# Incorporating set-based control within the singularity-robust multiple task-priority inverse kinematics

Gianluca Antonelli, Signe Moe, Kristin Y. Pettersen

***Abstract*— Inverse kinematics is an active research domain in robotics since several years due to its importance in multiple robotics application. Among the various approaches, differential inverse kinematics is widely used due to the possibility to real-time implementation. Redundant robotic systems exhibit more degrees of freedom than those strictly required to execute a given end-effector task, in such a case, multiple tasks can be handled simultaneously in, e.g., a task-priority architecture. This paper addresses the systematic extension of the multiple tasks singularity robust solution, also known as Null-space Based Behavioral control, to the case of set-based control tasks, i.e., tasks for which a range, rather than a specific value, is assigned. This is the case for several variables such as, for example, mechanical joint limits of robotic arms as well as obstacle avoidance for any kind of robots. Numerical validation are provided to support the solution proposed.**

## I. Introduction

In the recent years robotic systems with a large number of Degrees of Freedom (DOFs) are commonly used in several applications beyond the original industrial environment [1]. Examples are the underwater [2], [3], aerial [4] or humanoid [5].

Going out from the industrial, mainly structured, environment requires algorithms able to deal with real-time trajectory generation of the end-effector configuration. This requirement, together with the assumption of non-trivial robotic kinematic structure, practically imposes the use of differential approaches. Unstructured and non-repetitive environments, however, require that the algorithms are able to handle multiple control objectives such as, for example, the mechanical joint limits, the avoidance of obstacles, the orientation of directional sensors, the arm manipulability, etc.

In some cases it is of interest to assign a precise value to the controlled variable such as, e.g., the end-effector configuration while in some other it is of interest only to keep the variable within a certain region of interest (area of satisfaction according to [5]) such as in case of the joint limits. We will refer to the latter as *set-based control* while most of the literature uses the term inequality constraints due to the fact that, when on the boundary of the set, an inequality condition on the task space velocity holds. For the reasons that will be explained in the paper, the term set-based task will be preferred instead.

G. Antonelli is with the Department of Electrical and Information Engineering, University of Cassino and Southern Lazio, Italy, and with ISME (Integrated Systems for Marine Environment) antonelli@unicas.it.

S. Moe and K.Y. Pettersen are with the Center for Autonomous Marine Operations and Systems (AMOS), Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway {signe.moe, kristin.y.pettersen}@itk.ntnu.no

In [6], the use of the pseudoinverse is first recognized as a promising tool to achieve inverse kinematics in robotic applications. In [7] the null-space projector is considered in the solution to achieve secondary control objectives afforded via a gradient-based approach. Secondary objectives are defined and handled in a task-priority approach in [8], [9]. The work [10] extends this approach to multiple tasks. A first attempt to enlarge the solution to systematically include also inequality (or set-based) tasks is proposed in [11], further improved in [5]. In [2] inequalities are handled too, in the framework of task prioritization, by resorting to proper activation and regularization functions. A recent application is analyzed in [12].

Notice that the term *hierarchy* and *priority* are often used as synonyms in the literature. In the following, the latter will be used.

Starting from [8], a different path which guarantees singularity robustness is followed by [13], further extended and analyzed to multiple tasks in priority in [14], [15], [16]. As recognized in [11], inclusion of inequality constraints still is missing for this approach. This paper covers this gap by systematically handling them. A numerical case study shows the effectiveness of the approach.

## II. Background

### A. Singularity-robust-task-priority inverse kinematics

Let us define as $\boldsymbol{\sigma}(t) \in \mathbb{R}^m$ the task variable to be controlled and $\boldsymbol{q}(t) \in \mathbb{R}^n$ the vector of the system configuration, the following holds:

$$\boldsymbol{\sigma}(t) = \boldsymbol{f}(\boldsymbol{q}(t)) \tag{1}$$

with the corresponding differential relationship:

$$\dot{\boldsymbol{\sigma}}(t) \;=\; \frac{\partial \boldsymbol{f}(\boldsymbol{q}(t))}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}}(t) \;=\; \boldsymbol{J}(\boldsymbol{q}(t)) \dot{\boldsymbol{q}}(t)\,, \tag{2}$$

where $\boldsymbol{J}(\boldsymbol{q}(t)) \in \mathbb{R}^{m \times n}$ is the configuration-dependent analytical task Jacobian matrix and $\dot{\boldsymbol{q}}(t) \in \mathbb{R}^n$ is the system velocity.

The value of $n$ changes depending on the considered robotic system, a *classical* industrial manipulator exhibits $n = 6$, the underwater vehicle-manipulator system used in the project TRIDENT [2] has $n = 13$, being 6 the DOFs provided by the vehicle and 7 by the arm. Humanoids exhibits values up to $n = 30$ [5].

