2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Geometrical Interpretation and Detection of Multiple Task Conflicts using a Coordinate Invariant Index

Vincenzo Schettino[1], Mario D. Fiore[1], Claudia Pecorella[2], Fanny Ficuciello[2], Felix Allmendinger[1], Johannes Lachner[1,3], Stefano Stramigioli[3] and Bruno Siciliano[2]

***Abstract*— Modern robots act in dynamic and partially unknown environments where path replanning can be mandatory if changes in the environment are observed. Task-prioritized control strategies are well known and effective solutions to ensure local adaptation of robot behaviour. The highest priority in a stack of tasks is typically given to the management of correct robot operation or safe interaction with the environment such as obstacles or joint limits avoidance, that we can consider as constraints. If a constraint makes impossible achieving a certain task, such as tracking a Cartesian trajectory, a local control algorithm partially sacrifices the latter which is only accomplished to the best of the robot's ability to generate internal motions. In this control framework, problems may occur in some applications, like in the surgical domain, where it is not safe that some tasks are simply sacrificed without prior notice. The contribution of this work is to introduce a coordinate invariant index, that is used to provide a geometrical interpretation of task conflicts in a task-priority control framework and to develop a method for on-line detection of algorithmic singularities, with the goal of increasing safety and performances during robot operations.**

I. INTRODUCTION

Kinematic singularities are robot configurations where the Jacobian matrix is rank-deficient and cannot be inverted. This condition causes many control methods to fail and let the task assigned to the robot unfeasible. Hence, the study of such singular configurations is of significant importance for the application, control and design of robots. A common research objective has been to acquire tools/methods to detect and avoid such singularities during robot operations [1].

When multiple tasks are assigned to the robot, besides kinematic singularities, also singularities between tasks can occur, namely tasks conflicts better known as *algorithmic singularities*. Several solutions have been proposed in literature to assigns a hierarchical control structure and to manage multiple tasks, dealing with different aspects like: strict or soft priority hierarchy, static or dynamic priority hierarchy.

[1]are with KUKA Deutschland GmbH, Zugspitzstrae 140, 86165 Augsburg, DE. Vincenzo.Schettino, Felix.Allmendinger, Mario.Fiore, Johannes.Lachner @kuka.com

[2]are with PRISMA Lab, Department of Electrical Engineering and Information Technology, University of Naples Federico II, via Claudio 21, 80125 Naples, IT. claudia.pecorella, fanny.ficuciello, bruno.siciliano@unina.it.

[3]is with Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, The Netherlands S.Stramigioli@utwente.nl.

This research has been partially funded by the POR FESR 2014-2020 Italian National program within BARTOLO project (CUP B41C17000090007) and by the PNR 2015-2020 Italian National program within PROSCAN project (CUP E26C18000170005) and by the European Community, within the MURAB project (Grant 688188).

Fig. 1: Trocar use case with a KUKA LBR MED (Courtesy of KUKA Deutschland GmbH).

In this work, we consider the framework where multiple tasks are assigned, and handled by attributing to each of them a different level of priority using a Nullspace projection method. In this context, if a task with higher priority makes another task with lower priority unfeasible, the following situations can occur: (i) the task with lower priority can be partially accomplished, (ii) it could be completely neglected, (iii) the robot could get stuck because of the so called algorithmic singularity.

As a matter of fact, these reactions in some scenarios are not acceptable. If we consider medical applications, it is crucial for the patient healthiness that the surgeon can safely operate with the robot without abrupt interruptions. Let's consider the Trocar kinematics use case [2] (Fig. 1). In this operation context, the robot tool is inserted into the patient body, through a Trocar. The Trocar point is defined as a fix point in the world frame through which the shaft of the tool has to pass and that constitute a remote center of motion (RCM) for the manipulator. This kinematic constraint has to be obeyed while hand-guiding the robot. It is clear that the surgeon must be able to hand-guide the robot, while keeping away from all the kinematics constraints and workspace limitations (joint limits, singularities and so on), and avoiding any kind of unpredictable reactions from the robot. In such a context, possible task conflict should be promptly detected and an appropriate reaction triggered. Of course, many other use cases not only in the medical context can be found. Usually even in industrial context, in particular in human-robot collaborative applications, the problem of avoiding unpredictable robot reaction due to conflicts between multiple tasks and workspace limitations play an essential role for the design of the robot control law.

