Predictive Inverse Kinematics for Redundant Manipulators: Evaluation in Re-Planning Scenarios

Marco Faroni * Manuel Beschi ** Antonio Visioli*

* Dipartimento di Ingegneria Meccanica e Industriale, University of Brescia, Brescia, Italy (e-mail: {m.faroni003, antonio.visioli}@unibs.it).
** Istituto di Tecnologie Industriali e Automazione, National Research Council, Milan, Italy (e-mail: manuel.beschi@itia.cnr.it)

Abstract: In this paper, we analyze the effectiveness of a predictive redundancy resolution for constrained manipulators in case of on-line re-planning. The method is suitably modified to cope with re-planning issues, such as possible infeasible motions and position errors. Several re-planning scenarios are evaluated. Their definition is based on the smoothness of the re-planned task with respect to the current state of the robot. This allows a deep investigation of the behavior of the method under different conditions. Simulations results on a 7-degree-of-freedom KUKA LWR IV demonstrate remarkable advantages of the predictive method, both in terms of task error and redundancy exploitation.

Keywords: robot manipulators, redundant robots, model predictive control, inverse kinematics, joint limits, optimal redundancy handling, re-planning.

1. INTRODUCTION

Redundant manipulators have been deeply investigated by the robotics community due to the well-known advantages they present. As a matter of fact, the redundant degrees of freedom can be exploited to satisfy secondary goals, such as dexterity improvement (Lee and Chang, 2015), collision avoidance (Maciejewski and Klein, 1985), manipulability maximization (Faroni et al., 2016), joint range availability (Liegeois, 1977), imitation of human-like motions (Lamperti et al., 2015). By contrast, the presence of redundant degrees of freedom leads to more complex planning and control algorithms. In particular, the Inverse Kinematics (IK) allows infinite solutions and the choice of a suitable one is usually not trivial.

Different approaches have been proposed to cope with this issue. As a first classification, methods in the literature can be subdivided into global and local ones (Nakamura, 1990). The former ones typically consider the whole desired task in a single optimization problem to find a globally-optimal solution with respect to a given metrics. The latter ones solve the IK point-by-point, by taking into account just information about the current configuration of the system (i.e., the current joint configuration and the desired motion at the current cycle).

Obviously, global methods give better results in terms of global execution of the task. However, they present relevant drawbacks, mainly related to their high computational complexity. First of all, they lead to complex nonlinear optimization problems that are usually solved by means of approximated methods. Moreover, their computational burden does not make them suitable for online implementation, thus, they cannot cope with re-planning strategies. For these reasons, local methods are often preferred. They usually address the IK at differential level (Flacco et al., 2015). In the typical approach, the Jacobian of the manipulator is inverted by means of a generalized pseudo-inverse, in order to calculate joint velocities or accelerations. Then, joint configurations are computed by numerical forward integration (Siciliano, 1989). A Closed Loop Inverse Kinematics (CLIK) strategy is then used to recover from possible position errors given by numerical drift, out-of-path initial configuration, or re-planning (Sciavicco and Siciliano, 1988; Chiacchio et al., 1991). Secondary tasks are usually handled by projecting them into the null space of the Jacobian, so that they do not affect the execution of the primary task (Siciliano and Slotine, 1991; Ott et al., 2015). The main disadvantage of this approach lies in its inability of handling the robot kinematic and/or dynamic constraints. As a consequence, if the computed solution exceeds the physical limits of the robot, saturations may occur, leading to the deformation of the desired task.

Recent methods address this issue by converting the IK problem into a constrained Quadratic Program (QP) (Flacco et al., 2015; Kanoun et al., 2011; Rocchi et al., 2015; Zanchettin and Rocco, 2017). Constraints are included in the QP in the form of linear matrix inequalities, while hierarchical tasks can be handled in a priority framework (Kanoun et al., 2011; Flacco et al., 2015; Escande et al., 2014) or in terms of a multi-objective cost function with different weights (according to the task priority) (Liu et al., 2016).

The use of quadratic programming ensures the solution to be within the robot constraints and to minimize the
deviation with respect to the desired tasks. However, as this approach is purely local, the solution only takes into account the current information of the system. The use of information about the future evolution of tasks, constraints, and kinematic variables could be exploited to improve the performance of the IK in global terms. For this reason, few recent works apply model predictive techniques to address this kind of problems (Schuetz et al., 2014; Tassa et al., 2015; Zube, 2015; Faroni et al., 2017b).

