R}^{4\times4}$ the transformation matrix between the bases (see Figure 1), it results:

$$\mathbf{T}_{e_1e_2} = \mathbf{T}_{b_1e_1}^{-1} \mathbf{T}_{b_1b_2} \mathbf{T}_{b_2e_2} \tag{10}$$

Moreover, a relative Jacobian $\mathbf{J}_r$ may be defined that represents a linear transformation between joint and task velocities, such that:

$${}^{e_1}\mathbf{v}_r = \mathbf{J}_r\,\dot{\mathbf{q}} \tag{11}$$

where ${}^{e_1}\mathbf{v}_r = [\,\dot{x} \quad \dot{y} \quad \dot{z} \quad \omega_x \quad \omega_y \quad \omega_z\,]^T$ is the velocity of the second end-effector with respect to the first end-effector frame and $\dot{\mathbf{q}} = [\,\dot{\mathbf{q}}_1^T \quad \dot{\mathbf{q}}_2^T\,]^T$ is the vector composed by the joint velocities of the two arms. For the sake of clarity, superscript e_1 will be omitted in the remain of the paper when referring to the relative velocity ${}^{e_1}\mathbf{v}_r$.

Different ways to derive the relative Jacobian has been illustrated in [5], [12] and [17]. The present work applies the procedure presented in [12].

Once $\mathbf{J}_r$ has been obtained, the manipulability ellipsoids may be derived as shown previously for the single-arm case.

Moreover, due to the intrinsic redundancy of the system, the inverse kinematic solution becomes analogous to a single redundant manipulator.

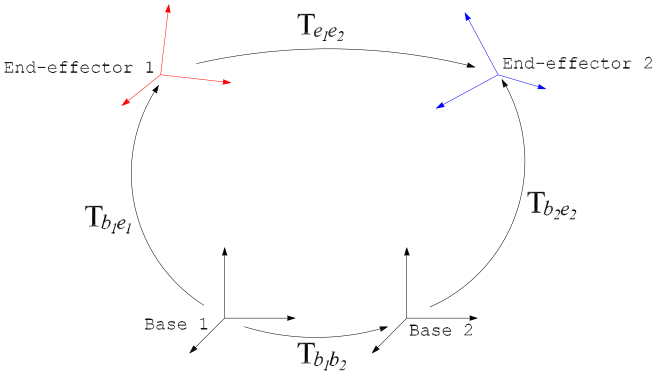

Fig. 1. Kinematic chain of the dual-arm system.

In particular, deriving the desired relative velocity $\mathbf{v}_r^{\mathrm{d}}$ from a desired task-space trajectory $\mathbf{T}^{\mathrm{d}}_{e_1e_2}(t)$, the inverse kinematic may be written as:

$$\dot{\mathbf{q}} = \mathbf{J}_r^{\dagger}\mathbf{v}_r^{\mathrm{d}} + (\mathbf{I} - \mathbf{J}_r^{\dagger}\mathbf{J}_r)\dot{\mathbf{q}}_0 \tag{12}$$

where $\mathbf{J}_r$ is the relative Jacobian of the dual-arm system, $\mathbf{J}_r^{\dagger} = \mathbf{J}_r^T(\mathbf{J}_r\mathbf{J}_r^T)^{-1}$ is its pseudo-inverse, $\mathbf{I}$ is the identity matrix and $\dot{\mathbf{q}}_0$ is an arbitrary vector, which is projected into the null space of $\mathbf{J}_r$ to satisfy a secondary goal. The projection of $\dot{\mathbf{q}}_0$ into the null space results in self-motions of the joints and, since the task is relative, it generates the motion of the end-effectors with respect to the world frame.

The vector $\dot{\mathbf{q}}_0$ is typically chosen at every step by implementing a gradient-type law, such that:

$$\dot{\mathbf{q}}_0 = -k_0\left(\frac{\partial c(\mathbf{q})}{\partial \mathbf{q}}\right)^T \tag{13}$$

where $c(\mathbf{q})$ is a cost function to be minimised and k_0 is a positive scalar.