Let us first assume a single $m$-dimensional task to be followed, for which a desired values $\boldsymbol{\sigma}_{\mathrm{des}}(t) \in \mathbb{R}^m$ is assigned. The motion references $\boldsymbol{q}_{\mathrm{des}}(t) \in \mathbb{R}^n$ for the robotic system may be computed by integrate the locally inverse mapping

of (2) achieved imposing minimum-norm velocity [17]. The following least-squares solution is given by (dependencies are dropped out to increase readability):

$$\dot{\boldsymbol{q}}_{\text{des}} = \boldsymbol{J}^{\dagger}\dot{\boldsymbol{\sigma}}_{\text{des}} = \boldsymbol{J}^{\text{T}}\left(\boldsymbol{J}\boldsymbol{J}^{\text{T}}\right)^{-1}\dot{\boldsymbol{\sigma}}_{\text{des}}\,, \tag{3}$$

where $\boldsymbol{J}^{\dagger}$, implicitly defined in the above equation for full rank matrices, is the right pseudoinverse of $\boldsymbol{J}$; in the general case, the pseudoinverse is the matrix that satisfies the four Moore-Penrose conditions [18].

The vector $\boldsymbol{q}_{\text{des}}$ achieved by time integral of (3) is prone to drift, i.e., used back in eq. (1) it does not result in $\boldsymbol{\sigma}_{\text{des}}$. A Closed Loop Inverse Kinematics (CLIK) version of the algorithm is usually implemented [13], namely,

$$\dot{\boldsymbol{q}}_{\text{des}} = \boldsymbol{J}^{\dagger}\left(\dot{\boldsymbol{\sigma}}_{\text{des}} + \boldsymbol{\Lambda}\widetilde{\boldsymbol{\sigma}}\right) = \boldsymbol{J}^{\dagger}\dot{\boldsymbol{\sigma}}_{\text{ref}}\,, \tag{4}$$

where $\widetilde{\boldsymbol{\sigma}} \in \mathbb{R}^m$ is the task error defined as

$$\widetilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_{\text{des}} - \boldsymbol{\sigma} \tag{5}$$

and $\boldsymbol{\Lambda} \in \mathbb{R}^{m\times m}$ is a positive-definite matrix of gains; the reference signal $\dot{\boldsymbol{\sigma}}_{\text{ref}}$ embeds thus the closed loop action.

In case of system redundancy, i.e., if $n > m$, the classic general solution contains a null projector operator [7]:

$$\dot{\boldsymbol{q}}_{\text{des}} = \boldsymbol{J}^{\dagger}\dot{\boldsymbol{\sigma}}_{\text{ref}} + \left(\boldsymbol{I}_n - \boldsymbol{J}^{\dagger}\boldsymbol{J}\right)\dot{\boldsymbol{q}}_{\text{null}}, \tag{6}$$

where $\boldsymbol{I}_n$ is the $(n \times n)$ Identity matrix and the vector $\dot{\boldsymbol{q}}_{\text{null}} \in \mathbb{R}^n$ is an arbitrary system velocity vector. It can be recognized that the operator $\left(\boldsymbol{I}_n - \boldsymbol{J}^{\dagger}\boldsymbol{J}\right)$ projects a generic velocity vector in the null space of the Jacobian matrix. This corresponds to generate a motion of the robotic system that does not affect that of the given task. A classic approach consists in using the null space projector to add to the sole task a gradient based term to perform some kind of optimization [19], [20].

For highly redundant systems, multiple tasks can be arranged in priority. Let us consider three tasks that will be denoted with the subscript a, b and c, respectively:

$$\boldsymbol{\sigma}_{\text{a}} = \boldsymbol{f}_{\text{a}}(\boldsymbol{q}) \in \mathbb{R}^{m_a} \tag{7}$$

$$\boldsymbol{\sigma}_{\text{b}} = \boldsymbol{f}_{\text{b}}(\boldsymbol{q}) \in \mathbb{R}^{m_b} \tag{8}$$

$$\boldsymbol{\sigma}_{\text{c}} = \boldsymbol{f}_{\text{c}}(\boldsymbol{q}) \in \mathbb{R}^{m_c}. \tag{9}$$

For each of the tasks a corresponding Jacobian matrix can be defined, in detail $\boldsymbol{J}_{\text{a}} \in \mathbb{R}^{m_a\times n}$, $\boldsymbol{J}_{\text{b}} \in \mathbb{R}^{m_b\times n}$ and $\boldsymbol{J}_{\text{c}} \in \mathbb{R}^{m_c\times n}$. Let us further define the corresponding null space projectors for the first two tasks as

$$\boldsymbol{N}_{\text{a}} = \left(\boldsymbol{I}_n - \boldsymbol{J}_{\text{a}}^{\dagger}\boldsymbol{J}_{\text{a}}\right) \tag{10}$$

$$\boldsymbol{N}_{\text{b}} = \left(\boldsymbol{I}_n - \boldsymbol{J}_{\text{b}}^{\dagger}\boldsymbol{J}_{\text{b}}\right). \tag{11}$$

