Design of singularity-robust and task-priority primitive controllers for cooperative manipulation using dual quaternion representation

Cristiana Miranda de Farias†, Yuri Gonçalves Rocha†, Luis Felipe Cruz Figueredo and Mariana Costa Bernardes

Abstract— **This paper revises and extends the problem of robust singularity and joint limits avoidance to the cooperative task-space using unit dual quaternion framework—ensuring singularity-free coupled representation of the cooperative space. The research is paramount to cooperative control applications within flexible manufacturing systems and poorly structured environments where robustness and reactiveness play a significant role. Singularity-robust techniques are proposed to control cooperative task primitives while dynamically avoiding kinematic singularities for redundant and non-redundant cooperative task trajectories. A task self-motion, which uses the nullspace of the cooperative task, is also exploited to avoid joint limits or singularities and to define task-priority controllers. Simulated and experimental results illustrate the effectiveness of the proposed singularity-robust and task-priority primitive controllers and the usefulness of singularity-robust solutions and joint limits avoidance in the cooperative task-space.**

I. Introduction

The integration of multiple robotics arms working together is not trivial [1], and great amount of research has focused on addressing the complexities of multi-arm manipulation [2]–[8]. With the hypothesis of a tightly grasped object, Uchiyama and Dauchez [3] introduced the concept of a symmetric control scheme based on the relationships of forces and velocities whereby four physically meaningful variables were indirectly extracted by integrating the velocities: absolute and relative positions; and absolute and relative orientations between the arms [1]. In [5], the same variables were taken to compose a new space, namely the cooperative task-space [5], [6]—relaxing the firmly grasping constraint which in turn empowered the analysis and enabled simply considering coordinated movements between arms. Whereas in [5], Cartesian coordinates and rotation matrices were used for representing translations and orientations in the cooperative space, Caccavale et al. [6] considered a unit quaternion-based singularity-free representation for orientations.

A more efficient representation was proposed by [7] based on unit dual quaternions and developed upon the foundations on the previous works of Connolly and Pfeiffer [4] and Khatib [2]. The cooperative dual task-space (CDTS) relies on dual quaternion algebra to represent the cooperative elements without singularities and without decoupling orientation and translation. The use of a coupled non-minimal and more efficient representation further enlightens task definition, system description, and eases extensions concerning multi-arm and human-robot manipulation. Moreso, the proposed coupled representation eases the deduction and analysis of geometric parameters related to the cooperative variables [1], [9], and is considerably more intuitive from the control point of view for being more connected to the sensor space [10].

The control technique proposed in [7] stems from a direct and instantaneous least square minimization of a task function [11]—therein defined as the cooperative manipulation task—similarly to classic task-space controllers. The solution thus neglects the significant role of kinematic singularities and undesired poses, and also disregards the existence of multiple paths for the robot links without disturbing the end-effector configuration due to task-space redundancy—namely the possibility of self-motion. However, it is well-known that applications in the cooperative task-space are prone to further interaction with joint limits, singularities and undesired configurations due to the increased complexity of the space and the additional constraints from the cooperative task. Hence, it is central for real cooperative manipulation applications to cope with task-space singularities, joint limits, and task-space redundancy; yet very few solutions have been extended to the cooperative task-space—despite advantages for such a complex framework. Particularly for the CDTS [7], there is no result in the literature concerning the design and implementation of singularity avoidance techniques.

In this context and in order to ensure proper manipulability through the whole cooperative task-space, this paper addresses the problem of singularity and joint limits avoidance for cooperative manipulation based on dual quaternion algebra. We design novel controllers for geometric cooperative task primitives using classic singularity-robust techniques for single-arm robotic systems which are efficient for cooperative tasks. The controllers also exploit the cooperative task self-motion, which consists on movements of the robot links that do not disturb the end effector configuration [12] and provides further reactiveness during execution as to avoid joint limits or singularities [13]. The original problem of cooperative control based on the efficient CDTS representation from [7] is enlivened by explicitly avoiding undesirable configurations from the arms that could degrade the overall cooperative manipulation performance. Moreso, we consider a controller invariant to base coordinate changes in contrast to [7]—which might be key to an efficient cooperative manipulation with different tasks and reference frames. To demonstrate the implementation simplicity and effectiveness, the solutions were evaluated within different simulated scenarios. Also, to better illustrate the usefulness of proper

† Both authors have contributed equally.

singularity-robust solutions and joint limits avoidance in the cooperative task-space, we performed experiments using the NAO H25 robot which is considerably liable to reach undesirable poses during the cooperative manipulation.

II. Mathematical Background and Kinematic Representation of Serial Manipulators

This section introduces concepts on dual quaternion representation and the CDTS.

A. Dual quaternions applied to rigid motion representation

The algebra of quaternions [14] is generated by the basis elements $1, \hat{\imath}, \hat{\jmath}$, and $\hat{k}$, which yields the set $\mathbb{H} \triangleq \left\{\eta + \boldsymbol{\mu} \ : \ \boldsymbol{\mu}{=}\mu_1\hat{\imath} + \mu_2\hat{\jmath} + \mu_3\hat{k},\ \eta, \mu_1, \mu_2, \mu_3{\in}\mathbb{R}\right\}$, where $\hat{\imath}, \hat{\jmath}, \hat{k}$ are quaternionic units such that $\hat{\imath}^2 = \hat{\jmath}^2 = \hat{k}^2 = \hat{\imath}\hat{\jmath}\hat{k} = -1$. Unit quaternions belong to a subset of $\mathbb{H}$ whose elements are constrained to the three-dimensional unit sphere in $\mathbb{R}^4$, that is, $\mathcal{S}^3 \triangleq \{\boldsymbol{x} \in \mathbb{H} : \|\boldsymbol{x}\| = 1\}$ where $\|\boldsymbol{x}\|^2 \triangleq \boldsymbol{x}\boldsymbol{x}^* = \boldsymbol{x}^*\boldsymbol{x}$ with $\boldsymbol{x}^* \triangleq \eta - \boldsymbol{\mu}$ being the conjugate of $\boldsymbol{x}$. Under multiplication the set forms the Lie group of unit quaternions, Spin(3), whose identity element is 1 and inverse is $\boldsymbol{x}^*$ [14]. Analogously to the way that complex numbers are used to represent rotations in the plane, unit quaternions yields three-dimensional rotations. An arbitrary rotation ϕ around the rotation axis $\boldsymbol{n} = n_x\hat{\imath} + n_y\hat{\jmath} + n_z\hat{k}$ is represented by the unit quaternion $\boldsymbol{x} = \cos(\phi/2) + \sin(\phi/2)\boldsymbol{n}$ [15].