The problem of finding suitable indexes to measure these kind of singularity is still open. An interesting question arises: does it make sense trying to extend the kinematic singularity index to algorithmic singularity? And if it makes sense, how can this be realized? In this work, we try to answer to these questions and we propose an approach to detect conflicts between tasks, considering a coordinate invariant measure used for the analysis of kinematic singularities of an open-chain mechanical structure already introduced in [3]. Then we give a geometrical interpretation of the detected algorithmic singularity.

The paper is organized as follows: Section II gives a brief overview of the multiple task-priority algorithms that could be found in literature for the management of a set of tasks. Section III presents a list of indexes for the analysis of robot kinematic singularities (Section III-A), followed by considerations about manipulability ellipsoids in Section III-B. In Section III-C we present the proposed method for the detection and the geometrical interpretation of task conflicts. Illustrative examples are presented in Section IV.

II. Related Works

Motion generation for highly redundant robots, either manipulators or humanoid robots, have been intensively studied in the past decades. To assign a hierarchical control structure to manage multiple tasks [1] several solutions have been proposed in literature, i.e. techniques based on extended Jacobian [4], [5] and augmented Jacobian [6], [7]. Multiple ways to define the desired hierarchy can be found in literature and are mainly divided into two categories: strict and non-strict priority hierarchy. To the first one belong the strategies that project lower priority tasks in the null-space of the higher priority tasks, through the usage of null-space projectors ([8], [9], [10], [11]). With these strategies the highest priority task is exactly fulfilled, while the others are performed just if they don't interfere with the ones at higher priority. Recursive projection onto the Nullspace of the higher priority task Jacobian have also been proposed in [12] and [13], where different projection matrices are used, depending on the chosen control framework, i.e., velocity, acceleration or force based. Flacco *et al.* in [14] propose a reverse priority method which allows to execute at best all tasks, preserving the desired hierarchy. They process higher priority tasks at the end, and add joint motion contribution following the reverse order of priorities.

Non-strict priority approaches replace idempotent projections with non-idempotent matrix operators that realize approximate projections. Soft priorities introduce coupling between tasks. [15] and [16] propose to adapt nullspace projectors to achieve dynamic adaptation of task priorities. A weighted mixture of multiple tasks implements a soft prioritization by assigning a scalar priority to each task ([17], [18]). Both strict and soft priorities hierarchies have also been formulated as a quadratic program ([19], [20]).

Liu *et al.* in [21] proposes an approach to handle both strict and non-strict priorities of an arbitrary number of tasks, completely projecting a task into the null-space of a set of tasks and partially into the null-space of some other tasks. [22] extends the work proposed by Liu *et al.* to include a weighting matrix in the computation of the nullspace projection, adding dynamic-consistency to the stack-of-tasks. The proposed extension is advantageous also in the case of non-strict priorities because it reduces the inertia coupling between tasks.

The methods to define multiple tasks have then been extended to include also set-based tasks (e.g. distance from joint limits) to the hierarchy in [23], [24], [25], [26], [27], in addition to equality tasks. Strategies to learn time-dependent priorities are under investigations. They can be learnt employing policy search relying on a user-defined cost function [28], [29] or employing programming-by-demonstration techniques [30].

When multiple tasks are assigned to the robot, singularities between tasks can occur, in addition to kinematic ones. [8], [31], [25] have proposed different methods to detect them. In particular, in [32] sufficient conditions regarding the rank of the augmented Jacobian for the stability of the regulation problem are presented.

III. Methodology

In the following we will refer to the high priority task as the *constraint* and to the secondary task (lower priority task) as the *task* to be executed in the Cartesian space. A robot joint configuration, $\mathbf{q}$(t), with $\mathbf{q} \in \mathbb{R}^n$ is a *conflict configuration* if and only if:

$$\min_{\sigma_i} SVD[\mathbf{J}_T(\mathbf{q}(t)) \cdot \mathbf{N}_C(\mathbf{q}(t))] = 0, \tag{1}$$

where $\mathbf{J}_T \in \mathbb{R}^{r \times n}$ is the Jacobian of the task and $\mathbf{N}_C \in \mathbb{R}^{n \times n}$ is the nullspace projector of the constraint identified by the jacobian $\mathbf{J}_C$ [8]. SVD is the singular values decomposition and σ_i is the i-th singular value. It is worth noting that the product in (1) might present a singularity even though the two jacobians related to the task and the constraint, $\mathbf{J}_T$ and $\mathbf{J}_C$, might be non-singular at the specific configuration $\mathbf{q}(t)$. Methods to analyze kinematic and algorithmic singularities have been widely investigated through years. In the next subsection we will present some of them.

A. Indexes for Kinematic Singularities

The literature is rich and comprehensive regarding indexes and measures of robot kinematic singularities [1]. A general overview of the most common indexes utilized for the analysis of kinematic singularity is shown in Table I. References for further details are reported as well.