Following the discussion in [14], [16], among the possible generalization of the singularity-robust task priority inverse kinematic solution proposed in [13], by defining $\boldsymbol{J}_{\text{ab}} \in \mathbb{R}^{(m_a+m_b)\times n}$ as

$$\boldsymbol{J}_{\text{ab}} = \begin{bmatrix}\boldsymbol{J}_{\text{a}}\\ \boldsymbol{J}_{\text{b}}\end{bmatrix}, \quad \boldsymbol{N}_{\text{ab}} = \left(\boldsymbol{I}_n - \boldsymbol{J}_{\text{ab}}^{\dagger}\boldsymbol{J}_{\text{ab}}\right) \tag{12}$$

the following will be considered:

$$\dot{\boldsymbol{q}}_{\text{des}} = \underbrace{\boldsymbol{J}_{\text{a}}^{\dagger}\dot{\boldsymbol{\sigma}}_{\text{a,ref}}}_{\dot{\boldsymbol{q}}_{\text{a,des}}} + \boldsymbol{N}_{\text{a}}\underbrace{\boldsymbol{J}_{\text{b}}^{\dagger}\dot{\boldsymbol{\sigma}}_{\text{b,ref}}}_{\dot{\boldsymbol{q}}_{\text{b,des}}} + \boldsymbol{N}_{\text{ab}}\underbrace{\boldsymbol{J}_{\text{c}}^{\dagger}\dot{\boldsymbol{\sigma}}_{\text{c,ref}}}_{\dot{\boldsymbol{q}}_{\text{c,des}}} \tag{13}$$

where the definition of $\dot{\boldsymbol{\sigma}}_{\text{x,ref}}$ can be easily extrapolated from (4), the reference vectors embed the positive definite matrices $\boldsymbol{\Lambda}_{\text{a}} \in \mathbb{R}^{m_a\times m_a}$, $\boldsymbol{\Lambda}_{\text{b}} \in \mathbb{R}^{m_b\times m_b}$ and $\boldsymbol{\Lambda}_{\text{c}} \in \mathbb{R}^{m_c\times m_c}$, the priority of the tasks follows the increasing alphabetical order, being a the highest-priority task. Eq. (13) also implicitly defines the joint velocities $\dot{\boldsymbol{q}}_{\text{x,des}} \in \mathbb{R}^n$ that represent the desired joint velocity of the x task as if it was alone.

The generalization to $N$ tasks is straightforward: let us assume that $\boldsymbol{\sigma}_{\text{y}} \in \mathbb{R}^{m_y}$ is the $N$ th task, moreover, the task x precedes the task y in priority; in other words, the task x is the $N-1$ th priority task. Equation (13) can be rewritten as:

$$\dot{\boldsymbol{q}}_{\text{des}} = \boldsymbol{J}_{\text{a}}^{\dagger}\dot{\boldsymbol{\sigma}}_{\text{a,ref}} + \boldsymbol{N}_{\text{a}}\boldsymbol{J}_{\text{b}}^{\dagger}\dot{\boldsymbol{\sigma}}_{\text{b,ref}} + \cdots + \boldsymbol{N}_{\text{ab...x}}\boldsymbol{J}_{\text{y}}^{\dagger}\dot{\boldsymbol{\sigma}}_{\text{y,ref}} \tag{14}$$

in which $\boldsymbol{N}_{\text{ab...x}}$ is the Null space of the matrix

$$\boldsymbol{J}_{\text{ab...x}} = \begin{bmatrix}\boldsymbol{J}_{\text{a}}\\ \boldsymbol{J}_{\text{b}}\\ \vdots\\ \boldsymbol{J}_{\text{x}}\end{bmatrix}, \tag{15}$$

### *B. Set-based definitions*

The formulation above allows to control the defined variables to a desired, *punctual*, although time varying, value. This is often an over constraint for the robotic system in the sense that instead of exactly assign a value the control problem is defined in reaching, or avoiding, a certain region. Two classical although crucial examples concern obstacle avoidance and mechanical joint limits. Additional tasks might concern directional sensors such as, for example, an end-effector mounted camera or a vehicle-fixed directional communication device.

The mapping still is defined by eq. (1) for which $\boldsymbol{\sigma}_{\text{des}}(t)$ is not an analytical function but a set to own to, in detail, a hyperrectangle in $\mathbb{R}^m$:

$$\boldsymbol{\sigma}(t) = \boldsymbol{f}(\boldsymbol{q}(t)) \in \mathcal{S} \tag{16}$$

When a task is satisfied no action is required to the robotic system, i.e., the task is not explicitly included in the computation of $\dot{\boldsymbol{q}}_{\text{des}}(t)$. The set $\mathcal{S}$ is defined by upper and lower bounds on each of the component of the vector $\boldsymbol{\sigma}$. By considering, for example, a set limited only on one extreme the following needs to be satisfied

$$\sigma_j(t) \leq \sigma_{\max,j}. \tag{17}$$

It is often useful to define a *bearing area* by resorting to a threshold $\sigma_{\varepsilon,j} < \sigma_{\max,j}$ in order to define a set where the task is *close to be violated*, i.e., it is active. Figure 1 graphically represents the above definitions. Notice that the term *active* is not to be intended as in the active set method for optimization problems of inequality constraints [21].