In particular, Faroni et al. (2017b) propose a fast predictive method to handle online redundancy resolution under kinematic constraints. This method models the robot joints as a series of multiple integrators. Then, the joints are coupled in a single linear state-space representation. The task is considered at velocity level at each point of the predictive horizon. At each cycle, the prediction of the joint configuration (calculated at the previous iteration) is used to compute the predicted Jacobians needed in the task expression. In this way, the task is linearized and the problem is converted into a QP. A continuous-time parametrization of inputs and outputs is applied to reduce the required computational time of the algorithm and this allows its online implementation.

As previously mentioned and as demonstrated in (Faroni et al., 2017b), the use of such strategy improves the purely local resolution of the IK by taking into account the future evolution of the system. However, the behavior of such technique in case of online task re-planning has not been investigated yet. This is of significant interest given that task re-planning is often exploited in recent developments in the field of collaborative robotics and robotic applications in unstructured environments. Re-planning strategies can be useful in several contexts. For example, the original task can be modified according to changes in the environment, in case of moving obstacles or when an unexpected event occurs in the proximity of the robot. However, the abrupt change of the planned task could be a potential source of instability behaviors, as the solution of predictive methods is based on the predicted task and the prediction of joint states (which can result to be incorrect in case of modification of the task). For these reasons, in this paper, we investigate the behavior and the performance of the predictive redundancy resolution proposed in (Faroni et al., 2017b) in case of different re-planning scenarios. For this purpose, the algorithm is suitably modified to cope with issues related to re-planning. In particular, the desired motion at the re-planning occurrence could result to be infeasible. This is the case, for example, of re-planned paths that present position and/or velocity discontinuities with respect to the current state of the robot. For this reason, the primary task is included in the online optimization problem as part of the quadratic cost function. This allows the task deformation when the desired motion is not realizable. Moreover, a CLIK scheme is implemented to recover from possible position errors. The evaluation of the method is organized as follows.

Firstly, the predictive method is compared against the local one in the case without re-planning to show the advantages of the technique. Secondly, the behavior in case of re-planning is investigated. Three re-planning scenarios are defined to simulate different situations. In the first case, the re-planned task has continuous velocity with respect to the current robot state. In the second one, the re-planned path has just continuous position, and in the third one the new desired path is discontinuous with respect to the current robot configuration, simulating an abrupt change in the planned motion. Moreover, different implementations are analyzed. Given the predictive nature of the algorithm, the re-planning can occur instantaneously or only after the end of the predictive horizon. In other words, in the second case, the re-planned task is visible to the controller as soon as it enters the predictive horizon.

The paper demonstrates the robustness of the method in all these different scenarios. Furthermore, a comparison against local methods is presented to highlight the improvements given by the predictive strategy in terms of tracking error and secondary task satisfaction.

2. BACKGROUND

The redundancy resolution method proposed in (Faroni et al., 2017b) is briefly recalled. Considering an \( n \)-degree-of-freedom manipulator, the joint-space kinematic model is derived by modeling each joint as a \( r \)-th order integrator in a single linear state-space representation. In this work, \( r = 2 \) will be used, so the outputs are given by the joint positions and velocities, while the control inputs are represented by the joint accelerations. The predictive equations are derived by considering the model in the continuous-time domain, as detailed in (Faroni et al., 2017a). The basic principle of the strategy consists in choosing a reduced number of predictive and control time instants along the respective horizons, in order to reduce the number of variables involved in the model and to speed up the execution of the resulting algorithm. In this work, the prediction and control time instants are chosen to be coincident and are denoted by the vector \( \tau \in \mathbb{R}^p \), where \( p \in \mathbb{N} \) is the number of time instants.