The eigenvalues (EIG) or singular values (SVD) (indexes of line 1, 2 and 3 of Tab. I) decomposition (for non-square matrix) are well-known methods for singularity analysis of a linear mapping [40]. In general, those methods are computationally expensive. Some low-cost methods for eigenvalues decomposition have been proposed during the years [41]. A natural extension, less expensive from a computational point of view, is the condition number (indexes of lines 4 and 5).

$\mathbf{M}(\mathbf{q})$ (reported in line 7) is the robot mass matrix, while $\Lambda(\mathbf{x})$ (in lines 3 and 8) is the robot inertia in the Cartesian Space. For sake of clarity, dependency from the robot joint configuration and from the end-effector position will be

TABLE I: List of indexes for the Kinematic Singularity

number	index	expression	reference
1	$\widetilde{\lambda}_m(\mathbf{J})$	$\min_{\lambda_i}[EIG(\mathbf{J}]$	[33]
2	$\widetilde{\sigma}_m(\mathbf{J})$	$\min_{\sigma_i}[SVD(\mathbf{J})]$	[34]
3	$\widetilde{\sigma}_m(\Lambda^{-1})$	$\min_{\sigma_i}[SVD(\Lambda^{-1})]$	[35]
4	$K_2(\mathbf{J})$	$\frac{\sigma_M}{\sigma_m}$	[36]
5	$K_F(\mathbf{J})$	$\frac{1}{n}\sqrt{tr(\mathbf{J}\mathbf{J}^T)}\sqrt{tr[(\mathbf{J}\mathbf{J}^T)^{-1}]}$	[37]
6	w	$\sqrt{det(\mathbf{J}\mathbf{J}^T)}$	[38]
7	w_d	$\sqrt{det[\mathbf{J}\mathbf{M}^{-1}(\mathbf{J}\mathbf{M}^{-1})^T]}$	[39]
8	w_m	$\sqrt{det(\Lambda^{-1})}$	-

omitted in the remainder of this paper for all the relative matrices.

A common property of all the mentioned indexes of Tab. I, is that their value is zero in a singular configuration and it increases as soon as the configuration of the robot moves out of the singularity. The way the value increases can be rather different from one index to another, depending on the robot kinematic structure and the physical entity the specific index can be related to. All the presented indexes normally deal with manipulability measures related to the kinematic structure of the robot. Indeed, they are typically used to analyse pure kinematic feasibility to arbitrarily generate end-effector velocity in a certain joint configuration. The concept of manipulability is always related to the one of manipulability ellipsoids.

What is not quite often remarked from other authors is that those ellipsoids are dependent from the joint coordinate choice [3]. A change of this parametrization will lead to a different ellipsoid, meaning that any desired ellipsoid can be generated given a suitable coordinate choice. Therefore, it is of crucial importance the choice of the kernel of the quadratic form which describes the ellipsoids equation, i.e., the matrix used for the manipulability analysis. This aspect is examinated more in detail in the next subsection.

B. Manipulability Ellipsoids

The derivation of manipulability ellipsoid typically starts by visualising the joint velocities $\dot{\mathbf{q}}$ as an hyper-sphere, given by the equation: $\dot{\mathbf{q}}^T\dot{\mathbf{q}} = 1$.

By (pseudo)-inverting the differential kinematic equation, an ellipsoid in the Cartesian Space is obtained:

$$\dot{\mathbf{x}}^T\mathbf{J}^{\#T}\mathbf{J}^{\#}\dot{\mathbf{x}} = 1, \tag{2}$$

where $\mathbf{J}^{\#}$ is a generalized inverse of the robot Jacobian.

As pointed out in [3], an aspect that is not often remarked in the context of manipulability analysis is that the matrix $\mathbf{J}^{\#T}\mathbf{J}^{\#}$ is in general not independent of joint coordinates. In fact, different joint parametrization would lead to a different sphere of joint velocities and thus, to a different ellipsoid in (2) for the same joint configuration. Furthermore, in case of redundant robot, the inverse of $\mathbf{J}$ does not exist and a pseudo-inversion is required. As a pseudo-inverse of $\mathbf{J}$ is not unique, arbitrarily high or poor manipulability can be obtained for the same joint configuration, depending on the selected pseudo-inverse. Thus, without a meaningful choice of joint parametrization, the manipulability ellipsoid embeds no physical meaning.