Similarly, the dual quaternion algebra completely describes the rigid body motion [14]. Thus, we consider the following subset of dual quaternions with unit norm $\underline{\boldsymbol{S}} \triangleq \left\{\boldsymbol{x} + \varepsilon\boldsymbol{x}' \ : \ \boldsymbol{x} \in \mathcal{S}^3,\ \boldsymbol{x}' \in \mathbb{H},\ \boldsymbol{x}\boldsymbol{x}'^*{+}\boldsymbol{x}'\boldsymbol{x}^*{=}0\right\}$, where ε is called dual unit and $\varepsilon^2 = 0$, $\varepsilon \neq 0$. Under multiplication, the subset $\underline{\boldsymbol{S}}$ forms the unit dual quaternion group Spin(3) $\ltimes \mathbb{R}^3$, whose identity element is 1 and group inverse is $\underline{\boldsymbol{x}}^*$—where $\underline{\boldsymbol{x}}^* \triangleq \boldsymbol{x}^* + \varepsilon\left(\boldsymbol{x}'\right)^*$ is the conjugate of $\underline{\boldsymbol{x}}$. An arbitrary rigid displacement may be represented in Spin(3) $\ltimes \mathbb{R}^3$ by a translation $\boldsymbol{p} = p_x\hat{\imath} + p_y\hat{\jmath} + p_z\hat{k}$—a pure quaternion (isomorphic to $\mathbb{R}^3$)—followed by a rotation $\boldsymbol{r} \in$ Spin(3),

$$\underline{\boldsymbol{x}} = \boldsymbol{r} + \varepsilon(1/2)\boldsymbol{p}\boldsymbol{r}. \tag{1}$$

It is known that $SE(3)$ is a non-commutative group, and the same is valid for the unit dual quaternion group which is a double cover of $SE(3)$—that is, if $\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{y}}$ are unit dual quaternions, then $\underline{\boldsymbol{x}}\underline{\boldsymbol{y}} \neq \underline{\boldsymbol{y}}\underline{\boldsymbol{x}}$. Nonetheless, Hamilton operators can be used for the commutation of dual quaternions multiplications [1]. Thus, we shall consider the dual quaternion mapping to $\mathbb{R}^8$ manifold vec : $\underline{\boldsymbol{S}} \to \mathbb{R}^8$, which satisfies $\operatorname{vec}\underline{\boldsymbol{z}} = \overset{+}{\boldsymbol{H}}\left(\underline{\boldsymbol{x}}\right)\operatorname{vec}\underline{\boldsymbol{y}} = \overset{-}{\boldsymbol{H}}\left(\underline{\boldsymbol{y}}\right)\operatorname{vec}\underline{\boldsymbol{x}}$, where $\overset{+}{\boldsymbol{H}}\left(\cdot\right), \overset{-}{\boldsymbol{H}}\left(\cdot\right)$ defined in [1] are the matrix form of the algebraic product. Also, using the definition of conjugate of dual quaternions it is easy to show that $\operatorname{vec}\underline{\boldsymbol{x}}^* = \boldsymbol{C}_8 \operatorname{vec}\underline{\boldsymbol{x}}$, where $\boldsymbol{C}_8 \triangleq diag(1, -1, -1, -1, 1, -1, -1, -1)$. An analogous mapping can be defined for the quaternion product, that is, $\operatorname{vec}_4 : \mathbb{H} \to \mathbb{R}_4$ such that for a quaternion $\boldsymbol{z} = \boldsymbol{x}\boldsymbol{y}$, one would have $\operatorname{vec}_4\boldsymbol{z} = \overset{+}{\boldsymbol{H}}_4\left(\boldsymbol{x}\right)\operatorname{vec}_4\boldsymbol{y} = \overset{-}{\boldsymbol{H}}_4\left(\boldsymbol{y}\right)\operatorname{vec}_4\boldsymbol{x}$, [1]. Similarly, for a quaternion conjugate, we also have $\operatorname{vec}\boldsymbol{x}^* = \boldsymbol{C}_4 \operatorname{vec}\boldsymbol{x}$, where $\boldsymbol{C}_4 \triangleq diag(1, -1, -1, -1)$.

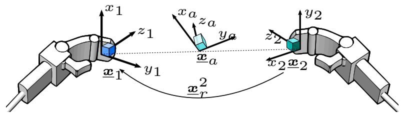

x_1
z_1
y_1
$\underline{\boldsymbol{x}}_1$
x_a
z_a
y_a
$\underline{\boldsymbol{x}}_a$
y_2
z_2
x_2
$\underline{\boldsymbol{x}}_2$
$\underline{\boldsymbol{x}}_r^2$

Fig. 1: CDTS representation of variables $\underline{\boldsymbol{x}}_a$ and $\underline{\boldsymbol{x}}_r$.

B. Cooperative dual task-space (CDTS)

The CDTS is fully described by the cooperative variables $\underline{\boldsymbol{x}}_a$ and $\underline{\boldsymbol{x}}_r$ defining respectively the relative and absolute dual positions between left and right end-effectors, $\underline{\boldsymbol{x}}_1$ and $\underline{\boldsymbol{x}}_2$, as shown in Fig. 1. The formal definition is as follows.

Definition 1: The relative and absolute dual positions are

$$\underline{\boldsymbol{x}}_r \triangleq \underline{\boldsymbol{x}}_2^*\underline{\boldsymbol{x}}_1 \tag{2}$$

$$\underline{\boldsymbol{x}}_a \triangleq \underline{\boldsymbol{x}}_2\underline{\boldsymbol{x}}_{r/2}, \tag{3}$$

where $\underline{\boldsymbol{x}}_{r/2}$ is the unit dual quaternion that defines the transformation corresponding to half of the angle ϕ_r around the axis $\boldsymbol{n}_r = \hat{\imath}n_x + \hat{\jmath}n_y + \hat{k}n_z$ of the quaternion $\mathcal{P}\left(\underline{\boldsymbol{x}}_r\right)$ and half of the translation between the two arms [7].

