

© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Hierarchical Tracking Control with Arbitrary Task Dimensions: Application to Trajectory Tracking on Submanifolds

Gianluca Garofalo and Christian Ott

Abstract—Hierarchical impedance control has been recently shown to effectively allow trajectory tracking, while guaranteeing the order of priorities during the execution. Nevertheless, the choice of the tasks is required to be such that, after being properly decoupled, they are all feasible and lead to an invertible Jacobian matrix. In this work, a modification is proposed that removes both these restrictions. The user is free to specify as many tasks as desired and especially without necessarily guaranteeing in advance that none of the tasks will become singular during the execution. Whenever tasks with higher priority use-up all the degrees of freedom, all the other tasks are naturally ignored. Still, as soon as some of the tasks with higher priority become singular, then the freed-up controllability is used to execute the next task in the stack. This is realized automatically, without any rearrangement of the tasks in the priority stack. As an application, the case of trajectory tracking on a submanifold of the workspace is considered, in which multiple charts of the atlas are used for the tasks. Simulations are used to validate the stability analysis.

I. INTRODUCTION

Due to the increasing number of tasks that robots are capable to perform, it is natural to require that more of them have to be executed simultaneously, see [1]–[13] and the references therein. When two or more tasks are in conflict, organizing them in a hierarchic stack allows to follow the order of priority in the execution. Due to its importance, several works can be found in the literature on this topic. As a first classification, one can distinguish between approaches that use soft and strict priorities. In the first case, the subtasks are weighted and the corresponding control actions are combined accordingly. Numerical optimization is often a good ally in these cases and in general when inequality constraints are also required to be formulated [1]–[3]. Additionally, the modification of the weights results in a smooth change in the task priorities [4]–[6]. Soft priorities can lead to tasks competing with each other independently of their priority level. This issue does not arise with strict priorities. Additionally, the latter methods are typically formulated in a way that allows to perform a stability analysis of the closed-loop system. A feature very important, particularly in human-robot interaction scenarios. Common methods to realize the hierarchical control with strict priorities use both feedback linearization and passivity based techniques [7]–[13]. By dynamically decoupling the tasks, stable tracking of the desired trajectory can be guaranteed. A key concept in the realization of multiple tasks is therefore the null

The authors are with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany. gianluca.garofalo(at)dlr.de

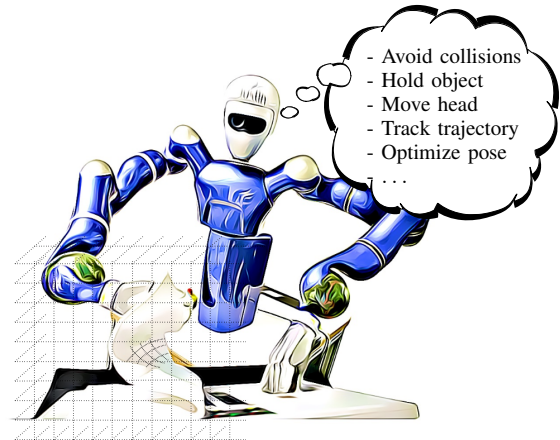


Fig. 1. The robot has to perform multiple tasks simultaneously, one of them is the tracking of a trajectory on a submanifold of its workspace (see the gridlines).

space projection [14], [15]. The projection of each task in the null space of the previous ones, i.e., those to execute with a higher priority, guarantees that the more important tasks are not disturbed by the less important ones. A typical example is that safety-critical objectives should never be compromised by other subtasks. While feedback linearization is historically the most prominent method thanks to its simplicity in the design and analysis [8], [9], passivity-based methods have few important features that are worth considering [11]–[13]. Namely, reduced compensation of terms in the dynamic model due to the preservation of the natural inertia of the robot, and realization of a compliant behavior without requiring measurements and feedback of the external interactions.

In this work, the hierarchical impedance-based tracking control recently developed in [13] is extended both from a practical and theoretical point of view. Therefore, the presented results contribute to the category of strict priority controllers via passivity-based methods. In particular, the contributions of this work can be summarized as follows. Firstly, the restrictions on the tasks assumed in [13] are removed. Unlike in [13], the user is free to specify as many tasks as desired and especially without necessarily guaranteeing in advance that none of the tasks will become singular during the execution. Whenever tasks with higher priority use-up all the degrees of freedom, all the other tasks are naturally ignored. Still, when some of the tasks with higher priority become singular, then the freed-up controllability is used to execute the next task in the stack. This is realized

automatically, without any rearrangement of the tasks in the priority stack. Secondly, the control law is reformulated in the spirit of the results presented in [16]. This is important for three reasons: it suggests possible extensions of this work by including adaptive and robust techniques, it considerably simplifies the stability analysis of the closed-loop system and, most importantly, it allows to show the much stronger property of uniform global asymptotic stability, rather than only uniform local asymptotic stability.