Therefore, the predictive equations at time \( t_0 \) can be written in matrix form as:

\[
\begin{align*}
q(t_0) &= L_0 \xi_0 + F_0 u(t_0) \quad (1) \\
\dot{q}(t_0) &= L_1 \xi_0 + F_1 u(t_0) \quad (2)
\end{align*}
\]

where \( q(t_0), q_\tau \in \mathbb{R}^{np} \) are respectively the joint position and velocity vectors predicted at time \( t_0 + \tau \), and \( u(t_0) \in \mathbb{R}^{np} \) is the input vector along the control horizon, \( \xi_0 \in \mathbb{R}^{2np} \) is the state vector at time \( t_0 \), \( \xi := [q_1, \ldots, q_{np}, \dot{q}_1, \ldots, \dot{q}_{np}]^T \), \( L_0, L_1 \in \mathbb{R}^{np \times 2n} \), and \( F_0, F_1 \in \mathbb{R}^{np \times np} \) are predictive matrices calculated as in (Faroni et al., 2017b).

Now, consider a \( m \)-dimensional primary task, with \( m < n \), which typically represents the desired pose of the robot in the Cartesian space. The satisfaction of the task along the predictive horizon is obtained by imposing that \( f(q(t)) = x(t) \) for all \( t \in \{t_0 + \tau\} \), where \( x \) is the desired Cartesian pose over time, and \( f: \mathbb{R}^n \rightarrow \mathbb{R}^m \) is the forward kinematics of the manipulator. Such task is considered at velocity level as:

\[
J(q(t)) \dot{q}(t) = \dot{x}(t) \quad \forall t \in \{t_0 + \tau\}. \quad (3)
\]

where \( J = df/dq \in \mathbb{R}^{m \times n} \) is the Jacobian matrix of the robot, and \( \dot{x} = dx/dt \) is the desired task velocity. This can be rewritten in matrix form as:

\[
J(q(t)) \dot{q}(t) = \dot{x}(t) \quad \forall t \in \{t_0 + \tau\}.
\]

where:

\[
\begin{align*}
\dot{J}(q(t)) := & \text{blkdiag} \left( J(q(t_0 + \tau_1)), \ldots, J(q(t_0 + \tau_p)) \right) \quad (5) \\
X(t) := & \left[ \dot{x}(t_0 + \tau_1)^T, \ldots, \dot{x}(t_0 + \tau_p)^T \right]^T. \quad (6)
\end{align*}
\]
The operator blkdiag(·) generates a block-diagonal matrix from the given matrices. Condition (4) is nonlinear in the control action $u_t$ for the given matrices. This condition is linearized by calculating the predicted joint configuration $\tilde{q}_t$ at each cycle by means of (1) and, then, using it to compute the predicted Jacobians at times $t_o + \tau$.

The analysis conducted in this paper deals with re-planning strategies. For this reason, infeasible desired motions and position errors may occur. In order to cope with this situations, changes in the original algorithm are introduced. Firstly, the linearized task (4) is rewritten in terms of a quadratic cost function that minimizes the deviation with respect to the desired task:

$$u_t = \arg\min \| \bar{J}(\tilde{q}_t) F_1 u_t + \bar{J}(\tilde{q}_t) L_1 \xi_0 - \bar{X}(t) \|^2$$ (7)

Secondly, a CLIK scheme is implemented to recover from position errors that could arise due to infeasible motions or to the re-planning itself (if the desired path changes discontinuously when re-planning occurs). The CLIK strategy consists in modifying the velocity reference by adding a term proportional to the position error $e_p$ at the current position. The position error is calculated as $e_p = x(t_0) - f(q(t_0))$. Details about the implementation of CLIK strategies with different Cartesian representations are well explained in (Caccavale et al., 1999).

Thus, (7) becomes:

$$u_t = \arg\min \| \bar{J}(\tilde{q}_t) F_1 u_t + \bar{J}(\tilde{q}_t) L_1 \xi_0 - (\bar{X}(t) + K_p e_p) \|^2$$ (8)

where $K_p \in \mathbb{R}^{m \times m}$ is a positive-definite matrix.

Kinematic constraints are now straightforwardly included in the formulation thanks to their linearity in the control action $u_t$. As a matter of fact, reminding the predictive equations (1) and (2), position, velocity, and acceleration limits can be imposed along the predictive horizon as:

$$q_{\min} \leq L_0 \xi_0 + F_0 u_t \leq q_{\max}$$

$$\dot{q}_{\min} \leq L_1 \xi_0 + F_1 u_t \leq \dot{q}_{\max}$$

$$\ddot{q}_{\min} \leq \dot{u}_t \leq \ddot{q}_{\max}$$ (9)

where $q_{\min}$, $q_{\max}$, $\dot{q}_{\min}$, $\dot{q}_{\max}$, $\ddot{q}_{\min}$, $\ddot{q}_{\max} \in \mathbb{R}^{n_p}$ are the minimum and maximum joint position, velocity, and acceleration limits respectively.

