On Null Space-Based Inverse Kinematics Techniques for Fleet Management: Toward Time-Varying Task Activation

Anna Mannucci, Danilo Caporale, and Lucia Pallottino

***Abstract*—Multirobot fleets play an important role in industrial logistics, surveillance, and exploration applications. A wide literature exists on the topic, both resorting to reactive (i.e. collision avoidance) and to deliberative (i.e. motion planning) techniques. In this work, null space-based inverse kinematics (NSB-IK) methods are applied to the problem of fleet management. Several NSB-IK approaches existing in the literature are reviewed, and compared with a reverse priority approach, which originated in manipulator control, and is here applied for the first time to the considered problem. All NSB-IK approaches are here described in a unified formalism, which allows (i) to encode the property of each controller into a set of seven main key features, (ii) to study possible new control laws with an opportune choice of these parameters. Furthermore, motivated by the envisioned application scenario, we tackle the problem of task-switching activation. Leveraging on the iCAT TPC technique Simetti and Casalino, 2016, in this article, we propose a method to obtain continuity in the control in face of activation or deactivation of tasks, and subtasks by defining suitable damped projection operators. The proposed approaches are evaluated formally, and via simulations. Performances with respect to standard methods are compared considering a specific case study for multivehicles management.**

***Index Terms*—Formal methods in robotics and automation, motion control, multirobot systems, path planning for multiple mobile robots or agents.**

I. Introduction

IN MANY applications, multirobot fleets are required to safely and autonomously navigate in partially known environments, while achieving some individual or shared objectives such as maintaining a specific formation, moving to a desired location, and avoiding obstacles, all at the same time. Typical scenarios arise, for instance, in intralogistics, service, exploration, search and rescue, and surveillance robotics. A possible way to manage complex objectives is to divide them into several subproblems (tasks); in doing so, the problem for the control designer consists in merging such tasks to obtain a satisfactory collective behavior. These approaches are appealing for online path planning of shared objectives: tasks are planned directly in the task space and mapped back to the system configuration by defining proper functions to relate these two spaces. This reduces the complexity required by decoupling task allocation (i.e., defining a proper goal for each agent at each time) and motion planning (i.e., finding a feasible, possibly optimal path to the goal state while ensuring constraint satisfaction). The resulting architecture allows the control of heterogeneous fleets: each basic unit can accomplish different tasks according to its characteristics, but it is supported by a unified modular functional control architecture. Agents can be treated as individual or cooperative ones by properly defining their active set of tasks, which is usually context dependent.

Task-based control strategies have been proved to be suitable tools to obtain feasible, locally optimal trajectories both in centralized and decentralized control architectures [2]–[12]. The main difference within these approaches consists in how behaviors are merged together. For this purpose, some techniques originally developed for manipulators, e.g., the null space-based inverse kinematics (NSB-IK), have been applied to fleets in [5]–[13] to explore redundancy (i.e., accomplish the greatest possible number of tasks) while managing the possible conflicts between prioritized tasks. Differently from other strategies, such as the *motor schema control* [3] or the *layered control system* [2], in fact, NSB-IK controllers ensure the correct task hierarchy by construction while, before being executed, each task is projected in the null space of its higher priority ones. As a consequence, in case of competitive (linearly dependent) tasks, the one with an higher priority will prevail on the conflicting parts.

In this article, we investigate the efficacy of different NSB-IK methods, such as the standard approach (SA) based on augmented Jacobian [14], the singularity robust (SR) [15], and the reverse priority (RP) algorithm [16], to map sets of prioritized tasks, originated for managing multivehicle fleets, into continuous robots' velocity profiles. While the RP algorithm was proven to achieve better performance in terms of task

Manuscript received May 12, 2020; revised July 23, 2020; accepted August 3, 2020. Date of publication September 10, 2020; date of current version February 4, 2021. This work was supported in part by the European Union's Horizon 2020 Research, and Innovation Program under Agreement 732737 (ILIAD), and in part by the Italian Ministry of Education, University, and Research (MIUR) in the framework of the CrossLab Project (Departments of Excellence). This article was recommended for publication by Associate Editor N. Gans and Editor P. Robuffo Giordano upon evaluation of the reviewers' comments. *(Corresponding author: Anna Mannucci.)*

Anna Mannucci is with the Research Center "E. Piaggio", University of Pisa, 56126 Pisa, Italy, with the Dipartimento di Ingegneria dell'Informazione, University of Pisa, 56126 Pisa, Italy, and also with the Multi-Robot Planning, and Control Lab, Center for Applied Autonomous Sensor Systems (AASS), Örebro University, 70281 Örebro, Sweden (e-mail: anna.mannucci@oru.se).

Danilo Caporale and Lucia Pallottino are with the Research Center "E. Piaggio", University of Pisa, 56126 Pisa, Italy, and also with the Dipartimento di Ingegneria dell'Informazione, University of Pisa, 56126 Pisa, Italy (e-mail: d.caporale@centropiaggio.unipi.it; lucia.pallottino@unipi.it).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2020.3018642

1552-3098 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Orebro University. Downloaded on September 28,2021 at 14:05:40 UTC from IEEE Xplore. Restrictions apply.

TABLE I
LIST OF THE MAIN SYMBOLS

Symbol	*Description*
$\{r_1, \ldots, r_N\}$	Robots in the fleet.
$\boldsymbol{q} = \{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N\}$	State of the system, with $\boldsymbol{q}_i \in \mathbb{R}^{n_i}$ being the n_i-DoF generalized coordinates of robot r_i.
$\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_l\}$	Set of tasks with decreasing priority (the lower the priority index, the higher the priority of the task).
$\boldsymbol{x}_i = \boldsymbol{f}_i(\boldsymbol{q})$	i-th task with task function $\boldsymbol{f}_i : \Omega \to \mathbb{R}^{m_i}$, $\Omega \in \mathbb{R}^{n_i}$.
$\boldsymbol{J}_i = \frac{\partial \boldsymbol{f}_i(q)}{\partial \boldsymbol{q}}$	Jacobian of the i-th task.
$\dot{\boldsymbol{x}}_i = \boldsymbol{J}_i \dot{\boldsymbol{q}}$	Task rate, with $\dot{\boldsymbol{q}}$ being the control vector.
x_M (x_m)	Upper (lower) bound of an inequality task $x \leq x_M$ $(x \geq x_m)$.
T_c	Sampling time.
$\boldsymbol{K}_i$	Positive definite matrix of gains for the i-th task.
$\mathcal{S}_i$	Task manifold at the i-th iteration.
$(\boldsymbol{x}_{d,i}, \dot{\boldsymbol{x}}_{d,i})$	Task and task rate reference values of the bilateral task $\boldsymbol{x}_i$.
$\boldsymbol{J}_{A,i}^T = (\boldsymbol{J}_1^T, \ldots, \boldsymbol{J}_i^T)$	Augmented Jacobian (transposed) at the i-th iteration.
$\boldsymbol{J}_{RA,i}^T = (\boldsymbol{J}_i^T, \ldots, \boldsymbol{J}_l^T)$	Reversed Augmented Jacobian (transposed) at the i-th iteration.
$\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, $\boldsymbol{\Sigma} = \text{diag}(\sigma_i)$	Singular Value Decomposition of a matrix.
$\square_{\{a:b,c:d\}}$	Matrix selector operator (extracting rows with indices from a to b and columns from c to d).
$\square^\sharp = \{\square^\#, \square^{\#,d}, \square^{\#,\boldsymbol{A},\boldsymbol{Q}}\}$	Pseudoinverse operators: $\square^\#$ Moore-Penrose, $\square^{\#,d}$ damped [16, 21] with damping parameters $(\epsilon, \lambda_{\max})$, $\square^{\#,\boldsymbol{A},\boldsymbol{Q}}$ regularized as defined in [1] (Equation 5).
$\boldsymbol{P}_*$, $\boldsymbol{\Psi}_i$, $\boldsymbol{\Phi}_i$	Projectors in a proper null space of tasks.
k_f	Forgetting factor [18, 22].
$\boldsymbol{A}_i = \text{diag}(a_{i,1}, \ldots, a_{i,m_i})$	Activation matrix of $\boldsymbol{x}_i$, with smooth sigmoidal functions $a_{i,j} \in [0, 1]$.
$\check{\square} = \{\square, \bar{\square}\}$	Rows-selection operators: $\square$ selects all the rows; $\bar{\square}$ selects only the rows s.t. $a_{i,j} > 0$.
$L = \{+, -\}$	Orders for merging tasks: $\{-\}$ from higher to lower priority (i.e., in Eq. (4) $i^- = i - 1, 0_L = 0, l_L = l$); $\{+\}$ from lower to higher priority (i.e., in Eq. (4) $i^- = i + 1, 0_L = l + 1, l_L = 1$).
$\boldsymbol{Y}_i$, $\boldsymbol{W}_i$	Weighting matrices.
$\mathcal{C}_* = (L, \check{\square}, \{\boldsymbol{Y}_i\}, \{\boldsymbol{W}_i\}, \{\square^\sharp\}, \{\boldsymbol{\Psi}_i\}, \{\boldsymbol{\Phi}_i\})$	Generalized notation for different NSB-IK control laws; $\{\boldsymbol{\Phi}_i\}$ is omitted whenever $\boldsymbol{\Phi}_i = \boldsymbol{\Psi}_i$ for each i.
$\boldsymbol{\Pi}_*$	Permutation matrix.
$\boldsymbol{e}_i = \boldsymbol{x}_{d,i} - \boldsymbol{x}_i$, $\dot{\boldsymbol{x}}_i - \boldsymbol{J}_i \dot{\boldsymbol{q}}$, ζ_i	i-th task error, error rate and performance index.

priority satisfaction with robotic manipulators [16], to the best of authors' knowledge, the approach has never been used for fleet management. Motivated by the application scenario, we tackle the problem of time-varying task activation, which may lead to discontinuities in controls using standard NSB-IK methods [1], [17]. In particular, a novel task priority technique named iCAT TPC (inequality Control objectives, Activation and Transition Task Priority Control) was proposed in [9], [11], extending the standard approach to cope with time-varying activation of multidimensional tasks. Conversely, this problem was only partially addressed in the RP and the SR methods.[1] Hereby, we investigate how to fit the core concepts of the iCAT TPC approach into the other NSB-IK families. Moreover, we aim to formally define new control laws and the related properties.

In summary, the novel contributions of this article are threefold:

1) We propose a unified formalism for the NSB-IK approaches, which allows to map performances into a set key features.
2) We extend the iCAT TPC [1] to the SR and the RP frameworks to handle time-varying activation of multidimensional tasks. To the best of the authors' knowledge, none of the existing SR- or RP-based approaches is currently achieving this objective while preserving good tracking, correctness of the task hierarchy, and continuity of controls.[1]
3) We investigate the effectiveness of different NSB-IK high-level controllers for managing fleets of heterogeneous robots subject to possibly coupling prioritized tasks.

We formally prove the properties and the conditions under which the new control laws ensure the optimal execution of tasks. For this purpose, the theory on pseudoinverse of partitioned matrices has been extended to understand the effect of introducing weighting matrices in null spaces.

This article is organized as follows. The problem is formally stated in Section II. Section III recalls the necessary background on null space-based approaches. Then, the iCAT TPC control is presented in Section IV and extended for the SR and the RP approaches. Moreover, Section V formalizes some useful tasks for controlling fleets. Simulations are reported in Section VI, where all the NSB techniques are compared. Finally, Section VII concludes this article. Also, some theorems for pseudoinverses of partitioned matrices and iterative formulations of projectors are formally derived in Appendix (Section VIII).

II. PROBLEM FORMULATION

Notation. Scalar values are denoted with lowercase normal font characters (e.g., α). Bold lowercase (e.g., $\boldsymbol{v}$) and bold uppercase symbols (e.g., $\boldsymbol{M}$) are used for vectors and matrices, respectively. The identity matrix is referred as $\boldsymbol{I}$. For reader convenience, Table I lists the main symbols used in this article.

The goal of this work is to design a centralized high-level control architecture to coordinate a fleet of N heterogeneous mobile robots $\{r_1, \ldots, r_N\}$, which are required to accomplish different activities simultaneously, such as avoiding an obstacle, tracking a reference trajectory, and maintaining a desired formation. Hence, the objective is to determine robots reference controls such that

[1] Specifically, only single-row tasks with time-varying activation have been integrated in the RP algorithm in [18] to model unilateral constraints while preserving continuity. Also, inequalities control objectives have been integrated in the SR approach using binary activation functions in [7] or via task scheduling in [19], [20], disregarding continuity.

Authorized licensed use limited to: Orebro University. Downloaded on September 28,2021 at 14:05:40 UTC from IEEE Xplore. Restrictions apply.

1) the hierarchical structure is ensured (i.e., in case of conflict between two tasks, the one with a higher priority will prevail);
2) for each task, the best possible execution will be achieved according to the given hierarchy (i.e., the portion of lower priority tasks that complies with the higher priority tasks will be preserved).

For this purpose, let $\boldsymbol{q}_i \in \mathbb{R}^{n_i}$ be the generalized coordinates of a generic n_i-DoF robot r_i obeying to nonlinear control affine dynamics $\dot{\boldsymbol{q}}_i = \boldsymbol{G}_i(\boldsymbol{q}_i)\boldsymbol{\nu}_i$, with $\boldsymbol{\nu}_i \in \mathbb{R}^{p_i}$, $p_i \leq n_i$ being the command velocities. Let $\boldsymbol{q} = (\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N)^T \in \mathbb{R}^n$ be the state of the system, with $n = \sum_{i=1}^{N} n_i$. For simplicity of discussion and without loss of generality, we consider the robots fully actuated, i.e., each $\boldsymbol{G}_i(\boldsymbol{q}_i)$ is full row rank $\forall \boldsymbol{q}_i$. Therefore, the vector $\dot{\boldsymbol{q}}$ will be used as the control vector from now on. Kinematic constraints and robot dynamics are not taken into account and low-level controllers are assumed to be able to opportunely track the given reference velocities.

An m-dimensional task is a control objective $\boldsymbol{f}(\boldsymbol{q}) = \boldsymbol{x}_d$ (i.e., bilateral task) or $\boldsymbol{f}(\boldsymbol{q}) \leq \boldsymbol{x}_M$ (i.e., unilateral task), where $\boldsymbol{f} : \Omega \to \mathbb{R}^m, \Omega \subset \mathbb{R}^n$ is the task function which allows to compute the level of accomplishment of the control objective and $\boldsymbol{x}_d$ (or $\boldsymbol{x}_M$) is a desired (maximal) value of the task variables $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q})$. Examples of useful tasks for controlling mobile robots can be found in [7], [9], [12], and [23] such as safety (e.g., ensuring a minimal distance from obstacles), connectivity (e.g., constraining the maximal reciprocal distance between agents), motion control (e.g., moving to a desired location, tracking a reference trajectory), and fleet formation (e.g., maintaining a rigid formation).

Tasks are ordered according to a priority list; tasks with the same priority are collected into multidimensional tasks.

Two tasks are *linearly independent* (l.i.) if there is no reciprocal interference between them (i.e., it possible to find a vector of commands which reduces both the task errors simultaneously).

The system is *redundant* with respect to a given m-dimensional task *A* if it is possible to define a task *B* such that *A* and *B* are linearly independent.

The problem of finding reference velocities, given a set of prioritized tasks, can be formalized as a hierarchical constrained optimization problem. This formulation is equivalent to the hierarchical constrained inverse kinematics (IK) problem for the cartesian control of manipulators, where the commands are joint velocities and the control objectives are usually related to the pose of the end effector, collision avoidance, and joint limits. Consequently, priority-based algorithms originally developed for manipulators are a suitable tool which can be applied to our purpose, as it will be shown next.

III. NSB-IK Framework

Null space-based algorithms are a class of hierarchical, prioritized IK solvers which exploit the kinematic redundancy of the system for the fulfillment of a number of constraints, while guaranteeing the correct task hierarchy with a low computational complexity (see [14]–[16], [24], and the references therein for details). The key point of these methods is the use of projection matrices to filter out the contribution of each task which does not comply with the given task hierarchy. Hereby, a generalized formulation is presented to summarize the state-of-the-art of the NSB methods while highlighting the main differences in terms of performances and robustness to singularities. Also, for the sake of clarity, necessary definitions and notation are reported.

A. Mathematical Background

Assuming task functions to be differentiable, the NSB methods solve the inverse mapping between tasks and commands at a differential level, obtaining a locally linear map between the vectors $\dot{\boldsymbol{x}}$ and $\dot{\boldsymbol{q}}$. Let $\boldsymbol{J}(\boldsymbol{q}) = \dfrac{\partial \boldsymbol{f}(\boldsymbol{q})}{\partial \boldsymbol{q}} \in \mathbb{R}^{m \times n}$ be the task Jacobian such that

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{1}$$

Hereafter, the dependence on $\boldsymbol{q}$ is omitted for notational convenience.

The vector $\dot{\boldsymbol{x}}$ can be used for the purpose of tracking a desired reference trajectory $(\boldsymbol{x}_d, \dot{\boldsymbol{x}}_d)$ by setting

$$\dot{\boldsymbol{x}} = \dot{\boldsymbol{x}}_d + \boldsymbol{K}\left(\boldsymbol{x}_d - \boldsymbol{x}\right) \tag{2}$$

where $\boldsymbol{K}$ is a positive definite matrix of gains.

From now on, the vectors $\dot{\boldsymbol{x}}$ and $\dot{\boldsymbol{x}} - \boldsymbol{J}\dot{\boldsymbol{q}}$ are referred to as *task rate* and *task error rate*, respectively. Let

$$\mathcal{S} = \left\{ \arg\min_{\dot{\boldsymbol{q}} \in \mathbb{R}^n} \|\dot{\boldsymbol{x}} - \boldsymbol{J}\dot{\boldsymbol{q}}\|^2 \right\}$$

be the *task manifold* containing the subspace of $\mathbb{R}^n$ on which the task error rate is minimized. Also, let $\dot{\boldsymbol{q}}_{\mathcal{N}} \in \mathbb{R}^n$ be a generic velocity. The vector $\dot{\boldsymbol{q}} \in \mathcal{S}$ and closest to $\dot{\boldsymbol{q}}_{\mathcal{N}}$ can be computed by solving the optimization problem

$$\arg\min_{\dot{\boldsymbol{q}} \in \mathcal{S}} \|\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_{\mathcal{N}}\|^2$$

and is given by

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\#}\dot{\boldsymbol{x}} + \boldsymbol{P}_{\mathcal{N}}\dot{\boldsymbol{q}}_{\mathcal{N}} = \dot{\boldsymbol{q}}_{\mathcal{N}} + \boldsymbol{J}^{\#}\left(\dot{\boldsymbol{x}} - \boldsymbol{J}\dot{\boldsymbol{q}}_{\mathcal{N}}\right) \tag{3}$$

where $\boldsymbol{P}_{\mathcal{N}} = \boldsymbol{I} - \boldsymbol{J}^{\#}\boldsymbol{J} \in \mathbb{R}^{n \times n}$ is the projector in the Jacobian's null space, $\square^{\#}$ is the Moore–Penrose pseudoinverse operator.[2]

Notably, if the task is *feasible* (i.e., $\dot{\boldsymbol{x}} \in \text{span}(\boldsymbol{J})$), then (3) and (2) ensure $\boldsymbol{x}$ converges exponentially to $\boldsymbol{x}_d$ with rate $\boldsymbol{K}$[3]. Conversely, if $\dot{\boldsymbol{x}} \notin \text{span}(\boldsymbol{J})$, then $\|\boldsymbol{x}_d - \boldsymbol{x}\| \geq 0$.

Moreover, close to a singularity (i.e., $\boldsymbol{q}$ such that $\boldsymbol{J}$ is not full row rank—frr from now on), the matrix $\boldsymbol{J}^{\#}$ will become ill-conditioned and the use of a regularized pseudoinverse $\boldsymbol{J}^{\#,d}$ is required to prevent high values of $\dot{\boldsymbol{q}}$. The task manifold is modified as follows:

$$\mathcal{S} = \left\{ \arg\min_{\dot{\boldsymbol{q}} \in \mathbb{R}^n} \|\dot{\boldsymbol{x}} - \boldsymbol{J}\dot{\boldsymbol{q}}\|^2 + \|\dot{\boldsymbol{q}}\|^2_{\boldsymbol{R}} \right\}$$

[2]Any generalized pseudoinverse operator $\square^{\dagger}$ ensures $\boldsymbol{J}\boldsymbol{P}_{\mathcal{N}} = \boldsymbol{0}$ since $\boldsymbol{J}\boldsymbol{J}^{\dagger}\boldsymbol{J} = \boldsymbol{J}$. However, $\dot{\boldsymbol{q}} \in \arg\min_{\dot{\boldsymbol{q}} \in \mathbb{R}^n} \|\dot{\boldsymbol{x}} - \boldsymbol{J}\dot{\boldsymbol{q}}\|^2$ is ensured only if $\square^{\dagger} = \square^{\#}$ in (3). See [25] for details.