This method implements an effective online redundancy resolution, allowing a local optimisation of the chosen performance index. However, it presents some drawbacks:

- it converges to a local optimum, but it does not ensure the convergence to a global one;
- it needs an initial configuration $\mathbf{q}_0$ and the found optimum is usually the nearest to this configuration;
- even if the desired trajectory was known a priori, the optimisation would take into account only the actual configuration of the system. Thus, the action $\dot{\mathbf{q}}_0$ might not be optimal for the overall task.

Note that the resolution of the kinematics for a relative task does not express any constraints with respect to the base frame. This means the manipulators are allowed to move with respect to the base frame as long as they respond to the relationships of the relative task. As a matter of fact, the joint self-motion makes the end-effectors move with respect to the base frame and the manipulators assume optimal postures depending on the optimisation criterion.

III. Proposed Method

According to the previous section, it is possible to maximise the manipulability of the system along a direction $\mathbf{u}$ by choosing a proper performance index.

Using the relative Jacobian $\mathbf{J}_r$ in (8) (where $\mathbf{W}$ has been omitted for the sake of simplicity), a performance index $c(\mathbf{q})$ may be defined that is inversely proportional to the force transmission ratio along $\mathbf{u}$ at each configuration $\mathbf{q}$ of the system (see also [18]):

$$c(\mathbf{q}) = \frac{1}{\alpha^2} = \mathbf{u}^T\mathbf{J}_r\mathbf{J}_r^T\mathbf{u} \tag{14}$$

Minimising $c(\mathbf{q})$, then, corresponds to maximising the force transmission ratio along $\mathbf{u}$.

From a practical point of view, a lower value of $c(\mathbf{q})$ is preferred when the relative task presents high forces on the tool along a given direction $\mathbf{u}$. For instance, a milling process implies elevated forces along the direction tangent to the desired path. Thus, a good value of $c(\mathbf{q})$ in each point of the task ensures lower torques in the joints and energy saving as well.

However, considering a desired path composed by h points and represented by a tensor $\mathbf{T}^{\mathrm{d}}_{e_1e_2} \in \mathbb{R}^{4\times4\times h}$, a global approach to the optimisation of $c(\mathbf{q})$ over the whole task would consist in finding the h configurations that minimise the total cost $\sum_{i=1}^{h} c_i$.

Remark 1: The choice of a different performance index does not affect the basic concept of the method. In any case, a total cost function $\sum_i c_i$ could be defined, in order to optimise the selected performance index $c(\mathbf{q})$ along the overall task.

It is clear that the computational burden of this approach would be too heavy. Thus, let $\Gamma = \{\gamma_1 \ldots \gamma_j\}$ be a subset of indices relative to j points on the path (with $j < h$) and let $w(\mathbf{q}_\Gamma)$ be a cost function, such that:

$$w(\mathbf{q}_\Gamma) = \sum_{i\in\Gamma} c(\mathbf{q}_i) \tag{15}$$

where $\mathbf{q}_\Gamma = [\mathbf{q}_{\gamma_1}^T, \ldots, \mathbf{q}_{\gamma_j}^T]^T$ is a column vector composed by the j configurations relative to each element of Γ.

If the chosen points are the most relevant in the definition of the overall task manipulability, then, (15) represents a satisfactory approximation of the complete cost function.

Remark 2: The choice of the relevant points depends on how the path curvature and the selected direction $\mathbf{u}$ change along the task. In other words, in each section of the path, the larger the variation of curvature and $\mathbf{u}$, the greater the number of points to consider.

The minimisation of (15) represents a nonlinear optimisation problem, which leads to the search of j configurations $\mathbf{q}_{i\in\Gamma}$ in the joint space. As a consequence, the optimisation problem is constrained, since:

- the solutions must lay between the mechanical boundaries of the joints:

$$\mathbf{lb} \leq \mathbf{q}_i \leq \mathbf{ub} \qquad \forall i \in \Gamma \tag{16}$$

- the solutions must obviously satisfy the nonlinear constraints given by the geometrical path:

$$\mathbf{T}_{e_1e_2}(\mathbf{q}_i) = \mathbf{T}^{\mathrm{d}}_{e_1e_2(i)} \qquad \forall i \in \Gamma \tag{17}$$