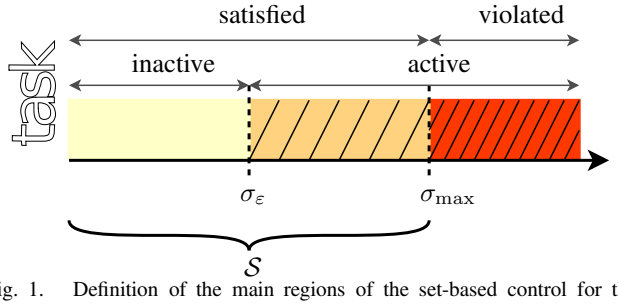


Fig. 1. Definition of the main regions of the set-based control for the example of an upper bounded set

When the $j$ component of the task lies in the bearing area, it is necessary to *push it away* from the boundary, i.e., to add an inequality constraint on the differential mapping of the kind:

$$0 \geq \dot{\sigma}_j(t) = \boldsymbol{J}_j(\boldsymbol{q}(t))\dot{\boldsymbol{q}}(t) \tag{18}$$

where $\boldsymbol{J}_j \in \mathbb{R}^{1\times n}$ is the corresponding row of the Jacobian associated to (16).

When this constraint can not be fulfilled (in terms that will be clarified in Section III) a vector $\dot{\boldsymbol{q}}_{\mathrm{x,des}}$ needs to be designed. One possibility is to select this velocity vector such as its projection on the task space is null, i.e., to *freeze* the task. The other possibility is to implement a feedback action aiming at smoothly move the variable to $\sigma_\varepsilon$.

It is worth noticing that values different from zero, bilateral set definition or absence of the bearing area might be easily considered if necessary. Also, each of the inequality above defines an hyperplane in $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ when exactly satisfied, i.e., with the equal sign.

A similar control problem is defined as *inequality objective* in [2] or inequality constraints in [11], [5]. The term constraint is focusing the attention on the differential relationship (18) while it is maybe important to keep in mind that the main control objective is at configuration level. The differential constraint, in fact, only arises when the task variable is in the bearing or the violated areas *and* eq. (18) is not satisfied. The latter, however, needs to be satisfied by the overall solution, i.e., a-posteriori, not necessarily when building the solution as done in [11]. This consideration will naturally lead to the proposed solution discussed in the Section III.

## III. Task-Priority Set-Based Objectives

In order to understand the proposed solution to handle set-based control let us consider a simple case study. A $n = 3$ DOF generic robotic system for which we have defined 3 mono-dimensional tasks characterized by unitary gains $\lambda_{\mathrm{x}}$ and constant desired value. At a certain time instant the following Jacobians and errors hold:

| task | Jacobian | $\tilde{\sigma}$ | kind |
|---|---|---|---|
| a | $\begin{bmatrix}1 & 0 & 1\end{bmatrix}$ | 0.01 | equality |
| b | $\begin{bmatrix}0 & 1 & 0\end{bmatrix}$ | -0.005 | set-based |
| c | $\begin{bmatrix}1 & 1 & 0\end{bmatrix}$ | -0.1 | equality |

and the second joint is *close* to its upper bound, i.e., we need to activate it. By locally inverting each of the tasks alone we can appreciate the velocity vectors required by each task

$$\dot{\boldsymbol{q}}_{\mathrm{a,des}} = \begin{bmatrix}0.5\\0\\0.5\end{bmatrix} \quad \dot{\boldsymbol{q}}_{\mathrm{b,des}} = \begin{bmatrix}0\\-0.005\\0\end{bmatrix} \quad \dot{\boldsymbol{q}}_{\mathrm{c,des}} = \begin{bmatrix}-0.05\\-0.05\\0\end{bmatrix}$$

which, mapped into the corresponding task spaces, gives back the errors of the tables above.

Coherently with the task-priority strategy the second and third third tasks need to be projected on the augmented higher-priority null space, i.e.:

$$\dot{\boldsymbol{q}}_{\mathrm{des}} = \dot{\boldsymbol{q}}_{\mathrm{a,des}} + \boldsymbol{N}_{\mathrm{a}}\dot{\boldsymbol{q}}_{\mathrm{b,des}} + \boldsymbol{N}_{\mathrm{ab}}\dot{\boldsymbol{q}}_{\mathrm{c,des}} = \begin{bmatrix}-0.02\\-0.005\\0.03\end{bmatrix}$$

which corresponds to

$$\dot{\sigma}_{\mathrm{a}} = \boldsymbol{J}_a\dot{\boldsymbol{q}}_{\mathrm{des}} = .01 \quad \dot{\sigma}_{\mathrm{b}} = -0.005 \quad \dot{\sigma}_{\mathrm{c}} = -0.025$$

i.e., the primary exactly fullfilled, the set-based also due to the orthogonality of the Jacobians and the third up to an error of $0.075$.