[3]Note that (2) and (3) ensure a null tracking error for any $(\boldsymbol{x}_d, \dot{\boldsymbol{x}}_d) \in \mathbb{R}^{2\tilde{}m}$ and for any temporal evolution of $\boldsymbol{q}$ iff $\boldsymbol{J}$ full row rank for each $\boldsymbol{q}$. However, the task error may be null for some $(\boldsymbol{x}_d, \dot{\boldsymbol{x}}_d)$ and $\boldsymbol{q}(t)$ even if $\text{span}(\boldsymbol{J}) \subset \mathbb{R}^m$.

Authorized licensed use limited to: Orebro University. Downloaded on September 28,2021 at 14:05:40 UTC from IEEE Xplore. Restrictions apply.

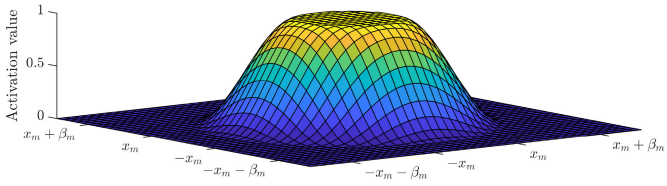


Fig. 1. Example of activation function $a_{i,j}$ for an inequality constraint $\|\boldsymbol{x}\| \geq x_m$, $\boldsymbol{x} \in \mathbb{R}^2$, with $\beta_m \geq 0$ being the related slack variable.

where $\boldsymbol{R} \geq 0$ is a regularization matrix which allows to weight the trade-off between performances and control effort. As in [15], [16], we consider the singular value-oriented regularization (SVO) with a variable damping factor [21], that is here briefly reported for reader convenience.

Let $\boldsymbol{J} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ be the singular value decomposition of $\boldsymbol{J}$, where $\boldsymbol{U} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ are unitary matrices with columns $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$, respectively, and $\boldsymbol{\Sigma}$ is a block diagonal $m \times n$ matrix of rank ρ, composed by a leading $\rho \times \rho$ diagonal matrix, containing the singular values $\sigma_i \geq 0$ $(i = 1, \ldots, \rho)$ of $\boldsymbol{J}$ in decreasing order ($\sigma_i \leq \sigma_j$ for $i > j$), and a $m - \rho$ zero matrix. Letting $\boldsymbol{R} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$, where $\boldsymbol{\Lambda}_{ii} = \lambda_i^2$ and $\boldsymbol{\Lambda}_{ij} = 0$ for $i \neq j$, the damped pseudoinverse is computed as

$$\boldsymbol{J}^{\#,d} = \sum_{i=1}^{\rho} \left(\frac{\sigma_i}{\sigma_i^2 + \lambda_i^2} \boldsymbol{v}_i \boldsymbol{u}_i^T \right).$$

Specifically, as in [15], [16], each λ_i is computed as

$$\lambda_i^2 = \begin{cases} \left(1 - \left(\frac{\sigma_m}{\epsilon}\right)^2\right) \lambda_{\max}^2 & \text{if } \sigma_i < \epsilon \\ 0 & \text{otherwise} \end{cases}$$

where σ_m is the smallest singular value and the parameters $\epsilon > 0$ and $\lambda_{\max}$ are used, respectively, to define the range of the damping action and the largest damping factor used near singularities. Accordingly, only the directions of the task that require high speed values, corresponding to small σ_i, are penalized. Note that when the damping is used, the projector $\boldsymbol{J}^{\#,d}$ is deformed, and the correct formulation of $\boldsymbol{P}_{\mathcal{N}}$ is given by $\boldsymbol{P}_{\mathcal{N}} = \boldsymbol{I} - \sum_{j=1}^{\rho} \boldsymbol{v}_j \boldsymbol{v}_j^T$.

B. Merging Different Tasks: A Unified Formalism

Let $\boldsymbol{x} = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_l^T)^T$ be an ordered list of multidimensional tasks, where $i < j$ when $\boldsymbol{x}_i$ has a higher priority than $\boldsymbol{x}_j$, $\dot{\boldsymbol{x}}_i = \boldsymbol{J}_i \dot{\boldsymbol{q}} \in \mathbb{R}^{m_i}$ and $\sum_{i=1}^{l} m_i \leq n$. Also, let $\boldsymbol{A}_i = \text{diag}(a_{i,1}, \ldots, a_{i,m_i})$ be the activation matrix of $\boldsymbol{x}_i$, where each $a_{i,j} \in [0, 1]$ is a smooth sigmoidal function used as weight of the related control objective in the optimization problem (Fig. 1). Each subtask $x_{i,j}$ is *active* when $a_{i,j} = 1$, *inactive* if $a_{i,j} = 0$, or *in transition* when $0 < a_{i,j} < 1$. The matrices $\{\boldsymbol{A}_i\}$ allow to integrate inequality and time-varying constraints in the NSB framework in a suboptimal but less computational-demanding way [1], [17], [18] when compared to state-of-the-art QP solvers (e.g., active sets based). In particular, inequality constraints will be transformed in equality ones (by introducing slack variables) and will be smoothly activated/deactivated when necessary.

Different NSB-IK methods have been proposed in the literature [1], [14]–[18] and are here defined by a unified notation with the tuple $\mathcal{C}_* = (L, \check{\square}, \{\boldsymbol{Y}_i\}, \{\boldsymbol{W}_i\}, \{\square^\sharp\}, \{\boldsymbol{\Psi}_i\}, \{\boldsymbol{\Phi}_i\})$ encoding the key features of each approach, where $L \in \{+, -\}$ is the order ($+$ ascendant, $-$ descendent) of going through the list of tasks (where, according to L, i^- is the predecessor of i, 0_L refers to initial values, and 1_L, l_L are the first and the last elements of the task list, respectively), $\check{\square}$ is a rows selector operator, $\boldsymbol{Y}_i$ and $\boldsymbol{W}_i$ are weighting matrices, $\square^\sharp$ is a pseudoinverse operator, and $\boldsymbol{\Psi}_i$ and $\boldsymbol{\Phi}_i$ are projection matrices. Consequently, the general iterative step can be written as[4]

$$\dot{\boldsymbol{q}}_i = \dot{\boldsymbol{q}}_{i^-} + \boldsymbol{\Psi}_i \left(\check{\boldsymbol{Y}}_i \check{\boldsymbol{J}}_i \boldsymbol{\Phi}_i\right)^\sharp \check{\boldsymbol{W}}_i \left(\dot{\check{\boldsymbol{x}}}_i - \check{\boldsymbol{J}}_i \dot{\boldsymbol{q}}_{i^-}\right)$$
$$\text{for } i \in \{1_L, \ldots, l_L\} \text{ s.t. } \check{\boldsymbol{W}}_i \check{\boldsymbol{J}}_i \neq \boldsymbol{0} \tag{4}$$

with $\boldsymbol{\Psi}_{0_L} = \boldsymbol{I}$, $\boldsymbol{\Phi}_{0_L} = \boldsymbol{I}$. Also, according to [22], at each discrete time k, $\dot{\boldsymbol{q}}_{0_L} = k_f \dot{\boldsymbol{q}}_{l_L}(\mathrm{k} - 1)$, with $k_f \in [0, 1]$ acting as a tunable forgetting factor. We denote as $\mathcal{S}_i(\mathrm{k})$ the search space at the ith step and $\mathcal{S}(\mathrm{k})$ the manifold of the solution at time k. Selected $\mathcal{C}_*$ relevant for this article are reported in Table II; for convenience, a simplified tuple $\mathcal{C}_* = (L, \check{\square}, \{\boldsymbol{Y}_i\}, \{\boldsymbol{W}_i\}, \{\square^\sharp\}, \{\boldsymbol{\Psi}_i\})$ is used whenever $\boldsymbol{\Phi}_i = \boldsymbol{\Psi}_i$ for all i. In particular, i) the tuple $(L, \{\boldsymbol{\Psi}_i\}, \{\boldsymbol{\Phi}_i\})$ selects between the three families of NSB-IK controllers, resulting in different performances in terms of optimality, correctness of the hierarchy, and robustness with respect to both kinematic and algorithmic singularities (see definitions in Table II); ii) the tuple $(\check{\square}, \{\boldsymbol{Y}_i\}, \{\boldsymbol{W}_i\}, \{\square^\sharp\})$ allows to explore different solutions to handle time-varying tasks, where key metrics are orthogonality of projectors (correctness), stability, and smoothness of the vector $\dot{\boldsymbol{q}}$.

One main concern in the introduction of task activation is that it may exist $i \in \{1, \ldots, l\}$ and a discrete time k s.t. $\mathcal{S}_i(\mathrm{k})$ changes discontinuously with respect to $\mathcal{S}_i(\mathrm{k} - 1)$. Therefore, projectors $(\check{\boldsymbol{Y}}_i \check{\boldsymbol{J}}_i \boldsymbol{\Phi}_i)^\sharp$, $\boldsymbol{\Psi}_i$, $\boldsymbol{\Phi}_i$ and controls may be discontinuous (see Fig. 2 for an example). Specifically, with reference to the notation given in Table II, single-row time-varying tasks can be smoothly activated/deactivated by the use of $\mathcal{C}_* = (L, \bar{\square}, \{\boldsymbol{I}\}, \{\boldsymbol{A}_i\}, \{\square^\#\}, \{\boldsymbol{\Psi}_i\}, \{\boldsymbol{\Phi}_i\})$ (e.g., [17], [18]), but extensions to the multidimensional case are not straightforward [1], [17].

Remark 1: Extensions $\mathcal{C}_* = (L, \bar{\square}, \{\boldsymbol{I}\}, \{\boldsymbol{A}_i\}, \{\square^\#\}, \{\boldsymbol{\Psi}_i\})$ of the methods in Table II with smooth activation functions and continuous references ensure the continuity of $\dot{\boldsymbol{q}}(\mathrm{k})$ *by construction* only if subtasks/tasks are perfectly decoupled [30] (*orthogonal* in [27]), i.e., only if:

1) $\boldsymbol{J}_i$ is f.r.r. $\wedge$ $\boldsymbol{J}_{i,j}\boldsymbol{J}_{i,k}^T = 0, \forall i, j \neq k$. **[SA, SR [15], RP]**
2) $\boldsymbol{J}_i \boldsymbol{J}_j^T = \boldsymbol{0}$ $\forall i, j \neq i$. **[SA, RP]**

Proof: See Appendix VIII-B. In short, (a) ensures the continuity of $\boldsymbol{J}_i^\#$ [30], while (b) prevents jumps of the optimal solutions due to changes of $\mathcal{S}(\mathrm{k})$ [17], [18]. Notably, the l.i. of tasks and subtasks is not sufficient to prevent discontinuities for all $\{(\boldsymbol{x}_{d,i}, \dot{\boldsymbol{x}}_{d,i}), \boldsymbol{A}_i\}$. ■

Smoother solutions can be obtained by choosing $\square^\sharp = \square^{\#,d}$ and high damping values [15], [31], [32]. However, the higher

[4] If $\boldsymbol{\Phi}_i = \boldsymbol{\Psi}_i$ for all i, then the term $\boldsymbol{\Psi}_i(\check{\boldsymbol{Y}}_i \check{\boldsymbol{J}}_i \boldsymbol{\Phi}_i)^\sharp$ of (4) is sometimes written as $(\check{\boldsymbol{Y}}_i \check{\boldsymbol{J}}_i \boldsymbol{\Psi}_i)^\sharp$. Note that the equivalence holds only if $\square^\sharp$ is a generalized pseudoinverse and $\boldsymbol{\Psi}_i$ is idempotent and Hermitian [26].

Authorized licensed use limited to: Orebro University. Downloaded on September 28,2021 at 14:05:40 UTC from IEEE Xplore. Restrictions apply.

TABLE II
NULL SPACE BASED APPROACHES - A GENERALIZED FORMULATION

NON TIME-VARYING APPROACHES (1)		
Standard Approach (SA) [14]	$\mathcal{C}_{\text{SA}} = (\{-\}, \square, \{\boldsymbol{I}\}, \{\boldsymbol{I}\}, \{\square^{\#}\}, \{\boldsymbol{P}_{A,i-1}\})$, where $\check{\boldsymbol{J}}_{A,0} = \check{\boldsymbol{J}}_i$, $\check{\boldsymbol{J}}_{A,i} = \begin{pmatrix} \check{\boldsymbol{J}}_{A,i-1}, \\ \check{\boldsymbol{J}}_i \end{pmatrix}$, $\boldsymbol{P}_{A,0} = \boldsymbol{I}$, $\boldsymbol{P}_{A,i} = \boldsymbol{I} - \check{\boldsymbol{J}}^{\#}_{A,i}\check{\boldsymbol{J}}_{A,i} = = \boldsymbol{P}_{A,i-1} - (\check{\boldsymbol{J}}_i\boldsymbol{P}_{A,i-1})^{\#}(\check{\boldsymbol{J}}_i\boldsymbol{P}_{A,i-1})$.	$\mathcal{S}_{i-1} = \{\dot{\boldsymbol{q}}_{i-1} + \boldsymbol{P}_{A,i-1}\boldsymbol{\nu}_{i-1}, \boldsymbol{\nu}_{i-1} \in \mathbb{R}^n\}$, $\mathcal{S}_0 = \mathbb{R}^n$, $\mathcal{S}_i = \{\arg\min_{\dot{\boldsymbol{q}} \in \mathcal{S}_{i-1}} \|\dot{\boldsymbol{x}}_i - \boldsymbol{J}_i\dot{\boldsymbol{q}}\|^2\}$. • Optimal only with f.r.r. and linearly independent tasks. • In case of singularities, the hierarchy is corrupted. • Extended to handle multi-dimensional unilateral tasks in [17]. • Extended to handle multi-dimensional time-varying tasks in [1] (iCAT TPC Approach).
Singularity Robust Approach (SR) [15][5]	$\mathcal{C}_{\text{SR}} = (\{+\}, \square, \{\boldsymbol{I}\}, \{\boldsymbol{I}\}, \{\square^{\#}\}, \{\boldsymbol{I}\})$.	$\mathcal{S}_i = \{\arg\min_{\dot{\boldsymbol{q}} \in \mathbb{R}^n} \|\dot{\boldsymbol{x}}_i - \boldsymbol{J}_i\dot{\boldsymbol{q}}\|^2\}$ is projected in the null space of the tasks $j < i$. • Optimal only under strict conditions [27]. • Algorithmic singularities are avoided by design. • Damping parameters can be specifically tuned for each task. • Unilateral tasks are handled through scheduling in [7]. • Extension to handle multi-dimensional time-varying tasks is one of the contribution of this paper (see Section IV-A).
Reverse Priority Approach (RP) [16] (2)	$\mathcal{C}_{\text{RP}} = (\{+\}, \square, \{\boldsymbol{I}\}, \{\boldsymbol{I}\}, \{\square^{\#}\}, \{\tilde{\boldsymbol{P}}_{i+1}\})$, where(3) $\tilde{\boldsymbol{P}}_{i+1}$ is the null space of those elements of the $l - i - 1$ lower priority tasks that are linearly independent from the i-th. Since $(\check{\boldsymbol{J}}_i\tilde{\boldsymbol{P}}_{i+1})^{\#} = \check{\boldsymbol{T}}_i(\check{\boldsymbol{J}}_i\check{\boldsymbol{T}}_i)^{\#}$ [16], practically, we should compute the matrix $\check{\boldsymbol{T}}_i = (\check{\boldsymbol{J}}^{\#}_{RA,i})_{\{:,1:\tilde{m}_1\}} \in \mathbb{R}^{n \times \tilde{m}_i}$, where $\check{\boldsymbol{J}}_{RA,i}$ is the Reverse Augmented Jacobian, recursively computed as $\check{\boldsymbol{J}}_{RA,l} = \check{\boldsymbol{J}}_i$, $\check{\boldsymbol{J}}^T_{RA,i} = (\check{\boldsymbol{J}}^T_i, \check{\boldsymbol{J}}^T_{RA,i+1})$.	$\mathcal{S}_{i+1} = \{\dot{\boldsymbol{q}}_{i+1} + \tilde{\boldsymbol{P}}_{i+1}\boldsymbol{\nu}_{i+1}, \boldsymbol{\nu}_{i+1} \in \mathbb{R}^n\}$, $\mathcal{S}_{l+1} = \mathbb{R}^n$, $\mathcal{S}_i = \{\arg\min_{\dot{\boldsymbol{q}} \in \mathcal{S}_{i+1}} \|\dot{\boldsymbol{x}}_i - \boldsymbol{J}_i\dot{\boldsymbol{q}}\|^2\}$. • Optimal with f.r.r. and linearly independent tasks. • The hierarchy is never corrupted in case of singularities. In particular, even if $\square^{\#,d}$ is required to compute $\check{\boldsymbol{T}}_i$ when $\check{\boldsymbol{J}}_{RA,i}$ is not f.r.r., the resulting task deformation is negligible [16]. • Extended to handle single-row unilateral tasks in [18]. • Extension to handle multi-dimensional time-varying tasks is one of the contribution of this paper (see Section IV-B).

Terminology

a. *Kinematic singularity:* $\boldsymbol{J}_i$ is not full row rank (f.r.r.).

b. *Algorithmic singularity:* $\exists i, j \neq i \in \{1, \ldots, l\}$ s.t. $(\boldsymbol{J}_i^T, \boldsymbol{J}_j^T)^T$ is not f.r.r. (tasks are not linearly independent).

Summary of issues

a. If $L = \{-\}$, then the method is optimal only with f.r.r. linearly independent tasks (projectors may be deformed due to the damping action, so high-priority tasks may not be executed at their best due to linearly dependent tasks with a lower priority).

b. When $\boldsymbol{\Phi}_i = \boldsymbol{I}$ for all i, the solution is searched in $\mathbb{R}^n$ and then projected in the null space of the higher priority tasks. Consequently, all the task errors will converge to zero only if the conditions in [27] are verified.

Computational complexity All of $\mathcal{C}_{\text{SA}}$, $\mathcal{C}_{\text{SR}}$, and $\mathcal{C}_{\text{RP}}$ (s.t. $\square^{\sharp} = \square^{\#}$) has computational complexity $O(mn^2)$, with $\boldsymbol{x} \in \mathbb{R}^m$, $\boldsymbol{q} \in \mathbb{R}^n$. Proofs are trivial and are omitted for brevity. In particular, the computational complexity of each control law $\mathcal{C}_*$ mostly depends on the complexity of computing pseudoinverses of matrices (and hence by the definition of $\square^{\sharp}$—see also [1], [17] for further details).

(1) Original formulation of each control law. Use $\square^{\#,d}$ instead of $\square^{\#}$ to handle singularities.

(2) The mathematical support is based on the Greville's formula for partitioned matrices [25], [28] (see also Appendix VIII-A and [16] for details.

(3) Differently from $\check{\square}$ and $\tilde{\square}$ that are operators, $\tilde{\square}$ is just a superscript, and is here maintained to be consistent with the notation in [16], [18].

the damping factor, the lower the accuracy in tracking references [30]. This trade-off between performance and smoothness of controls is actually limiting the use of NSB controllers (which ensure to execute tasks correctly according to the hierarchy) to time-varying 1-D tasks [18], [30]. An exception is given for the SA in [11], [17], where generic m-dimensional tasks have been successfully integrated in the framework while maintaining both optimality and continuity. Therefore, leveraging on the approach of [11] (namely, the iCAT TPC), we aim to overcome this limitation also in the SR and the RP methods, as shown in the following sections.

[5]A coupled yet still suboptimal formulation of the SR approach with $\boldsymbol{\Phi}_i = \boldsymbol{I}$, $\boldsymbol{\Psi}_i = \boldsymbol{I} - \boldsymbol{J}^{\#}_{A,i-1}\boldsymbol{J}_{A,i-1}$ is given in [24], with the stability analysis in [27] and the extension for set-based scalar inequality constraints in [19], [20]. To remark differences between coupled/decoupled methods, in this article, we opt for the decoupled formulation as also in [7], [16], [27], [29].

IV. iCAT TPC

To prevent discontinuities in the vector $\dot{\boldsymbol{q}}$ when the manifold $\mathcal{S}(\mathrm{k})$ is changing, two conditions need to be fulfilled [18].