A. Smoothness in joint motion

In order to avoid abrupt changes in the configurations obtained from the optimisation resolution, it is necessary to limit the excursion between the configurations $\mathbf{q}_{i\in\Gamma}$. For this reason, consider a term $w'(\mathbf{q}_\Gamma)$, such that:

$$w'(\mathbf{q}_\Gamma) = \sum_{i\in\Gamma\backslash\{\gamma_1\}} \|\mathbf{q}_i - \mathbf{q}_{i-1}\|^4 \tag{18}$$

Note that $w'(\mathbf{q}_\Gamma)$ depends on the distance between the j configurations derived from the optimisation and may be utilised to ensure that the optimisation solutions are not abruptly different.

Considering (15) and (18), the cost function becomes:

$$w(\mathbf{q}_\Gamma) = \sum_{i\in\Gamma} \mathbf{u}_i^T \mathbf{J}_r(\mathbf{q}_i)\mathbf{J}_r^T(\mathbf{q}_i)\mathbf{u}_i + \lambda \sum_{i\in\Gamma\backslash\{\gamma_1\}} \|\mathbf{q}_i - \mathbf{q}_{i-1}\|^4 \tag{19}$$

where $\lambda \in \mathbb{R}_{\geq 0}$ is a weighting factor for the smoothness term.

It is important to mention that the smoothness term in the cost function avoids large differences between consecutive configurations, but does not directly take into account the velocity and acceleration constraints. On the other hand, the proposed method aims at obtaining a series of configurations, each one relative to a geometrical point on the path, and without defining the motion over time. These boundaries have to be taken into account when defining the motion law.

B. Interpolation between the obtained configurations

The optimisation problem presented in (19) returns the optimal configurations $\mathbf{q}_\Gamma$, but does not calculate the configurations relative to the remaining ones. In order to solve this issue, it is necessary to find a criteria to interpolate the given $\mathbf{q}_{i\in\Gamma}$. The chosen method is the iterative optimisation of the cost function (19) for a certain number k of inner points between each couple of $\mathbf{q}_{i\in\Gamma}$. However, in this case, the cost function must take into account the distance from the initial and final configurations of the interpolation.

In other words, consider each couple of consecutive configurations $\bar{\mathbf{q}}_\gamma$, $\bar{\mathbf{q}}_{\gamma_{+1}}$, with $\gamma \in \Gamma$ and let $\Delta = \{\delta_1 \dots \delta_k\}$ be the set of k internal points between $\bar{\mathbf{q}}_\gamma$ and $\bar{\mathbf{q}}_{\gamma_{+1}}$. The cost function to optimise becomes:

$$\hat{w}(\mathbf{q}_\Delta) = \sum_{i\in\Delta} \mathbf{u}_i^T \mathbf{J}_r(\mathbf{q}_i)\mathbf{J}_r^T(\mathbf{q}_i)\mathbf{u}_i + \lambda \sum_{i\in\Delta\backslash\{\delta_1\}} \|\mathbf{q}_i - \mathbf{q}_{i-1}\|^4 + \lambda\left(\|\mathbf{q}_{\delta_1} - \bar{\mathbf{q}}_\gamma\|^4 + \|\bar{\mathbf{q}}_{\gamma_{+1}} - \mathbf{q}_{\delta_k}\|^4\right) \tag{20}$$

In this way the algorithm tends to maximise the manipulability between each configuration. As mentioned, this optimisation has to be performed for all the intervals between each couple of consecutive configuration $\mathbf{q}_{i\in\Gamma}$.

This method may be applied recursively to interpolate between the new k+1 intervals until the total amount of points reaches the points of the path (i.e. h).