However, an a-posteriori reasoning leads to the conclusion that the third task can be satisfied and the second, for this specific configuration, may be simply ignored (notice the null-space projector):

$$\dot{\boldsymbol{q}}_{\mathrm{des}} = \dot{\boldsymbol{q}}_{\mathrm{a,des}} + \boldsymbol{N}_{\mathrm{a}}\dot{\boldsymbol{q}}_{\mathrm{c,des}} = \begin{bmatrix}-0.02\\-0.05\\0.03\end{bmatrix}.$$

which corresponds to

$$\dot{\sigma}_{\mathrm{a}} = \boldsymbol{J}_a\dot{\boldsymbol{q}}_{\mathrm{des}} = .01 \quad \dot{\sigma}_{\mathrm{b}} = -0.05 \quad \dot{\sigma}_{\mathrm{c}} = -0.07$$

i.e., the primary still exactly totally fullfilled and the third with an improved accuracy since the error decreased from $0.075$ to $0.03$. In addition, it can be noticed that it *helps* the second-priority, set-based, to move away from the boundary (in fact, the secondary task is simply the joint position and the constrain is thus: $\dot{q}_{2,\mathrm{des}} < 0$). The key idea is that the inclusion of the set-based tasks is not necessarily the best idea since the test of its necessity needs to be done a-posteriori, i.e., on the overall solution.

The generalization to a generic number of equality/set-based tasks follows. Let us assume that a number $n_i$ of set-based tasks is defined with a given priority for a total of $p$ tasks. At the generic sampling time eq. (16) is verified for each, a number $n_{set,a}$ are active and *should* be considered in the solution. However, instead of including each of the set-based task, with the corresponding null-space projector, we consider all the possible combinations of active/absent inequalities, i.e., for $2^{n_{set,a}}$ cases, by computing the solution (14). We thus have $2^{n_{set,a}}$ possible joint velocities among which the solution needs to be selected *a-posteriori*.

The generalization to a different number of tasks or to a different priority is then possible. The algorithm builds a set of possible solutions $\mathcal{Q}$ and null spaces $\mathcal{N}$. The initial desired

velocity is zero and the null space projector is the Identity matrix. Starting from the higher priority task, at each level of priority, the following is implemented:

1: Initialize $\mathcal{Q}$ with one null $n \times 1$ vector
2: Initialize $\mathcal{N}$ with one Identity matrix
3: **for** $i = 1$ to $p$ **do**
4: Compute $\boldsymbol{J}_{\mathrm{x}}$
5: Compute $\dot{\boldsymbol{q}}_{\mathrm{x,des}}$ according to eq. (4)
6: **if** i==set-based **then**
7: $\mathcal{Q}_{\mathrm{temp}} = \mathcal{Q}$
8: $\mathcal{N}_{\mathrm{temp}} = \mathcal{N}$
9: **end if**
10: Update $\mathcal{Q}$ summing to each the vector $\dot{\boldsymbol{q}}_{\mathrm{x,des}}$ pre-multiplied by the corresponding null space projectors in $\mathcal{N}$
11: Update $\mathcal{N}$ properly including, in each, the contribution of $\boldsymbol{J}_{\mathrm{x}}$
12: **if** i==set-based **then**
13: $\mathcal{Q} = \mathcal{Q} \bigcup \mathcal{Q}_{\mathrm{temp}}$
14: $\mathcal{N} = \mathcal{N} \bigcup \mathcal{N}_{\mathrm{temp}}$
15: **end if**
16: **end for**
17: **return** $\mathcal{Q}$ $\{$it contains $2^{n_{set,a}}$ elements$\}$

It is now necessary to select the *best* velocity among the possible solutions. A-posteriori it is sufficient to verify if a set-based task need to be activated or if it can be relaxed. Obviously, there is no guarantee that the lower priority tasks are fullfilled since stability conditions arise [16], however, if one of the potential velocities *pushes away* a set-based task from the boundary the latter can be ignored.

Selection of the solution is achieved in two steps: a) all the velocities that do not satisfy the set-based tasks put in the top priorities are erased and b) the maximum-norm solution of the remaining is selected.

Even if several top priority, set-based tasks are defined, for example all the joint limits, it is appropriate to consider them individually in order to handle only the critical joint and leave the other the possibility to move. The *worst* case, all the joint limits activated, would simply *stop* the robot motion until the lower priority velocities do not push it away from some limit.

The rational for picking the maximum-norm solution is due to the observation that each null-space projection filter out some velocity components and thus the less restrictive solution is the one with largest norm. Different policies may be adopted and are currently under investigation.

The proposed algorithm is thus deterministic, on the contrary to [5], which is recursive. Also, with respect to [11] and [22], together with [5] and [2], it does not need to have the set-based tasks all at the higher priority levels. Finally, with respect to [2], it never loses the priority among tasks.