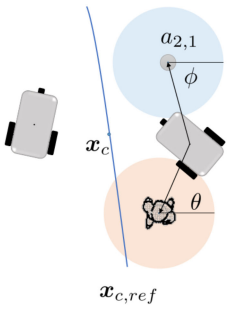
1) Smooth changes of the manifold $\mathcal{S}(\mathrm{k})$ are required, i.e., the projectors $\boldsymbol{\Psi}_i$, $\boldsymbol{\Phi}_i$ and $(\check{\boldsymbol{Y}}_i\check{\boldsymbol{J}}_i\boldsymbol{\Phi}_i)^{\sharp}$ should be continuous with activations.
2) The solution should move with continuity inside $\mathcal{S}(\mathrm{k})$, i.e., jumps of the optimal solution should be avoided.

To address both 1) and 2) and to extend the SA approach for with time-varying tasks activation, the solution proposed in [11] exploits the regularized operator of [1], defined as

$$\boldsymbol{X}^{\#,\boldsymbol{A},\boldsymbol{Q}} \triangleq \left(\boldsymbol{X}^T\boldsymbol{A}\boldsymbol{X} + \boldsymbol{V}^T\boldsymbol{\Lambda}\boldsymbol{V} + \gamma\boldsymbol{C}^T\boldsymbol{C}\right)^{\#}\boldsymbol{X}^T\boldsymbol{A}\boldsymbol{A} \quad (5)$$

where $\boldsymbol{A}$ is the diagonal activation matrix, $\boldsymbol{Q}$ is a projection matrix, $\boldsymbol{C} = \boldsymbol{I} - \boldsymbol{Q}$, $\boldsymbol{V}$ is the right orthonormal matrix of the SVD

Authorized licensed use limited to: Orebro University. Downloaded on September 28,2021 at 14:05:40 UTC from IEEE Xplore. Restrictions apply.



$$\boldsymbol{J}_1 = \begin{pmatrix}\cos\theta & \sin\theta & 0 & 0\end{pmatrix}$$

$$\boldsymbol{J}_{2,1} = \begin{pmatrix}\cos\phi & \sin\phi & 0 & 0\end{pmatrix}$$

$$\boldsymbol{J}_{2,2} = \begin{pmatrix}\frac{1}{2} & 0 & \frac{1}{2} & 0\\ 0 & \frac{1}{2} & 0 & \frac{1}{2}\end{pmatrix}$$

$$\boldsymbol{J}_2 = \begin{cases}\begin{pmatrix}\boldsymbol{J}_{2,1}\\ \boldsymbol{J}_{2,2}\end{pmatrix} & \text{if } a_{2,1} > 0\\ \boldsymbol{J}_{2,2} & \text{otherwise}\end{cases}$$

Fig. 2. Discontinuity of $(\boldsymbol{\Psi}_i, \boldsymbol{\Phi}_i)$: Let $\mathcal{C}_{\text{RP}} = (\{+\}, \bar{\square}, \{\boldsymbol{I}\}, \{\boldsymbol{A}_i\}, \{\square^{\#}\}, \{\tilde{\boldsymbol{P}}_{i+1}\})$, with $\tilde{\boldsymbol{P}}_{i+1}$ defined as shown in Table II. $\tilde{\boldsymbol{P}}_{i+1}$ does not preserve the continuity if $a_1 = 1, a_{2,2} = 1$ and $a_{2,1}$ changes from 0 to $\epsilon > 0$ (provable by computation). As we will see in Section V, $\boldsymbol{J}_1$ and $\boldsymbol{J}_{2,1}$ formalize a task of collision avoidance, while $\boldsymbol{J}_{2,2}$ requires to maintain the centroid of the fleet on a reference trajectory $\boldsymbol{x}_{c,ref}$.

decomposition of $\boldsymbol{X}^T\boldsymbol{A}\boldsymbol{X} + \gamma\boldsymbol{C}^T\boldsymbol{C}$, $\boldsymbol{\Lambda}$ is the SVO damping matrix described in Section III, and γ is a weighting factor.

Specifically, (5) results from solving the following optimization problem:

$$\arg\min_{\dot{\boldsymbol{q}}\in\mathbb{R}^n}\{\|\boldsymbol{A}\left(\dot{\boldsymbol{x}} - \boldsymbol{J}\dot{\boldsymbol{q}}\right)\|^2 + \underbrace{\|\boldsymbol{J}\dot{\boldsymbol{q}}\|^2_{\boldsymbol{A}(\boldsymbol{I}-\boldsymbol{A})}}_{\text{task oriented regularization}} + \underbrace{\left\|\boldsymbol{V}^T\dot{\boldsymbol{q}}\right\|^2}_{\text{SVO}} + \underbrace{\gamma\left\|\boldsymbol{C}\dot{\boldsymbol{q}}\right\|^2}_{\text{minim. controls}}\}$$

which allows to cope with discontinuities and chattering phenomena caused by insertion and deletion of subtasks *at the same priority level* [**C1**]. Then, to prevent discontinuities which may arise between tasks *at different priority levels* **[C1+C2]**, a bi-level optimization problem is posed. Considering $L = \{-\}$ and $\boldsymbol{\Psi}_i = \boldsymbol{P}_{A,i-1}$, the *auxiliary problem*

$$\begin{aligned}&\arg\min_{\boldsymbol{\nu}_i\in\mathbb{R}^n}\{\|\boldsymbol{A}_i\left[\dot{\boldsymbol{x}}_i - \boldsymbol{J}_i\left(\dot{\boldsymbol{q}}_{i^-} + \boldsymbol{\Psi}_i\boldsymbol{\nu}_i\right)\right]\|^2\\ &+ \|\boldsymbol{J}_i\boldsymbol{\Psi}_i\boldsymbol{\nu}_i\|^2_{\boldsymbol{A}_i(\boldsymbol{I}-\boldsymbol{A}_i)}\\ &+ \left\|\boldsymbol{V}_i^T\boldsymbol{\nu}_i\right\|^2_{\boldsymbol{\Lambda}_i} + \gamma\|(\boldsymbol{I} - \boldsymbol{\Psi}_i)\,\boldsymbol{\nu}_i\|^2\}\end{aligned} \tag{6}$$

seeks for the lowest possible task error rate which also minimizes the use of control directions of the higher priority tasks with transient components. $\boldsymbol{V}_i$ is the right orthonormal matrix of the SVD decomposition of

$$\boldsymbol{\Psi}_i^T\boldsymbol{J}_i^T\boldsymbol{A}_i\boldsymbol{J}_i\boldsymbol{\Psi}_i + \gamma\left(\boldsymbol{I} - \boldsymbol{\Psi}_i\right)^T\left(\boldsymbol{I} - \boldsymbol{\Psi}_i\right)$$

and $\boldsymbol{\Lambda}_i$ is the related SVO damping matrix. Therefore, the task reference rate is modified to ensure the best possible execution of the current task while accounting that the null space of the higher priority tasks is only partially exploitable.

As a result of the bi-level optimization, (4) is formalized as follows:

$$\begin{aligned}\mathcal{C}_{\text{iCAT TPC}} &= (\{-\}, \square, \{\boldsymbol{I}\}, \{\boldsymbol{W}_i\}, \{\square^{\#,\boldsymbol{A}_i,\boldsymbol{I}}\}, \{\boldsymbol{P}_{A,i-1}\})\\ \boldsymbol{W}_i &= \boldsymbol{J}_i\boldsymbol{P}_{A,i-1}\left(\boldsymbol{J}_i\boldsymbol{P}_{A,i-1}\right)^{\#,\boldsymbol{A}_i,\boldsymbol{P}_{A,i-1}}\\ \boldsymbol{P}_{A,i} &= \boldsymbol{P}_{A,i-1}\left(\boldsymbol{I} - \left(\boldsymbol{J}_i\boldsymbol{P}_{A,i-1}\right)^{\#,\boldsymbol{A}_i,\boldsymbol{I}}\boldsymbol{J}_i\boldsymbol{P}_{A,i-1}\right)\\ k_f &= 0,\ \boldsymbol{P}_{A,0} = \boldsymbol{I}.\end{aligned}$$

To address **C2**, the algorithm also requires the introduction of an additional task $\boldsymbol{J}_{l+1} = \boldsymbol{I}$ to saturate the residual degrees of freedom to minimize $\|\dot{\boldsymbol{q}}\|$.

As opposed to other techniques such as [17], (5) ensures a polynomial complexity in the number of tasks [1]. However, since $L = \{-\}$, the method still suffers from both kinematic and algorithmic singularities, so the task hierarchy may be corrupted due to the damping action. This issue may be solved with reverse methods. For this purpose, an extension of the iCAT approach is formalized in the following to properly handle time-varying multidimensional tasks within the SR and the RP control laws.

A. iCAT SR

As shown in Remark 1 (see also Appendix VIII-B), the main source of discontinuities of the method of [15] is related to the projector $\check{\boldsymbol{J}}_i^{\sharp}$. To achieve **C1**, the regularized pseudoinverse operator of [1] is applied to the method of [15], leading to a control law $\mathcal{C}_{\text{iCAT SR}} = (\{+\}, \square, \{\boldsymbol{I}\}, \{\boldsymbol{I}\}, \{\square^{\#,\boldsymbol{A}_i,\boldsymbol{I}}\}, \{\boldsymbol{I}\})$. Therefore, at each time k, the vector $\boldsymbol{q}$ is given by

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_1^{\#,\boldsymbol{A}_1,\boldsymbol{I}}\dot{\boldsymbol{x}}_1 + \sum_{i=2}^{l}\left(\boldsymbol{P}_i\boldsymbol{J}_i^{\#,\boldsymbol{A}_i,\boldsymbol{I}}\dot{\boldsymbol{x}}_i\right) + \boldsymbol{P}_{l+1}\dot{\boldsymbol{q}}_{l+1}$$

$$\boldsymbol{P}_i = \left(\boldsymbol{I} - \boldsymbol{J}_1^{\#,\boldsymbol{A}_1,\boldsymbol{I}}\boldsymbol{J}_1\right)\ldots\left(\boldsymbol{I} - \boldsymbol{J}_{i-1}^{\#,\boldsymbol{A}_{i-1},\boldsymbol{I}}\boldsymbol{J}_{i-1}\right)$$

so the operator $\square^{\#,\boldsymbol{A}_i,\boldsymbol{I}}$ ensures the continuity of projectors $\boldsymbol{J}^{\#,\boldsymbol{A}_i,\boldsymbol{I}}$ and $\boldsymbol{P}_i$.

Moreover, since $L = \{+\}$, results of [18], [22] can be exploited to achieve **C2**. According to [18], [22], in fact, a forgetting factor k_f close to 1 may prevent jumps between pareto-optimal solutions at different time k, forcing the solution at time k to be searched as closest as possible to the one computed at time k − 1 (see [18], [22] for details).

However, the method may be still suboptimal with respect to minimizing all the task errors of subtasks with $a_{i,j} = 1$ (e.g., note that if $\boldsymbol{A}_i = \boldsymbol{I}$, then $\mathcal{C}_{\text{iCAT SR}} \equiv \mathcal{C}_{\text{SR}}$, so the hypotheses of [27] are required to ensure that all the task errors converge to **0**).

B. iCAT RP

To to achieve C1 and according to Remark 1, we now exploit (5) to ensure the continuity projectors $(\boldsymbol{J}_i\tilde{\boldsymbol{P}}_{i+1})^{\#}$ (Section IV-B1) and $\tilde{\boldsymbol{P}}_{i+1}$ (Section IV-B2) with respect to activation of tasks. Section IV-B3 leverages on the results of [18] to also ensure **C2**. See Fig. 3 for a guide to theorems. Similarly to [1], formal properties will be proved considering just the task-oriented contribution in (5), that is, for each matrix $\boldsymbol{J}$ with activation matrix $\boldsymbol{A}$, then

$$\boldsymbol{J}^{\#,\boldsymbol{A},\boldsymbol{I}} = \left(\sqrt{\boldsymbol{A}}\boldsymbol{J}\right)^{\#}\sqrt{\boldsymbol{A}}\boldsymbol{A} = \left(\bar{\boldsymbol{J}}^{\#}\ \boldsymbol{0}\right)\begin{pmatrix}\bar{\boldsymbol{A}} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{pmatrix}.^{6} \tag{7}$$

[6] See (32) and (41) of [1]. Also, note that (7) is equivalent to (5) when $\boldsymbol{Q} = \boldsymbol{I}$ and $\sqrt{\boldsymbol{A}}\boldsymbol{J}$ is not singular. While $\boldsymbol{Q} = \boldsymbol{I}$ is verified (the auxiliary problem is not used in the proposed formulation), the SVO contribution will be exploited to preserve the continuity of pseudoinverses when exists $a_{i,j} \to 0$. However, if $\boldsymbol{\Lambda}$ is correctly tuned and tasks are regular, then errors caused by this approximation are relevant only for small values $a_{i,j}$.

Authorized licensed use limited to: Orebro University. Downloaded on September 28,2021 at 14:05:40 UTC from IEEE Xplore. Restrictions apply.

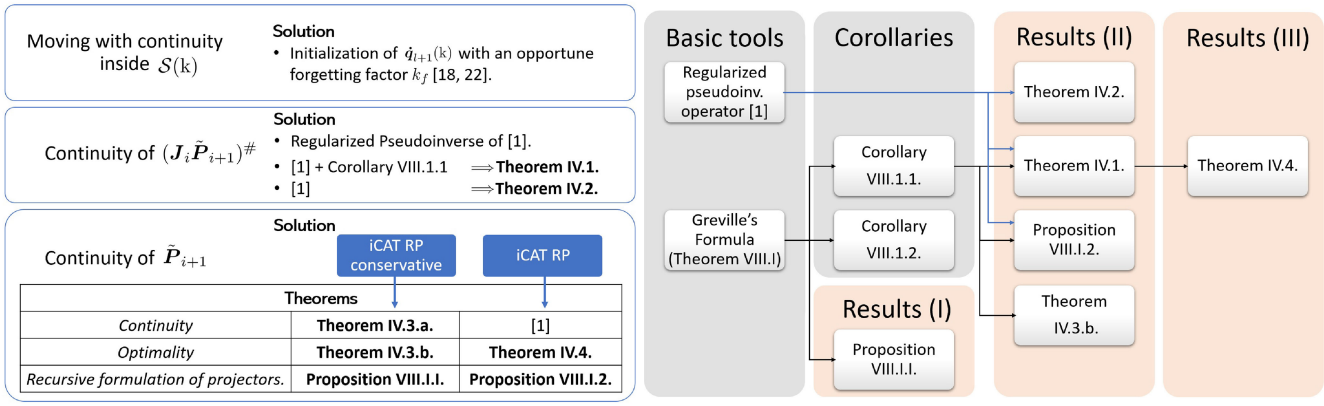


Fig. 3. iCAT reverse priority control laws: a guide to theorems.

1) Continuity of $(\boldsymbol{J}_i\tilde{\boldsymbol{P}}_{i+1})^{\#}$*:* Choosing $\square^{\sharp} = \square^{\#,\boldsymbol{A}_i,\boldsymbol{I}}$, we ensure continuity with respect to activation and deactivation of tasks at the same level of priority [1]. Hence, in the following, we analyze the requirements for ensuring null task errors with the method defined by the tuple $\mathcal{C}_{\text{iCAT RP}} = (\{+\}, \square, \{\boldsymbol{I}\}, \{\boldsymbol{I}\}, \{\square^{\#,\boldsymbol{A}_i,\boldsymbol{I}}\}, \{\tilde{\boldsymbol{P}}_{i+1}\})$ and related to the set of optimization problems in (6), with $\gamma = 0$.

Consider a control objective $x_{i,j}$ corresponding to the jth row of the task $\boldsymbol{x}_i$. Given a desired value $x_{d_{i,j}}$, let $e_{i,j} = x_{d_{i,j}} - x_{i,j}$ be the task error, so that $\dot{e}_{i,j} = \dot{x}_{d_{i,j}} - \boldsymbol{J}_{i,j}\dot{\boldsymbol{q}}$. By using (2), we have that $\dot{e}_{i,j} = -K_{i,j}e_{i,j} + \varepsilon_{i,j}$, with $K_{i,j} > 0$ and $\varepsilon_{i,j} = \dot{x}_{i,j} - \boldsymbol{J}_{i,j}\dot{\boldsymbol{q}}$. At each discrete time k, according to $\mathcal{C}_{\text{iCAT-RP}}$, the control $\dot{\boldsymbol{q}}$ that can be written as

$$\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_i + \sum_{k=2}^{i} \tilde{\boldsymbol{P}}_{k+1}\boldsymbol{\nu}_k, \text{ where}$$

$$\boldsymbol{\nu}_k = \left(\boldsymbol{J}_k\tilde{\boldsymbol{P}}_{k+1}\right)^{\#,\boldsymbol{A}_k,\boldsymbol{I}}\left(\dot{\boldsymbol{x}}_k - \boldsymbol{J}_k\dot{\boldsymbol{q}}_{k+1}\right)$$

so $\varepsilon_{i,j}$ can be split as $\varepsilon_{i,j} = \varepsilon_{i,j}^{(i)} + \delta_{i,j}$, with

1) $\varepsilon_{i,j}^{(i)} = \dot{x}_{i,j} - \boldsymbol{J}_{i,j}\dot{\boldsymbol{q}}_i$ (i.e., the value of $\varepsilon_{i,j}$ at iteration i) defining the dynamics of $e_{i,j}$ due to subtasks at the same priority level and tasks with lower priority.
2) $\delta_{i,j} = -\boldsymbol{J}_{i,j}\sum_{k=2}^{i}\tilde{\boldsymbol{P}}_{k+1}\boldsymbol{\nu}_k$ collecting the dynamics of $e_{i,j}$ due to the higher priority tasks.

Consequently, $\dot{e}_{i,j} = -K_{i,j}e_{i,j}$ ensures $e_{i,j} \to 0$ whenever $\varepsilon_{ij} \to 0$. We show how the task error has a stable, exponentially decreasing dynamics, whenever it is fully active, i.e., $a_{i,j} = 1$, and it is not affected by higher priority tasks, i.e., $\delta_{i,j} = 0$.

Theorem IV.1: Consider the $\mathcal{C}_{\text{iCAT RP}}$ method, with activation matrices $\{\boldsymbol{A}_h\}, h \in \{1, \ldots, l\}$ and $\square^{\sharp}$ defined according to (7). Let $(x_{d_{i,j}}, \dot{x}_{d_{i,j}}) \in \mathbb{R}^2$ be a generic task reference. The error $e_{i,j}$ for the ith task and its jth components, with $\delta_{i,j} = 0$, converges exponentially to zero if and only if $a_{i,j} = 1$ and $\bar{\boldsymbol{J}}_i$ is f.r.r.