The paper is organized as follows. Section II recalls some useful definitions and introduces the notation. Section III provides a brief description of the considered systems and defines the control objective. In Section IV the reader can find the proposed control design. Subsequently, the main results of this work are summarized as theorems in Section V and then their proof with the stability analysis is given in Section VI. To validate the results, Section VII presents two simulations that show a particular interesting application. The possibility of having an arbitrary number of tasks is used to globally track a trajectory on a submanifold, by using multiple charts of the maximal atlas for different tasks. Trajectory tracking on a submanifold of the workspace, as sketched in Fig. 1, handling of singularities and orientation tracking on $SO(3)$ are considered. Finally, Section VIII summarizes the work and points to possible future extensions.

II. PRELIMINARIES

The usual Euclidean norm is denoted by $|\cdot|$ and $A \succ 0$ ($A \succeq 0$, respectively) denotes that the matrix A is positive definite (semi-definite, respectively).

Local and global asymptotic stability are denoted in short as LAS and GAS, respectively. The letter U in the acronyms denotes uniformity, i.e., independence from the initial time t_0 in case of a time-varying system. Finally, ISS denotes input-to-state stability. When not specifically stated, these properties are always meant to hold for the origin of the state space.

A. Weighted Moore-Penrose inverse

Given a matrix $A \in \mathbb{R}^{m \times n}$ and two symmetric positive definite matrices $W_1 \in \mathbb{R}^{m \times m}$ and $W_2 \in \mathbb{R}^{n \times n}$, the following problem for an unknown matrix $X \in \mathbb{R}^{n \times m}$:

$$AXA = A \quad (1a)$$

$$XAX = X \quad (1b)$$

$$(W_1AX)^\top = W_1AX \quad (1c)$$

$$(W_2XA)^\top = W_2XA \quad (1d)$$

has a unique solution $X = A_{W_1, W_2}^\dagger$ called the weighted Moore–Penrose inverse¹ (WMPI) of A [17], [18]. One can compute A_{W_1, W_2}^\dagger from A^\dagger , i.e., from the classic Moore–Penrose inverse (MPI) of A and the Cholesky decomposition² of the weight matrices $W_1 = R_1^\top R_1$ and

$W_2 = R_2^\top R_2$. Notice that A is not assumed to have full rank and therefore A_{W_1, W_2}^\dagger cannot be computed as a left or right weighted inverse. Using the defining properties of the MPI, one can easily show by direct computation that

$$A_{W_1, W_2}^\dagger = R_2^{-1} (R_1 A R_2^{-1})^\dagger R_1 \quad (2)$$

satisfies (1).

In this work, the simplified notation A^+ will be used when W_1 and W_2 are assumed to be the identity matrix and the inertia matrix (see next section), respectively.

III. ROBOT MODEL AND CONTROL OBJECTIVE

The considered fully-actuated robotic system is modeled by the nonlinear differential equations:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau, \quad (3)$$

where the state of the robot is given by generalized positions and velocities $q, \dot{q} \in \mathbb{R}^n$, n being the number of degrees of freedom (DoF). The dynamic matrices are the symmetric and positive definite inertia matrix $M \in \mathbb{R}^{n \times n}$, a Coriolis matrix $C \in \mathbb{R}^{n \times n}$ satisfying the passivity property $\dot{M} = C + C^\top$ and the gravity torque vector $g \in \mathbb{R}^n$. Finally, the control input $\tau \in \mathbb{R}^n$ is realized through the motors of the robot. Due to the use of rotational joints or prismatic joints with end-stops, the configuration space of the robot is bounded. This implies the boundedness of the eigenvalues of M .

A. Task specifications

Assume that the user provides a finite number of tasks with a level of priority. Each task is assumed to be expressed in coordinates as

$$x_i = f_i(q) \quad (4a)$$

$$\dot{x}_i = J_i(q)\dot{q}, \quad (4b)$$

for $i \in \{1, \dots, r\}$ and $f_i: \mathbb{R}^n \rightarrow \mathbb{R}^{m_i}$. The tasks are organized in a priority stack such that x_i has a higher priority than x_j , for all $i < j$. The total task dimension m is given by $m = \sum_{i=1}^r m_i$. The whole stack will be denoted by suppressing the subscript and organized in a column array, e.g., $x = [x_1^\top \dots x_r^\top]^\top \in \mathbb{R}^m$.

Assumption 1: The total task dimension is not less than the DoF of the robot, i.e., $m \geq n$.

Notice that compared to [13], the total task dimension is not forced to be exactly n . Actually, the use of a joint-space configuration as last task, i.e., $x_r = q$, is recommended to avoid that at any time the stack of tasks leaves some DoF uncontrolled. In this case, the total task dimension is clearly greater than n . The considered scenario is also more general than [11], where the condition $\sum_{i=1}^{r-1} m_i \leq n$ and $m_r \geq 0$ was used. Additionally, in [11] only the regulation case was addressed and a minimal set of velocities was extracted in order to obtain a square Jacobian matrix, which was further assumed to be invertible. On the contrary, no restriction to a singularity-free workspace is required in this work and therefore $J_i(q)$ does not need to be a full-rank matrix.