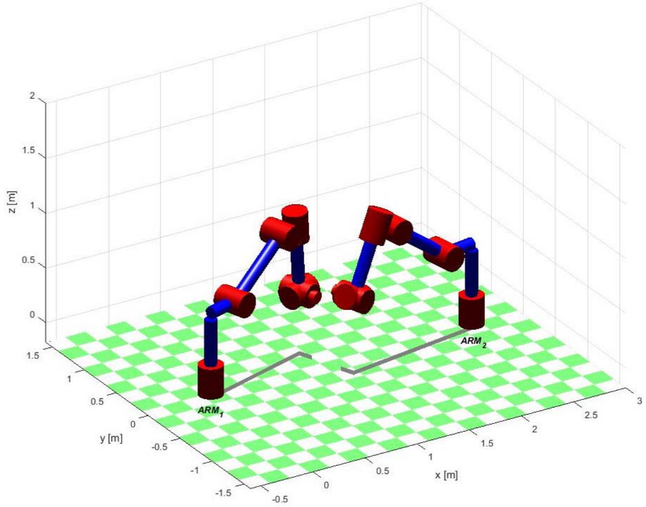

Fig. 2. Simulation of the dual-arm manipulator setup by means of the Matlab Robotics Toolbox.

IV. Illustrative example

A. System setup

The method has been simulated on a dual-arm system composed by two 6-dof Comau NS16 manipulators. The robots lay one opposite to the other, at a distance of 2.5 m, as shown in Figure 2, where the Matlab Robotics Toolbox [19] has been used as simulation environment.

Thus, the transformation matrix between the bases is:

$$\mathbf{T}_{b_1 b_2} = \begin{bmatrix} -1 & 0 & 0 & 2.5 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Considering the first robot base as the world frame, (10) is still equal to:

$$\mathbf{T}_{e_1 e_2} = \mathbf{T}_{b_1 e_1}^{-1} \mathbf{T}_{b_1 b_2} \mathbf{T}_{b_2 e_2} \tag{21}$$

The mechanical bounds of the joints are the same for both arms and are reported in Table I.

B. Desired path

The desired path is defined with respect to the first end-effector frame and consists of a corner with side equal to 150 mm in the horizontal plane $z = 0.3$ m, with a fillet of radius $r = 10$ mm, as shown in Figure 3. Fillets and changes of direction represent typical issues in technological application, and this motivates the choice of this path.

Therefore, considering the Cartesian path as composed by $h = h_1 + h_2 + h_3$ points, where h_1, h_2, h_3 are the number

TABLE I

Mechanical bounds of the joints for the two arms.

Arm	Bound	q_1	q_2	q_3	q_4	q_5	q_6
1	Lower [deg]	-60	-45	-150	-90	-90	-90
	Upper [deg]	$+60$	$+90$	$+45$	$+90$	$+90$	$+90$
2	Lower [deg]	-60	-45	-150	-90	-90	-90
	Upper [deg]	$+60$	$+90$	$+45$	$+90$	$+90$	$+90$

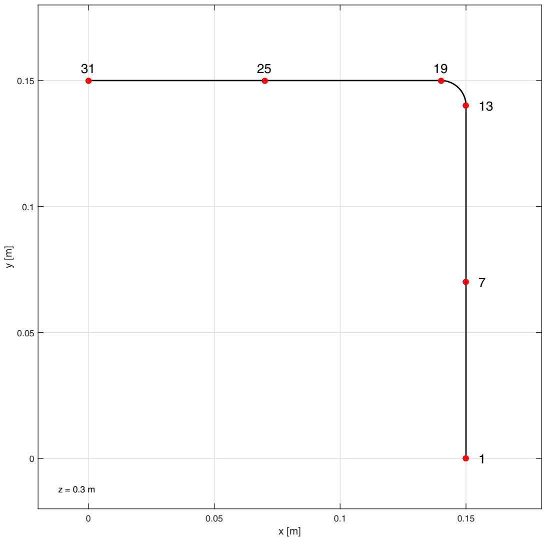

Fig. 3. Relative path to be performed by the manipulators.