Proof: According to (4), we have that

$$\varepsilon_{i,j}^{(i)} = \dot{x}_{i,j} - \boldsymbol{J}_{i,j}\dot{\boldsymbol{q}}_{i+1} + \boldsymbol{J}_{i,j}\tilde{\boldsymbol{P}}_{i+1}\left(\boldsymbol{J}_i\tilde{\boldsymbol{P}}_{i+1}\right)^{\#,\boldsymbol{A}_i,\boldsymbol{I}}\left(\dot{\boldsymbol{x}}_i - \boldsymbol{J}_i\dot{\boldsymbol{q}}_{i+1}\right). \quad (8)$$

Sufficiency: $a_{i,j} = 1$ and $\bar{\boldsymbol{J}}_i$ f.r.r. $\Rightarrow \forall(x_{d_{i,j}}, \dot{x}_{d_{i,j}})$, $\varepsilon_{i,j}^{(i)} = 0$. Since $\text{rank}(\bar{\boldsymbol{J}}_i) = \text{rank}(\bar{\boldsymbol{J}}_i\tilde{\boldsymbol{P}}_{i+1})$ [16], (8) can be simplified as follows. Assume $\boldsymbol{J}_i$ be f.r.r. and let $\bar{m}_i \leq m_i$ be the rank of $\boldsymbol{A}_i$ ($\bar{m}_i < m_i$ if and only if some rows are inactive). Also, let $\boldsymbol{\Pi}_i \in \mathbb{R}^{m_i \times m_i}$ be a permutation matrix such that $\boldsymbol{\Pi}_i\boldsymbol{A}_i\boldsymbol{\Pi}_i^T = \text{diag}(\bar{\boldsymbol{A}}_i, \boldsymbol{0})$. Then, according to (7), we have that

$$\left(\boldsymbol{J}_i\tilde{\boldsymbol{P}}_{i+1}\right)^{\#,\boldsymbol{A}_i,\boldsymbol{I}} = \begin{bmatrix}\left(\bar{\boldsymbol{J}}_i\tilde{\boldsymbol{P}}_{i+1}\right)^{\#} & \boldsymbol{0}\end{bmatrix}\begin{pmatrix}\bar{\boldsymbol{A}}_i & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{pmatrix}\boldsymbol{\Pi}_i.$$

Since $\bar{\boldsymbol{J}}_i$ is f.r.r., then

$$\boldsymbol{J}_{i,j}\tilde{\boldsymbol{P}}_{i+1}\begin{bmatrix}\left(\bar{\boldsymbol{J}}_i\tilde{\boldsymbol{P}}_{i+1}\right)^{\#} & \boldsymbol{0}\end{bmatrix} = \begin{bmatrix}\boldsymbol{e}_{\varsigma}^T & \boldsymbol{0}\end{bmatrix}$$

where ς is defined by $\boldsymbol{J}_{i,j} = (\bar{\boldsymbol{J}}_i)_{\{\varsigma,:\}}$ and $\boldsymbol{e}_{\varsigma} \in \mathbb{R}^{\bar{m}_i}$ is the vector of the canonical base of $\mathbb{R}^{\bar{m}_i}$ with the 1 in the ςth row (let us use this notation from now on). Consequently,

$$\boldsymbol{J}_{i,j}\tilde{\boldsymbol{P}}_{i+1}\left(\boldsymbol{J}_i\tilde{\boldsymbol{P}}_{i+1}\right)^{\#,\boldsymbol{A}_i,\boldsymbol{I}} = \begin{pmatrix}\boldsymbol{e}_{\varsigma}^T\bar{\boldsymbol{A}}_i & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{pmatrix}\boldsymbol{\Pi}_i$$

and, since $\boldsymbol{J}_i = \boldsymbol{\Pi}_i^T\left(\bar{\boldsymbol{J}}_i\right)$ and $\dot{\boldsymbol{x}}_i = \boldsymbol{\Pi}_i^T\left(\dot{\bar{\boldsymbol{x}}}_i\right)$, (8) is equivalent to

$$\begin{aligned}\varepsilon_{i,j}^{(i)} &= \dot{x}_{i,j} - \boldsymbol{J}_{i,j}\dot{\boldsymbol{q}}_{i+1} + \boldsymbol{e}_{\varsigma}^T\bar{\boldsymbol{A}}_i\left(\dot{\bar{\boldsymbol{x}}}_i - \bar{\boldsymbol{J}}_i\dot{\boldsymbol{q}}_{i+1}\right)\\ &= (1 - a_{i,j})\left(\dot{x}_{i,j} - \boldsymbol{J}_{i,j}\dot{\boldsymbol{q}}_{i+1}\right)\end{aligned}$$

so $a_{i,j} = 1 \Rightarrow \varepsilon_{i,j}^{(i)} = 0$.

Necessity: $a_{i,j} = 1$ and $\bar{\boldsymbol{J}}_i$ f.r.r. $\Leftarrow \forall(x_{d_{i,j}}, \dot{x}_{d_{i,j}})$, $\varepsilon_{i,j}^{(i)} = 0$. Let us drop the assumption of $\bar{\boldsymbol{J}}_i$ f.r.r. Let $\bar{\boldsymbol{J}}_i$ be partitioned as $\bar{\boldsymbol{J}}_i = \begin{pmatrix}\boldsymbol{R}\\ \boldsymbol{S}\end{pmatrix}$, where $\boldsymbol{R} \in \mathbb{R}^{m_r \times n}$ is f.r.r., $\boldsymbol{S} = \boldsymbol{\chi}\boldsymbol{R} \in \mathbb{R}^{m_s \times n}$ and $\bar{m}_i = m_r + m_s$. According to Corollary VII.1.1 (see Appendix), $\exists \boldsymbol{W} \in \mathbb{R}^{m_r \times \bar{m}_i}$ such that

$$\begin{bmatrix}\left(\bar{\boldsymbol{J}}_i\tilde{\boldsymbol{P}}_{i+1}\right)^{\#} & \boldsymbol{0}\end{bmatrix} = \begin{bmatrix}\left(\boldsymbol{R}\tilde{\boldsymbol{P}}_{i+1}\right)^{\#}\boldsymbol{W} & \boldsymbol{0}\end{bmatrix}.$$

Let the matrices $\bar{\boldsymbol{A}}_i$ and $\boldsymbol{W}$ to be partitioned as $\bar{\boldsymbol{A}}_i = \text{diag}(\bar{\boldsymbol{A}}_{r,i}, \bar{\boldsymbol{A}}_{s,i})$ and $\boldsymbol{W} = [\boldsymbol{W}_r \quad \boldsymbol{W}_s]$, where $\boldsymbol{W}_r \in \mathbb{R}^{m_r \times m_r}$

Authorized licensed use limited to: Orebro University. Downloaded on September 28,2021 at 14:05:40 UTC from IEEE Xplore. Restrictions apply.

and $\boldsymbol{W}_s \in \mathbb{R}^{m_r \times m_s}$ are defined as

$$\boldsymbol{W}_r = \boldsymbol{I} - \boldsymbol{\eta}\boldsymbol{\chi} = \boldsymbol{I} - \boldsymbol{\chi}^T \left[\boldsymbol{I} + \boldsymbol{\chi}\boldsymbol{\chi}^T\right]^{-1}$$
$$\boldsymbol{W}_s = \boldsymbol{\eta} = \boldsymbol{\chi}^T \left[\boldsymbol{I} + \boldsymbol{\chi}\boldsymbol{\chi}^T\right]^{-1}.$$

Since the matrix $\boldsymbol{R}\tilde{\boldsymbol{P}}_{i+1}$ is f.r.r., we have that

$$\boldsymbol{J}_{i,j}\tilde{\boldsymbol{P}}_{i+1}\left(\boldsymbol{R}\tilde{\boldsymbol{P}}_{i+1}\right)^{\#} = \begin{cases} 2\boldsymbol{e}_{\varsigma'}^T & \text{if } \boldsymbol{J}_{i,j} = \boldsymbol{R}_{\varsigma',:} \\ \boldsymbol{\chi}_{i,j} & \text{if } \boldsymbol{J}_{i,j} \subseteq \boldsymbol{S} \end{cases}$$

where $\boldsymbol{e}_{\varsigma'} \in \mathbb{R}^{m_r}$, and $\boldsymbol{\chi}_{i,j}$ is the row of the matrix $\boldsymbol{\chi}$ corresponding to $\boldsymbol{J}_{i,j}$. Consider the following two cases:

1) $\boldsymbol{J}_{i,j} \subseteq \boldsymbol{R}$. The portion of the task error rate related to $\dot{x}_{i,j}$ depends on the value of

$$\boldsymbol{e}_{\varsigma'}^T \boldsymbol{W}\bar{\boldsymbol{A}}_i\dot{\boldsymbol{x}}_i = \boldsymbol{e}_{\varsigma'}^T \begin{bmatrix} \boldsymbol{W}_r\bar{\boldsymbol{A}}_{r,i} & \boldsymbol{W}_s\bar{\boldsymbol{A}}_{s,i} \end{bmatrix} \begin{pmatrix} \dot{\tilde{\boldsymbol{x}}}_{r,i} \\ \dot{\tilde{\boldsymbol{x}}}_{s,i} \end{pmatrix}.$$

Hence, since $\dot{\tilde{\boldsymbol{x}}}_{s,i}$ is an arbitrary vector, then

$$\boldsymbol{e}_{\varsigma'}^T \boldsymbol{W}\bar{\boldsymbol{A}}_i\dot{\tilde{\boldsymbol{x}}}_i = \dot{x}_{i,j}\ \forall \dot{x}_{i,j} \iff \begin{cases} \boldsymbol{e}_{\varsigma'}^T \boldsymbol{W}_r\bar{\boldsymbol{A}}_{r,i} = \boldsymbol{e}_{\varsigma'}^T \\ \boldsymbol{e}_{\varsigma'}^T \boldsymbol{W}_s\bar{\boldsymbol{A}}_{s,i} = \boldsymbol{0} \end{cases}$$

that holds only if $\boldsymbol{W}_s = \boldsymbol{0} \iff \boldsymbol{\chi} = \boldsymbol{0}$, so if and only if $\boldsymbol{W}_r = \boldsymbol{I}$ (due to the definition of $\boldsymbol{W}$) and $a_{i,j} = 1$. The same conclusion is obtained when the portion of the task error rate related to $\dot{\boldsymbol{q}}_{i+1}$ is considered.

2) $\boldsymbol{J}_{i,j} \subseteq \boldsymbol{S}$. If $\boldsymbol{R}$ is f.r.r., then $\|\boldsymbol{\chi}\boldsymbol{\eta}\| \leq 1$ and $\boldsymbol{\chi}_{i,j}^T \boldsymbol{W}\bar{\boldsymbol{A}}_i\dot{\tilde{\boldsymbol{x}}}_i < \dot{x}_{i,j}$ (i.e., the task error rate related to rows which are linear dependent is never null). Consequently, the optimality can be ensured only if $\bar{\boldsymbol{J}}_i$ is f.r.r. ■

It is worth noting that $\boldsymbol{J}_i$ f.r.r. $\Rightarrow \bar{\boldsymbol{J}}_i$ frr, while $\bar{\boldsymbol{J}}_i$ frr $\not\Rightarrow$ $\boldsymbol{J}_i$ f.r.r. Hence, $m_i > n$ is allowed if we ensure the set of active tasks to be always full row rank.

Theorem IV.1 gives us conditions for a good task tracking performance neglecting the effect of higher priority tasks. We now show under which conditions task error is not affected by linearly independent higher priority tasks, which both leads to $\delta_{i,j} = 0$ and ensures the correct task hierarchy execution.

Theorem IV.2: Given the ith task, consider higher priority kth tasks which are not inactive, $\left\{(\bar{\boldsymbol{x}}_{d_k}, \dot{\tilde{\boldsymbol{x}}}_{d_k}), \bar{\boldsymbol{A}}_k\right\}$, $2 \leq k \leq i$. Then, $\delta_{i,j} = 0$ if and only if $\boldsymbol{J}_{i,j}\tilde{\boldsymbol{P}}_k = 0$.

Proof: Sufficiency: It holds due to the definition of $\delta_{i,j}$.

Necessity: Since $\left\{(\bar{\boldsymbol{x}}_{d_k}, \dot{\tilde{\boldsymbol{x}}}_{d_k}), \bar{\boldsymbol{A}}_k\right\}$ are arbitrary, then $\boldsymbol{J}_{i,j}\sum_{k=2}^{i}\tilde{\boldsymbol{P}}_k\boldsymbol{\nu}_k = 0$ only if $\boldsymbol{J}_{i,j}\tilde{\boldsymbol{P}}_k = \boldsymbol{0}$ for each k. The proof can be given by contradiction. Note that $\tilde{\boldsymbol{P}}_k\boldsymbol{\nu}_k = \dot{\boldsymbol{q}}_k - \dot{\boldsymbol{q}}_{k+1}$ is not null whenever $\dot{\boldsymbol{q}}_{k+1}$ does not minimize the cost function of the iCAT RP optimization problem at the kth step. Also, by using (2) and (7), each $\boldsymbol{\nu}_k$ can be computed as

$$\left[(\bar{\boldsymbol{J}}_k\tilde{\boldsymbol{P}}_{k+1})^{\#}\ \boldsymbol{0}\right] \boldsymbol{A}_k\left(\dot{\boldsymbol{x}}_{d_k} + \boldsymbol{K}_k\left(\boldsymbol{x}_{d,k} - \boldsymbol{x}_k\right) - \boldsymbol{J}_k\dot{\boldsymbol{q}}_{k+1}\right)$$

so activation/deactivation and transient dynamics of high-priority tasks affect $\dot{e}_{i,j}$ whenever $\boldsymbol{J}_{i,j}\tilde{\boldsymbol{P}}_k \neq \boldsymbol{0}$ and $\bar{\boldsymbol{A}}_k \neq \boldsymbol{0}$. ■

Note that Theorems IV.1 and IV.2 give the conditions for exponential stability of the error dynamics.

2) Continuity of $\tilde{\boldsymbol{P}}_{i+1}$: Leveraging upon the solution proposed in [1], our objective is to shape the null space of $\tilde{\boldsymbol{J}}_{i+1}$ (i.e., the matrix $\tilde{\boldsymbol{P}}_{i+1}$) by taking into account the activation matrix (i.e., $\tilde{\boldsymbol{\mathcal{A}}}_{i+1}$), to obtain smooth transitions of tasks. For this purpose, we propose two methods called *iCAT conservative RP* and *iCAT RP*.

a) iCAT Conservative RP: The *conservative* approach computes the matrix $\tilde{\boldsymbol{J}}_{i+1}$ considering all the tasks as active. In doing so, the continuity of the pseudoinverse is granted, and the resulting projection matrix induces the ith tasks to live (if possible) in the portion of the null space which minimally interfere with the lower priority tasks. Then, the contribution of the activation is added to weigh the directions of the null space as follows:

$$\tilde{\boldsymbol{P}}_{i+1} = \boldsymbol{I} - \tilde{\boldsymbol{J}}_{i+1}^{\#}\tilde{\boldsymbol{\mathcal{A}}}_{i+1}\tilde{\boldsymbol{J}}_{i+1}. \tag{9}$$

This leads to a suboptimal but continuous control law, with a recursive formulation given as shown in Appendix VIII-D. Continuity and optimality are analyzed in the following theorem.

Theorem IV.3: Given any $\{(\boldsymbol{x}_{d_h}, \dot{\boldsymbol{x}}_{d,h}), \boldsymbol{A}_h\}$, the projector $\tilde{\boldsymbol{P}}_{i+1} = \boldsymbol{I} - \tilde{\boldsymbol{J}}_{i+1}^{\#}\tilde{\boldsymbol{\mathcal{A}}}_{i+1}\tilde{\boldsymbol{J}}_{i+1}$

1) is continuous with respect to the matrix $\boldsymbol{\mathcal{A}}_i$;
2) ensures $\varepsilon_{i,j} = 0$ for each $x_{i,j}$ such that $a_{i,j} = 1$, $x_{i,j}$ l.i. from the set $\{\boldsymbol{x}_k : k < i,\ \mathrm{tr}(\boldsymbol{A}_k) > 0\}$, if and only if $\boldsymbol{J}_i$ is f.r.r. and l.i. to all the other $\boldsymbol{J}_h$.

Proof:

1) *Continuity:* Since the matrix $\tilde{\boldsymbol{J}}_{i+1}$ is independent from the activation values, the use of continuous activation functions ensures the matrix $\tilde{\boldsymbol{P}}_{i+1}$ to be continuous with respect to task activation.
2) *Optimality: Sufficiency:* Theorem IV.1 ensures $\varepsilon_{i,j}^{(i)} = 0$ since $a_{i,j} = 1$ and $\boldsymbol{J}_i$ f.r.r. $\Rightarrow \bar{\boldsymbol{J}}_i$ f.r.r.. Also, $\boldsymbol{J}_i$ f.r.r. and for each (i,h), $h \neq i$, $\mathrm{span}(\boldsymbol{J}_i^T) \cap \mathrm{span}(\boldsymbol{J}_h^T) = \{\boldsymbol{0}\}$ implies that $\tilde{\boldsymbol{J}}_{i+1}$ is f.r.r. for each i. Consequently, since $\boldsymbol{J}_{i,j}$ is l.i. to $\{\boldsymbol{J}_k\}$, $k < i$, then $\boldsymbol{J}_{i,j} \subseteq \tilde{\boldsymbol{J}}_{k+1}$ for each $k \in [2, i]$. Hence, according to Corollary VII.1.2, $\tilde{\boldsymbol{J}}_{k+1}$ f.r.r. and $a_{i,j} = 1$ ensures Theorem IV.2 to hold, so $\delta_{i,j} = 0 \Rightarrow \varepsilon_{i,j} = 0$.
   Necessity: Theorem IV.1 requires $a_{i,j} = 1$ and $\bar{\boldsymbol{J}}_i$ f.r.r. to ensure $\varepsilon_{i,j} = 0$ whenever $\delta_{i,j} = 0$. Also, linearly, independence between all the tasks and subtasks is required since inactive rows affect the computation of matrices $\{\tilde{\boldsymbol{J}}_{k+1}\}$, $k \in [2, i]$. The proof is given considering the effect of: (a) inactive rows of $\boldsymbol{J}_i$ (SP subtasks) and inactive low-priority (LP) tasks; (b) inactive higher priority (HP) tasks.
   a) $\tilde{\boldsymbol{J}}_{k+1}$ may not be f.r.r. if $\boldsymbol{J}_i$ is not f.r.r. or it exists $h > i$ such that $\mathrm{span}(\boldsymbol{J}_i^T) \cap \mathrm{span}(\boldsymbol{J}_h^T) \neq \{\boldsymbol{0}\}$. Consequently, according to Corollary VII.1.2, $\boldsymbol{J}_{i,j}\tilde{\boldsymbol{P}}_{k+1} = \boldsymbol{0}$ if $\boldsymbol{J}_{i,j}$ l.i. to $\boldsymbol{J}_k$ and $a_{i,j} = 1$ only if all the l.d. LP tasks and SP subtasks are active.
   b) It may be $\delta_{i,j} > 0$ if $\boldsymbol{J}_{i,j}$ l.i. to $\{\bar{\boldsymbol{J}}_k\}$ and $a_{i,j} = 1$, but $\boldsymbol{J}_{i,j}$ l.d. to $\{\boldsymbol{J}_k\}$, $k \in [2, i]$. In such case, $\boldsymbol{J}_{i,j} \not\subseteq \tilde{\boldsymbol{J}}_{k+1} \Rightarrow \boldsymbol{J}_{i,j}^T \notin \mathrm{span}(\tilde{\boldsymbol{J}}_{k+1}^T)$. Consequently, $\boldsymbol{J}_{i,j}\tilde{\boldsymbol{P}}_{k+1}|_{\tilde{\boldsymbol{A}}_{k+1} = \boldsymbol{I}} \neq \boldsymbol{0} \Rightarrow \delta_{i,j} \neq 0$ in some cases. ■

Authorized licensed use limited to: Orebro University. Downloaded on September 28,2021 at 14:05:40 UTC from IEEE Xplore. Restrictions apply.

Remark 2: Note that:

1) $\square^{\#,d}$ is still required to preserve the continuity of $\{\tilde{\boldsymbol{P}}_{i+1}\}$ when relaxing the hypothesis of f.r.r. and l.i. of Jacobians.
2) If $\sum_{i=1}^{l} m_i > n$, then $\exists h \in \{l, \ldots, 1\}$ such that $i < h \Rightarrow \tilde{\boldsymbol{P}}_{i+1} = \boldsymbol{I}$. Consequently, tasks $\{\boldsymbol{x}_i\}_{i=1}^{h-1}$ are as in the $\mathcal{C}_{\text{iCAT SR}}$ approach.

b) iCAT RP: The tight constraints imposed by the inactive tasks, which have been recognized as the source of suboptimality of the iCAT RP conservative method, can be relaxed by the use of the following projection matrix:

$$\tilde{\boldsymbol{P}}_{i+1} := \boldsymbol{I} - \left(\tilde{\boldsymbol{J}}_{i+1}\right)^{\#,\tilde{\boldsymbol{\mathcal{A}}}_{i+1},\boldsymbol{I}} \tilde{\boldsymbol{J}}_{i+1} \tag{10}$$

where the operator of (5) automatically removes the contribution of the inactive rows from the computation of $\tilde{\boldsymbol{J}}_{i+1}^{\#}$ while preserving the continuity. A recursive formulation of the projector is given in Section VIII-E.

With the projector proposed in (10), continuity of the $\mathcal{C}_{\text{iCAT RP}}$ method is ensured by the definition of the regularized pseudoinverse operator in (7). As for the optimality of the method, we want to guarantee that the new projector exploits the results of the previous theorems, in particular that it still holds that active lower priority tasks linearly independent from higher priority ones are guaranteed to achieve stable tracking.

Theorem IV.4: Given a set $\{\boldsymbol{x}_{d_h}, \dot{\boldsymbol{x}}_{d_h}, \boldsymbol{A}_h\}$, consider a task $x_{i,j}$ which is active, $a_{i,j} = 1$, and linearly independent from higher priority tasks which are not inactive, hence in the set $\{\boldsymbol{x}_k : k < i, \text{tr}(\boldsymbol{A}_k) > 0\}$. Projector (10) guarantees exponential convergence of task error $\varepsilon_{i,j}$ if and only if, at each time k, $\bar{\boldsymbol{J}}_i$ is f.r.r. and l.i. to all the other $\bar{\boldsymbol{J}}_j$.