¹In the more general setting the transpose is replaced with the conjugate transpose.

²Once again, only Cholesky decomposition of real matrices are considered here.

B. Control objective

For the i -th task a desired trajectory $x_{i,d}(t) \in \mathbb{R}^{m_i}$ is provided, together with its first and second time derivatives, i.e., $\dot{x}_{i,d}(t)$ and $\ddot{x}_{i,d}(t)$. As for the task coordinates, the stack of desired trajectories will be denoted by omitting the subscript, e.g., $x_d \in \mathbb{R}^m$. The desired values are required to be compatible³, meaning that it exists⁴ a smooth, bounded joint-space trajectory $q_d(t)$ that simultaneously satisfies all the task trajectories x_d , i.e., $x(q_d) = x_d$. A controller has to be designed such that the closed-loop system has an asymptotically stable equilibrium point in $(\tilde{x}, \dot{\tilde{x}}) = (0, 0)$, where $\tilde{x} = x - x_d$.

Loosely speaking, the stack of tasks completely defines the behavior of the robot and obviously it would be impossible to perform a stack of tasks that would require more DoFs than the available ones. The goal is rather to realize a compliant behavior in a given set of coordinates x_i , but still having the possibility of using a different set of coordinates x_j when the first set of coordinates x_i fails to provide a proper description of the behavior to execute.

IV. CONTROLLER DESIGN

The design of the control law relies on null space projection matrices to obtain modified task velocities, which are consistent with the priority stack. The idea was originally proposed in [15] to kinematically decouple the tasks, where a generalized inverse is used to this end. By adding weights as in (1), it will be shown that the approach can be extended to achieve dynamic decoupling. Dynamically decoupled task velocities had already been used in [11], where they were computed starting from augmented Jacobian matrices. As in [15], a recursive formulation is adopted in this work to compute the needed quantities. Nevertheless, one can show that this procedure is equivalent to compute the null space projection matrices of the augmented Jacobian matrices as in [11], [13], see Appendix B.

A. Dynamically consistent Jacobian matrices

Also in this and the next sections, stacks of matrices will be denoted by omitting the subscript.

Definition 1: The dynamically consistent Jacobian matrix $\bar{J}_i(q)$ is computed using the orthogonal projection matrix $P_{i-1}(q)$ with metric M and they are defined as⁵

$$\bar{J}_i = J_i P_{i-1} \quad (5)$$

$$P_i = P_{i-1} (E - \bar{J}_i^+ \bar{J}_i) \quad (6)$$

for $i \in \{1, \dots, r\}$ and with $P_0 = E$, being $E \in \mathbb{R}^{n \times n}$ the identity matrix.

A consequence of the definition is that the Jacobian matrix of the first task is unchanged. This is expected, since the first task has the highest priority. The singularities of \bar{J}_i do

³Since incompatible trajectories are unfeasible, there is no loss of generality in assuming that they have been modified offline into compatible ones. For example, by minimizing for each task the difference to the original trajectory, given the constraints imposed by the higher priority tasks.

⁴Only the existence is required, but not the availability to the controller.

⁵Remember the notation $(\cdot)^+ = (\cdot)_{E,M}^\dagger$ and E the identity matrix.

not need to be explicitly detected. They simply yield zero singular values when performing the SVD needed to compute the WMPI and consequently effect P_i , which will reflect the increased null space dimension. Few important properties of the projection matrices P_i are reported here, which are useful to obtain the main results of this work.

Proposition 1: The matrix P_i satisfies the properties

$$P_i P_i = P_i \quad (7)$$

$$P_i^\top M = M P_i \quad (8)$$

$$P_i P_j = P_j P_i = P_i \quad \forall j < i \quad (9)$$

Proof: See Appendix A.1. ■

The first two are what qualifies P_i as an orthogonal projection matrix with metric M . The last one guarantees that each task is projected in the null space of all the previous ones. These properties can be proved either by induction, as shown in Appendix A.1, or by relating the recursive formulation to the augmented Jacobian matrix approach.

Proposition 2: The matrix \bar{J}_i and its WMPI \bar{J}_i^+ satisfy the following properties for all $j \neq i$

$$\bar{J}_j \bar{J}_i^+ = 0 \quad (10)$$

$$\bar{J}_j^{+\top} M \bar{J}_i^+ = 0 \quad (11)$$

$$\bar{J}_j M^{-1} \bar{J}_i^\top = 0 \quad (12)$$

Proof: See Appendix A.2. ■

These properties are fundamental for deriving the expression of the model in the coordinates introduced later on and then used for the design of the control law. In particular, (10) allows to verify that

$$\bar{J}^+ = [\bar{J}_1^+ \quad \dots \quad \bar{J}_r^+] , \quad (13)$$

i.e., the WMPI of \bar{J} is obtained from the WMPI of each single task. In order to guarantee that the tasks can be used for the coordinate transformation in Section IV-B, the following additional assumption will be made.