of points of the vertical, rounded and horizontal segments respectively, the relative position $\mathbf{x}^{\mathrm{d}}_{e_1e_2} = [x_i \quad y_i \quad z_i]^T$ is given by:

$$
x_i = \begin{cases} l & \forall i = 1 \dots h_1 \\ (l-r) + r\cos\left(\frac{\pi}{2}\frac{i-h_1}{h_2+1}\right) & \forall i = h_1{+}1 \dots h_1{+}h_2 \\ (l-r)\left(1-\frac{i-(h_1+h_2+1)}{h_3-1}\right) & \forall i = h_1{+}h_2{+}1 \dots h \end{cases}
$$

$$
y_i = \begin{cases} (l-r)\frac{i-1}{h_1-1} & \forall i = 1 \dots h_1 \\ (l-r) + r\sin\left(\frac{\pi}{2}\frac{i-h_1}{h_2+1}\right) & \forall i = h_1{+}1 \dots h_1{+}h_2 \\ l & \forall i = h_1{+}h_2{+}1 \dots h \end{cases}
$$

$$
z_i = \quad 0.3 \qquad \forall i = 1 \dots h
$$

The relative orientation between the end-effectors is constant and is given by the following rotation matrix:

$$
\mathbf{R}^{\mathrm{d}}_{e_1e_2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}
$$

Thereby, the desired path may be written as:

$$
\mathbf{T}^{\mathrm{d}}_{e_1e_2} = \left[\begin{array}{c|c} \mathbf{R}^{\mathrm{d}}_{e_1e_2} & \mathbf{x}^{\mathrm{d}}_{e_1e_2} \\ \hline \mathbf{0} & 1 \end{array}\right] \qquad \forall i = 1 \dots h
$$

Moreover, the unit vector tangent to each point of the path is:

$$
\mathbf{t}_i = \begin{cases} [0 \quad 1 \quad 0]^T & \forall i = 1 \dots h_1 \\ [-\sin\left(\frac{\pi}{2}\frac{i-h_1}{h_2+1}\right) \quad \cos\left(\frac{\pi}{2}\frac{i-h_1}{h_2+1}\right) \quad 0]^T & \forall i = h_1{+}1 \dots h_1{+}h_2 \\ [-1 \quad 0 \quad 0]^T & \forall i = h_1{+}h_2{+}1 \dots h \end{cases}
$$

In the simulation, h_1, h_2, h_3 were set to 13, 5, 13 respectively, so that $h = 31$ results.

C. Algorithm setup

First of all, the algorithm has to optimise the cost function (19), which considers the manipulability indexes of a subset of points Γ. As mentioned previously, the manipulability ellipsoid may be decomposed in a linear and an angular one. In this simulation, the cost function takes into account only the linear ellipsoid, which means the weighting matrix introduced in (8) has to be $\mathbf{W} = \mathrm{diag}(1,1,1,0,0,0)$.

Moreover, the manipulability index has to be optimised along the direction tangent to the path. Thereby, for all points, it must be $\mathbf{u}_i = [\mathbf{t}_i^T, \ \mathbf{0}]^T$.

Following the guidelines given in Remark 2, the number of elements of Γ is chosen equal to six, which is a good trade-off between computational burden and effectiveness. These points are shown in Figure 3 and correspond to $\Gamma = \{1;\, 7;\, 13;\, 19;\, 25;\, 31\}$.

For this first step, the weight for joint smoothness λ was selected equal to 1.5 in order to ensure a good trade-off between the smoothness and the manipulability terms. In the second step, the algorithm minimises the cost function (19) in order to optimise the manipulability index of the five inner points between each of the previous configurations. In this case, λ was set to 3.

The optimisation has been performed by means of a multiple start gradient-based algorithm, starting from 100 random initial configurations for each problem (for details see [20]).

V. Results

The purpose of this section is the comparison between the results obtained with the proposed method and the typical approach presented in Section II. In order to do this, the proposed algorithm is executed as described in the previous section. The optimisation determines h configurations, relative to each point of the path.

In the meanwhile, the pseudo-inverse resolution presented in Section II is performed with $k_0 = 1$ and using an initial configuration $\mathbf{q}_0$ equal to the first configuration obtained by the proposed algorithm. The cost function $c(\mathbf{q})$ is the same as that defined in (14).

Figure 4 shows the value of the manipulability index (14) along the path, obtained by means of the proposed method. Similarly, Figure 4 shows the same index obtained through the pseudo-inverse-based method.