Proof: The proof can be derived from Theorem IV.3.b considering that the contribution of inactive subtasks is automatically removed by using the regularized pseudoinverse operator. ■

Remark 3: $\tilde{\boldsymbol{P}}_{i+1} = \boldsymbol{I} - \tilde{\boldsymbol{J}}_{i+1}^{\#} \tilde{\boldsymbol{J}}_{i+1} \Rightarrow \delta_{i,j} = 0$ whenever $a_{i,j} > 0$ if tasks are f.r.r. and l.i. Differently, (9) and (10) ensure $\delta_{i,j} = 0$ only if $a_{i,j} = 1$. If $0 < a_{i,j} < 1$, then $e_{i,j}$ is partially affected by each $\boldsymbol{\nu}_k$, $k < i$ s.t. $\text{tr}(\boldsymbol{A}_k) > 0$. As in [1], in fact, the nonorthogonality of $\tilde{\boldsymbol{P}}_{i+1}$ during transitions is exploited to achieve continuity.

3) Moving with Continuity Inside $\mathcal{S}(\text{k})$: Similarly to Section IV-A and according to [18], [22], a forgetting factor close to 1 can be used to account for 2. Thus, at each time k, the optimal solution is searched as closest as possible to the one computed at time $\text{k} - 1$. Also, thanks to the reverse order, the role of the auxiliary problem is now *automatically* played by the projector $\tilde{\boldsymbol{P}}_{i+1}$.

Collecting all the pieces together, and considering the formulation of the projector given in VIII-E, the iterative control law of the iCAT RP is formalized as follows:

$$\begin{aligned}
&\mathcal{C}_{\text{iCAT RP}} = (+, \square, \{\boldsymbol{I}\}, \{\boldsymbol{I}\}, \{\square^{\#,\boldsymbol{A}_i,\boldsymbol{I}}\}, \{\tilde{\boldsymbol{P}}_{i+1}\}) \\
&\tilde{\boldsymbol{P}}_{i+1} = \left(\boldsymbol{I} - \boldsymbol{B}_i \boldsymbol{B}_i{}^{\#}\right) \boldsymbol{Q}_i^r + \boldsymbol{B}_i \boldsymbol{B}_i^{\#} \\
&\boldsymbol{B}_i = \left(\boldsymbol{J}_{RA,i}^{\#,\boldsymbol{\mathcal{A}}_i,\boldsymbol{I}} \boldsymbol{\mathcal{A}}_i^{\#}\right)_{\{:,1:m_i\}} \\
&\boldsymbol{Q}_i^r = \boldsymbol{I} - \boldsymbol{J}_{RA,i}^{\#,\boldsymbol{\mathcal{A}}_i,\boldsymbol{I}} \boldsymbol{J}_{RA,i}
\end{aligned} \tag{11}$$

and, thanks to (5), inherits a polynomial complexity in the number of tasks. Note that, similarly to the original approach [16], the computation of $\tilde{\boldsymbol{P}}_{i+i}$ requires the pseudoinversion of the reverse augmented Jacobian $\boldsymbol{J}_{RA,i}$. Therefore, $\tilde{\boldsymbol{P}}_{i+i}$ may be deformed by the damping action in case of kinematic or algorithmic singularities. Experiments in [16] show that errors are practically negligible with small damping (and consequently for each time k such that $\mathcal{C}_{\text{RP}} \equiv \mathcal{C}_{\text{iCAT RP}}$ at time k – i.e., if $\boldsymbol{A}_i(\text{k}) \in \{\boldsymbol{0}; \boldsymbol{I}\}$ for each $i \in \{1, \ldots, l\}$). However, the questions of how enforcing robustness to singularities of iCAT approaches based on augmented Jacobians and how to determine optimal damping parameters are still open and left for deeper investigations.

Remark 4: The use of weighted projectors couples performances to the activation of linearly dependent subtasks (see Corollary VII.1.1 in Section VIII). Dropping the assumptions of f.r.r. and l.i. of Jacobians may prevent correctness and optimality guarantees of all iCAT methods based on augmented Jacobians (i.e., the iCAT TPC [1], [11] and the iCAT RPs approaches proposed in this article).

As a result of the theoretical analysis, the properties of the iCAT RP approach defined in (11) can be summarized as follows.

1) The regularized pseudoinverse operator of [1] with smooth activation functions ensures the continuity of pseudoinverses.
2) Projectors $\tilde{\boldsymbol{P}}_{i+1}$ defined as in (10) ensure smooth changes of the manifold $\mathcal{S}(\text{k})$.
3) Each active tasks is optimally executed if it is f.r.r. and l.i. from all its higher priority ones.
4) Whenever $\mathcal{C}_{\text{iCAT RP}} \equiv \mathcal{C}_{\text{RP}}$, correctness and optimality are ensured even in case of kinematic or algorithmic singularities and with small damping action [16].

V. Fleet Management Tasks

Some examples of NSB-based controllers (centralized or decentralized) have already been applied to multirobot fleets in [5]–[13], [33]. For convenience, we recall the mathematical definition of some common tasks (according to the aforementioned literature). In particular, considering our scenario, the generalized coordinates of the ith robot are defined by the vector $\boldsymbol{q}_i = (\boldsymbol{p}_i \ \theta_i)^T = (x_i \ y_i \ \theta_i)^T$ (i.e., the Cartesian position and orientation of the mobile robot with respect to a fixed reference frame).

A. Collision Avoidance

As in [4], [5], [7], [8], [23], we use a discrete bounded-shaped representation of obstacles $\boldsymbol{\mathcal{O}} = (o_1, \ldots, o_p)$, where obstacles are gathered together in circle-shaped clusters. Let d_m be the safety distance to obstacles; each constraint

$$\left\| \boldsymbol{p}_i - \boldsymbol{p}_{o_j} \right\| \geq d_m, \text{for } i = 1, \ldots, N, \ j = 1, \ldots, p,$$

is formalized by the following unilateral task:

$$x_{i,j} = \left\| \boldsymbol{p}_i - \boldsymbol{p}_{o_j} \right\|, x_{d_{i,j}} = d_m + \beta_m^7, \dot{x}_{d_{i,j}} = 0$$

[7]The task is here formalized according to [8]. Better results (e.g., avoiding local minima, or chattering phenomena) can be obtained using different planning techniques, such that $x_{d_{i,j}}$ will drive the robot to an arbitrary point where the inequality constraint is satisfied.

Authorized licensed use limited to: Orebro University. Downloaded on September 28,2021 at 14:05:40 UTC from IEEE Xplore. Restrictions apply.

where β_m is a positive slack variable that transforms the inequality constraint into an equality one. Note that, $x_{i,j} \in [0, d_m + \beta_m]$ is the activation region, with $a_{i,j} = 1$ if $x_{i,j} \leq d_m$, $a_{i,j} = 0$ if $x_{i,j} \geq d_m + \beta_m$ and $a_{i,j} \in (0, 1)$ being a smooth decreasing sigmoid in $(d_m, d_m + \beta_m)$. As in [1], we use $a_{i,j} = 0.5(\cos(\alpha\pi) + 1)$, with $\alpha = \beta_m^{-1}(x_{i,j} - d_m)$.[8] The collective $Np \times n$ Jacobian is computed from differentiation, and task errors are defined according to (2). This task is set as the one with highest priority [8], [12], [23], so that when a robot may hit an obstacle, safety issues locally prevail on tracking performances. Even if simple, this formulation allows to prevent collisions only if the following assumptions are verified.

1) Each robot should never be engaged in avoiding more than two obstacles, and obstacles are not collinear, otherwise the Jacobian will be singular. This requires to opportunely select the subtasks to be activated at each time.
2) The slack variable β_m should be greater than the minimum braking distance.

We remark that the approach may suffer of local minima with symmetric configurations of obstacles. In this case, the issue can be tackled by adding a small random noise or with an opportune goal selection, as common in the literature of reactive approaches (e.g., [34]). Note that more complex strategies could be implemented [35], especially when the NSB method is moved to the dynamic level [36].

B. Keeping the Centroid of the Formation Along a Desired Trajectory

The task of maintaining the centroid of the fleet along a desired trajectory [5], [7] is formalized as

$$\boldsymbol{x} = \boldsymbol{p}_{c,d} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{p}_i \in \mathbb{R}^2, \text{ so that}$$

$$\dot{\boldsymbol{x}} = \dot{\boldsymbol{p}}_{c,d} + \boldsymbol{K}\left(\boldsymbol{p}_{c,d} - \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{p}_i\right) \in \mathbb{R}^2$$

where $K > 0$ is a suitable positive gain,[9] $\boldsymbol{p}_{c,d}$ and $\dot{\boldsymbol{p}}_{c,d}$ are the reference position and velocity of the centroid, respectively. The resulting $2 \times n$ Jacobian is f.r.r. and independent from $\boldsymbol{q}$, with singular values $\sigma_1 = \sigma_2 = \frac{1}{\sqrt{N}}$. However, to increase the robustness with respect to damping, a scaled task $\sqrt{N}\boldsymbol{x}$ is here considered. Small singular values related to f.r.r. Jacobians, in fact, may unnecessarily constraint the damping threshold ϵ.

C. Radial Distance from the Centroid (Platoon on a Circle)

Let $\boldsymbol{p}_m = \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{p}_j$ be the current position of the centroid. Each robot i may be asked to lay on a circle of radius r by imposing the following task [27]:

$$x_i = \frac{1}{2}\left\|\boldsymbol{p}_i - \boldsymbol{p}_m\right\|^2 \in \mathbb{R}, \text{ and}$$

$$\dot{x}_i = K\left(\frac{r^2}{2} - x_i\right) \in \mathbb{R}. \tag{12}$$

Consequently, the collective task rate is written as $\dot{\boldsymbol{x}} = (\dot{x}_1^T, \ldots, \dot{x}_N^T)^T \in \mathbb{R}^N$ and $\boldsymbol{J} \in \mathbb{R}^{N \times n}$ can be computed by differentiation. In particular, the resulting Jacobian is singular for some configurations (e.g., $\exists\, \boldsymbol{p}_i = \boldsymbol{p}_m$, all the vehicles are along a line, or $N = 2$).

D. Escape Space Minimization (Perimeter)

In an escorting mission, the fleet may be required to maintain a target in the centroid of the formation, while minimizing the intruding/escaping space. For this purpose, the fleet should lay on a N-vertex polygon, and the task can be formalized as in [7], [27] as

$$x = \frac{1}{2}\left[\left\|\boldsymbol{p}_1 - \boldsymbol{p}_N\right\|^2 + \sum_{i=2}^{N}\left\|\boldsymbol{p}_i - \boldsymbol{p}_{i-1}\right\|^2\right]$$

$$\boldsymbol{J} = \left(\ldots\left(\left(\boldsymbol{p}_i - \boldsymbol{p}_{i+1}\right)^T + \left(\boldsymbol{p}_i - \boldsymbol{p}_{i-1}\right)^T\right)\ldots\right)$$

$$\dot{x} = K\left(\frac{Nl^2}{2} - x\right) \tag{13}$$

where $l = 2\,r\cos(\frac{\pi}{2} - \frac{\pi}{N})$ is the distance between consecutive vertices of the polygon and r is the radius of the circle circumscribed about it.

E. Remarks.

1) iCAT RP Conservative + Collision Avoidance as Primary Task: All the collision avoidance constraints are assumed to be active when $i = 1$ and the projector $\tilde{\boldsymbol{P}}_2$ is computed. Since the hypotheses required by Theorem IV.3.b. may not be satisfied, a further analysis is required. Low-priority tasks are merged in an optimal way according to the task hierarchy as in the original reverse priority controller. Then, if $Np \geq n \Rightarrow \tilde{\boldsymbol{P}}_2 = \boldsymbol{I}$. Hence, the resulting control law is suboptimal, but under A1, $\boldsymbol{x}_1$ is never corrupted by algorithmic singularities (see Section VI-C for further details).

2) Circular + Perimeter: An algorithmic singularity occurs when the tasks defined in (12) and (13) are jointly satisfied. However, a similar behavior is obtained replacing the circular task with an appropriate convex cost function $\mathcal{H}(\boldsymbol{q})$ to minimize applying, for example, the projected gradient method. In particular, we consider

$$\mathcal{H}(\boldsymbol{q}) = \sum_{i=1}^{N}\left\|r^2 - \left\|\boldsymbol{p}_i - \boldsymbol{p}_m\right\|^2\right\|^2 \tag{14}$$

$$\dot{\boldsymbol{q}}_{0_L} = \alpha\boldsymbol{\nabla}_q\mathcal{H}(q) \tag{15}$$

where α is the optimization step updated with a backtracking line search.

[8] Similar expressions can be derived for control objectives $x_{i,j} \leq x_{i,j}^M$ and $x_{i,j}^m \leq x_{i,j} \leq x_{i,j}^M$; see [1], [16], [17] for details. Also, velocity considerations can be included with reverse controllers to avoid useless activations of unilateral constraints as described in [18].

[9] For simplicity, a constant gain has been used. However, many strategies can be used to obtain numerically stable execution and/or good tracking of the desired trajectory.

Authorized licensed use limited to: Orebro University. Downloaded on September 28,2021 at 14:05:40 UTC from IEEE Xplore. Restrictions apply.

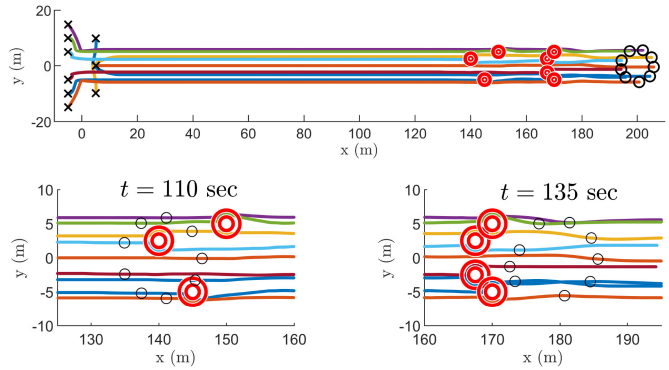


Fig. 4. Tracking the reference trajectory $\boldsymbol{p}_{c,d}$ while maintaining a desired formation and avoiding obstacles (red circles).

TABLE III
PERFORMANCE INDICES

Collision avoidance (m): $\zeta_i = \sum_{i=1}^{N} \max\left\{0,\ d_m - \left\|\boldsymbol{p}_i - \boldsymbol{p}_{o_j}\right\|\right\}$.
Centroid (m): $\zeta_i = \left\|\boldsymbol{p}_{c,d} - \boldsymbol{p}_m\right\|$.
Circular (m^2): $\zeta_i = \frac{1}{2}\sum_{i=1}^{N}\left\|r^2 - \left\|\boldsymbol{p}_i - \boldsymbol{p}_m\right\|^2\right\|$.
Perimeter (m^2): $\zeta_i = \frac{1}{2}\left\|Nl^2 - \left\|\boldsymbol{p}_1 - \boldsymbol{p}_N\right\|^2 - \sum_{i=2}^{N}\left\|\boldsymbol{p}_i - \boldsymbol{p}_{i-1}\right\|^2\right\|$.

VI. SIMULATIONS

The nine homogeneous holonomic vehicles[10] scenario proposed in [7] is used as a common test case and it is reproduced for reader convenience in Fig. 4. The centroid of the formation is required to follow a linear trajectory from a starting position $\boldsymbol{p}_{c,d}(0) = (0 \quad 0)$ to a final position $\boldsymbol{p}_{c,d}(180) = (200 \quad 0)$. The trajectory is computed using a fifth-order interpolating polynomial law, considering $\dot{\boldsymbol{p}}_{c,d}(t) = \boldsymbol{0}$, $\ddot{\boldsymbol{p}}_{c,d}(t) = \boldsymbol{0}$ when $t = 0$ or $t > 180$ s. During the mission, the vehicles may be required to stay on a circle, to minimize the escape space and/or to avoid some static obstacles, which poses are assumed to be known (see Section V). Two concentric red circles are used in Figs. 4, 6, and nine to display each obstacle, with the internal and the external circles representing the obstacle itself and the activation disk, respectively.

All the experiments are implemented in MATLAB R2018a and have been run on an Intel Core i7-5500 U CPU @ 2.40 GHz $\times$ 4 processor. Different control laws $\mathcal{C}_*$ are compared,[11] by defining scalar performance indices $\{\zeta_i\}$ (see Table III) so that for each $\boldsymbol{x}_i$, $\zeta_i \to 0$ iff $\boldsymbol{x}_i \to \boldsymbol{x}_{d,i}$. Statistical indices reported in Tables IV–VII are evaluated over the time horizon $t \in [0, 200]$.

A. Non Time-Varying Mission (Without Obstacles): Centroid + Circular + Perimeter

The controllers are first compared considering non-time-varying activations. The set of tasks satisfies the hypotheses for the simultaneous asymptotic stability of the three task's error [27], i.e., for each $i \in \{1, 2, 3\}$ $\boldsymbol{e}_i \to \boldsymbol{0}$ (exponentially) with a speed depending on gains $\{\boldsymbol{K}_i\}$. Small damping values are required for the augmented Jacobian-based controllers to handle the algorithmic singularity that may arise when the

[10] We remark that the approach is general with respect to the fleet composition since it only requires control affine kinematics (see Section II).

[11] As in [16], we use a common set of parameters, that is, we fix task errors characteristics (e.g., the speed in which they converge to 0 and the maximum error rates due to damping actions) when tasks are separately executed. The question of how to compute the optimal parameters (e.g., see [37] for SR controllers) when tasks are merged together is interesting, but beyond the scope of this work which, indeed, focuses on comparing structural properties—specifically, γ and k_f are set according to [11] and [22], respectively.

TABLE IV
CENTROID + CIRCULAR + PERIMETER

Method		ζ_1	ζ_2	ζ_3
All the controllers	max	1.10e-3	6.00	1.25e-12
	mean	6.51e-4	2.88	1.78e-13
	std	3.84e-2	2.21	1.51e-13

Order of tasks: 1. Centroid, 2. Circular, 3. Perimeter.
Initial configuration: optimal (i.e., $\boldsymbol{e}_i(t=0) = \boldsymbol{0}$ for all the tasks).
Parameters: $\boldsymbol{K}_1 = \text{diag}(0.8)$, $\boldsymbol{K}_2 = \text{diag}(0.4)$, $K_3 = 0.5$, $T_c = 0.05$ sec, $\epsilon = 0.1$, $\lambda^2_{\max} = 0.1$, $r = 10$ m, $\gamma = 1$, $k_f = 0$.

TABLE V
COLLISION AVOIDANCE + CENTROID

Method		ζ_1	ζ_2
Standard Approach, iCAT TPC, iCAT RP	max	0	1.10e-3
	mean	0	6.51e-4
	std	0	3.87e-4
Singularity Robust	max	0	8.97e-2
	mean	0	3.00e-3
	std	0	9.90e-3
iCAT SR, iCAT RP conservative	max	0	5.92e-2
	mean	0	2.70e-3
	std	0	7.90e-3

Order of tasks: 1. Collision avoidance, 2. Centroid.
Initial configuration: optimal (i.e., $\boldsymbol{e}_i(t=0) = \boldsymbol{0}$ for all the active tasks).
Parameters: $\boldsymbol{K}_1 = \text{diag}(0.8)$, $\boldsymbol{K}_2 = \text{diag}(0.8)$, $d_m = 1$ m, $\beta_m = 1$ m, $T_c = 0.05$ s, $\epsilon = 0.1$, $\lambda^2_{\max} = 0.1$, $r = 10$ m, $\gamma = 1$, $k_f = 0$.
Results are practically comparable when $\epsilon = 1, \lambda^2_{\max} = 1$.

TABLE VI
COLLISION AVOIDANCE + CENTROID + PERIMETER + COST FUNCTION FOR CIRCULAR

Method		ζ_2	ζ_3	ζ_4
Singularity Robust	max	5.68e-1	31.2	51.4
	mean	4.00e-2	2.14	7.58
	std	1.00e-1	5.52	8.90
Reverse Priority	max	1.10e-3	3.71e-1	22.9
	mean	5.92e-4	3.16e-2	6.23
	std	4.13e-4	6.85e-2	4.35
iCAT TPC	max	1.10e-3	3.72e-1	20.9
	mean	5.87e-4	2.98e-2	5.79
	std	4.10e-4	6.74e-2	3.75
iCAT SR, iCAT RP conserv.	max	5.66e-1	27.3	45.7
	mean	3.55e-2	1.82	6.73
	std	9.17e-2	4.66	7.08
iCAT RP	max	1.10e-3	3.69e-1	22.8
	mean	5.92e.4	3.00e-2	6.13
	std	4.13-4	6.66e-2	4.25

Order of tasks: 1. Collision avoidance. 2. Centroid. 3. Perimeter.
Initial configuration: not optimal (see Fig. 4); however, the initial transient has not been considered when performance indices have been computed.
Parameters: $\boldsymbol{K}_1 = \text{diag}(0.8)$, $\boldsymbol{K}_2 = \text{diag}(0.8)$, $K_3 = 0.5$, $d_m = 1$ m, $\beta_m = 1$ m, $T_c = 0.05$ s, $\epsilon = 0.1$, $\lambda^2_{\max} = 0.1$, $r = 6$ m, $\gamma = 1$, $k_f = 0$.
The cost function (14) is used to specify the requirement of laying on a circle. Let ζ_4 be the corresponding performance index definition according to Table III. Results of the standard approach, which is not able to find a feasible solution (see also Figs. 11 and 12), are not displayed. All the other controllers satisfy the safety constraints in this simulation.