Definition 2: The active tasks x_A are given by the entries of x corresponding to the independent rows of \bar{J} in a subset \mathcal{U} of the configuration space.

Assumption 2: \bar{J} is always full rank, $\text{rank}(\bar{J}) = n$, and given any two active tasks $x_{A,1}$ in \mathcal{U}_1 and $x_{A,2}$ in \mathcal{U}_2 , then there exist $\Omega = \mathcal{U}_1 \cap \mathcal{U}_2 \neq \emptyset$ where $x_{A,1} \circ x_{A,2}^{-1}$ is smooth.

Intuitively, this means that tasks cannot be all singular in the same points. With a posture task at the end of the stack, Assumption 2 is always verified. Notice the similarity between this requirement and the compatibility of coordinate charts in a smooth manifold [19].

The two projection matrices obtained from \bar{J} and \bar{J}^+ satisfy $\bar{J}^+ \bar{J} = E$ and $\bar{J} \bar{J}^+ = \text{diag}(\bar{J}_i \bar{J}_i^+)$, i.e., a block-diagonal matrix with entries $\bar{J}_i \bar{J}_i^+$ on the diagonal. Finally, the following proposition guarantees that whenever tasks with higher priority use-up all the DoFs, all the other tasks are naturally ignored, i.e., they are not in x_A .

Proposition 3: If $\text{rank}([\bar{J}_1^\top \quad \dots \quad \bar{J}_k^\top]^\top) = n$, then both $\bar{J}_i = 0$ and $\bar{J}_i^+ = 0$ for all $i > k$.

Proof: This is a direct consequence of the fact that $P_k = 0$ and the application of (9). ■

B. Change of coordinates

In this section, the model will be represented by replacing the joint velocities \dot{q} , with task velocities. Nevertheless, to guarantee the dynamic decoupling of the tasks, for each of them the velocity $v_i = \bar{J}_i \dot{q}$ will be considered rather than \dot{x}_i . The following proposition establishes the relationships between all the different quantities.

Proposition 4: Given the dynamically decoupled task space velocities $v_i = \bar{J}_i \dot{q}$, then

$$\dot{q} = \bar{J}^+ v \quad (14a)$$

$$\dot{x} = T v \quad (14b)$$

where

$$T = \begin{bmatrix} E & 0 & 0 & \dots & 0 \\ J_2 \bar{J}_1^+ & E & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 \\ J_r \bar{J}_1^+ & J_r \bar{J}_2^+ & \dots & \dots & E \end{bmatrix} \quad (15)$$

is a lower block-triangular matrix with identities on the diagonal (therefore invertible) such that $J = T \bar{J}$.

Proof: See Appendix A.3. \blacksquare

Notice that in general $T \neq J \bar{J}^+$ and the equality holds only if each \bar{J}_i is full-rank. In the special case of \bar{J} invertible, $J = T \bar{J} \implies T = J \bar{J}^{-1}$, which is the relationship given in [13]. Using the relationships (14) and pre-multiplying (3) by $\bar{J}^{+\top}$, the model in the new coordinates is

$$\dot{q} = \bar{J}^+ v \quad (16a)$$

$$\Lambda \dot{v} + \Gamma v = \bar{J}^{+\top} (\tau - g), \quad (16b)$$

where $\Lambda = \bar{J}^{+\top} M \bar{J}^+$ and $\Gamma = \bar{J}^{+\top} (C \bar{J}^+ + M \dot{\bar{J}}^+)$. The advantage of using v rather than \dot{x} manifests in the resulting structure of the transformed inertia matrix, i.e., $\Lambda = \text{diag}(\bar{J}_i^{+\top} M \bar{J}_i^+)$, which is a direct consequence of (11)⁶. Therefore, in these coordinates, there is no coupling between the tasks at the acceleration level. Additionally, thanks to the passivity property of (3), one can write Γ as $\Gamma = \Gamma_d + \Gamma_s$ with block-diagonal matrix $\Gamma_d = \text{diag}(\Gamma_{d,i})$, $\Gamma_{d,i} = \bar{J}_i^{+\top} (C \bar{J}_i^+ + M \dot{\bar{J}}_i^+)$ and

$$\dot{\Lambda} = \Gamma_d + \Gamma_d^\top \quad (17)$$

$$\Gamma_s = -\Gamma_s^\top. \quad (18)$$

Therefore, the matrix Γ_s given by the outer diagonal blocks of Γ is skew-symmetric, see also Remark 1.

C. Control law and closed-loop system

The final step in the design of the control law leverages on the properties of the projector $\bar{J} \bar{J}^+ = (\bar{J} \bar{J}^+)^{\top}$.