## IV. Numerical validation

A simple 4-DOFs planar manipulator is considered to illustrate the results. Despite the simplicity of the robotic structure, all the considerations made in this paper arises. A *gymn* test has been designed in order to stress the algorithm by exciting several possible situations: several set-based higher priority tasks needed, all the set-based higher priority tasks excited (*block* of the robot for safety reason), an equality and a set-based lower-priority task. Also, the numerical values of the set boundaries as well as the desired values have been widely varied. One single case study will be reported here corresponding to:

| pr. | task | type | gain | min | max |
|---|---|---|---|---|---|
| 1 | mech. joint 1 | set-based | 2 | $-90°$ | $90°$ |
| 2 | mech. joint 2 | set-based | 2 | $-100°$ | $100°$ |
| 3 | mech. joint 3 | set-based | 2 | $-90°$ | $90°$ |
| 4 | mech. joint 4 | set-based | 2 | $-120°$ | $120°$ |
| 5 | e.e. obst. av. | set-based | 2 | 0.3 m | – |
| 6 | e.e. position | equality | 500 | – | – |
| 7 | e.e. orientation | set-based | 0.1 | $-45°$ | $45°$ |

Figure 2 shows the initial configuration, the blue line denotes the desired end-effector's path, it first goes up, then moves to the opposite extreme and finally comes back to the initial value. An animation of the simulation is available http://webuser.unicas.it/lai/robotica/video/antonelli_med15_01.avi. It is suggested to watch the video before continuing the reading to appreciate the movement.

The second link is red since it is intentionally placed in the joint's limit bearing area. An obstacle (dashed grey) is also present in the environment.

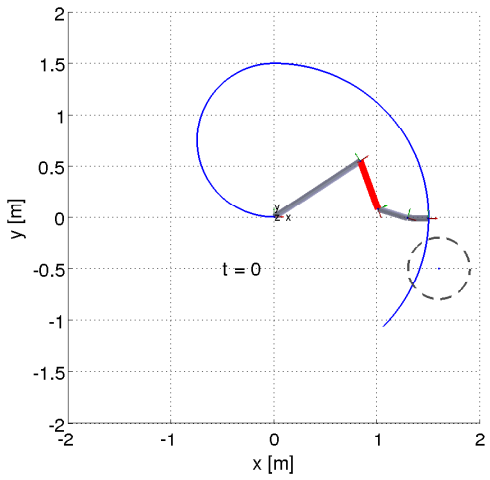


Fig. 2. Initial configuration, the blue line denotes the desired end-effector's path, it first goes up, then moves to the opposite extreme and finally comes back to the initial value. The second link is red since it is intentionally placed in the joint's limit bearing area. In dashed grey an obstacle

Figure 3 and 4 show two snapshots. It can be appreciated that the manipulator moves with the second link in the bearing area while following the desired end-effector trajectory until it goes out of the workspace and 3 out of 4 joints are in their corresponding bearing areas.

An additional snapshots is shown in figure 5 when the arm is avoiding the obstacle.

Activations (in red) of the set-based tasks can be appreciated by figure 6. It can be noticed that the equality task is always active while the set-based are activated or not

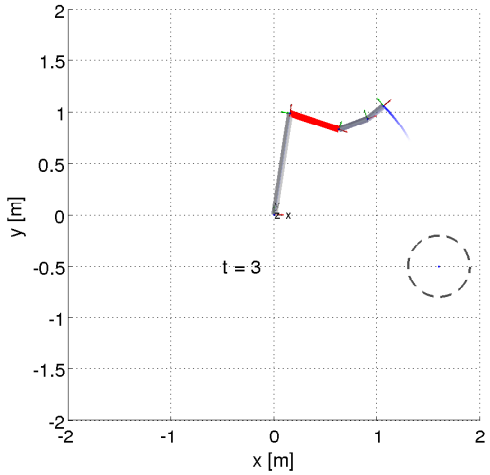


Fig. 3. At $t = 3\,\mathrm{s}$ the end-effector is following the trajectory with the second joint always *close* to its mechanical limit and thus active

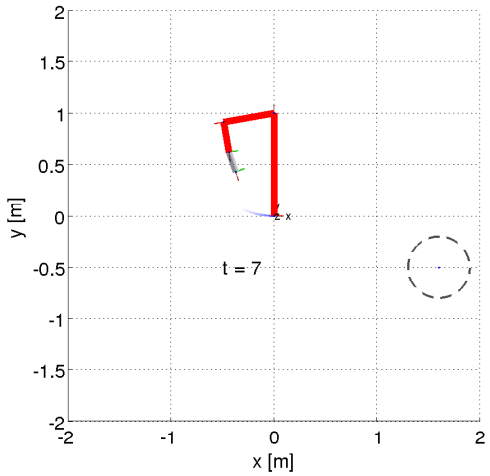


Fig. 4. At $t = 7\,\mathrm{s}$ the desired trajectory is out of the workspace and 3 of the 4 links are in their bearing area