As shown in [7], [16], the relative dual quaternion Jacobian satisfies the differential relation $\operatorname{vec}\dot{\underline{\boldsymbol{x}}}_r = \boldsymbol{J}_{\underline{\boldsymbol{x}}_r}\dot{\boldsymbol{\Theta}}$, where

$$\boldsymbol{\Theta} \triangleq \begin{bmatrix}\boldsymbol{\theta}_1^T & \boldsymbol{\theta}_2^T\end{bmatrix}^T \tag{4}$$

is the vector composed by both arm's joints

$$\boldsymbol{J}_{\underline{\boldsymbol{x}}_r} = \left[\begin{array}{cc}\overset{+}{\boldsymbol{H}}\left(\underline{\boldsymbol{x}}_2^*\right)\boldsymbol{J}_{\underline{\boldsymbol{x}}_1} & \overset{-}{\boldsymbol{H}}\left(\underline{\boldsymbol{x}}_1\right)\boldsymbol{J}_{\underline{\boldsymbol{x}}_2}^*\end{array}\right]. \tag{5}$$

Analogously, for absolute variables

$$\operatorname{vec}\dot{\underline{\boldsymbol{x}}}_a = \boldsymbol{J}_{\underline{\boldsymbol{x}}_a}\dot{\boldsymbol{\Theta}}, \tag{6}$$

where $\boldsymbol{J}_{\underline{\boldsymbol{x}}_a} = \overset{-}{\boldsymbol{H}}\left(\underline{\boldsymbol{x}}_{r/2}\right)\boldsymbol{J}_{\underline{\boldsymbol{x}}_{2\text{ext}}} + \overset{+}{\boldsymbol{H}}\left(\underline{\boldsymbol{x}}_2\right)\boldsymbol{J}_{\underline{\boldsymbol{x}}_{r/2}}$ in which $\boldsymbol{J}_{\underline{\boldsymbol{x}}_{2\text{ext}}} = \begin{bmatrix}\boldsymbol{0}_{8\times\dim\boldsymbol{\theta}_1} & \boldsymbol{J}_{\underline{\boldsymbol{x}}_2}\end{bmatrix}$, $\dim\boldsymbol{\theta}_1$ corresponds to the dimension of the vector $\boldsymbol{\theta}_1$,

$$\boldsymbol{J}_{\underline{\boldsymbol{x}}_{r/2}} = \left[\begin{array}{c}\frac{1}{2}\overset{-}{\boldsymbol{H}}_4\left(\boldsymbol{r}_{r/2}^*\right)\boldsymbol{J}_{\mathcal{P}\left(\underline{\boldsymbol{x}}_r\right)} \\ \frac{1}{4}\left(\overset{-}{\boldsymbol{H}}_4\left(\boldsymbol{r}_{r/2}\right)\boldsymbol{J}_{\boldsymbol{p}_r} + \overset{+}{\boldsymbol{H}}_4\left(\boldsymbol{p}_r\right)\boldsymbol{J}_{\mathcal{P}\left(\underline{\boldsymbol{x}}_{r/2}\right)}\right)\end{array}\right]$$

$\boldsymbol{J}_{\mathcal{P}\left(\underline{\boldsymbol{x}}_{r/2}\right)}$ corresponds to the four upper rows of $\boldsymbol{J}_{\underline{\boldsymbol{x}}_{r/2}}$ (5).

Although this paper does not address the issue of the forces involved in the manipulation, we remark that forces and moments can be represented directly in the CDTS [1].

III. Singularity-robust and task-priority control law for cooperative primitives

In the context of Definition 1, the forward kinematics model (FKM) of all feasible bi-manual tasks must be related to the kinematics from both manipulators and, consequently, they are function of the augmented joint vector $\boldsymbol{\Theta}$ defined in (4). Thus, any task primitive $\boldsymbol{h}_{task}$ in the CDTS can be described as $\boldsymbol{h}_{task} = \boldsymbol{f}_{task}(\boldsymbol{\Theta})$, and the differential kinematics related to each task primitive yields $\operatorname{vec}\dot{\underline{\boldsymbol{x}}}_{task} = \boldsymbol{J}_{task}\dot{\boldsymbol{\Theta}}$,where $\boldsymbol{J}_{task}$ is a predefined Jacobian that relates $\boldsymbol{\Theta}$ (4) with the derivative of the task primitive.

From the relative and absolute dual positions defined in (2)-(3), and the corresponding relative and absolute dual quaternion Jacobians (5), (6), the lemmas [1] define the differential kinematics for the cooperative geometric primitives.

Lemma 1: Consider the relative pose between arms defined by the unit dual quaternion $\underline{\boldsymbol{x}}_r$ with differential kinematics given by $\operatorname{vec}\dot{\underline{\boldsymbol{x}}}_r = \boldsymbol{J}_{\underline{\boldsymbol{x}}_r}\dot{\boldsymbol{\Theta}}$ (2),(5), then the relative orientation between arms is given by the unit quaternion $\boldsymbol{r}_{rel} = \mathcal{P}\left(\underline{\boldsymbol{x}}_r\right)$ and the differential kinematics by $\operatorname{vec}_4 \dot{\boldsymbol{r}}_{rel} = \boldsymbol{J}_{\mathcal{P}\left(\underline{\boldsymbol{x}}_r\right)}\dot{\boldsymbol{\Theta}}$, where $\boldsymbol{J}_{\mathcal{P}\left(\underline{\boldsymbol{x}}_r\right)}$ corresponds to the four upper rows of the relative dual quaternion Jacobian $\boldsymbol{J}_{\underline{\boldsymbol{x}}_r}$.