A coordinate-invariant approach consists in recalling that joint velocities $\dot{\mathbf{q}}$ are elements on a manifold, namely the Joint Configuration Space. To speak about length or norm of an element, a metric has to be defined [3]. An appropriate choice is the mass matrix $\mathbf{M}$, which leads to the redefinition of the sphere related to joint velocities as:

$$\dot{\mathbf{q}}^T\mathbf{M}\dot{\mathbf{q}} = 1, \tag{3}$$

which represents an ellipsoid in the space of joint velocities for a given joint configuration. It can be noticed that in this way equation (3) provides the same ellipsoid, no matter the parametrization or the units chosen for $\dot{\mathbf{q}}$ [3]. Note that the problem still remain on how to invert the differential kinematics equation in order to obtain a velocity ellipsoid in the Cartesian Space.

Following the reasoning from above, it is possible to define the ellipsoid of generalized joint forces τ, i.e. $\tau^T\mathbf{M}^{-1}\tau = 1$.

Note that $\mathbf{M}$ is positive definite and can always be inverted. If now we map the joints generalized forces to end-effector generalized forces, f, by using $\tau = \mathbf{J}^T\mathbf{f}$, we obtain the ellipsoid:

$$\mathbf{f}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)\mathbf{f} = 1. \tag{4}$$

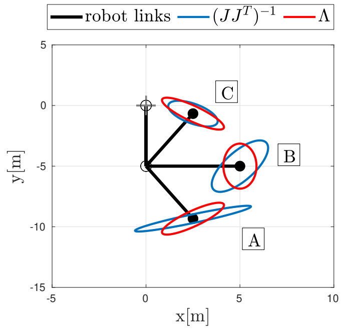
This time no inversion of the Jacobian is required. The kernel of the quadratic form in (4) is the inverse of $\Lambda = (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}$ already seen in Tab. I (indexes of lines 3 and 8). This matrix represents the end-effector inertia in the Cartesian Space and it is the induced metric in this space, obtained by using $\mathbf{M}$ as a metric in the Joint Configuration Space. Note that Λ is also independent of joint parametrization [3].

Being consistent with the metric definition adopted so far, we can define the velocity ellipsoid as:

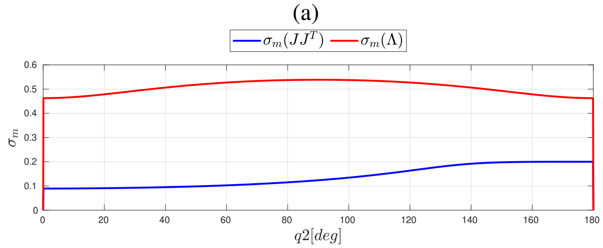
$$\dot{\mathbf{x}}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}\dot{\mathbf{x}} = 1, \tag{5}$$

which can also be obtained from (3) by adoperating the *dynamically consistent* inverse of the Jacobian [42]. For this reason we call the ellipsoid defined in (5) *dynamically consistent velocity ellipsoid*, the kernel of which is Λ itself.

The velocity and force ellipsoids in (5) and (4) incorporate a strong physical meaning: their principal axis describe the capability of the robot to produce end-effector velocities/forces in certain directions taking into account not only the kinematic structure of the robot, but also the dynamic constraints intrinsically expressed by the inertia matrix Λ; furthermore, they are independent of the joint parametrization, which comes as consequence of choosing a proper metrics in the Joint Configuration Space [3]. It can be recognized, that the core of the quadratic form of the velocity ellipsoid is constituted by the inverse of the matrix core of force ellipsoid. Therefore, the principal axis of the two ellipsoids coincide, while their respective dimensions are

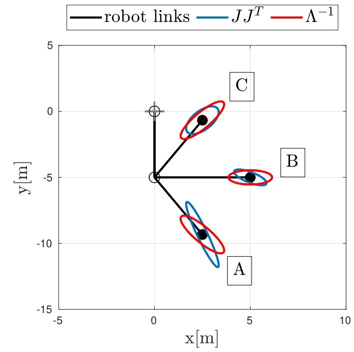


(a)

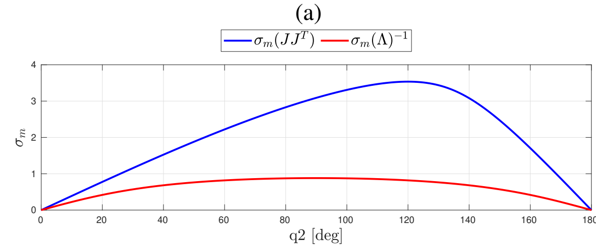


(b)

Fig. 2: Comparison between the classical velocity ellipsoid (blue) and the one obtained with Λ^{-1} (red) in (2a), for a 2-DOF planar robot in different configurations: $q_1 = -90°$ and $q_2 = 30°$ in A, $q_2 = 90°$ in B and $q_2 = 150°$ in C. Comparison between the minimum singular value for the classical velocity ellipsoid (blue) and the one obtained with Λ^{-1} (red) in (2b), for the same robot, with $q_1 = -90°$ and $q_2 \in [0°, 180°]$.