Proposition 5: The projection matrix $\bar{J} \bar{J}^+$ satisfies

$$\bar{J} \bar{J}^+ \Lambda = \Lambda \quad (19a)$$

$$\bar{J} \bar{J}^+ \Gamma = \Gamma \quad (19b)$$

$$\bar{J} \bar{J}^+ \Gamma_d = \Gamma_d \quad (19c)$$

$$\bar{J} \bar{J}^+ \Gamma_s = \Gamma_s. \quad (19d)$$

⁶While (11) implies that Λ is block-diagonal, (12) allows to conclude that so is also the transformed mobility matrix $\bar{J} M^{-1} \bar{J}^\top$.

Proof: Both (19a) and (19b) follow directly from (1b). Since both $\bar{J} \bar{J}^+$ and Γ_d are block-diagonal, then (19c) follows from applying (1b) to each entry on the diagonal. Finally, (19d) is a consequence of linearity and of $\Gamma_s = \Gamma - \Gamma_d$. \blacksquare

At this point, one can introduce the projected gains K , $D \in \mathbb{R}^{m \times m}$ and the auxiliary variables $v_r, s \in \mathbb{R}^m$. The first are defined as $K = \bar{J} \bar{J}^+ K' \bar{J} \bar{J}^+$ and $D = \bar{J} \bar{J}^+ D' \bar{J} \bar{J}^+$, with $K' = \text{diag}(K'_i)$, $D' = \text{diag}(D'_i)$ and $K'_i > 0$, $D'_i > 0$. The latter are the reference velocity $v_r = v_d - K \tilde{x}$, where $v_d = \bar{J} \bar{J}^+ T^{-1} \dot{x}_d$, and the sliding variable $s = v - v_r$, such that the control law can be written as

$$\tau = g + \bar{J}^\top (\Lambda \dot{v}_r + \Gamma_d v_r + \Gamma_s v - D s). \quad (20)$$

Applying this control law to the system (16) and using $\tilde{v} = v - v_d$ yields the closed-loop system

$$\Lambda \dot{s} = -(\Gamma_d + D) s \quad (21a)$$

$$\tilde{v} = -K \tilde{x} + s, \quad (21b)$$

where the properties (19) have been used. Notice that $s = \bar{J} \bar{J}^+ s$ follows from the definition of s .

Remark 1: A very important observation should be made at this point. In the control law (20) the compensation $\Gamma_s v$ is only necessary to ensure the decoupling between the tasks, but not for stability purposes. Replacing that term with $\Gamma_s v_r$ would not impair the following stability analysis thanks to the skew-symmetry of Γ_s . If a strict decoupling between the tasks is not a requirement, then the control law

$$\tau = g + \bar{J}^\top (\Lambda \dot{v}_r + \Gamma v_r - D s) \quad (22)$$

has the advantage of replacing a cancellation of dynamic terms with a feedforward torque.

V. MAIN RESULTS

When referring to a singularity-free case, both task and algorithmic singularities are excluded [15].

Results will be presented in an increasing level of difficulty, with the omitted proofs postponed to Section VI. The simplest situation is obtained when \bar{J} is assumed to be a square and invertible matrix, as it was done in [11], [13].

Theorem 1: Given Assumption 2 with $m = n$ and no singularities, the closed-loop system (21), obtained by applying the control law (20) to (3), is UGAS.

Compared to [13], the previous theorem has two important advantages. First and most importantly, it leads to the much stronger property of UGAS, rather than its local counterpart ULAS. Secondly, the control law is derived in the spirit of [16] and therefore it easily suggests extensions of the control law to robust and adaptive techniques. Topics not extensively explored yet within the context of hierarchical control.

Theorem 2: Given Assumptions 1-2, with the i -th task being either singularity-free or having $\bar{J}_i = 0$, then the closed-loop system (21), obtained by applying the control law (20) to (3), is UGAS.

The previous theorem considers the situation in which there are more tasks than DoF, but each \bar{J}_i is either zero

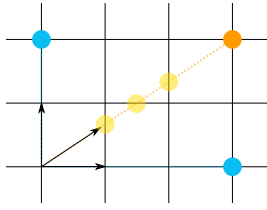


Fig. 2. Whenever three task directions are linearly dependent, the only way the desired values can all be achieved is if, given two independent ones (cyan), the third is the same linear combination of the desired values (orange). Any other goal like the yellow ones, instead, can never be obtained.

or nonsingular. Therefore, a task is allowed to be completely replaced by another one in the priority stack when it becomes singular, without any rearrangement of their priority level.

Corollary 1: If for system (3) only 1-dimensional tasks are designed, with the last n being n independent joint-space tasks, then the closed-loop system (21) is UGAS.

Proof: The requirement on the last n tasks implies that Assumption 1-2 are automatically satisfied. Also, if each task is 1-dimensional, then the only way it can be singular is if $\bar{J}_i = 0$. Therefore, the result follows from Theorem 2. ■

Considering only 1-dimensional tasks is not a too strong restriction since tasks that are not competing with each other will still be fully satisfied. The situations excluded by this restriction are those in which fully-coupled gain matrices would be needed. However, these matrices are typically chosen to be diagonal.