Lemma 2: From (2), (5), the relative translation and differential kinematics between arms are given by the quaternion $\boldsymbol{p}_{rel}{=}2\,\mathcal{D}\left(\underline{\boldsymbol{x}}_r\right)\mathcal{P}\left(\underline{\boldsymbol{x}}_r\right)^*$ and by $\operatorname{vec}_4 \dot{\boldsymbol{p}}_{rel}{=}\boldsymbol{J}_{\boldsymbol{p}_r}\dot{\boldsymbol{\Theta}}$, where $\boldsymbol{J}_{\boldsymbol{p}_r} = 2\overset{-}{\boldsymbol{H}}_4\left(\mathcal{P}\left(\underline{\boldsymbol{x}}_r^*\right)\right)\boldsymbol{J}_{\mathcal{D}\left(\underline{\boldsymbol{x}}_r\right)} + 2\overset{+}{\boldsymbol{H}}_4\left(\mathcal{D}\left(\underline{\boldsymbol{x}}_r\right)\right)\boldsymbol{C}_4\boldsymbol{J}_{\mathcal{P}\left(\underline{\boldsymbol{x}}_r\right)}$ with $\boldsymbol{J}_{\mathcal{D}\left(\underline{\boldsymbol{x}}_r\right)}$ corresponding to the four lower rows of $\boldsymbol{J}_{\underline{\boldsymbol{x}}_r}$.

Lemma 3: Let $\boldsymbol{p}_{rel}$ be the relative translation, the square distance Jacobian $\boldsymbol{J}_d$ is the Jacobian that satisfies $\dot{c} = \boldsymbol{J}_d\dot{\boldsymbol{\Theta}}$, where $c \triangleq \left\|\boldsymbol{p}_{rel}\right\|^2$ and $\boldsymbol{J}_d = 2\left(\operatorname{vec}_4 \boldsymbol{p}_{rel}\right)^T\boldsymbol{J}_{\boldsymbol{p}_r}$.

Lemma 4: From (3), (6), the absolute orientation and differential kinematics are given by the unit quaternion $\boldsymbol{r}_{abs} = \mathcal{P}\left(\underline{\boldsymbol{x}}_a\right)$ and by $\operatorname{vec}_4 \dot{\boldsymbol{r}}_{abs} = \boldsymbol{J}_{\mathcal{P}\left(\underline{\boldsymbol{x}}_a\right)}\dot{\boldsymbol{\Theta}}$, where $\boldsymbol{J}_{\mathcal{P}\left(\underline{\boldsymbol{x}}_a\right)}$ corresponds to the four upper rows from $\boldsymbol{J}_{\underline{\boldsymbol{x}}_a}$.

Lemma 5: From (3), (6), the absolute translation and differential kinematics between arms are given by $\boldsymbol{p}_{abs}{=}2\,\mathcal{D}\left(\underline{\boldsymbol{x}}_a\right)\mathcal{P}\left(\underline{\boldsymbol{x}}_a\right)^*$ and by $\operatorname{vec}_4 \dot{\boldsymbol{p}}_{abs}{=}\boldsymbol{J}_{\boldsymbol{p}_a}\dot{\boldsymbol{\Theta}}$, where $\boldsymbol{J}_{\boldsymbol{p}_a} = 2\overset{-}{\boldsymbol{H}}_4\left(\mathcal{P}\left(\underline{\boldsymbol{x}}_a^*\right)\right)\boldsymbol{J}_{\mathcal{D}\left(\underline{\boldsymbol{x}}_a\right)} + 2\overset{+}{\boldsymbol{H}}_4\left(\mathcal{D}\left(\underline{\boldsymbol{x}}_a\right)\right)\boldsymbol{C}_4\boldsymbol{J}_{\mathcal{P}\left(\underline{\boldsymbol{x}}_a\right)}$ with $\boldsymbol{J}_{\mathcal{D}\left(\underline{\boldsymbol{x}}_a\right)}$ being the lower rows rows from $\boldsymbol{J}_{\underline{\boldsymbol{x}}_a}$.

In this section, we address different control design strategies for the geometric primitives derived from the CDTS.

A. Cooperative task primitive stability

To perform the two-arm control based on the previously described primitives and possible combinations, we state an error function based on the spatial difference between the desired task, $\boldsymbol{h}_{des}$, and the current cooperative primitive, $\boldsymbol{h}_{task}$, in dual quaternion space, that is, $\underline{\boldsymbol{h}}_e = \underline{\boldsymbol{h}}_{task}^*\underline{\boldsymbol{h}}_{des}$. When $\underline{\boldsymbol{h}}_{task}$equals to $\underline{\boldsymbol{h}}_{des}$, the spatial difference $\underline{\boldsymbol{h}}_e$ is 1. This allows us to exploit an error function from [17] which is invariant to coordinate changes, $\underline{\boldsymbol{e}}_{task} = 1 - \underline{\boldsymbol{h}}_e = 1 - \underline{\boldsymbol{h}}_{task}^*\underline{\boldsymbol{h}}_{des}$,if $\underline{\boldsymbol{h}}_{task}$ converges to $\underline{\boldsymbol{h}}_{des}$, the dual quaternion error $\underline{\boldsymbol{e}}_{task} \rightarrow 0$. Rewriting the equation as $\underline{\boldsymbol{e}}_{task} = \left(\underline{\boldsymbol{h}}_{des}^* - \underline{\boldsymbol{h}}_{task}^*\right)\underline{\boldsymbol{h}}_{des}$, and regarding a constant desired task objective, the error dynamics yields $\operatorname{vec}\dot{\underline{\boldsymbol{e}}}_{task} = -\overset{-}{\boldsymbol{H}}\left(\underline{\boldsymbol{h}}_{des}\right)\operatorname{vec}\dot{\boldsymbol{h}}_{task}^*$. From the differential kinematics of the cooperative primitives, Lemmas 1-5, the dynamics of the task error is given by $\operatorname{vec}\dot{\underline{\boldsymbol{e}}}_{task} = -\boldsymbol{N}_{task}\dot{\boldsymbol{\Theta}}$ where $\boldsymbol{N}_{task} = \overset{-}{\boldsymbol{H}}\left(\underline{\boldsymbol{h}}_{des}\right)\boldsymbol{J}_{task}^*$ with $\boldsymbol{J}_{task}^* = \boldsymbol{C}_8\boldsymbol{J}_{task}$ for the relative and absolute dual quaternion Jacobians; and $\boldsymbol{J}_{task}^* = \boldsymbol{C}_4\boldsymbol{J}_{task}$ for the dual quaternion Jacobians from Lemmas 1,2,3 and 5.