Comparing the figures, the proposed method shows a flatter trend of the performance index, although the preferred direction $\mathbf{u}$ changes through the path. On the contrary, the pseudo-inverse method shows an abrupt increment of the performance index, which corresponds to the change of direction due to the fillet. As a matter of fact, the average performance indexes are equal to 1.17 and 1.52 for the first and second method respectively. This shows that the proposed method ensures a better global manipulability, since it takes into account the whole task during the optimisation.

Moreover, the pseudo-inverse method optimises the cost function from a given initial configuration. This means it does not give any information about the optimal initial posture of the manipulators for a given task. The first configuration found by the proposed method, instead, represents an optimal solution in this sense. This means the task will be performed starting from an optimal initial posture of the manipulators. Thus, a critical issue for the pseudo-inverse

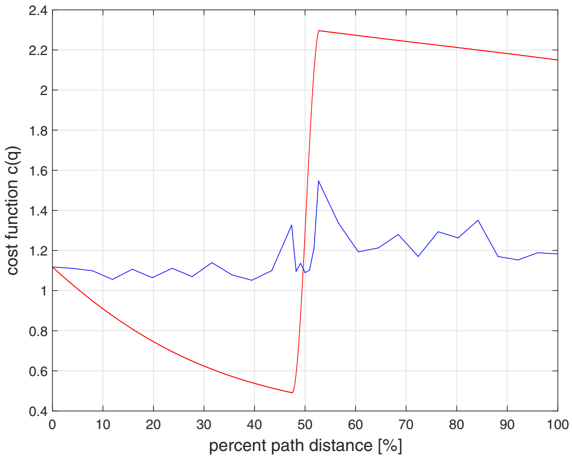

Fig. 4. Comparison of the proposed method and the pseudo-inverse method using the same initial configuration. (Blue: cost function trend using the proposed method. Red: cost function trend using the pseudo-inverse-based method).

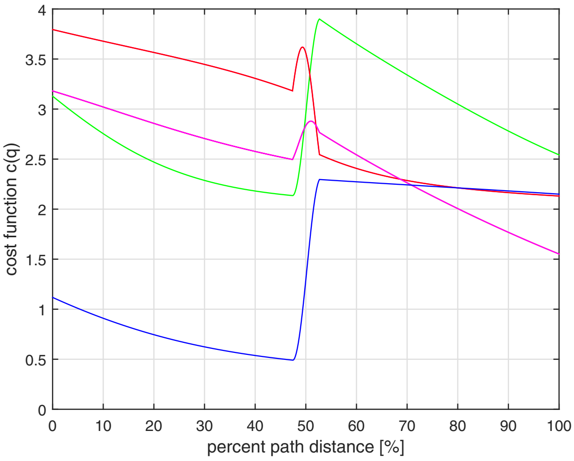

Fig. 5. Comparison of the pseudo-inverse method for different initial configurations (Blue: solution using the first configuration of the proposed method as initial configuration. Red, green, magenta: solution using random initial configurations; note that the initial configurations must satisfy the constraints (16)-(17)).

method is the choice of a suitable initial configuration of the system. As a matter of fact, the use of a non-optimal $\mathbf{q}_0$ leads to a remarkable decay in the trend of the performance index. This is depicted in Figure 5, which shows the trend of the manipulability index, using different initial configurations. In particular, it is easy to note that the trend worsens with respect to the case in which the first configuration of the proposed method is used as initial one (blue line in Figure 5).

Finally, the proposed method allows an easy implementation of joint and Cartesian constraints, which are not easily handled by the pseudo-inverse method.

VI. Conclusions

A method for the global optimisation of the task manipulability of a dual-arm system has been presented. This approach is based on the optimisation of a cost function, which depends on several points along the desired path.

Since the task is expressed in terms of relative poses between the end-effectors, the dual-arm kinematics is treated by using the relative Jacobian, which makes the resolution of the redundancy similar to a single-arm manipulator.

As shown by simulation, the proposed method allows a better overall manipulability along the task. Moreover, this approach is able to find an optimal initial configuration for the system, which represents a great advantage respect to other methods, like the pseudo-inverse-based one.