Remark 2: The case in which \bar{J}_i can be singular even if it is nonzero will not be considered in detail in the stability analysis, but only in simulation. Once again the compatibility of the desired trajectory is necessary, as exemplified in Fig. 2.

VI. STABILITY ANALYSIS

The analysis of the system is carried out by considering (21) as the feedforward interconnection of two subsystems. For the first subsystem (21a), consider the functions

$$\begin{aligned} V_s &= \frac{1}{2} s^\top \Lambda s \\ V_1 &\leq V_s \leq V_2, \end{aligned} \quad (23)$$

where $V_l = \lambda_l s^\top \bar{J}^{+\top} \bar{J}^+ s$, with $l \in \{1, 2\}$ and λ_1, λ_2 are coefficients proportional to the minimum and maximum eigenvalues of M , respectively. Differentiating V_s along the flow of (21) yields

$$\dot{V}_s = -s^\top (\bar{J} \bar{J}^+)^{\top} D' (\bar{J} \bar{J}^+) s \leq -\lambda_3 V_s, \quad (24)$$

being λ_3 a coefficient directly proportional to the minimum eigenvalue of D' and inversely proportional to the maximum eigenvalue of M . The second subsystem (21b), is itself the feedforward interconnection of subsystems. This becomes clear when writing subsystem (21b) as

$$\dot{\tilde{x}} = -T K \tilde{x} + u, \quad (25)$$

with $u = T s$, due to the triangular structure of T . A schematic representation of the connections between the subsystems is given in Fig. 3.

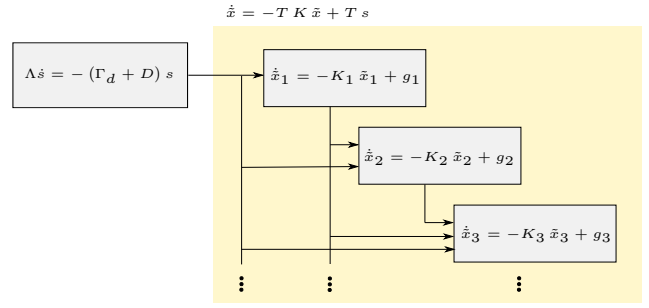


Fig. 3. Schematic cascaded and triangular structure of the subsystems, with $g_1 = g_1(s_1)$, $g_2 = g_2(s_2, \tilde{x}_1)$, $g_3 = g_3(s_3, \tilde{x}_1, \tilde{x}_2)$ and so on.

The proofs of Theorems 1-2 rely on the results in [20, Theorem 4], where a small gain theorem is presented to infer ISS of an interconnected time-varying system and therefore UGAS in absence of inputs.

Proof: [Theorem 1] Due to the hypothesis of Theorem 1, $\bar{J}^+ = \bar{J}^{-1}$, $\bar{J} \bar{J}^+ = E$ and therefore V_s is an ISS-Lyapunov function for (21a), for any Lyapunov gain χ_1 . On the other hand, since $K \succ 0$, (21b) is ISS with respect to the input s . This is because (21b) is itself the feedforward interconnection of autonomous ISS systems. Therefore, it exists an ISS-Lyapunov function with a given Lyapunov gain χ_2 . Since χ_1 is arbitrary, the composition of the Lyapunov gains can be made into a simple contraction, i.e., $\chi_1 \circ \chi_2(r) < r$ for all $r > 0$. Then, by [20, Theorem 4] the system is UGAS. ■

Proof: [Theorem 2] Due to the hypothesis of Theorem 2, each \bar{J}_i is either zero or nonsingular and correspondingly $\bar{J}_i \bar{J}_i^+$ is either zero or E . Therefore, for all the tasks having $\bar{J}_i = 0$ the equations in (21a) and in (21b) are identities of the type $0 = 0$. Clearly these equations can be safely removed and the rest of the proof is identical to Theorem 1. ■

As final comment, it is worth to mention that the proofs presented here not only lead to stronger results, but are also much more compact than the one presented in [13]. The main difference is that the control law (20) leads to a closed-loop system, where the triangular structure has been “moved” into the kinematic equation (21b) only. On the contrary in [13], to show only ULAS a recursive application of the results in [21] is necessary.

VII. TRAJECTORY TRACKING ON A SUBMANIFOLD

Two examples are considered in this section⁷. In the first simulation, a simple planar manipulator, $n = 2$, is considered. Albeit its simplicity, this example captures the essence of the novelty of the proposed approach and shows the improvement over related methods, e.g., [13]. Suppose that one would like to realize a Cartesian impedance, via the task x_1 , and track a desired trajectory $x_{1,d}$ that forces the robot to be fully stretched, Fig. 4-(a). Since this is a singular configuration and in [13] the Jacobian matrix is inverted, the robot fails to complete the task as soon as

⁷Simulations use forward Euler method with a 1 ms sampling time. The gains K_i^l were set to 100 and D_i^l proportional to the inertia, i.e., $D' = 5 \Lambda$. When performing the singular value decomposition needed to compute the WMPI, the singular values were set to zero if smaller than 0.05.