In addition to the cooperative primitives describes in Lemmas 1-5, we can define new cooperative tasks and easily derive their error and differential kinematics by simply combining the primitives. For instance, the full dual pose whereby we aim to control both relative and absolute poses stems from both $\underline{\boldsymbol{x}}_r$ and $\underline{\boldsymbol{x}}_a$, that is, $\operatorname{vec}\underline{\boldsymbol{h}}_{task} = \begin{bmatrix} \operatorname{vec}\underline{\boldsymbol{x}}_r^T & \operatorname{vec}\underline{\boldsymbol{x}}_a^T \end{bmatrix}^T$ and $\boldsymbol{N}_{task} = \begin{bmatrix} \boldsymbol{N}_{\underline{\boldsymbol{x}}_r}^T & \boldsymbol{N}_{\underline{\boldsymbol{x}}_a}^T \end{bmatrix}^T$.

The aim of the control scheme is to synthesize a feedback controller using the augmented joint vector (4) to make the current pose converge to a desired reference using the dual quaternion framework to avoid decoupling the rotational and translational dynamics. In this sense, let us choose a classic least square solution control law based on the inverse of the task differential kinematics and using the right pseudo-inverse of the cooperative task Jacobian $\boldsymbol{N}_{task}^+$, that is,

$$\dot{\boldsymbol{\Theta}} = \boldsymbol{N}_{task}^+\boldsymbol{K}\operatorname{vec}\underline{\boldsymbol{e}}_{task}. \tag{7}$$

To address the stability issues of the closed-loop system, let us choose the positive definite function $V_{task}{=}(1/2)\operatorname{vec}\underline{\boldsymbol{e}}_{task}^T\operatorname{vec}\underline{\boldsymbol{e}}_{task}$ as Lyapunov function candidate, for any given cooperative task. From the Lyapunov function time-derivative along t, that is, $\dot{V}_{task} = -\operatorname{vec}\underline{\boldsymbol{e}}_{task}^T\boldsymbol{N}_{task}\boldsymbol{N}_{task}^+\boldsymbol{K}\operatorname{vec}\underline{\boldsymbol{e}}_{task}$, it clear that the error dynamics is exponentially stable for any positive definite $\boldsymbol{K}$—considering a well-posed cooperative task Jacobian $\boldsymbol{N}_{task}$. The gain $\boldsymbol{K}$ defines the convergence rate and an H_∞ performance regarding the noise to error attenuation can be easily derived from [17] for any cooperative task.

B. Cooperative task-redundancy

It is interesting to highlight that the cooperative primitives require different number of DOFs to be fully controlled. The more complex the cooperative task, the more DOFs it uses. Yet, some tasks may require less DOFs than available from the augmented joint vector (4). The resulting task-redundancy may be used to raise a bimanual self-motion whereby motions of both manipulators links do not disturb the cooperative task variable trajectory configuration.

Nonetheless, the least-square control law defined in (7) do not exploits the available self-motion. Its attractiveness lies on the least square property of the pseudoinverse that generates minimum norm joint velocities for a given trajectory. However, [18] proved that joint velocities are only instantaneously minimized and can become arbitrarily large near singular configurations.

Among the feasible solutions, we can regard optimization-based techniques exploiting the existing task-redundancy to perform secondary tasks and enhance the the flexibility and reactiveness of the task execution.Thus, a more general solution proposed by [19] for single-arm manipulation can be extended to consider the cooperative task-redundancy,

$$\dot{\boldsymbol{\Theta}} = \boldsymbol{N}_{task}^+\boldsymbol{K}\operatorname{vec}\underline{\boldsymbol{e}}_{task} + \boldsymbol{P}_{task}\boldsymbol{z}_{null}, \tag{8}$$

where $\boldsymbol{z}_{null}$ is an arbitrary control input, and $\boldsymbol{P}_{task} = \boldsymbol{I} - \boldsymbol{N}_{task}^+\boldsymbol{N}_{task}$ is the operator that projects $\boldsymbol{z}_{null}$ onto the null space of $\boldsymbol{N}_{task}$. The secondary task will be performed at the best using the self-motion of the CDTS—that is, without disturbing the trajectory of the main cooperative task—and a secondary constraint can be optimized using a least-square solution similarly to the main task optimization. Hence, given a desired task $\boldsymbol{s}_{null}$ with dynamics described by $\dot{\boldsymbol{s}_{null}} = \boldsymbol{J}_{null}\dot{\boldsymbol{\Theta}}$, we can easily track its dynamics along the null space of the first task with the input $\boldsymbol{z}_{null}$ from (8), that is, $\dot{\boldsymbol{s}} = \boldsymbol{J}_{null}\boldsymbol{N}_{task}^+\boldsymbol{K}\operatorname{vec}\underline{\boldsymbol{e}}_{task} + \boldsymbol{J}_{null}\boldsymbol{P}_{task}\boldsymbol{z}_{null}$, such that the minimal solution for $\left\|\dot{\boldsymbol{s}}_{null} - \boldsymbol{J}_{null}\dot{\boldsymbol{\Theta}}\right\|$, yields $\boldsymbol{z}_{null} = \left(\boldsymbol{J}_{null}\boldsymbol{P}_{task}\right)^+\left(\dot{\boldsymbol{s}}_{null} - \boldsymbol{N}_{task}^+\boldsymbol{K}\operatorname{vec}\underline{\boldsymbol{e}}_{task}\right)$.

The task-redundancy self-motion can be exploited to recover from escapable singularities. Thus, the desired task

$\boldsymbol{s}_{null}$ must be defined as a manipulability function that quantifies the proximity of a singular configuration–usually, with the determinant or using the minimum singular value. We can extend the classic manipulability function from [20], [21] to address the singularity avoidance within the CDTS, $s_{null} = \sqrt{\det\left\{\boldsymbol{N}_{task}\boldsymbol{N}_{task}^{T}\right\}}$. It is easy to see that s_{null} is a non-negative scalar with zero value only when the Jacobian is not full rank. The gradient of the manipulability function over (4) may be inferred using Jacobi's formula: $\dot{\boldsymbol{s}}_{null} = \nabla s_{null} = \det\left\{\boldsymbol{N}_{task}\boldsymbol{N}_{task}^{T}\right\} \mathrm{tr}\{\frac{\partial \boldsymbol{N}_{task}}{\partial \boldsymbol{\Theta}}\boldsymbol{N}_{task}^{+}\}$, where the tr is the matrix trace. Hence, we can exploit the the cooperative self-motion to optimize the manipulability function as to escape singularities from both arms. The advantages over traditional singularity avoidance technique for a single-arm is the use of a single DOF optimization task acting on both manipulators—using classic techniques, one would be required to define a manipulability function for each arm demanding the use of two DOFs at nullspace.