(a)



(b)

Fig. 3: Comparison between the classical force ellipsoid (blue) and the one obtained with Λ^{-1} (red) in (3a), for a 2-DOF planar robot in different configurations: $q_1 = -90°$ and $q_2 = 30°$ in A, $q_2 = 90°$ in B and $q_2 = 150°$ in C. Comparison between the minimum singular value for the classical force ellipsoid (blue) and the one obtained with Λ^{-1} (red) in (3b), for the same robot, with $q_1 = -90°$ and $q_2 \in [0°, 180°]$.

in inverse. Good velocity manipulability is then obtained in directions of poor force manipulability and viceversa, which is a known property according to force/velocity duality (Fig. 2a, Fig. 2b, Fig. 3a and Fig. 3b).

The analysis conducted above, remarks that only ellipsoids obtained by choosing a proper metric in the Joint Configuration Space reflects physical properties of the robot. Other ellipsoids could infer for the same configuration arbitrarily high or poor manipulability, depending on the joint coordinates and the units chosen to parametrize the Joint Configuration Space. Based on this idea, it is reasonable for the analysis of kinematic singularities to consider only the indexes based on Λ (indexes 3 and 8 from Tab. I)

C. Task-Conflict Singularities

The feasibility of tasks assigned to a robotic manipulator is strictly related to the invertibility of the associated Jacobian (or others) matrix that is required in the control loop [43]. We show that to investigate task-constraint singularities in a natural way it is possible to consider the same index of the kinematic singularity (Tab. I) but with a different "input" matrix carrying information about the tasks feasibility.

Without losing generality let consider the feasibility of two tasks performed simultaneously. These tasks can be feasible for the robot when performed separately, but become unfeasible when performed together despite the assigned level of priority for each task. In (1) a condition for a task-conflict to occur was given. The product $\mathbf{J}_{T|C} = \mathbf{J}_T \cdot \mathbf{N}_C$, $\mathbf{J}_{T|C} \in \mathbb{R}^{r \times n}$, represents the restriction of the mapping $\mathbf{J}_T$ to the nullspace of Jacobian matrix related to the constraint, i.e. to the task subspace that is consistent with the constraint. It can then be interpreted as a new task Jacobian (*constraint-consistent task Jacobian*), associated to a virtual manipulator that can only produce motion in the range of $\mathbf{J}_{T|C}$.

A task-conflict singularity for the real manipulator corresponds to a kinematic singularity for the virtual one. Thus, it seems reasonable to apply the same index identified for kinematic singularity to $\mathbf{J}_{T|C}$, in order to analyse task-constraint conflict situations:

$$\widetilde{\sigma}_m(\Lambda_{T|C}^{-1}) = \min_{\sigma_i}[SVD(\Lambda_{T|C}^{-1})] \tag{6}$$

$$w_m = \sqrt{det(\Lambda_{T|C}^{-1})}, \tag{7}$$

where $\Lambda_{T|C} = (\mathbf{J}_{T|C}\mathbf{M}^{-1}\mathbf{J}_{T|C}^T)^{-1} \in \mathbb{R}^{r \times r}$ is the inertia matrix in the task space that is consistent with the constraint. It must be pointed out that $\Lambda_{T|C}$ can be inverted just if there is no conflict between the task and the constraint, and thus if the virtual manipulator is not in a singular configuration.

For some simple cases, the virtual manipulator can be easily identified as a substructure of the real manipulator. Some practical examples are shown in the next section.

IV. Illustrative Examples

The singularity between different tasks assigned to a robot manipulator can be recognized as a pure kinematic singularity of a particular sub-structure of the manipulator

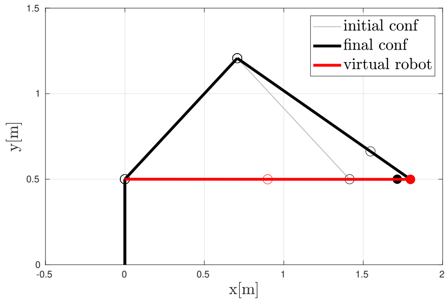


Fig. 4: Task-Constraint conflict geometrical example with a 3-DOF manipulator

itself. In this way, to analyze such a singularity, it is possible to extend the same indexes and measure which normally are used to detect pure kinematic ones. As shown before (in sect. III-C), we can use the same indexes with a different input matrix. The input matrix this time has to bring information about the tasks and constraints involved. This information restricts the indexes into the subspace of the task which is consistent with the assigned constraint defined by $\mathbf{J}_{T|C}$.