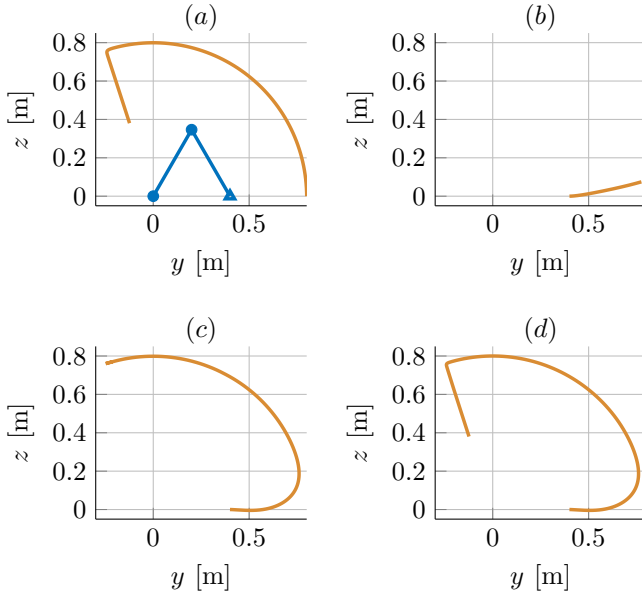


Fig. 4. Comparison with [13]. Desired trajectory and robot in the initial configuration (a). Executed trajectory using [13] (b). Result using the proposed approach with only Cartesian coordinates (c). Result using the proposed approach with both Cartesian and joint coordinates (d).

such a configuration is reached, Fig. 4-(b). With the proposed approach, a first improvement can be obtained since the use of WMPI is numerically more robust; still the trajectory cannot be completed when the robot is asked to move in a direction which is in the nullspace of the Jacobian matrix, Fig. 4-(c). The proposed method can do actually even better. While so far $m = n = 2$, one can add another task, e.g., $x_2 = q$ with $x_{2,d} = q_d$ being q_d the desired trajectory $x_{1,d}$ expressed in joint coordinates. Therefore now $m = 4$. By using two sets of coordinates instead of one, which is only possible with $m > n$, the robot can continue to track the trajectory even when the Cartesian coordinates fail, because the joint coordinates can be used as a backup, Fig. 4-(d). Using the Cartesian coordinates in the first task guarantees that they have priority over the joint coordinates.

Trajectory tracking on a submanifold, like the configuration space of a robot, is in other words a clear case in which the possibility of having $m > n$ pays off. This is due to the typical local nature of the coordinates and the necessity of using multiple sets of coordinates to have a global description of the manifold [19]. If each of these set of coordinates is used for a task of the priority stack, it is clear that the number of tasks might become easily larger than the available DoFs. This is exemplified also in the next example. The end-effector of one of the arms in Fig. 1, $n = 7$, has to track a trajectory on a circle, while changing its orientation and trying to keep a given posture q_d . Both the position and orientation task involve manifold that cannot be described globally by only one chart, i.e., S^1 and $SO(3)$ respectively [19]. Therefore, the same concepts will be used in both cases and exemplified using S^1 , for which the charts depicted in Fig. 5 and corresponding parametrization are adopted.

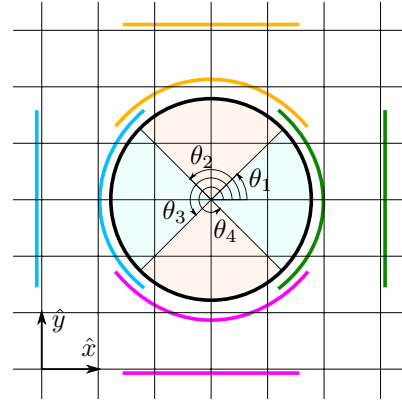


Fig. 5. Each of the four charts, in different colors, maps part of the circle to an open set. Together they cover the whole circle. Quadrants are grouped by color to denote the used parametrization of the circle. Although there are four charts, only two different parameters are used, i.e., x_4 and x_5 .

Denoting by p the position of the end-effector, ϕ_{zYZ} and ϕ_{xYZ} two different sets of Euler angles (with singularities when the second angle ϕ_Y is 0 and $\pi/2$, respectively), the tasks are

$$\begin{aligned}
 x_1(q) &= \beta \phi_{zYZ}(q) & x_2(q) &= \phi_{xYZ}(q) \\
 x_3(q) &= (p_x(q) - p_{x,0})^2 + (p_y(q) - p_{y,0})^2 \\
 x_4(q) &= b (p_x(q) - p_{x,0}) & x_5(q) &= p_y(q) - p_{y,0} \\
 x_6(q) &= p_z(q) & x_7(q) &= q,
 \end{aligned}$$