The idea of optimizing a secondary task can be extended to avoid joint limits as in [19]. This can be extended to the cooperative space by considering the augmented joint vector (4), that is, $s_{null} = \frac{1}{2}(\boldsymbol{\Theta} - \bar{\boldsymbol{\Theta}})^{T}(\boldsymbol{\Theta} - \bar{\boldsymbol{\Theta}})$, with $\bar{\boldsymbol{\Theta}}$ being the central joint values. The differential forward kinematics in this case is given by $\dot{s}_{null} = (\boldsymbol{\Theta} - \bar{\boldsymbol{\Theta}})^{T}\dot{\boldsymbol{\Theta}}$.

Moreover, during the trajectory execution, we may want to prioritize some particular cooperative variables in contrast to others. For instance, for a particular cooperative task, the absolute position may be more relevant than both manipulators orientation. Using a task-priority controller both relative and absolute orientations may be modeled as secondary optimization problems relaxing the number of constraints for the main control problem. In addition, the cooperative self-motion can also be used to optimize secondary constraints in the lack of available DOFs—for instance, in the case where there is only six DOFs available but we seek to control the absolute cooperative position and both relative and absolute orientations. In this scenario, the tasks may compete for the motion in the null space hierarchically [22], [23] or within a convex context as described in [24]. The latter technique will be used throughout this paper, as it provides a better enlightenment of the competition process. The resulting cooperative task-priority controller for the primitives yields

$$\dot{\boldsymbol{\Theta}} = \boldsymbol{N}_{task}^{+}\boldsymbol{K}\,\mathrm{vec}\,\underline{\boldsymbol{e}}_{task} + \boldsymbol{P}_{task}\left(\chi\boldsymbol{z}_{null,1} + (1-\chi)\boldsymbol{z}_{null,2}\right), \tag{9}$$

where the cooperative self-motion dynamics is projected upon both vector inputs with $\chi\in[0,1]$ being a smooth function that yields importance to either secondary tasks,

$$\begin{aligned} \boldsymbol{z}_{null,1} &= (\boldsymbol{J}_{null,1}\boldsymbol{P}_{task})^{+}\left(\dot{\boldsymbol{s}}_{null,1} - \boldsymbol{N}_{task}^{+}\boldsymbol{K}\mathrm{vec}\,\underline{\boldsymbol{e}}_{task}\right) \\ \boldsymbol{z}_{null,2} &= (\boldsymbol{J}_{null,2}\boldsymbol{P}_{task})^{+}\left(\dot{\boldsymbol{s}}_{null,2} - \boldsymbol{N}_{task}^{+}\boldsymbol{K}\mathrm{vec}\,\underline{\boldsymbol{e}}_{task}\right), \end{aligned} \tag{10}$$

where $\dot{\boldsymbol{s}}_{null,i}$ and $\boldsymbol{J}_{null,i}$, $i=\{1,2\}$, are the secondary task to be optimized and its corresponding Jacobian.

Remark 1: The cooperative task-redundancy based controller can also be easily redefined as to use recent self-motion strategies for single-arm manipulation, such as [22] that regards k levels of the hierarchical subtasks or as in [13] that regards a directional null space approach.

C. Singularity-robust task primitive controller

In the case of non-redudant cooperative tasks or inescapable singularities, self-motion can not be used since there is only one feasible joint configuration for both arms to achieve a desired cooperative task trajectory. Hence, the task trajectory must be modified in order to avoid singularities. In this scenario, the most classic techniques for singularity avoidance are the use of truncated SVD pseudo-inverse or Singularity Robust Inverses (SRIs) which yields a new task-trajectory modifying the differential kinematics to avoid singular configurations. Despite the importance within a complex manipulation scenario, to the best of the authors knowledge, there is no result in the cooperative manipulation literature that explicitly concerns singularity-robustness techniques—nor within existing control solutions based on dual quaternion algebra. The attractiveness of combining both unit dual quaternion representation and singularity robust techniques lies in a fully representation and kinematic singularity-free cooperative manipulation for redundant and non-redundant tasks.

So, we must regard the least square solution for the differential kinematics with the Moore-Penrose pseudoinvese, $\boldsymbol{N}_{task}^{\dagger} = \boldsymbol{N}_{task}^{T}\left(\boldsymbol{N}_{task}\ \boldsymbol{N}_{task}^{T}\right)^{-1}$.This, however, will still yield problematic results when near singularity regions. Analyzing the Jacobian through singular value decomposition (SVD) results in $\boldsymbol{N}_{task}^{\dagger} = \sum_{i=1}^{\min\{m,n\}}\sigma_i u_i v_i^{T}$, where σ_i is a singular value with associated m-sized output and n-sized input vectors u_i and v_i. When inverted we have $\boldsymbol{N}_{task}^{\dagger} = \sum_{i=1}^{\min\{m,n\}}\frac{1}{\sigma_i}u_i v_i^{T}$,that will result in infinite velocities whenever $\sigma_i \longrightarrow 0$. As a solution we may truncate the velocity functions to zero when near any manipulator singularities—known as truncated SVD. This might preserve both arms' integrity, keeping it from trying to reach impossible positions, but at the expense of a smooth, continuous trajectory.

A second strategy lies in modifying the trajectory of the end effector to avoid singularities. This can be done by adding a dampening factor λ_{DLS} to the Jacobian, such as $\boldsymbol{N}_{task}^{\dagger,DLS} = \boldsymbol{N}_{task}^{T}\left(\boldsymbol{N}_{task}\ \boldsymbol{N}_{task}^{T} + \lambda_{DLS}^{2}\boldsymbol{I}\right)^{-1}$.With the addition of λ_{DLS}, the SVD of the inverted damped least square (DLS) Jacobian then becomes$\boldsymbol{N}_{task}^{\dagger,DLS} = \sum_{i=1}^{\min\{m,n\}}\frac{\sigma_i}{\sigma_i^2+\lambda^2}u_i v_i^{T}$.We then notice that when σ_i is considerably higher than λ_{DLS}, the output of the Jacobian will remain mostly unchanged and when σ_i approaches λ_{DLS} the resulting map will be half of the one obtained from the Moore-Penrose pseudoinvese, only slightly altering the cooperative task trajectory. This damped least square solution ensures a continuity in the cooperative task trajectory, but as a disadvantage it will add to the error's norm.