It is worth to show our proposed geometrical interpretation of *task-conflict singularities* with different kinematic structures. We first show the above mentioned concept with an example considering a 3-DoF manipulator (see the Appendix for further details). For the sake of simplicity, in this example let consider as a constraint to not move the joint 1, and as task to move the end-effector along a line through the positive direction of the x-axis, starting from the initial configuration ($\mathbf{q}_{init} = [45°, -90°, 45°]^T$). At some point both tasks become unfeasible because the robot will not be able to move the end-effector along a straight line taking into account the constraint, see Fig. 4. It is exactly in that moment which we can recognize that the matrix $\mathbf{J}_{T|C}$ loses full rank. In this specific configuration, such matrix describes a robot manipulator of 2-DOF which is in the outstretched kinematic singular configuration (see the red link in Fig. 4).

It is now interesting to present our method for a more complex kinematic structure such the one of a redundant manipulator. Let consider the following scenarios with a robot KUKA LBR iiwa 7kg:

- Example 1: the 3 dimensional task consists to move the end-effector along a straight line in the negative direction of the x-axis of the robot base frame. The constraint is to keep joint 2 fixed at its initial joint position (Fig. 5).
- Example 2: the 3 dimensional task consists to move the end-effector along a straight line in the positive direction of the x-axis of the robot base frame. The constraint is to keep joint 4 fixed at its initial joint position (Fig. 6).

In each of them the task and the constraint are both feasible for the robot, but when performed together the robot will reach a configuration in which they are in conflict. For such a scenario, it has been emphasized the novel geometrical interpretation of these singularities. The virtual manipulator, highlighted in black in Fig. 5 and 6, represents the virtual kinematic structure described by the Jacobian of the task

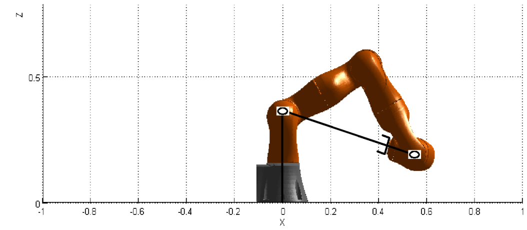


Fig. 5: Task-Constraint conflict geometrical interpretation with KUKA LBR iiwa (during *example 1*)

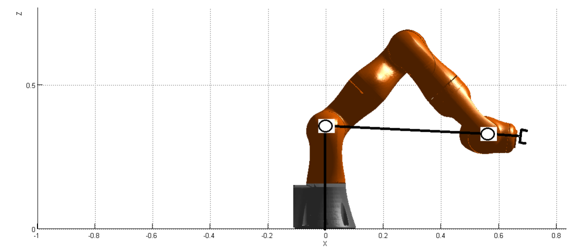


Fig. 6: Task-Constraint conflict geometrical interpretation with KUKA LBR iiwa (during *example 2*)

projected into the nullspace of the constraint, which is in a kinematic singularity.

In the end, for a given kinematic structure it is not trivial to determine whether a task-conflict singularity would arise. There are conditions related to tasks and their jacobians to prove the feasibility of multiple tasks, and these have been widely investigated by G. Antonelli through years [32].

V. Conclusions

The problem of on-line unfeasibility detection of a stack of tasks has been addressed. Thanks to a geometrical interpretation of the singularities between tasks assigned to the robot with different priorities, it has been possible to state that any valid measure for the kinematic singularities can be used to detect the conflict between tasks. The above mentioned measure has to be restricted into the space of the task which is consistent to the constraint. Finally, a coordinate invariant index to on-line measure such a kind of singularities has been proposed.

Further research direction would be to investigate an appropriate reactive control schemes (to be decided according to the specific application) to manage the conflict situations.