Finally, secondary tasks should be written in terms of a quadratic cost function, in order to include it in the optimization problem. For example, if the secondary objective task is the maximization of the joint range availability (Liegois, 1977; Atawneh et al., 2016), the online optimization problem can be expressed in quadratic form as:

$$\min (F_0 u_t + L_0 \xi_0 - \bar{q})^TW_{JA}(F_0 u_t + L_0 \xi_0 - \bar{q})$$ (10)

where $\bar{q} \in \mathbb{R}^{n_p}$ is given by the middle value of the joint range, and $W_{JA} \in \mathbb{R}^{n_p \times n_p}$ is a suitable positive-definite weighting matrix. Alternatively, if the secondary task is the minimization of the joint velocity norm, the quadratic problem results to be

$$\min (F_1 u_t + L_1 \xi_0)^TW_{VEL}(F_1 u_t + L_1 \xi_0)$$ (11)

where $W_{VEL} \in \mathbb{R}^{n_p \times n_p}$ is a suitable positive-definite weighting matrix.

Equations (8), (9), and (10) (or (11)) can now be clustered in a constrained QP that has to be updated and solved at each cycle. The higher priority of the primary task (8) with respect to the secondary one (10) (or (11)) can be handled in a hierarchical-QP framework (see (Kanoun et al., 2011; Escande et al., 2014) for details) or by using a weighted approach (as well exposed in (Liu et al., 2016)).

3. RE-PLANNING SCENARIOS

Consider the task function $x : \mathbb{R} \rightarrow \mathbb{R}^m$, $\gamma \rightarrow x(\gamma)$, where $\gamma$ is a curvilinear abscissa that parametrizes the task curve and is typically utilized to define the timing law of the trajectory. For the sake of simplicity and without loss of generality, assume $\gamma$ corresponds to the normalized longitudinal length along the curve. Time dependency is given by $\gamma : \mathbb{R} \rightarrow [0,1], t \rightarrow \gamma(t)$, assuming $\gamma \in C^1(\mathbb{R})$.

In case online re-planning occurs at time $t = t_{re-pl}$, the resulting task function can be expressed as:

$$x(\gamma) = \begin{cases} x_{orig}(\gamma(t)) & \text{if } t < t_{re-pl} \\ x_{re-pl}(\gamma(t)) & \text{if } t \geq t_{re-pl} \end{cases}$$ (12)

where $x_{orig}$ is the original desired task and $x_{re-pl}$ is the new task after the occurrence of re-planning. Functions $x_{orig}$ and $x_{re-pl}$ are reasonably assumed to be at least of class $C^1([0,1])$.

The smoothness of the resulting task $x$ affects the system response. For this reason, three re-planning scenarios are devised, depending on the smoothness of $x(\gamma)$:

1. $x \in C^1([0,1])$. In this case, discontinuity may occur at acceleration level, thus, the task can potentially be performed without generating tracking errors. This scenario is sketched in Figure 1a.

2. $x \in C^0([0,1])$. The task can present discontinuous velocity with respect to the current state of the robot. As the manipulator presents bounded accelerations, the velocity discontinuity is not feasible, therefore a position error will surely occur. The magnitude of such error can depend on the state of the robot when the re-planning occurs. This scenario is sketched in Figure 1b.

3. $x$ is discontinuous at $\gamma = \gamma(t_{re-pl})$. This means that, right after the re-planning occurs, this scenario intrinsically introduces a large position error, as the robot results to be far away from the new desired task position. In this phase, the robot motion will be mainly driven by the CLIK controller, until the position error is recovered. Note that, in a real application, such scenario is not likely to happen, as the high-level planner should be in charge of generating a continuous path (linking the discontinuous path if needed). However, this evaluation is important for testing the robustness of the algorithm in case a sudden and abrupt change of the task occurs. This scenario is sketched in Figure 1c.