For lower errors, Chiaverini [25] introduced a SRI strategy for single-arms based on an adaptive dampening factor,

$$\lambda_{SRI} = \begin{cases} 0, & if\ \sigma_{min} \geq \mu, \\ \left(1 - \left(\frac{\sigma_{min}}{\mu}\right)^{2}\right)\lambda_{Max} & if\ \sigma_{min} < \mu. \end{cases} \tag{11}$$

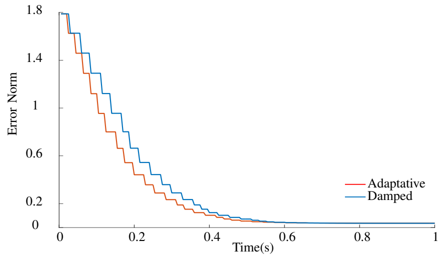


Fig. 2: Trajectory error norm for SRI and DLS techniques.

When the minimum singular value σ_{min} is far from the singular region μ, we do not add the dampening factor. Also, when σ_{min} is inside μ, we add a dampening factor proportional to the distance from the singular point. To further filter the result we define theSRI for the cooperative space as $\boldsymbol{N}^{\dagger,SRI}_{task} = \boldsymbol{N}^T_{task}\left(\boldsymbol{N}_{task}\ \boldsymbol{N}^T_{task} + \lambda^2_{SRI} u_{min} u^T_{min} + \beta^2 I\right)^{-1}$. In which u_{min} is the direction of the minimal singular value and β is an additional constant dampening factor such as $\beta^2 \ll \lambda^2$ that serves as a contingency for the cases where there are more than one singular value.

IV. Simulation and Experiments

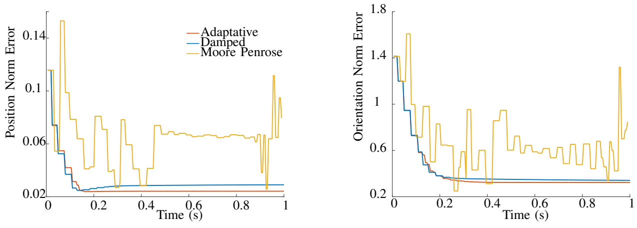
This section presents experiments using the NAO robot, that has five DOFs in each arm—thus being underactuated and prone to kinematic singularities. The experiments will take place on simulation, with tasks being performed on the Virtual Robot Experimentation Platform (V-REP) as well as on the physical robot, where we evaluate robustness when there are real-world disturbances.

We propose that the robot perform a task such as carrying a tray, in which we control absolute position, relative and absolute orientation and relative distance between the arms. Those tasks, together, will use all 10 DOFs of the augmented task space forthe NAO's arms. To evaluate the effects of the dampening factors and how it modifies the trajectory and avoids singularities, we propose a comparison between the norm of resulting trajectory errors associated with each method. We also aim to analyze this setup on the physical robotand visualize the effects of the robot's self motion, both to describe secondary tasks and to avoid joint limits.

The simulation was set considering an absolute pose reference and setting a relative distance of 0.3m, and a mirrored configuration between the arms. It was configured with a sample time of 5ms and was ran for both the DLS inversion method, with $\lambda_{DLS} = 10^{-4}$ and for the SRI, with $\lambda_{Max} = 10^{-2}$, $\mu = 10^{-2}$ and $\beta = 10^{-4}$.

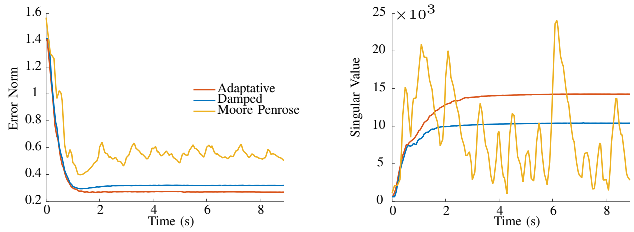
As we can observe in the graph of Fig.2, the SRI converges faster than the DLS. This can be understood by the fact that the adaptive dampening factor has a lesser effect on the trajectory for larger minimal singular values. Indeed this difference is less noticeable when we reach the steady state. The effector will converge to the desired pose in both cases, but due to the greater impact the DLS solution has on the trajectory, its convergence takes longer.

To further reduce the error we introduced the notion of self motion, used here primarily to avoid joint limits and performing less priority tasks —in this case, absolute and relative orientation control. The joint limit avoidance was defined with a gain of 0.2, as for the the secondary tasks, in accordance to (9), we used a gain of 0.8. The main task gain was set as $K = 0.4$. We then repeated the experiment for all three inversion methods in order to analyze their influence on the cooperative trajectory error—the dampening values were kept as in the previous experiment.



(a) (b)

Fig. 3: (a) shows the main task-error for each pseudo-inverse and(b) the secondary task error for the same application.



(a) (b)

Fig. 4: Pseudoinverse techniques in the NAO, in (a) the error norm and in (b) the minimum singular value on the trajectory.

From Fig.3, the convergence speed for the main task controller is faster than the one for the secondary task, which could be explained by the nullspace action. Because of the projector, we only have movements related to the secondary task which will not affect the main task's, resulting in idle time for that specific motion.Also, when comparing all three methods of inversion, it is clear that the Truncated Moore-Penrose yields the worst result, as it introduces discontinuities in every singular position.

In the physical platform we reduced the controlled primitives to the absolute position and relative and absolute orientations, leaving one DOF free, used to avoid joint limits in the nullspace. This time we used $\lambda_{DLS} = 10^{-3}$ and, for the SRI, $\lambda_{Max} = 10^{-2}$, $\mu = 10^{-2}$ and $\beta = 10^{-4}$.

Comparing the influence of the inversion methods on the trajectory,as can be seen in Fig.4(a), the results were similar to what we obtained from simulation. The truncated Moore-Penrose did not converge due to discontinuities on its output. Both dampened results converged, and the SRI kept the trajectory further from singularities and with smaller steady-state errors. This can be explained by the lesser influence of λ on the SRI. From Fig.4(b) we see that no singular configuration was reached.