Authorized licensed use limited to: Orebro University. Downloaded on September 28,2021 at 14:05:40 UTC from IEEE Xplore. Restrictions apply.

TABLE VII
COLLISION AVOIDANCE + CENTROID + CIRCULAR + PERIMETER

Method		ζ_2	ζ_3	ζ_4
Singularity Robust	max	4.67e-1	44.1	31.2
	mean	3.03e-2	4.69	1.15
	std	7.92e-2	8.21	5.29
Reverse Priority	max	9.84e-2	8.14	1.92
	mean	5.7e-3	1.82	1.47e-1
	std	1.38e-2	1.69	2.91e-1
iCAT TPC	max	1.10e-3	59.8	179.2
	mean	5.88e-4	4.89	9.56
	std	4.11e-4	9.89	28.3
iCAT SR	max	4.67e.1	44.1	31.2
	mean	3.03e-2	4.69	3.15
	std	7.92e-3	8.21	5.29
iCAT RP cons.	max	4.74e.1	33.8	22.2
	mean	2.05e-2	3.63	1.20
	std	6.11e-3	5.79	3.21
iCAT RP	max	4.49e-2	7.03	1.55
	mean	2.1e-3	1.73	9.32e-2
	std	5.5e-3	1.57	2.24e-1

Order of tasks: 1. Collision avoidance. 2. Centroid. 3. Circular. 4. Perimeter.
Initial configuration: not optimal to show differences between the iCAT RP conservative approach and the iCAT SR (the initial transient has not been considered for computing performance indices).
Parameters: $\boldsymbol{K}_1 = \text{diag}(0.8)$, $\boldsymbol{K}_2 = \text{diag}(0.8)$, $\boldsymbol{K}_3 = \text{diag}(0.4)$, $K_4 = 0.5$, $d_m = 1$ m, $\beta_m = 1$ m, $T_c = 0.05$ s, $\epsilon = 1, \lambda^2_{\max} = 1, r = 6$ m, $\gamma = 1, k_f = 0$.
High damping is required to handle singularities. Results of the standard approach, which is not able to find a feasible solution, are not displayed.

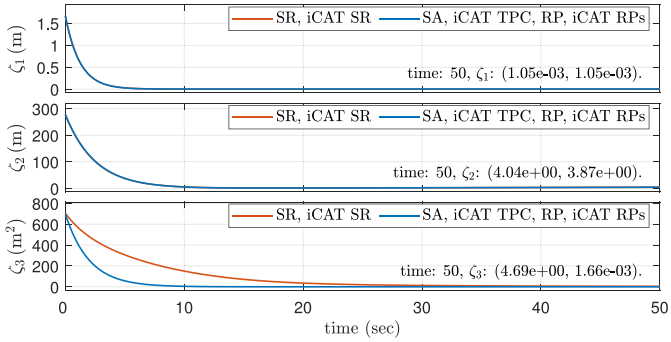


Fig. 5. Centroid + Circular + Perimeter. Performance indices $\zeta_1, \zeta_2, \zeta_3$ when starting from the not optimal configuration displayed in Fig. 4. Parameters are listed in Table IV.

desired formation is reached. Performances when starting from an optimal and a not optimal configuration are shown in Table IV and Fig. 5, respectively.

B. A Simple Time-Varying Mission: Collision Avoidance + Centroid

With reference to the scenario reported in Fig. 4, the fleet is now required to maintain its centroid along a desired trajectory while avoiding some obstacles. Parameters and results of the test are listed in Table V. For simplicity, results are clustered in three groups with similar performance. It is worth noting that even though the SA performs well, it does not always provide a feasible solution as reported next.

To show an interesting case, input parameters are selected so that at least one robot will be subject to two safety constraints at the same time, that is, $\boldsymbol{J}_1$ does not satisfy Remark 1(a). As expected, the controllers $\mathcal{C}_* = (L, \bar{\square}, \{\boldsymbol{I}\}, \{\boldsymbol{A}_i\}, \{\square^{\#,d}\}, \{\boldsymbol{\Psi}_i\})$ are not able to preserve the continuity of the vector $\dot{\boldsymbol{q}}$ when small damping values are used (see Figs. 6 and 7). Specifically, discontinuities may happen:

1) **[SA]** whenever a robot enters/exits a safety disc (e.g., at $t = 107$ s in Fig. 7). At such times, in fact, the projector $\boldsymbol{P}_{A,1} = \boldsymbol{I} - \bar{\boldsymbol{J}}_1^{\#}\bar{\boldsymbol{J}}_1$ is not continuous. As shown in Fig. 6, SA does not preserve continuity even if $\boldsymbol{J}_1$ verifies Remark 1(a), i.e., when each robot is subject to at most one safety constraint at the same time.
2) **[SR, RP]** when a robot, which was previously avoiding two obstacles, exits one safety disc (e.g., at $t = 134$ s in Fig. 7). At such times, in fact, the task manifold $\mathcal{S}(\mathrm{k})$ may change without continuity, since the robot can now account for the lower priority task (tracking the centroid). However, in this specific case, reverse approaches preserve the continuity when Remark 1(a) holds (see Fig. 6).

On the contrary, iCAT-based extensions provide continuous smoother solutions as it can be seen from Fig. 7. When increasing the damping parameters, discontinuities are smoothed, as it can be appreciated in Fig. 8, but task errors may increase and collisions may happen since high damping may corrupt the solution. Results with k_f set as suggested in [18] and [22] are shown in Fig. 9 (high damping) and Fig. 10 (small damping). However, even if continuity of velocities is preserved, the minimum norm solution ($k_f = 0$) seems to provide better results in this case.

C. A Complex Time-Varying Mission: Collision Avoidance + Centroid + Perimeter + Circular

The aim of this test is to show how the tasks' modeling may jeopardize performance of different approaches. This may occur in case of the Jacobian full row rankness loss even in case of high damping parameters (see Remark 4). For this purpose, the fleet is now simultaneously subject to all the tasks presented on Section V. First, in Table VI, the requirement of laying on a circle is not specified as a task, while the cost function in (14) and the projected gradient method are used for this purpose. All the controllers (with the exception of SA, which is not able to find a feasible solution) do not violate the collision avoidance constraints (Fig. 11). Note that, also in this case, iCAT-based controllers are able to remove the discontinuities due to multiple activation (see Fig. 12 at $t = 124$ s).

On the contrary, in Table VII, the requirement of laying on a circle is now considered as a task and the mission is performed using (12) instead of (14). It is worth noting that, in this case, the augmented Jacobian is not f.r.r. when the fleet is laying on the regular polygon (see Section V-E2). Consequently, high damping values are required to tackle with the related algorithmic singularity and this may lead to a corruption of higher priority tasks (e.g, collisions as shown in Fig. 13).

Authorized licensed use limited to: Orebro University. Downloaded on September 28,2021 at 14:05:40 UTC from IEEE Xplore. Restrictions apply.

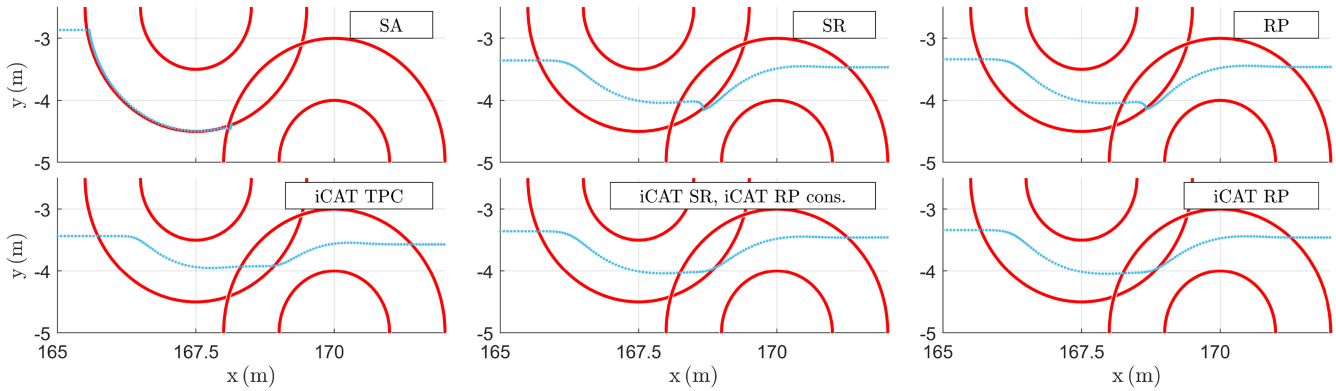


Fig. 6. Coll. avoidance + Centroid. Detail on the two bottom obstacles in Fig. 4. Trajectories with $k_f = 0$ (minimum norm solution) and small damping ($\epsilon = 0.1$, $\lambda^2_{max} = 0.1$).

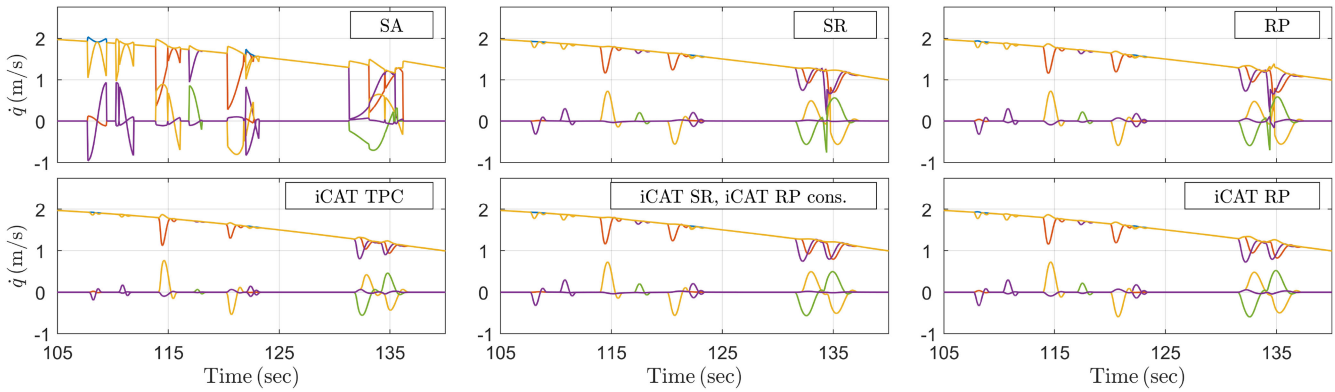


Fig. 7. Coll. avoidance + Centroid. Velocities with $k_f = 0$ (minimum norm solution) and small damping ($\epsilon = 0.1$, $\lambda^2_{\max} = 0.1$).

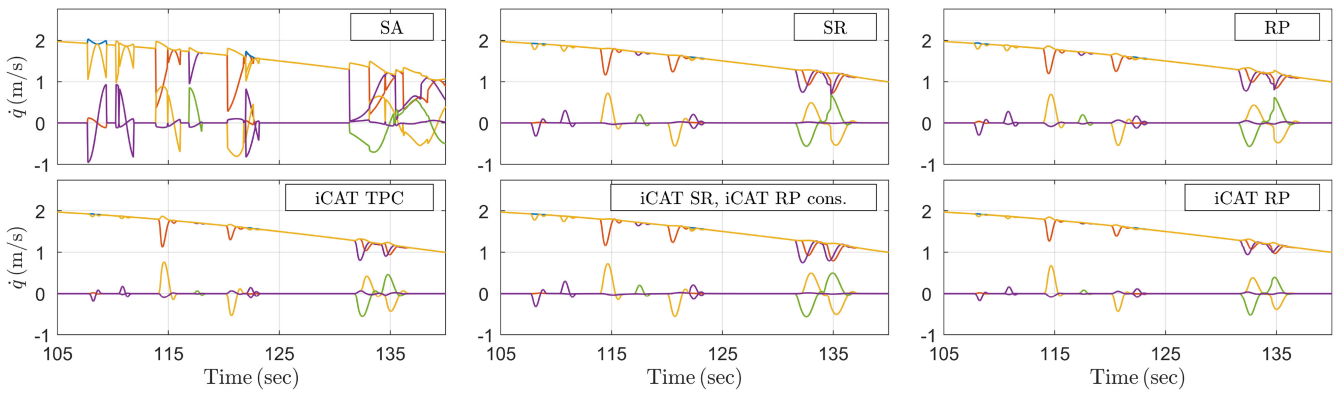


Fig. 8. Coll. avoidance + Centroid. Velocities with $k_f = 0$ (minimum norm solution) and high damping ($\epsilon = 1$, $\lambda^2_{\max} = 1$).

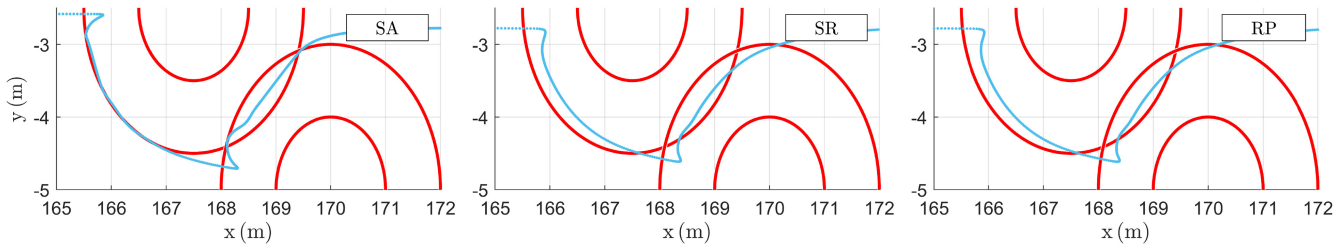


Fig. 9. Coll. avoidance + Centroid. Detail on the two bottom obstacles in Fig. 4. Trajectories with $k_f = 1 - \frac{T_c}{2}$ [18] and high damping ($\epsilon = 1$, $\lambda^2_{\max} = 1$).

Authorized licensed use limited to: Orebro University. Downloaded on September 28,2021 at 14:05:40 UTC from IEEE Xplore. Restrictions apply.

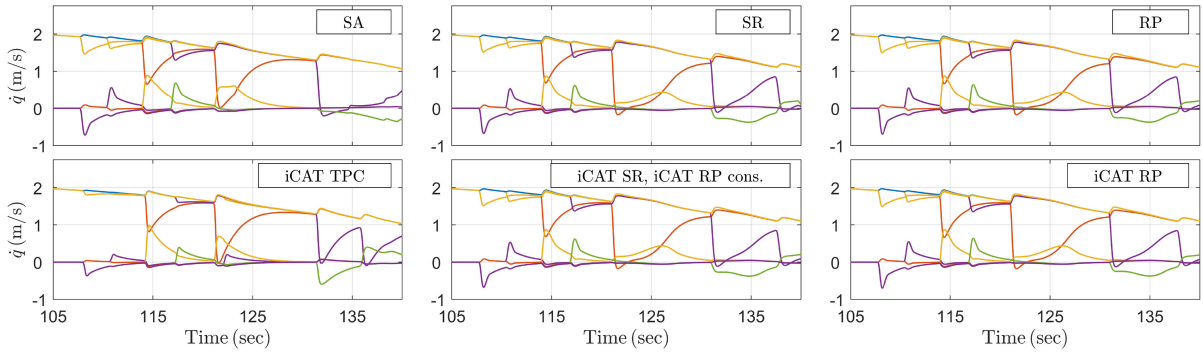


Fig. 10. COLL. AVOIDANCE + CENTROID. Velocities with $k_f = 1 - \frac{T_c}{2}$ and low damping ($\epsilon = 0.1$, $\lambda^2_{\max} = 0.1$).

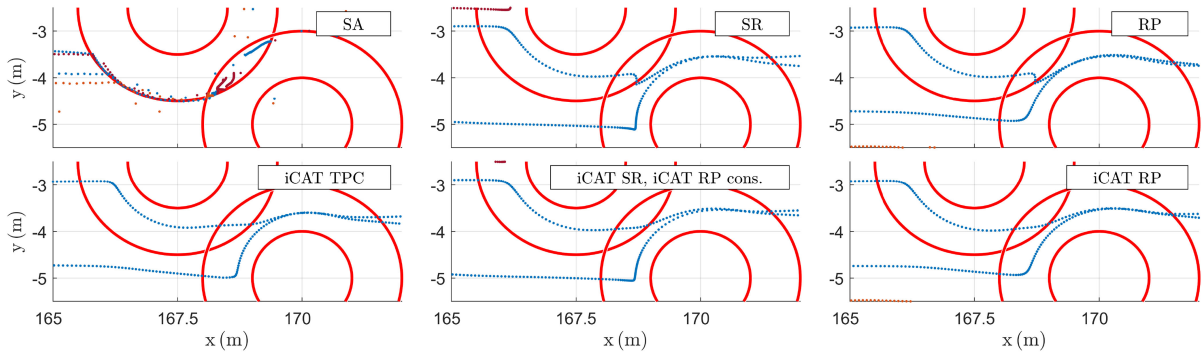


Fig. 11. COLL. AVOIDANCE + CENTROID + PERIMETER + MINIM. $\mathcal{H}(q)$. Detail on the two bottom obstacles in Fig. 4. Trajectories for the case described in Table VI.

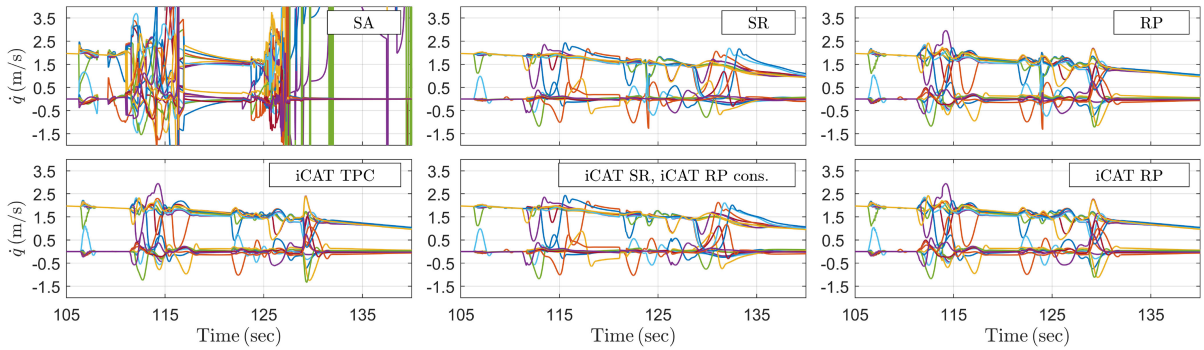


Fig. 12. COLL. AVOIDANCE + CENTROID + CIRCULAR + MINIM. $\mathcal{H}(q)$. Velocities.

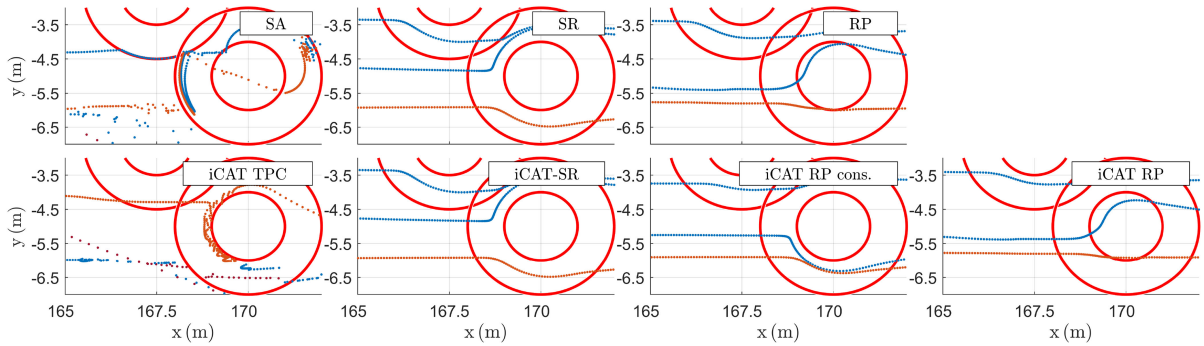


Fig. 13. Coll. avoidance + Centroid + Circular + Perimeter. Detail on the two bottom obstacles in Fig. 4. Trajectories for the case described in Table VII. The set of tasks is not f.r.r. SA and iCAT TPC methods are not able to deal with singularities, even with high damping parameters. Collisions may happen when using the RP and the iCAT RP, since high damping parameters may corrupt the computation of matrices $\{\boldsymbol{T}_i\}$.