Appendix

In the following details about the experiment with the 3 DoFs planar manipulator of Fig. 4 are shown. As explained in Sec. IV the considered constraint is to keep fixed the first joint and move the end-effector along a straight line in the positive direction of x axis, starting from initial configuration $\mathbf{q}_{init} = [45°, -90°, 45°]$. The jacobian of a 3 DoFs manipulator is:

$$\mathbf{J} = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} - a_3 s_{123} & -a_2 s_{12} - a_3 s_{123} & -a_3 s_{123} \\ a_1 c_1 + a_2 c_{12} + a_3 c_{123} & a_2 c_{12} + a_3 c_{123} & a_3 c_{123} \end{bmatrix}$$

where a_i is the length of the i-th manipulator's link, c_{ij} and s_{ij} are respectively $\cos(q_i + q_j)$ and $\sin(q_i + q_j)$. For the considered constraint, the respective jacobian is $\mathbf{J}_c = [1 \quad 0 \quad 0]$, and the associated nullspace is $\mathbf{N}_C = \mathbf{I}_{3\times 3} - \mathbf{J}_C^{\#}\mathbf{J}_C$. The *constraint consistent task jacobian* is then:

$$\mathbf{J}_{T|C} = \mathbf{J}_T\mathbf{N}_C = \begin{bmatrix} 0 & -a_2s_{12} - a_3s_{123} & -a_3s_{123} \\ 0 & a_2c_{12} - a_3c_{123} & a_3c_{123} \end{bmatrix}$$

which is equivalent to the jacobian of a 2DoFs manipulator, the virtual one. For $q_3 = 0$ it is in a singular configuration since the columns become linearly dependents, and thus the virtual manipulator is in the outstretched configuration. Considering the aforementioned task, the obtained configuration is the one highlighted in red in Fig. 4.

References

[1] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, 2nd ed. Springer Publishing Company, Incorporated, 2016.

[2] P. From, "On the kinematics of robotic-assisted minimally invasive surgery," *Modeling, Identification and Control*, vol. 34, pp. 69–82, 2013.

[3] J. Lachner, V. Schettino, F. Allmendinger, M. D. Fiore, F. Ficuciello, B. Siciliano, and S. Stramigioli, "The influence of coordinates in robotic manipulability analysis," *Mechanism and Machine Theory*, vol. 146, p. 103722, 2020.

[4] P. Chang, "A closed-form solution for inverse kinematics of robot manipulators with redundancy," in *IEEE Journal on Robotics and Automation*, vol. 3, no. 5, October 1987, pp. 393–403.

[5] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," in *IEEE Int. Conf. on Robotics and Automation*, vol. 2, March 1985, pp. 722–728.

[6] L. Sciavicco and B. Siciliano, "A dynamic solution to the inverse kinematic problem for redundant manipulators," in *IEEE Int. Conf. on Robotics and Automation*, vol. 4, March 1987, pp. 1081–1087.

[7] O. Egeland, "Task-space tracking with redundant manipulators," *IEEE Journal on Robotics and Automation*, vol. 3, no. 5, pp. 471–475, October 1987.

[8] B. Siciliano and J. . E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Fifth Int. Conf. on Advanced Robotics 'Robots in Unstructured Environments*, June 1991, pp. 1211–1216 vol.2.

[9] A. Dietrich, C. Ott, and A. Albu-Schffer, "An overview of null space projections for redundant, torque-controlled robots," *Int J Robot Res*, vol. 34, no. 11, pp. 1385–1400, 2015.

[10] F. Ficuciello, L. Villani, and B. Siciliano, "Variable impedance control of redundant manipulators for intuitive humanrobot physical interaction," *IEEE Transactions on Robotics*, vol. 31, no. 4, 2015.

[11] A. Cirillo, F. Ficuciello, C. Natale, S. Pirozzi, and L. Villani, "A conformable force/tactile skin for physical humanrobot interaction," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 41–48, 2016.

[12] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int J Robot Res*, vol. 4, no. 3, pp. 109–117, 1985.

[13] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy," *Int J Robot Res*, vol. 10, pp. 410–425, 08 1991.

[14] F. Flacco and A. De Luca, "A reverse priority approach to multi-task control of redundant robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Sep. 2014, pp. 2421–2427.

[15] A. Dietrich, A. Albu-Schffer, and G. Hirzinger, "On continuous null space projections for torque-based, hierarchical, multi-objective manipulation," in *IEEE Int. Conf. on Robotics and Automation*, May 2012, pp. 2978–2985.

[16] N. Dehio, D. Kubus, and J. J. Steil, "Continuously shaping projections and operational space tasks," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Oct 2018, pp. 5995–6002.

[17] K. Bouyarmane and A. Kheddar, "Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Sep. 2011, pp. 4414–4419.

[18] J. Salini, V. Padois, and P. Bidaud, "Synthesis of complex humanoid whole-body behavior: A focus on sequencing and tasks transitions," in *IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 1283–1290.

[19] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *Int J Robot Res*, vol. 33, no. 7, pp. 1006–1028, 2014.