Moreover, re-planning is handled by the predictive method by following two counterposed approaches:

- an instantaneous approach, in the sense that the predicted task seen by the predictive controller changes abruptly when the re-planning occurs (similarly to local methods).
- a “predictive” approach, that is, the predictive controller is able to foresee the re-planned task as soon as it enters the predictive horizon.
the following timing law
\[ \tau = \frac{6}{T_{\text{tot}}} t^5 - \frac{15}{T_{\text{tot}}} t^4 + \frac{10}{T_{\text{tot}}} t^3 \]  

(13)

where \( T_{\text{tot}} = 12 \) s is the execution time of the task. Two secondary tasks are tested. In the first case, the robot should maximize the joint range availability (i.e., \( (10) \), with \( W_{\text{JA}} = I \)). In the second case, it should minimize the norm of the velocity vector (i.e., \( (11) \), with \( W_{\text{VEL}} = I \)). The following parameters are evaluated: the maximum and mean position error along the task (\( e_{\text{max}} \) and \( e_{\text{mean}} \) respectively), and the secondary task satisfaction. In case of joint availability maximization, the task satisfaction is represented by the index
\[ i_{\text{JA}} = \frac{1}{T_{\text{tot}}} \int_0^{T_{\text{tot}}} \| \vec{q} - \vec{q}(t) \| \, dt, \]  

(14)

where \( \vec{q} \) is the vector given by the middle values of the joint ranges. In case of velocity minimization, the secondary task is evaluated by means of the index
\[ i_{\text{VEL}} = \frac{1}{T_{\text{tot}}} \int_0^{T_{\text{tot}}} \| \dot{\vec{q}}(t) \| \, dt. \]  

(15)

The smaller \( i_{\text{JA}} \) and \( i_{\text{VEL}} \), the better the secondary task is satisfied. Results are shown in Tables 1 and 2, which show that the predictive approach ensures better satisfaction of the secondary tasks, compared to the local approach. The error values for the joint-availability case are comparable. In the case of velocity minimization, however, the local approach gives a larger maximum error, which is avoided by the predictive method.

A second analysis is conducted to evaluate the behavior and the performance of the predictive method in case of re-planning. Referring to (12), the original desired primary task is given by the circle \( x_{\text{orig}} = [0.35, 0.35 + 0.125 \cos(4\pi\gamma), 0.4 + 0.125 \sin(4\pi\gamma)]^T \) m, and \( \gamma \) is calculated by using the timing law (13). The secondary task consists in the maximization of the joint range availability. The re-planning occurs at time \( t_{\text{re-pl}} = 6 \) s. From that moment on, depending on the re-planning scenario, the new desired task is:

- Scenario 1: \( x_{\text{re-pl}} = [0.35, 0.6 - 0.125 \cos(4\pi\gamma), 0.4 + 0.125 \sin(4\pi\gamma)]^T \) m.
- Scenario 2: \( x_{\text{re-pl}} = [0.35, 0.6 - 0.125 \cos(4\pi\gamma), 0.4 - 0.125 \sin(4\pi\gamma)]^T \) m (note that the motion direction becomes counterclockwise).
- Scenario 3: \( x_{\text{re-pl}} = [0.35, 0.725 - 0.125 \cos(4\pi\gamma), 0.4 + 0.125 \sin(4\pi\gamma)]^T \) m.

Note that the definition of the three scenarios is compliant with that given in Section 3. Moreover, as mentioned in Section 3, the re-planning is handled both in instantaneous and predictive manner.

Results are shown in Tables 3, 4, and 5. The evaluated parameters are the same as those in Table 1. As expected, note that the first scenario always shows a small error. This is clear also from Figure 2a, where the Cartesian tasks...
Fig. 2. Task executions. Dotted black: original path. Dotted red: re-planned path. Solid blue: task executed by the predictive method. Solid magenta: task executed by the local method (blue and magenta lines are sometimes indistinguishable due to overlapping).