As a drawback, the optimisation problem presents a heavy computational burden, and needs to be performed offline.

References

[1] C. Smith *et al.*, “Dual arm manipulation - a survey,” *Robotics and Autonomous Systems*, vol. 60, pp. 1340–1353, 2012.

[2] J. Lee, P. H. Chang, and R. S. Jamisola, “Relative task prioritization for dual-arm with multiple, conflicting tasks: Derivation and experiments,” in *IEEE International Conference on Robotics and Automation*, 2013, pp. 1928 – 1933.

[3] W. S. Owen, E. A. Croft, and B. Benhabib, “Acceleration and torque redistribution for a dual-manipulator system,” *IEEE Transactions on Robotics and Automation*, vol. 21, pp. 1226–1230, 2005.

[4] ——, “On-line trajectory resolution for two-armed systems with conflicting performance criteria,” *Mechanism and Machine Theory*, vol. 44, pp. 949–965, 2009.

[5] C. L. Lewis and A. A. Maciejewski, “Trajectory generation for cooperating robots,” in *IEEE International Conference on Systems Engineering*, 1990, pp. 300–303.

[6] J. Lee, O. H. Chang, and D. G. Gweon, “A cost function inspired by human arms movement for a bimanual robotic machining,” in *IEEE International Conference on Robotics and Automation*, 2012, pp. 5431–5436.

[7] C. Lamperti, A. M. Zanchettin, and P. Rocco, “A redundancy resolution method for an anthropomorphic dual-arm manipulator based on a musculoskeletal criterion,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 1846–1851.

[8] S. Lee, “Dual redundant arm configuration optimization with task-oriented dual arm manipulability,” *IEEE Transaction on Robotics and Automation*, vol. 5, pp. 78–97, 1989.

[9] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, “Global task space manipulability ellipsoids for multiple-arm systems,” *IEEE Transaction on Robotics and Automation*, vol. 7, pp. 678–685, 1991.

[10] A. Bicchi, C. Melchiorri, and D. Balluchi, “On the mobility and manipulability of general multiple limb robots,” *IEEE Transactions on Robotics and Automation*, vol. 11, pp. 215–228, 1995.

[11] A. Bicchi and D. Prattichizzo, “Manipulability of cooperating robots with unactuated joints and closed-chain mechanisms,” *IEEE Transactions on Robotics and Automation*, vol. 16, pp. 336–345, 2000.

[12] R. S. Jamisola, P. Kormushev, D. G. Caldwell, and F. Ibikunle, “Modular relative jacobian for dual-arms and the wrench transformation matrix,” in *IEEE International Conference on Cybernetics and Intelligent Systems*, 2015, pp. 181–186.

[13] A. M. Zanchettin and P. Rocco, “Dual-arm redundancy resolution based on null-space dynamically-scaled posture optimization,” in *IEEE International Conference on Robotics and Automation*, 2012, pp. 311–316.

[14] P. Chiacchio, S. Chiaverini, and B. Siciliano, “Dexterous reconfiguration of a two-arm robot system,” in *International Conference on Control*, 1991, pp. 347–351.

[15] T. Yoshikawa, “Manipulability of robotics mechanisms,” *The International Journal of Robotics Research*, vol. 4, pp. 3–9, 1985.

[16] E. Schwartz, R. Manseur, and K. Doty, “Non-commensurate systems in robotics,” *International Journal of Robotics and Automation*, vol. 17, pp. 86–92, 2002.

[17] A. Mohri, M. Yamamoto, and G. Hirano, “Cooperative path planning for two manipulators,” in *IEEE International Conference on Robotics and Automation*, 1996, pp. 2853–2858.

[18] S. L. Chiu, “Task compatibility of manipulator postures,” *The International Journal of Robotics Research*, vol. 7, pp. 13–21, 1988.

[19] P. I. Corke, *Robotics, Vision & Control: Fundamental Algorithms in Matlab*. Springer, 2011.

[20] A. Messac, *Optimization in Practice with MATLAB*®. Cambridge University Press, 2015.