[20] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 473–491, Mar 2016.

[21] M. Liu, Y. Tan, and V. Padois, "Generalized hierarchical control," *Autonomous Robots*, vol. 40, no. 1, pp. 17–31, Jan 2016.

[22] N. Dehio and J. J. Steil, "Dynamically-consistent generalized hierarchical control," in *IEEE Int. Conf. on Robotics and Automation*, May 2019, pp. 1141–1147.

[23] E. Simetti and G. Casalino, "A novel practical technique to integrate inequality control objectives and task transitions in priority based control," *J Intell Robot Syst*, vol. 84, no. 1, pp. 877–902, Dec 2016.

[24] S. Moe, A. R. Teel, G. Antonelli, and K. Y. Pettersen, "Stability analysis for set-based control within the singularity-robust multiple task-priority inverse kinematics framework," in *IEEE Conf. on Decision and Control*, Dec 2015, pp. 171–178.

[25] S. Moe, G. Antonelli, A. Teel, K. Pettersen, and J. Schrimpf, "Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results," *Frontiers in Robotics and AI*, vol. 3, 04 2016.

[26] P. D. Lillo, S. Chiaverini, and G. Antonelli, "Handling robot constraints within a set-based multi-task priority inverse kinematics framework," in *IEEE Int. Conf. on Robotics and Automation*, 2019, pp. 7477–7483.

[27] J. Sverdrup-Thygeson, S. Moe, K. Y. Pettersen, and J. T. Gravdahl, "Kinematic singularity avoidance for robot manipulators using set-based manipulability tasks," in *IEEE Conf. on Control Technology and Applications*, Aug 2017, pp. 142–149.

[28] N. Dehio, R. F. Reinhart, and J. J. Steil, "Multiple task optimization with a mixture of controllers for motion generation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Sep. 2015, pp. 6416–6421.

[29] V. Modugno, G. Neumann, E. Rueckert, G. Oriolo, J. Peters, and S. Ivaldi, "Learning soft task priorities for control of redundant robots," in *IEEE Int. Conf. on Robotics and Automation*, 2016, pp. 221–226.

[30] J. Silvrio, Y. Huang, L. Rozo, S. Calinon, and D. G. Caldwell, "Probabilistic learning of torque controllers from kinematic and force constraints," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Oct 2018, pp. 1–8.

[31] O. Khatib, L. Sentis, J. Park, and J. Warren, "Whole-body dynamic behavior and control of human-like robots." *Int J Hum Robot*, vol. 10, pp. 29–43, 03 2004.

[32] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems," *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 985–994, Oct 2009.

[33] C. A. Klein and B. E. Blaho, "Dexterity measures for the design and control of kinematically redundant manipulators," *Int J Robot Res*, vol. 6, no. 2, pp. 72–83, 1987.

[34] R. P. Paul and C. N. Stevenson, "Kinematics of robot wrists," *Int J Robot Res*, vol. 2, no. 1, pp. 31–38, 1983.

[35] O. Khatib, "Inertial properties in robotic manipulation: An object-level framework," *Int J Robot Res*, vol. 14, no. 1, pp. 19–36, 1995.

[36] J. K. Salisbury and J. J. Craig, "Articulated hands: Force control and kinematic issues," *Int J Robot Res*, vol. 1, no. 1, pp. 4–17, 1982.

[37] W. A. Khan and J. Angeles, "The Kinetostatic Optimization of Robotic Manipulators: The Inverse and the Direct Problems," *J Mech Des*, vol. 128, no. 1, pp. 168–178, 08 2005.

[38] T. Yoshikawa, "Manipulability of robotic mechanisms," *Int J Robot Res*, vol. 4, no. 2, pp. 3–9, 1985.

[39] H. Asada, "A Geometrical Representation of Manipulator Dynamics and Its Application to Arm Design," *J Dyn Syst*, vol. 105, no. 3, pp. 131–142, 09 1983.

[40] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, "Differential kinematics and statics," in *Robotics: Modelling, Planning and Control*. London: Springer London, 2009, pp. 105–160.

[41] A. Bunse-Gerstner, "An analysis of the HR algorithm for computing the eigenvalues of a matrix," *Linear Algebra and its Applications*, vol. 35, pp. 155 – 173, 1981.

[42] H. Bruyninckx and O. Khatib, "Gauss' principle and the dynamics of redundant and constrained manipulators," in *IEEE Int. Conf. on Robotics and Automation*, vol. 3, April 2000, pp. 2563–2568 vol.3.

[43] J. Angeles, *Fundamentals of robotic mechanical systems*. Springer, 2002.