Table 3. Scenario 1: Comparison of local and predictive methods.

|          | $\epsilon_{\text{max}}$ [m] | $\epsilon_{\text{mean}}$ [m] | $i_{\text{JA}}$ [rad] |
|----------|-------------------------------|------------------------------|------------------------|
| Local    | $2.47 \cdot 10^{-4}$         | $1.30 \cdot 10^{-4}$        | 1.91                   |
| Predictive (instant.) | $2.47 \cdot 10^{-4}$         | $1.31 \cdot 10^{-4}$        | 1.82                   |
| Predictive (predict.)  | $2.47 \cdot 10^{-4}$         | $1.31 \cdot 10^{-4}$        | 1.82                   |

Table 4. Scenario 2: Comparison of local and predictive methods.

|          | $\epsilon_{\text{max}}$ [m] | $\epsilon_{\text{mean}}$ [m] | $i_{\text{JA}}$ [rad] |
|----------|-------------------------------|------------------------------|------------------------|
| Local    | $1.60 \cdot 10^{-2}$         | $3.06 \cdot 10^{-4}$        | 2.28                   |
| Predictive (instant.) | $1.57 \cdot 10^{-2}$         | $2.98 \cdot 10^{-4}$        | 1.81                   |
| Predictive (predict.)  | $1.24 \cdot 10^{-2}$         | $2.45 \cdot 10^{-4}$        | 1.81                   |

Table 5. Scenario 3: Comparison of local and predictive methods.

|          | $\epsilon_{\text{max}}$ [m] | $\epsilon_{\text{mean}}$ [m] | $i_{\text{JA}}$ [rad] |
|----------|-------------------------------|------------------------------|------------------------|
| Local    | $1.25 \cdot 10^{-1}$         | $2.2 \cdot 10^{-3}$         | 2.55                   |
| Predictive (instant.) | $1.25 \cdot 10^{-1}$         | $2.3 \cdot 10^{-3}$         | 1.78                   |
| Predictive (predict.)  | $1.25 \cdot 10^{-1}$         | $2.3 \cdot 10^{-3}$         | 1.78                   |

obtained by the predictive and the local methods almost overlap the desired path. The second scenario presents a significant maximum deformation of the task (due to the infeasible velocity discontinuity), and the third one has a maximum error equal to the distance of the original and the re-planned task when the re-planning occurs. In almost all cases, the maximum and mean error values are comparable for the two methods, showing that the occurrence of re-planning does not generate unpredictable behaviors of the predictive method. Moreover, in case the re-planning is handled in the predictive way, it is also possible to reduce the position error that arises in the second scenario. This behavior is clarified in Figure 2b, which shows the Cartesian task performed by the robot with instantaneous and predictive re-planning. The zoomed window shows that the predictive approach converges to the new path with a smaller deformation of the task. This does not happen in the third scenario, for the position discontinuity is mainly recovered by the CLIK scheme, which has no predictive nature. In this case, the two methods converges to the new task in a similar way, as shown in Figure 2c. Finally, for all scenarios, the predictive redundancy handling ensures remarkable improvements of the secondary task satisfaction, similarly to the case without re-planning. The case of velocity norm minimization leads to analogous findings, however, numerical results are omitted for the sake of brevity.

5. CONCLUSIONS

This paper has presented an evaluation of the predictive redundancy resolution in case of different re-planning scenarios. First of all, the original method was slightly modified to cope with re-planning issues. In particular, the primary task was handled in the cost function of the QP. Secondary tasks can be handled in a priority framework. Then, an outer CLIK controller was implemented to recover from position errors. Simulations on a 7-degree-of-freedom KUKA LWR IV were performed to assess the performance of the method. The predictive method was compared to the local one without
re-planning. The predictive method shows a significant improvement of the secondary task satisfaction, as the predictive nature of the algorithm allows a better exploitation of the redundancy.

In case of re-planning, the predictive method still ensures better satisfaction of the secondary task, compared to the local method. As regards the tracking error, the predictive and the local methods give similar results. This is an important result to assess the robustness of the predictive method, as a sudden change of the predicted task does not give rise to unstable or undesired behaviors. The only difference in terms of tracking error arises when the re-planned task presents a velocity discontinuity. In this case, by handling the re-planning in a predictive way, the method is able to reduce the maximum deformation of the desired task.

Summarizing, the predictive method shows significant advantages over its local counterpart in all re-planning cases, confirming the benefits assessed without re-planning.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union’s Horizon2020 research and innovation program under grant agreement No. 780488 (PickPlace).

REFERENCES

Atawnih, A., Papageorgiou, D., and Doulgeri, Z. (2016). Kinematic control of redundant robots with guaranteed joint limit avoidance. *Robotics and Autonomous Systems*, 79, 122–131.

Caccavale, F., Siciliano, B., and Villani, L. (1999). The role of Euler parameters in robot control. *Asian Journal of Control*, 1, 25–34.