With the SRI we also analyzed the effects of optimization-based techniques in the nullspace for avoiding joint limits. When joint limits are reached the robot has a mechanical restrain that keeps it from continuing to move, affecting the

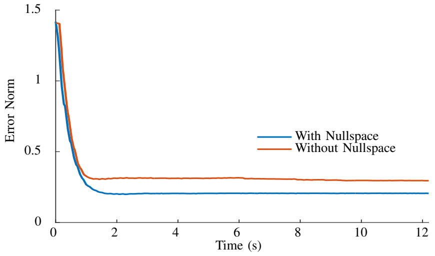
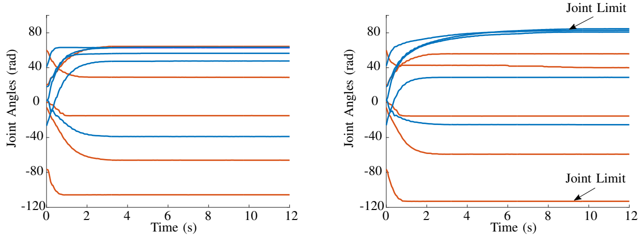


Fig. 5: Error norm on the NAO robot with and without the use of nullspace optimization aiming joint limits avoidance.



(a)

(b)

Fig. 6: The plot (a) shows the robot joints along the trajectory using the nullspace optimization whereas plot (b) shows that without the nullspace the robot movement is hampered by two joints reaching their limits.

steady state error. From Fig.6, we see that two joints reached their limits, resulting in a larger error. When we add a task to avoid those limits on the nullspace, it works without affecting the effector's position and thus the cooperative variables. In Fig.5 we see that there is a reduction in the steady-state error. In Fig.6(a) the joints previously limited are better positioned, allowing the effector to move more freely, approach its target pose and achieve a desired cooperative task.

V. Conclusion

This work used the algebra of dual quaternions for the cooperative context of bi-manual manipulation. We extended the strategies for joint limit and robust singularity avoidance to a cooperative, task-oriented context, dynamically avoiding kinematic singularities for redundant and non-redundant cooperative task trajectories. A task self-motion defined over the CDTS further enhanced reactiveness during execution to avoid joint limits or singularities and allowed the definition of task-priority controllers. Strategies proposed on the cooperative space were validated on an underactuated platform, which adds to the model's complexity for not having extra DOFs when executing a combination of tasks—highlighting the importance of proper singularity-robust solutions and joint limits avoidance in the cooperative task-space.

Acknowledgements

The authors would like to thank the support of CAPES, CNPQ as well as the Foundation for Research Support of the Federal District FAP-DF.

References

[1] B. V. Adorno, "Two-arm Manipulation: From Manipulators to Enhanced Human-Robot Collaboration," PhD thesis, LIRMM - Université Montpellier 2, Montpellier, France, 2011.

[2] O. Khatib, "Object manipulation in a multi-effector robot system," in *Proceedings of the 4th International Symposium on Robotics Research*. Cambridge, MA, USA: MIT Press, 1988, pp. 137–144.

[3] M. Uchiyama and P. Dauchez, "A symmetric hybrid position/force control scheme for the coordination of two robots," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, Apr 1988, pp. 350–356 vol.1.

[4] T. Connolly and F. Pfeiffer, "Cooperating manipulator control using dual quaternion coordinates," in *IEEE Conference on Decision and Control*, Lake Buena Vista, 1994, pp. 2417–2418.

[5] P. Chiacchio, S. Chiaverini, and B. Siciliano, "Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system," *J. Dyn. Sys., Meas., Control*, vol. 118, no. 4, pp. 691–697, 1996.

[6] F. Caccavale, P. Chiacchio, and S. Chiaverini, "Task-space regulation of cooperative manipulators," *Automatica*, vol. 36, no. 6, pp. 879–887, 2000.

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

[8] G. P. Incremona, G. D. Felici, A. Ferrara, and E. Bassi, "A supervisory sliding mode control approach for cooperative robotic system of systems," *IEEE Systems Journal*, vol. 9, no. 1, pp. 263–272, March 2015.

[9] L. F. C. Figueredo, "Kinematic control based on dual quaternion algebra and its application to robot manipulators," Ph.D. dissertation, University of Brasilia, Brazil, 2016.

[10] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, 1992.

[11] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Verlag, 2009.

[12] N. Bedrossian, "Classification of singular configurations for redundant manipulators," in *IEEE International Conference on Robotics and Automation*, may 1990, pp. 818 –823.

[13] N. Mansard and F. Chaumette, "Directional redundancy for robot control," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1179 – 1192, 2009.

[14] J. M. Selig, *Geometric Fundamentals of Robotics*, 2nd ed. Springer-Verlag New York Inc., 2005.

[15] J. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*. Princeton University Press, 1999.

[16] L. F. C. Figueredo, B. V. Adorno, J. Y. Ishihara, and G. A. Borges, "Switching strategy for flexible task execution using the cooperative dual task-space framework," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 1703–1709.

[17] ——, "Robust kinematic control of manipulator robots using dual quaternion representation," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 1949–1955.

[18] J. Baillieul, J. Hollerbach, and R. Brockett, "Programming and control of kinematically redundant manipulators," in *The 23rd IEEE Conference on Decision and Control*, Dec 1984, pp. 768–774.

[19] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, no. 12, pp. 868–871, 1977.

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

[21] G. Marani, J. Kim, J. Yuh, and W. K. Chung, "A real-time approach for singularity avoidance in resolved motion rate control of robotic manipulators," in *IEEE International Conference on Robotics and Automation*, no. May, 2002, pp. 1–6.

[22] P. Baerlocher and R. Boulic, "An inverse kinematics architecture enforcing an arbitrary number of strict priority levels," *The Visual Computer*, vol. 20, no. 6, pp. 402–417, 2004.

[23] J. Kim, G. Marani, W. K. Chung, and J. Yuh, "A general singularity avoidance framework for robot manipulators: task reconstruction method," in *IEEE International Conference on Robotics and Automation*, April 2004.

[24] A. Cherubini and F. Chaumette, "A redundancy-based approach for obstacle avoidance in mobile robot navigation," in *IEEE/RSJ International Conf. on Intelligent Robots and Systems*, 2010, pp. 5700–5705.

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