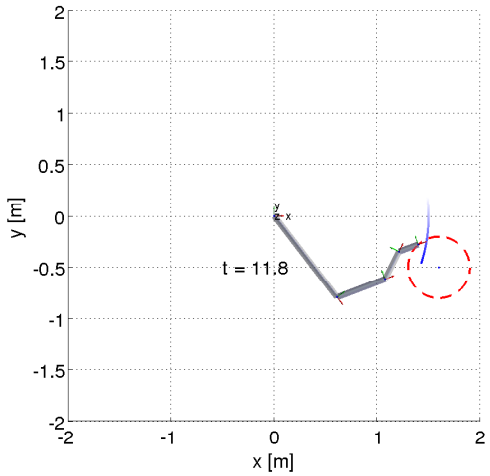


Fig. 5. At $t = 11.8\,\mathrm{s}$ the end-effector obstacle avoidance needs to be activated

depending on the configuration. In particular, for $t = 7\,\mathrm{s}$ the arm is in the configuration shown in figure 4 with 3 joints *locked*. For $t = 12\,\mathrm{s}$ and $t = 14\,\mathrm{s}$ the arm is avoiding the obstacle.

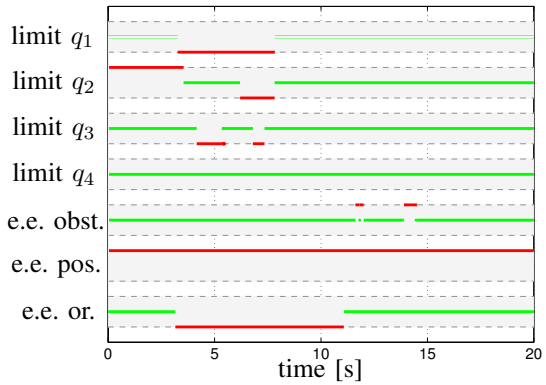


Fig. 6. Task activation (in red) during the movement. The second joint is intentionally in the bearing area at start, the equality task is marked as red for all the movement. For $t = 7\,\mathrm{s}$ the arm is in the configuration shown in figure 4 with 3 joints *locked*. For $t = 12\,\mathrm{s}$ and $t = 14\,\mathrm{s}$ the arm is avoiding the obstacle

Figure 7 shows the joints values with the corresponding mechanical limits. The set-based control is nicely working and keeping the joints all within their admissible values even for *stressing* trajectories.

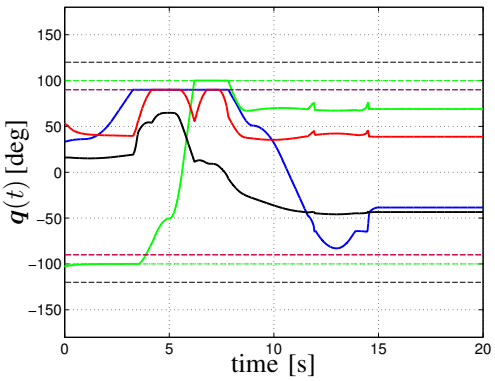


Fig. 7. Joints variables (colored solid) with corresponding limits (corresponding color, dashed). The activation can be appreciated

Distance from the obstacle, a lower-bounded set-based control variable, is shown in Figure 8.

Figure 9 shows the end-effector tracking error. It can be noticed that the error is significant when the higher-priority, safety-related, tasks are active during obstacle avoidance or when the joint limits prevent the robot to reach the desired values.

## V. Conclusions

This paper addressed the systematic extension of the multiple tasks singularity robust solution, also known as Null-space Based Behavioral control, to the case of set-based control tasks, i.e., tasks for which a range, rather than a specific value, is assigned.

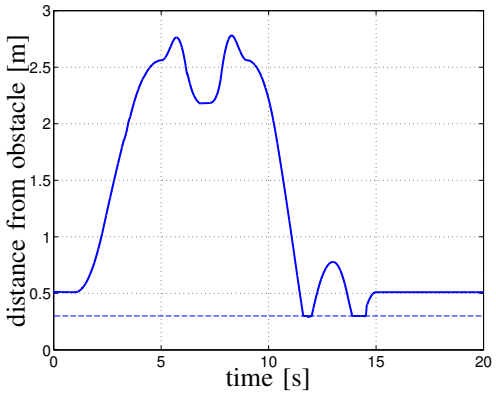


Fig. 8. End-effector distance from the obstacle (solid) with corresponding limits (dashed)