Chiachio, P., Chiaverini, S., Sciavicco, L., and Siciliano, B. (1991). Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy. *The International Journal of Robotics Research*, 10, 410–425.

Escande, A., Mansard, N., and Weber, P.B. (2014). Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33, 1006–1028.

Faroni, M., Beschi, M., Berenguel, M., and Visioli, A. (2017a). Fast MPC with staircase parametrization of the inputs: Continuous Input Blocking. In *Proceedings IEEE International Conference on Emerging Technologies and Factory Automation*, 1–8. Limassol (Cyprus).

Faroni, M., Beschi, M., Visioli, A., and Molinari Tosatti, L. (2016). A global approach to manipulability optimisation for a dual-arm manipulator. In *Proceedings IEEE International Conference on Emerging Technologies and Factory Automation*, 1–6. Berlin (Germany).

Faroni, M., Beschi, M., Visioli, A., and Molinari Tosatti, L. (2017b). A predictive approach to redundancy resolution for robot manipulators. In *Proceedings IFAC World Congress*, 8975–8980. Toulouse (France).

Flacco, F., De Luca, A., and Khatib, O. (2015). Control of redundant robots under hard joint constraints: Saturation in the null space. *IEEE Transaction on Robotics*, 31, 637–654.

Kanoun, O., Lamiraux, F., and Wieber, P.B. (2011). Kinematic control of redundant manipulators: generalizing the task priority framework to inequality tasks. *IEEE Transactions on Robotics*, 27, 785–792.

Lamperti, C., Zanchettin, A.M., and Rocco, P. (2015). A redundancy resolution method for an anthropomorphic dual-arm manipulator based on a musculoskeletal criterion. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1846–1851. Hamburg (Germany).

Lee, J. and Chang, P.H. (2015). Redundancy resolution for dual-arm robots inspired by human asymmetric bimanual action: Formulation and experiments. In *Proceedings IEEE International Conference on Robotics and Automation*, 6058–6065. Seattle (USA).

Liegeois, A. (1977). Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man and Cybernetics*, 7, 868–871.

Liu, M., Tan, Y., and Padois, V. (2016). Generalized hierarchical control. *Autonomous Robots*, 40, 17–31.

Maciejewski, A.A. and Klein, C.A. (1985). Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. *The International Journal of Robotics Research*, 4, 109–117.

Nakamura, Y. (1990). *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley, MA, USA.

Ott, C., Dietrich, A., and Albu-Schäffer, A. (2015). Prioritized multi-task compliance control of redundant manipulators. *Automatica*, 53, 416–423.

Rocchi, A., Hoffman, E.M., Caldwell, D.G., and Tsagarakis, N.G. (2015). Opensoft: a whole-body control library for the compliant humanoid robot Coman. In *Proceedings IEEE International Conference on Robotics and Automation*, 6248–6253. Seattle (USA).

Schuetz, C., Buschmann, T., Baur, J., Pfaff, J., and Ulbrich, H. (2014). Predictive online inverse kinematics for redundant manipulators. In *Proceedings IEEE International Conference on Robotics and Automation*, 5056–5061. Hong Kong (China).

Sciavicco, L. and Siciliano, B. (1988). A solution algorithm to the inverse kinematic problem for redundant manipulators. *IEEE Journal on Robotics and Automation*, 4, 403–410.

Siciliano, B. (1989). Kinematic control of redundant robot manipulators: A tutorial. *Journal of Intelligent and Robotic Systems*, 3, 201–212.

Siciliano, B. and Slotine, J.J.E. (1991). A general framework for managing multiple tasks in highly redundant robotic systems. In *Proceedings International Conference on Advanced Robotics*, 1211–1216. Pisa (Italy).

Tassa, Y., Mansard, N., and Todorov, E. (2015). Controlled-differential dynamic programming. In *Proceedings IEEE International Conference on Robotics and Automation*, 1168–1175. Seattle (USA).

Zanchettin, A.M. and Rocco, P. (2017). Motion planning for robotic manipulators using robust constrained control. *Control Engineering Practice*, 59, 127–136.

Zube, A. (2015). Cartesian nonlinear model predictive control of redundant manipulators considering obstacles. In *Proceedings IEEE International Conference on Industrial Technology*, 137–142. Seville (Spain).