Authorized licensed use limited to: Orebro University. Downloaded on September 28,2021 at 14:05:40 UTC from IEEE Xplore. Restrictions apply.

VII. Conclusion

In this article, we investigated the efficacy of using NSB-IK techniques, such as the standard approach, the SR method, and the RP algorithm. We proposed an iCAT extension to deal with time-varying activation of tasks, for the kinematic control of multivehicle systems subject to prioritized cartesian tasks. These methods exploit null space projectors to ensure the correct task hierarchy (i.e., high priority tasks never being corrupted by lower priority ones). All the NSB controllers were integrated here in a novel unified formulation, which allowed to easily relate their properties to the choice of a minimal set of key features. The set of features was given by the following: i) the priority order (ascendant or descendent) to merge tasks; ii) a rows selector defined according to activations; iii) weighting matrices for both projectors and task error rates; iv) the pseudoinverse operator, and v) the null space projection matrices. Eliciting strengths and weakness of each NSB approach via formal analysis or from the literature, we analyzed how different feature choices may impact on optimality, correctness of the hierarchy, robustness with respect to both kinematic and algorithmic singularities, and continuity of controls. In this article, we showed how different feature combinations produced different control laws with different performance on tasks. Moreover, leveraging upon the regularized pseudoinverse operator of the iCAT TPC technique, we proposed extensions to handle time-varying activation of multidimensional tasks while ensuring continuity of controls within the SR (i.e., the iCAT SR approach) and the RP approaches (i.e., the iCAT RP approaches).

A theoretical analysis of iCAT RP methods formally proved the following properties. Active tasks were optimally executed if linearly independent to all the other tasks which are active or in activation. In such cases, the continuity in controls and the correct task hierarchy were also preserved. However, when tasks were linearly dependent, weighted null spaces (which are a key tool to ensure continuity) coupled performances of linearly dependent tasks to activations (i.e., the error due to the higher priority constraint was null only if all the same or lower dependent constraints were active). As a result, dependent tasks and high damping parameters may prevent high-priority linearly dependent tasks to be correctly executed. The questions of how to enforce robustness to singularities while ensuring continuity and how to determine optimal damping parameters are still open and left for future work. Note that this is a current limitation of all Jacobian augmented-based methods with time-varying activations (i.e., [1], [11], [19] and the iCAT RPs approaches proposed in this article).

Simulations confirmed the effectiveness of the proposed approaches when the aforementioned conditions were satisfied, but also their limitations when relaxing these assumptions, enforcing the importance not only of a correct modeling of tasks, but also of supervising the set of active tasks which were concurrently executed.

Further extensions of this work will be also devoted to apply the theoretical results of this article for designing a more specific NSB-based control architecture for multirobot fleet management. The focus was on a centralized control design, to illustrate the key points of the method. A decentralized architecture for efficient control of large fleets and second-order dynamics to deal with kinodynamic constraints and force inputs will be investigated as a generalization of this work.

Appendix

A. Greville's Formula for Partitioned Matrices

For convenience, Corollary 1 of [28], p. 51 is here recalled in Theorem VII.1 and it will be extended in Corollaries VII.1.1 and VII.1.2 for providing general properties of pseudoinverses of not f.r.r. matrices.

Theorem VII.1 (from [28] pag. 51): Let $\boldsymbol{J}$ be any real $m \times n$ matrix, partitioned as $\boldsymbol{J} = \left[\boldsymbol{R}^T, \boldsymbol{S}^T\right]^T$, where $\boldsymbol{R} \in \mathbb{R}^{m_r \times n}$, $\boldsymbol{S} \in \mathbb{R}^{m_s \times n}$ and $m = m_r + m_s$; then

$$\boldsymbol{J}^{\#} = \begin{bmatrix} \boldsymbol{H} & \boldsymbol{T} \end{bmatrix} \tag{16}$$

where

$$\begin{aligned}
\boldsymbol{H} &= \boldsymbol{R}^{\#} - \boldsymbol{T}\boldsymbol{S}\boldsymbol{R}^{\#} \\
\boldsymbol{T} &= \boldsymbol{E}^{\#} + (\boldsymbol{I} - \boldsymbol{E}^{\#}\boldsymbol{S})\boldsymbol{R}^{\#}\boldsymbol{R}^{\#,T}\boldsymbol{S}^T\boldsymbol{K}(\boldsymbol{I} - \boldsymbol{E}\boldsymbol{E}^{\#}) \\
\boldsymbol{E} &= \boldsymbol{S}(\boldsymbol{I} - \boldsymbol{R}^{\#}\boldsymbol{R}) \\
\boldsymbol{K} &= [\boldsymbol{I} + (\boldsymbol{I} - \boldsymbol{E}\boldsymbol{E}^{\#})\boldsymbol{S}\boldsymbol{R}^{\#}\boldsymbol{R}^{\#,T}\boldsymbol{S}^T(\boldsymbol{I} - \boldsymbol{E}\boldsymbol{E}^{\#})]^{-1}.
\end{aligned}$$

If $\exists \boldsymbol{\chi} : \boldsymbol{S} = \boldsymbol{\chi}\boldsymbol{R}$, then we can state the following results.

Corollary VII.1.1: $\exists \boldsymbol{W} \neq \boldsymbol{0} : \boldsymbol{J}^{\#} = \boldsymbol{R}^{\#}\boldsymbol{W}$ and

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{I} - \boldsymbol{\eta}\boldsymbol{\chi}\boldsymbol{R}\boldsymbol{R}^{\#} & \boldsymbol{\eta} \end{bmatrix} \tag{17}$$

$$\boldsymbol{\eta} = \boldsymbol{\chi}^T\boldsymbol{K}. \tag{18}$$

Proof: From Theorem VII.1, we have that $\boldsymbol{E} = \boldsymbol{\chi}\boldsymbol{R}(\boldsymbol{I} - \boldsymbol{R}^{\#}\boldsymbol{R}) \Rightarrow \boldsymbol{E} = \boldsymbol{0}$. Consequently,

$$\begin{aligned}
\boldsymbol{K} &= [\boldsymbol{I} + \boldsymbol{S}\boldsymbol{R}^{\#}\boldsymbol{R}^{\#,T}\boldsymbol{S}^T]^{-1} = [\boldsymbol{I} + \boldsymbol{\chi}\boldsymbol{R}\boldsymbol{R}^{\#}\boldsymbol{\chi}^T]^{-1} \\
\boldsymbol{T} &= \boldsymbol{R}^{\#}\boldsymbol{R}^{\#,T}\boldsymbol{S}^T\boldsymbol{K} = \boldsymbol{R}^{\#}\boldsymbol{\chi}^T\boldsymbol{K} = \boldsymbol{R}^{\#}\boldsymbol{\eta}
\end{aligned}$$

since $\boldsymbol{R}\boldsymbol{R}^{\#}$ is symmetric and idempotent and $\boldsymbol{R}^{\#}\boldsymbol{R}\boldsymbol{R}^{\#} = \boldsymbol{R}^{\#}$. Hence, (16) can be computed as

$$\begin{aligned}
\boldsymbol{J}^{\#} &= \begin{bmatrix} \boldsymbol{R}^{\#} - \boldsymbol{R}^{\#}\boldsymbol{\eta}\boldsymbol{\chi}\boldsymbol{R}\boldsymbol{R}^{\#} & \boldsymbol{R}^{\#}\boldsymbol{\eta} \end{bmatrix} \\
&= \boldsymbol{R}^{\#}\begin{bmatrix} \boldsymbol{I} - \boldsymbol{\eta}\boldsymbol{\chi}\boldsymbol{R}\boldsymbol{R}^{\#} & \boldsymbol{\eta} \end{bmatrix}.
\end{aligned}$$

■

Corollary VII.1.2: Let $\boldsymbol{A} = \mathrm{diag}(\boldsymbol{A}_R, \boldsymbol{A}_S) \in \mathbb{R}^{m \times n}$, where $\boldsymbol{A}_R \in \mathbb{R}^{m_r \times m_r}$ and $\boldsymbol{A}_S \in \mathbb{R}^{m_s \times m_s}$ are diagonal matrices whose elements are defined in the interval [0,1]. If $\boldsymbol{R}$ is f.r.r., then

$$\boldsymbol{J}(\boldsymbol{I} - \boldsymbol{J}^{\#}\boldsymbol{A}\boldsymbol{J}) = \boldsymbol{0} \iff \begin{array}{l} \boldsymbol{A} = \boldsymbol{I} \text{ or} \\ (\boldsymbol{A}_R = \boldsymbol{I} \text{ and } \boldsymbol{\chi} = \boldsymbol{0}) \end{array}. \tag{19}$$

Proof: Sufficiency.: It can be easily verified since $\boldsymbol{J} = \boldsymbol{J}\boldsymbol{J}^{\#}\boldsymbol{J}$.

Authorized licensed use limited to: Orebro University. Downloaded on September 28,2021 at 14:05:40 UTC from IEEE Xplore. Restrictions apply.

Necessity.: Let $\boldsymbol{W}$ be partitioned as $[\boldsymbol{W}_R \;\; \boldsymbol{W}_S] = [\boldsymbol{I} - \boldsymbol{\eta\chi} \;\; \boldsymbol{\chi}]$. Then, the left-hand side of (19) can be rewritten as

$$\boldsymbol{J}(\boldsymbol{I} - \boldsymbol{J}^{\#}\boldsymbol{A}\boldsymbol{J}) = \begin{pmatrix} \boldsymbol{I} \\ \boldsymbol{\chi} \end{pmatrix} \left[\boldsymbol{I} - \boldsymbol{W} \begin{pmatrix} \boldsymbol{A}_R \\ \boldsymbol{A}_S\boldsymbol{\chi} \end{pmatrix} \right] \boldsymbol{R}$$

so

$$(19) \text{ holds } \iff \begin{cases} (\boldsymbol{I} - \boldsymbol{W}_R\boldsymbol{A}_R - \boldsymbol{W}_S\boldsymbol{A}_S\boldsymbol{\chi})\,\boldsymbol{R} = \boldsymbol{0} \\ \boldsymbol{\chi}\,(\boldsymbol{I} - \boldsymbol{W}_R\boldsymbol{A}_R - \boldsymbol{W}_S\boldsymbol{A}_S\boldsymbol{\chi})\,\boldsymbol{R} = \boldsymbol{0}. \end{cases}$$

which is verified only if

$$\begin{aligned} &\boldsymbol{I} - \boldsymbol{W}_R\boldsymbol{A}_R - \boldsymbol{W}_S\boldsymbol{A}_S\boldsymbol{\chi} = \boldsymbol{0} \\ &\iff (\boldsymbol{I} - \boldsymbol{\eta\chi})\boldsymbol{A}_R = \boldsymbol{I} - \boldsymbol{\eta}\boldsymbol{A}_S\boldsymbol{\chi}. \end{aligned} \tag{20}$$

since $\boldsymbol{R} \neq \boldsymbol{0}$. Also, we have that $\boldsymbol{\chi}^T\boldsymbol{\chi} = (\boldsymbol{I} + \boldsymbol{\chi}^T\boldsymbol{\chi})\boldsymbol{\eta\chi}$, so (20) can be rewritten as $\boldsymbol{A}_R = \boldsymbol{I} + \boldsymbol{\chi}^T(\boldsymbol{I} - \boldsymbol{A}_S)\boldsymbol{\chi}$.

Since $\boldsymbol{A}_R$ and $\boldsymbol{A}_S$ are diagonal matrices with values in [0,1], then $\boldsymbol{\chi}^T(\boldsymbol{I} - \boldsymbol{A}_S)\boldsymbol{\chi}$ is a positive matrix. Hence, (19) admits only the set of solutions ($\boldsymbol{A}_R = \boldsymbol{I}$ and $\boldsymbol{A}_S = \boldsymbol{I}$) or ($\boldsymbol{\chi} = \boldsymbol{0}$ and $\boldsymbol{A}_R = \boldsymbol{I}$). ■

B. Discontinuities of $\mathcal{C}_* = (L, \bar{\square}, \{\boldsymbol{I}\}, \{\boldsymbol{A}_i\}, \{\square^{\#}\}, \{\boldsymbol{\Psi}_i\})$

Proof sketch of Remark 1 (a) Required to ensure the continuity of $\boldsymbol{J}_i^{\#}$ (see [30]).

(b) Required to prevent jumps of the optimal solutions due to changes of $\mathcal{S}(\mathrm{k})$ [17], [18]. In particular, assuming (a):

1) SA: $\boldsymbol{P}_{A,i-1} = \boldsymbol{I} - \bar{\boldsymbol{J}}_{A,i-1}^{\#}\bar{\boldsymbol{J}}_{A,i-1}$, so discontinuities are prevented for all $\{(\boldsymbol{x}_{d,i}, \dot{\boldsymbol{x}}_{d,i}), \boldsymbol{A}_i\}$ only if $\bar{\boldsymbol{J}}_i\boldsymbol{P}_{A,i-1} = \bar{\boldsymbol{J}}_i$, that is, if $\boldsymbol{J}_i\boldsymbol{J}_j^T = \boldsymbol{0}$ for each $i, j < i$, i.e., if (b) holds.
2) SR: Each $[\Pi_{j=1}^{i-1}(\boldsymbol{I} - \bar{\boldsymbol{J}}_j^{\#}\bar{\boldsymbol{A}}_j\bar{\boldsymbol{J}}_j)]\bar{\boldsymbol{J}}_i^{\#}\bar{\boldsymbol{A}}_i\dot{\boldsymbol{x}}_i$ is continuous with activation, so (b) is not required.
3) RP: $\tilde{\boldsymbol{P}}_{i+1} = \boldsymbol{I} - \tilde{\boldsymbol{J}}_{i+1}^{\#}\tilde{\boldsymbol{J}}_{i+1}$, with $\tilde{\boldsymbol{J}}_{i+1}$ computed from $\bar{\boldsymbol{J}}_{RA,i}$ (see Table II). Thus, discontinuities are removed by construction only if $\boldsymbol{J}_i\tilde{\boldsymbol{P}}_{i+1} = \boldsymbol{J}_i$, i.e., if $\boldsymbol{J}_i\boldsymbol{J}_j^T = \boldsymbol{0}$ for each $i, j > i$, i.e., if (b) holds. ■

C. iCAT RP

Assumptions: As highlighted in Section IV-B, formal properties of this section are proved considering the definition of regularized pseudoinverse $\square^{\boldsymbol{I},\boldsymbol{A},\boldsymbol{Q}}$ according to (7) (without the SVO contribution).

Preliminaries: According to [16], the matrix $\tilde{\boldsymbol{J}}_{i+1}$ collects the rows of $\boldsymbol{J}_{RA,i+1}$ that are linearly independent from $\boldsymbol{J}_i$, so let $\boldsymbol{\Pi}_{i+1}$ be an appropriate permutation matrix such that $\boldsymbol{\Pi}_{i+1}\boldsymbol{J}_{RA,i+1} = \begin{pmatrix} \tilde{\boldsymbol{J}}_{i+1} \end{pmatrix}$. Also, as introduced to Section IV-B, let $\boldsymbol{\mathcal{A}}_i = \mathrm{diag}(\boldsymbol{A}_i, \boldsymbol{\mathcal{A}}_{i+1})$ be a recursive formulation of the augmented activation matrix, i.e., the activation matrix related to the reverse augmented Jacobian $\boldsymbol{J}_{RA,i}$. Specifically, we define

$$\boldsymbol{\mathcal{A}}_{\boldsymbol{\Pi}_{i+1}} := \mathrm{diag}(\tilde{\boldsymbol{\mathcal{A}}}_{i+1}, *) = \boldsymbol{\Pi}_{i+1}\boldsymbol{\mathcal{A}}_{i+1}{\boldsymbol{\Pi}_{i+1}}^T \tag{21}$$

so that

$$\boldsymbol{\Pi}_{i+1}\boldsymbol{\mathcal{A}}_{i+1}\boldsymbol{J}_{RA,i+1} = \boldsymbol{\mathcal{A}}_{\boldsymbol{\Pi}_{i+1}} \begin{pmatrix} \tilde{\boldsymbol{J}}_{i+1} \\ \hat{\boldsymbol{J}}_{i+1} \end{pmatrix}. \tag{22}$$

Moreover, according to the Greville's formula (Appendix VIII-A), the pseudoinverse of the matrix $\boldsymbol{J}_{RA,i}$ can be written as $\boldsymbol{J}_{RA,i}^{\#} = \begin{bmatrix} \boldsymbol{T}_i & \boldsymbol{H}_i \end{bmatrix}$, where $\boldsymbol{H}_i = \boldsymbol{J}_{RA,i+1}^{\#} - \boldsymbol{T}_i\boldsymbol{J}_i\boldsymbol{J}_{RA,i+1}^{\#}$. From [28], p. 19, the following equality holds:

$$\begin{pmatrix} \tilde{\boldsymbol{J}}_{i+1}^{\#} & \boldsymbol{0} \end{pmatrix} = \left(\boldsymbol{I} - \boldsymbol{T}_i\boldsymbol{T}_i^{\#}\right)\boldsymbol{H}_i\boldsymbol{\Pi}_{i+1}^T. \tag{23}$$

D. Iterative Formulation of the Projector in (9)

Let $\boldsymbol{Q}_i$ be the projector in the weighted null space of the reverse Jacobian $\boldsymbol{J}_{RA,i}$, defined as

$$\boldsymbol{Q}_i = \boldsymbol{I} - \boldsymbol{J}_{RA,i}^{\#}\boldsymbol{\mathcal{A}}_i\boldsymbol{J}_{RA,i}.$$

Proposition VII.1.1: A recursive formulation for (9) is given by

$$\tilde{\boldsymbol{P}}_{i+1} = \left(\boldsymbol{I} - \boldsymbol{T}_i{\boldsymbol{T}_i}^{\#}\right)\boldsymbol{Q}_{i+1} + \boldsymbol{T}_i{\boldsymbol{T}_i}^{\#} \tag{24}$$

$$\boldsymbol{Q}_i = (\boldsymbol{I} - \boldsymbol{T}_i\boldsymbol{J}_i)\,\boldsymbol{Q}_{i+1} + \boldsymbol{T}_i\,(\boldsymbol{I} - \boldsymbol{A}_i)\,\boldsymbol{J}_i. \tag{25}$$

Proof: Considering (22) and (23), (24) is proven as follows:

$$\begin{aligned} \tilde{\boldsymbol{P}}_{i+1} &= \boldsymbol{I} - \begin{bmatrix} \tilde{\boldsymbol{J}}_{i+1}^{\#} & \boldsymbol{0} \end{bmatrix} \boldsymbol{\mathcal{A}}_{\boldsymbol{\Pi}_{i+1}} \begin{pmatrix} \tilde{\boldsymbol{J}}_{i+1} \\ * \end{pmatrix} \\ &= \boldsymbol{I} - \left(\boldsymbol{I} - \boldsymbol{T}_i{\boldsymbol{T}_i}^{\#}\right)\boldsymbol{H}_i\boldsymbol{\mathcal{A}}_{i+1}\boldsymbol{J}_{RA,i+1} \\ &= \boldsymbol{I} - (\boldsymbol{I} - \boldsymbol{T}_i{\boldsymbol{T}_i}^{\#})(\boldsymbol{I} - \boldsymbol{T}_i\boldsymbol{J}_i)\boldsymbol{J}_{RA,i+1}^{\#}\boldsymbol{\mathcal{A}}_{i+1}\boldsymbol{J}_{RA,i+1} \\ &= (\boldsymbol{I} - \boldsymbol{T}_i{\boldsymbol{T}_i}^{\#})(\boldsymbol{I} - \boldsymbol{J}_{RA,i+1}^{\#}\boldsymbol{\mathcal{A}}_{i+1}\boldsymbol{J}_{RA,i+1}) + \boldsymbol{T}_i{\boldsymbol{T}_i}^{\#} \\ &= \left(\boldsymbol{I} - \boldsymbol{T}_i{\boldsymbol{T}_i}^{\#}\right)\boldsymbol{Q}_{i+1} + \boldsymbol{T}_i{\boldsymbol{T}_i}^{\#}. \end{aligned}$$

Using the Greville's formula and a bit of math, it is possible to prove the iterative formulation of the projector given in (25) as follows:

$$\begin{aligned} \boldsymbol{Q}_i &= \boldsymbol{I} - \boldsymbol{J}_{RA,i}^{\#}\boldsymbol{\mathcal{A}}_i\boldsymbol{J}_{RA,i} \\ &= \boldsymbol{I} - \left[\boldsymbol{T}_i,\; (\boldsymbol{I} - \boldsymbol{T}_i\boldsymbol{J}_i)\,\boldsymbol{J}_{RA,i+1}^{\#}\right]\boldsymbol{\mathcal{A}}_i \begin{pmatrix} \boldsymbol{J}_i \\ \boldsymbol{J}_{RA,i+1} \end{pmatrix} \\ &= (\boldsymbol{I} - \boldsymbol{T}_i\boldsymbol{J}_i)\,\boldsymbol{Q}_{i+1} + \boldsymbol{T}_i\,(\boldsymbol{I} - \boldsymbol{A}_i)\,\boldsymbol{J}_i. \end{aligned}$$