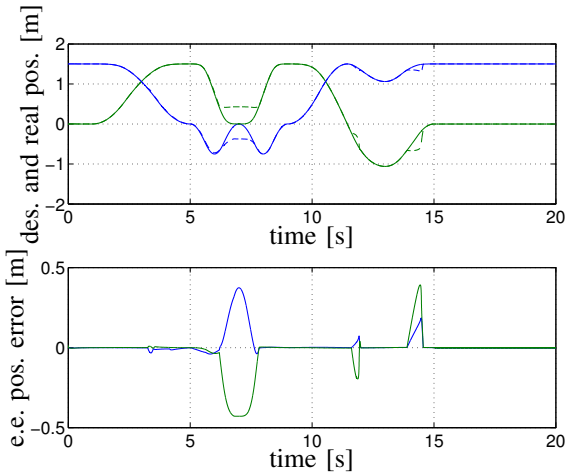


Fig. 9. Desired and real end-effector position (top) and tracking errors (bottom). Significant errors can be appreciated when the joint limits and the obstacle avoidance limit the arm movement

The proposed algorithm has been widely tested on a simple robotic structure and demonstrated to be efficient and to fit nicely the NSB approach.

In this work no consideration has been made on the computational load or on the possibility to speed up the solution computation by adopting proper numerical algorithms. In particular, it is known that it is possible to compute recursively the null-space with respect to the number of tasks [23]. In case an additional task needs to be add to eq. (14), for example, the following might be adopted:

$$\boldsymbol{N}_{\mathrm{ab}\ldots\mathrm{y}} = \boldsymbol{N}_{\mathrm{ab}\ldots\mathrm{x}} \left[ \boldsymbol{I} - \left( \boldsymbol{J}_{\mathrm{y}} \boldsymbol{N}_{\mathrm{ab}\ldots\mathrm{x}} \right)^{\dagger} \left( \boldsymbol{J}_{\mathrm{y}} \boldsymbol{N}_{\mathrm{ab}\ldots\mathrm{x}} \right) \right].$$

Future works concern analytical discussion, comparison with the state of the art and test on more elaborated robotic structures.

## Acknowledgements

This work was supported by the Research Council of Norway through the Center of Excellence funding scheme, project number 223254, by the Italian *Ministero dell'Istruzione, dell'Università e della Ricerca, PRIN 2010-2011* through the project MARIS (protocol 2010FBLHRJ) and by the European Community through the projects ARCAS (FP7-287617), EuRoC (FP7-608849), DexROV (H2020-635491) and AEROARMS (H2020-644271).

## References

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

[2] E. Simetti, G. Casalino, S. Torelli, A. Sperindé, and A. Turetta, "Floating underwater manipulation: Developed control methodology and experimental validation within the TRIDENT project," *Journal of Field Robotics*, vol. 31(3), pp. 364–385, 2013.

[3] G. Antonelli, *Underwater robots*. Heidelberg, D: Springer Tracts in Advanced Robotics, Springer-Verlag, 3rd ed., January 2014.

[4] A. Jimenez-Cano, J. Martin, G. Heredia, and A. Ollero, "Control of an aerial robot with multi-link arm for assembly tasks," in *2013 IEEE International Conference on Robotics and Automation*, (Karlsruhe, Germany), May 2013.

[5] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *International Journal of Robotics Research*, 2013.

[6] D. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Transactions on Man-Machine Systems*, vol. 10, no. 2, pp. 47–52, 1969.

[7] A. Liégeois, "Automatic supervisory control of the configuration and behavior of muldibody mechanisms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 7, pp. 868–871, 1977.

[8] A. Maciejewski and C. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 109–117, 1985.

[9] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *The International Journal Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.

[10] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Proceedings 5th International Conference on Advanced Robotics*, (Pisa, I), pp. 1211–1216, 1991.

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

[12] H. Azimian, T. Looi, and J. Drake, "Closed-loop inverse kinematics under inequality constraints: Application to concentric-tube manipulators," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 498–503, IEEE, 2014.

[13] 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.

[14] N. Mansard and F. Chaumette, "Task sequencing for high-level sensor-based control," *IEEE Transactions on Robotics and Automation*, vol. 23, no. 1, pp. 60–72, 2007.

[15] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The Null-Space-based Behavioral control for autonomous robotic systems," *Journal of Intelligent Service Robotics*, vol. 1, pp. 27–39, Jan. 2008.

[16] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems," *IEEE Transactions on Robotics*, vol. 25, pp. 985–994, October 2009.

[17] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *Journal of Intelligent Robotic Systems*, vol. 3, no. 3, pp. 201–212, 1990.

[18] G. Golub and C. Van Loan, *Matrix Computations*. Baltimore, MD: The Johns Hopkins University Press, third ed., 1996.

[19] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*. Reading, MA: Addison-Wesley, 1991.

[20] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*. London, UK: Springer-Verlag, 2nd ed., 2000.

[21] J. Nocedal and S. Wright, *Numerical Optimization 2nd*. Heidelberg, D: Springer, 2nd ed., 2006.

[22] M. de Lasa, I. Mordatch, and A. Hertzmann, "Feature-based locomotion controllers," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, 2010.

[23] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, vol. 2, no. 04, pp. 505–518, 2005.