■

E. Iterative Formulation of the Projector in (10)

Let $\boldsymbol{Q}_i^r$ be the regularized projector in the weighted null space of the reverse augmented Jacobian, defined as

$$\boldsymbol{Q}_i^r = \boldsymbol{I} - \boldsymbol{J}_{RA,i}^{\#,\boldsymbol{\mathcal{A}}_i,\boldsymbol{I}}\boldsymbol{J}_{RA,i}.$$

Proposition VII.1.2: A recursive formulation for (10) is given by

$$\begin{aligned} &\tilde{\boldsymbol{P}}_{i+1} = \left(\boldsymbol{I} - \boldsymbol{T}_i{\boldsymbol{T}_i}^{\#}\right)\boldsymbol{Q}_i^r + \boldsymbol{T}_i\boldsymbol{T}_i^{\#} \\ &\text{where } \boldsymbol{T}_i = \left(\sqrt{\boldsymbol{\mathcal{A}}_i}\boldsymbol{J}_{RA,i}\right)^{\#}_{\{:,1:m_i\}}. \end{aligned} \tag{26}$$

Proof: The proof is based on the definition of the regularized operator given in (7), which automatically removes the contribution of the inactive rows from the computation of the

Authorized licensed use limited to: Orebro University. Downloaded on September 28,2021 at 14:05:40 UTC from IEEE Xplore. Restrictions apply.

pseudoinverse. However, the original formulation of the Greville's formula should be modified opportunely to care with the weighting matrix $\sqrt{\boldsymbol{A}}$. Without loosing generality, the reverse ordering of indices is maintained. Let $\boldsymbol{A} = \text{diag}(\boldsymbol{A}_i, \boldsymbol{A}_{i+1})$, $\boldsymbol{J} = \begin{pmatrix} \boldsymbol{J}_i \\ \boldsymbol{J}_{i+1} \end{pmatrix}$, so that

$$\begin{aligned}\left(\sqrt{\boldsymbol{A}}\boldsymbol{J}\right)^{\#} &= \begin{pmatrix} \sqrt{\boldsymbol{A}_i}\boldsymbol{J}_i \\ \sqrt{\boldsymbol{A}_{i+1}}\boldsymbol{J}_{i+1} \end{pmatrix}^{\#} \\ &= \begin{bmatrix} \boldsymbol{T}_i & \left(\boldsymbol{I} - \boldsymbol{T}_i\sqrt{\boldsymbol{A}_i}\boldsymbol{J}_i\right)\boldsymbol{H}_i \end{bmatrix} \end{aligned} \tag{27}$$

where $\boldsymbol{H}_i = (\sqrt{\boldsymbol{A}_{i+1}}\boldsymbol{J}_{i+1})^{\#}$. Hence, (7) is equivalent to

$$\begin{aligned}\boldsymbol{J}^{\#,\boldsymbol{A},\boldsymbol{I}} &= \left(\sqrt{\boldsymbol{A}}\boldsymbol{J}\right)^{\#}\sqrt{\boldsymbol{A}}\boldsymbol{A} \\ &= \begin{bmatrix} \boldsymbol{T}_i\sqrt{\boldsymbol{A}_i} & \left(\boldsymbol{I} - \boldsymbol{T}_i\sqrt{\boldsymbol{A}_i}\boldsymbol{J}_i\right)\boldsymbol{H}_i\sqrt{\boldsymbol{A}_{i+1}} \end{bmatrix}\boldsymbol{A}.\end{aligned}$$

Then, comparing the matrices

$$\boldsymbol{J}^{\#}\boldsymbol{A}\boldsymbol{J} = \begin{bmatrix} \boldsymbol{T}_i & \left(\boldsymbol{I} - \boldsymbol{T}_i\boldsymbol{J}_i\right)\boldsymbol{J}_{i+1}^{\#} \end{bmatrix}\boldsymbol{A}\boldsymbol{J} \tag{28}$$

$$\boldsymbol{J}^{\#,\boldsymbol{A},\boldsymbol{I}}\boldsymbol{J} = \begin{bmatrix} \boldsymbol{T}_i\sqrt{\boldsymbol{A}_i} & \left(\boldsymbol{I} - \boldsymbol{T}_i\sqrt{\boldsymbol{A}_i}\boldsymbol{J}_i\right)\boldsymbol{H}_i\sqrt{\boldsymbol{A}_{i+1}} \end{bmatrix}\boldsymbol{A}\boldsymbol{J} \tag{29}$$

it can be noticed that they have a similar structure, but the role of the matrix $\boldsymbol{T}_i$ in the first equation is played by the matrix $\boldsymbol{B}_i = \boldsymbol{T}_i\sqrt{\boldsymbol{A}_i}$ in the second one. The application of the previous result to (10) is then straightforward, leading to the following formulation of projector:

$$\begin{aligned}\tilde{\boldsymbol{P}}_{i+1} &= \left(\boldsymbol{I} - \boldsymbol{B}_i\boldsymbol{B}_i{}^{\#}\right)\boldsymbol{Q}_{i+1}^{r} + \boldsymbol{B}_i\boldsymbol{B}_i^{\#} \\ &= \left(\boldsymbol{I} - \boldsymbol{T}_i\boldsymbol{T}_i{}^{\#}\right)\boldsymbol{Q}_{i+1}^{r} + \boldsymbol{T}_i\boldsymbol{T}_i^{\#}\end{aligned}$$

where $\boldsymbol{B}_i = \boldsymbol{T}_i\sqrt{\boldsymbol{A}}_i = (\boldsymbol{J}_{RA,i}^{\#,\boldsymbol{\mathcal{A}}_i,\boldsymbol{I}}\boldsymbol{\mathcal{A}}_i^{\#})_{\{:,1:m_i\}}$ and equivalence holds since the factor $\sqrt{\boldsymbol{A}_i}$ get simplified in the product $\boldsymbol{B}_i\boldsymbol{B}_i^{\#}$.

However, for the sake of completeness, the proof is given in the following:

$$\begin{aligned}\tilde{\boldsymbol{P}}_{i+1} &= \boldsymbol{I} - \tilde{\boldsymbol{J}}_{i+1}^{\#,\boldsymbol{\mathcal{A}}_{i+1},\boldsymbol{I}}\tilde{\boldsymbol{J}}_{i+1} \\ &= \boldsymbol{I} - \left(\tilde{\boldsymbol{\mathcal{A}}}_{i+1}^{\frac{1}{2}}\tilde{\boldsymbol{J}}_{i+1}\right)^{\#}\left(\tilde{\boldsymbol{\mathcal{A}}}_{i+1}\right)^{\frac{3}{2}}\tilde{\boldsymbol{J}}_{i+1} \\ &= \boldsymbol{I} - \begin{bmatrix} \left(\tilde{\boldsymbol{\mathcal{A}}}_{i+1}^{\frac{1}{2}}\tilde{\boldsymbol{J}}_{i+1}\right)^{\#} & \boldsymbol{0} \end{bmatrix}\left(\boldsymbol{\mathcal{A}}_{\boldsymbol{\Pi}_{i+1}}\right)^{\frac{3}{2}}\begin{pmatrix} \tilde{\boldsymbol{J}}_{i+1} \\ \hat{\boldsymbol{J}}_{i+1} \end{pmatrix}\end{aligned}$$

then, using (21) and (22), we have that

$$\tilde{\boldsymbol{P}}_{i+1} = \boldsymbol{I} - \begin{bmatrix} \left(\tilde{\boldsymbol{\mathcal{A}}}_{i+1}^{\frac{1}{2}}\tilde{\boldsymbol{J}}_{i+1}\right)^{\#} & \boldsymbol{0} \end{bmatrix}\boldsymbol{\Pi}_{i+1}\left(\boldsymbol{\mathcal{A}}_{i+1}\right)^{\frac{3}{2}}\boldsymbol{J}_{RA,i+1}.$$

Then, (23) is applied to the matrix $(\sqrt{\boldsymbol{\mathcal{A}}_i}\boldsymbol{J}_{RA,i})$, leading to $\boldsymbol{H}_i = (\sqrt{\boldsymbol{\mathcal{A}}_{i+1}}\boldsymbol{J}_{RA,i+1})^{\#}$ and $\boldsymbol{T}_i = (\sqrt{\boldsymbol{\mathcal{A}}_i}\boldsymbol{J}_{RA,i})^{\#}{}_{\{:,1:m_i\}}$. Considering that

$$\begin{bmatrix} \left(\tilde{\boldsymbol{\mathcal{A}}}_{i+1}^{\frac{1}{2}}\tilde{\boldsymbol{J}}_{i+1}\right)^{\#} & \boldsymbol{0} \end{bmatrix} = \left(\boldsymbol{I} - \boldsymbol{T}_i\boldsymbol{T}_i^{\#}\right)\boldsymbol{H}_i\boldsymbol{\Pi}_{i+1}^{T}$$

the following formulation of the projector is obtained:

$$\begin{aligned}\tilde{\boldsymbol{P}}_{i+1} &= \boldsymbol{I} - \left(\boldsymbol{I} - \boldsymbol{T}_i\boldsymbol{T}_i^{\#}\right)\left(\boldsymbol{\mathcal{A}}_{i+1}^{\frac{1}{2}}\boldsymbol{J}_{RA,i+1}\right)^{\#}\boldsymbol{\mathcal{A}}_{i+1}^{\frac{3}{2}}\boldsymbol{J}_{RA,i+1} = \\ &= \boldsymbol{I} - \left(\boldsymbol{I} - \boldsymbol{T}_i\boldsymbol{T}_i^{\#}\right)\boldsymbol{J}_{RA,i+1}^{\#,\boldsymbol{\mathcal{A}}_{i+1},\boldsymbol{I}}\boldsymbol{J}_{RA,i+1} = \\ &= \left(\boldsymbol{I} - \boldsymbol{T}_i\boldsymbol{T}_i^{\#}\right)\boldsymbol{Q}_{i+1}^{r} + \boldsymbol{T}_i\boldsymbol{T}_i^{\#}.\end{aligned} \tag{30}$$

Finally, similarly to (25), a recursive formulation of the regularized null space of the reverse augmented Jacobian can be computed as follows (pedices have been omitted for the purpose of readability):

$$\begin{aligned}\boldsymbol{Q}_i^{r} &= \boldsymbol{I} - \begin{bmatrix} \boldsymbol{B}_i\boldsymbol{A}_i & \left(\boldsymbol{I} - \boldsymbol{B}_i\boldsymbol{J}_i\right)\boldsymbol{J}_{i+1}^{\#,\boldsymbol{\mathcal{A}}_{i+1},\boldsymbol{I}} \end{bmatrix}\begin{pmatrix} \boldsymbol{J}_i \\ \boldsymbol{J}_{i+1} \end{pmatrix} \\ &= \left(\boldsymbol{I} - \boldsymbol{B}_i\boldsymbol{J}_i\right)\boldsymbol{Q}_{i+1}^{r} + \boldsymbol{B}_i\left(\boldsymbol{I} - \boldsymbol{A}_i\right)\boldsymbol{J}_i\end{aligned}$$

which applied to (30) leads to the formulation of $\tilde{\boldsymbol{P}}_{i+1}$ given in (26), since

$$\begin{aligned}\left(\boldsymbol{I} - \boldsymbol{B}_i\boldsymbol{B}_i{}^{\#}\right)\left(\boldsymbol{I} - \boldsymbol{B}_i\boldsymbol{J}_i\right) &= \left(\boldsymbol{I} - \boldsymbol{B}_i\boldsymbol{B}_i{}^{\#}\right) \\ \left(\boldsymbol{I} - \boldsymbol{B}_i\boldsymbol{B}_i{}^{\#}\right)\boldsymbol{B}_i &= \boldsymbol{0}.\end{aligned}$$

■

REFERENCES

[1] 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. 1–4, pp. 877–902, 2016.

[2] R. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Autom.*, vol. 2, no. 1, pp. 14–23, Mar. 1986.

[3] R. C. Arkin, "Motor schema-based mobile robot navigation," *Int. J. Robot. Res.*, vol. 8, no. 4, pp. 92–112, 1989.

[4] R. C. Arkin," *Behavior-Based Robot.*, Cambridge, MA, USA: MIT Press, 1998.

[5] B. E. Bishop, "On the use of redundant manipulator techniques for control of platoons of cooperating robotic vehicles," *IEEE Trans. Syst. Man Cybern. A Syst. Humans*, vol. 33, no. 5, pp. 608–615, Sep. 2003.

[6] D. J. Stilwell, B. E. Bishop, and C. A. Sylvester, "Redundant manipulator techniques for partially decentralized path planning and control of a platoon of autonomous vehicles," *IEEE Trans. Syst. Man Cybern. B Cybern.*, vol. 35, no. 4, pp. 842–848, Aug. 2005.

[7] G. Antonelli and S. Chiaverini, "Kinematic control of platoons of autonomous vehicles," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1285–1292, Dec. 2006.

[8] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The null-space-based behavioral control for autonomous robotic systems," *Intell. Service Robot.*, vol. 1, no. 1, pp. 27–39, 2008.

[9] T. Fabbri, E. Simetti, G. Casalino, L. Pallottino, and A. Caiti, "Distributed task-priority based control in area coverage & adaptive sampling," in *Proc. OCEANS 2017-Aberdeen*, Scotland, IEEE, 2017, pp. 1–8.

[10] A. Marino, L. E. Parker, G. Antonelli, and F. Caccavale, "A decentralized architecture for multi-robot systems based on the null-space-behavioral control with application to multi-robot border patrolling," *J. Intell. Robot. Syst.*, vol. 71, no. 3–4, pp. 423–444, 2013.

[11] E. Simetti and G. Casalino, "Manipulation and transportation with cooperative underwater vehicle manipulator systems," *IEEE J. Ocean. Eng.*, vol. 42, no. 4, pp. 782–799, Oct. 2017.

[12] G. Casalino, E. Simetti, F. Wanderlingh, K. Darvish, B. Bruno, and F. Mastrogiovanni, "On autonomous robotic cooperation capabilities within factory and logistic scenarios," *Procedia Manuf.*, vol. 11, pp. 147–163, 2017.

[13] B. E. Bishop, "Formation control of underactuated autonomous surface vessels using redundant manipulator analogs," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 4892–4897.

Authorized licensed use limited to: Orebro University. Downloaded on September 28,2021 at 14:05:40 UTC from IEEE Xplore. Restrictions apply.

[14] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Proc. 5th Int. Conf. Adv. Robot.*, 1991, vol. 2, pp. 1211–1216.
[15] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. Robot. Autom.*, vol. 13, no. 3, pp. 398–410, Jun. 1997.
[16] F. Flacco and A. De Luca, "A reverse priority approach to multi-task control of redundant robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 2421–2427.
[17] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Trans. Robot.*, vol. 25 no. 3, pp. 670–685, Jun. 2009.
[18] F. Flacco and A. De Luca, "Unilateral constraints in the reverse priority redundancy resolution method," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 2564–2571.
[19] S. Moe, G. Antonelli, A. R. Teel, K. Y. 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 Robot. AI*, vol. 3, p. 16, 2016.
[20] P. Di Lillo, F. Arrichiello, G. Antonelli, and S. Chiaverini, "Safety-related tasks within the set-based task-priority inverse kinematics framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 6130–6135.
[21] A. A. Maciejewski and C. A. Klein, "Numerical filtering for the operation of robotic manipulators through kinematically singular configurations," *J. Field Robot.*, vol. 5, no. 6, pp. 527–552, 1988.
[22] F. Flacco and A. De Luca, "Discrete-time velocity control of redundant robots with acceleration/torque optimization properties," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 5139–5144.
[23] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. Auton. Robot Vehicles*, Springer, 1986, pp. 396–404.
[24] P. Baerlocher and R. Boulic, "Task-priority formulations for the kinematic control of highly redundant articulated structures," in *Proc. Int. Conf. Intell. Robots Syst.*, 1998, vol. 1, pp. 323–329.
[25] A. Ben-Israel and T. NE Greville, *Generalized Inverses: Theory and Applications*, vol. 15. Springer Science & Business Media, 2003.
[26] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Intell. J. Robot. Res.*, vol. 4, no. 3, pp. 109–117, 1985.
[27] G. Antonelli, F. Arrichiello, and S. Chiaverini, "Stability analysis for the null-space-based behavioral control for multi-robot systems," in *Proc. 47th IEEE Conf. Decis. Control*, 2008, pp. 2463–2468.
[28] T. L. Boullion and P. L. Odell, *Generalized Inverse Matrices*. Wiley, 1971.
[29] G. Antonelli, F. Arrichiello, and S. Chiaverini, "Experimental kinematic comparison of behavioral approaches for mobile robots," *IFAC Proc. Volumes*, vol. 38, no. 1, pp. 295–300, 2005.
[30] N. Mansard, A. Remazeilles, and F. Chaumette, "Continuity of varying-feature-set control laws," *IEEE Trans. Autom. Control*, vol. 54, no. 11, pp. 2493–2505, Nov. 2009.
[31] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *J. Dyn. Syst. Meas. Control*, vol. 108, no. 3, pp. 163–171, 1986.
[32] A. S Deo and I. D Walker, "Robot subtask performance with singularity robustness using optimal damped least-squares." in *Proc. Proc. IEEE Int. Conf. Robot. Autom.*, 1992, pp. 434–441.
[33] F. Arrichiello, A. Marino, and F. Pierri, "Distributed fault-tolerant control for networked robots in the presence of recoverable/unrecoverable faults and reactive behaviors," *Frontiers Robot. AI*, vol. 4, 2017, Art. no. 2, doi: 10.3389/frobt.2017.00002.
[34] D. Bareiss and J. van den Berg, "Generalized reciprocal collision avoidance," *Intell. J. Robot. Res.*, vol. 34, no. 12, pp. 1501–1514, 2015.
[35] F. Flacco, A. De Luca, and O. Khatib, "Motion control of redundant robots under joint constraints: Saturation in the null space," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 285–292.
[36] A. Dietrich, T. Wimbock, A. Albu-Schaffer, and G. Hirzinger, "Integration of reactive, torque-based self-collision avoidance into a task hierarchy," *IEEE Trans. Robot.*, vol. 28, no. 6, pp. 1278–1293, Dec. 2012.
[37] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems," *IEEE Trans. Robot.*, vol. 25, no. 5, pp. 985–994, Oct. 2009.

Anna Mannucci received the bachelor's degree in electronic, the master's and Ph.D. degrees in robotics and automation from the University of Pisa, Pisa, Italy, in 2013, 2016, and 2020, respectively.

She is currently a Research Assistant with the Multi-Robot Planning and Control Laboratory, Örebro University, Örebro, Sweden. She is one of the principal developers and maintainers of the coordination_oru library, a general-purpose tool for multi-vehicle coordination. The library is used in the ILIAD European project and in industrial R&D by third parties such as Volvo, Scania, and Epiroc. Her research interests focus on integrated task allocation, coordination, motion planning, and control algorithms for single and multirobot systems.

Danilo Caporale was born in Lanciano, Italy, in 1987. He received the B.S. and M.S. degrees in automation and control engineering in 2009 and 2011, respectively, and the Ph.D. degree in information technology, with a thesis on robust control devices for railway traction and braking, all from the Politecnico di Milano, Milan, Italy.

Since 2015, he has been a Postdoc Researcher with Centro di Ricerca E. Piaggio, Universita'di Pisa, Pisa, Italy. His research interests include motion planning and control, with applications to vehicle, biological and humanoid robotic systems.

Lucia Pallottino received the "Laurea" degree in mathematics and the Ph.D. degree in robotics and industrial automation from the University of Pisa, Pisa, Italy, in 1998 and 2002, respectively.

She is currently an Associate Professor with the Centro di Ricerca "E. Piaggio" and the Dipartimento di Ingegneria dell'Informazione, University of Pisa, Pisa, Italy. She is the Deputy Director of Centro di Ricerca "E. Piaggio". Her main research interests within robotics include motion planning and optimal control of multirobot systems and coordination of multirobot vehicles.

Dr. Pallottino has been an Associate Editor for the IEEE ROBOTICS AND AUTOMATION LETTERS since 2017 and was an Associate Editor for the IEEE TRANSACTION ON ROBOTICS from 2014 to 2017.

Authorized licensed use limited to: Orebro University. Downloaded on September 28,2021 at 14:05:40 UTC from IEEE Xplore. Restrictions apply.