with the corresponding velocities and for a total of $m = 17$. The desired value $x_{3,d}$ will determine the radius of the circle centered at $(p_{x,0}, p_{y,0})$, while $x_{4,d}$ and $x_{5,d}$ the time evolution of the end-effector of the robot along the circle, since x_4 and x_5 represent the parametrization of the circle according to the current quadrant. The coefficient b is used to render the task x_4 singular when the parametrization provided by x_4 has to be replaced by the one provided by x_5 , i.e., $b = 1$ when $\theta_1 \leq \theta < \theta_2$ or $\theta_3 \leq \theta < \theta_4$ and $b = 0$ otherwise, see Fig. 5. Similarly, β is used to pass from one set of Euler angles to the other. In particular, $\beta = 0$ when the second angle of ϕ_{zYZ} is close to zero, e.g., $\phi_Y \in [-0.15, 0.15]$ rad. Finally, notice that the use of x_4 or x_5 alone would have not resulted in a continuous rotation along the circle. Similarly, since $\phi_{Y,d}$ in Fig. 6 assumes both the values 0 and $\pi/2$, there exists no set of Euler angles that can be used without being in singularity. While in Fig. 6 only 3 of the 17 signals are shown, in Fig. 7 the overall convergence to zero of the orientation, position and configuration error norms is reported. This is possible because the references are compatible and therefore feasible, otherwise only a local minimum would have been reached. For example, $x_{4,d}$ and $x_{5,d}$ have been generated using the well-known trigonometric identities and therefore are consistent with each other.

To summarize, since coordinates are typically valid only locally, using multiple sets of coordinates allows to cover the whole underlying manifold on which the tracking task is defined [19]. This can be seen both when using Cartesian

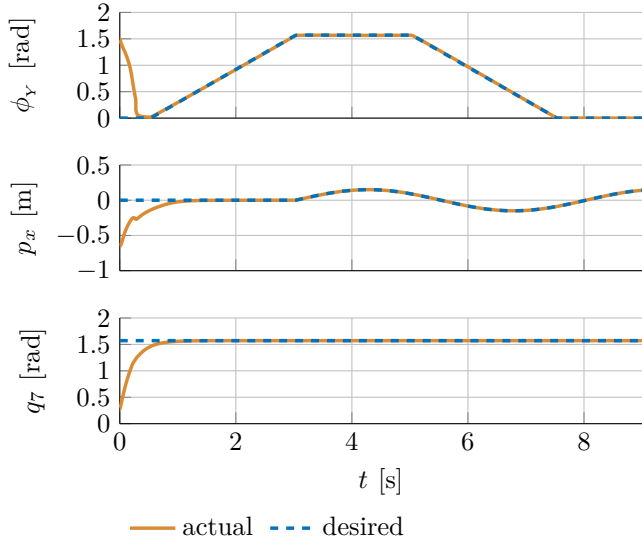


Fig. 6. The robot has to change orientation, move on a circle, keep a certain height and configuration. Since ϕ_Y goes from 0 to $\pi/2$, there exists no singularity-free set of Euler angles for the task. Instead, two sets are used. At $t = 3$, the end-effector starts turning at a constant angular speed along a circle too. To exemplify the tracking on a submanifold of the workspace, x_3 forces the system on S^1 and x_4, x_5 provide a global parametrization.

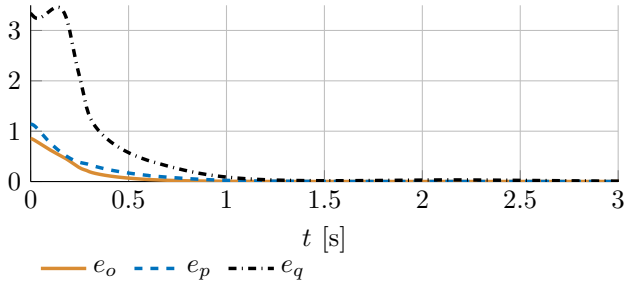


Fig. 7. Convergence of the overall error norms for the orientation e_o , position e_p and configuration e_q . The vector part of the quaternion error was used for e_o . To improve the visualization, only the first 3s are shown.

coordinates in the first example and when using Euler angles on $SO(3)$ or the charts in Fig. 5 for S^1 in the second example. Only in this way it was possible to achieve UGAS on $SO(3)$ via Euler angles. While the resulting trajectory is continuous, there is no guarantee that the control input will be too. Actually, it is well known that for this case no continuous feedback exists to solve the problem.

VIII. CONCLUSION AND FUTURE WORKS

This work has addressed the problem of stabilizing a stack of tasks with a given level of priorities. The proposed control law implements a strict hierarchy and therefore guarantees that no task is affected by the execution of those with a lower priority. Compared to similar approaches in the literature, no restrictions on the number of the tasks and the absence of singularity are necessary. In [13], these assumptions are required to guarantee that an inertially decoupled coordinate transformation can be performed. Here, instead, the use of WMPIs has been shown to achieve the same result, without

those restrictions. The control law leads to a closed-loop system, whose stability analysis is also greatly simplified compared to the state of the art and in the same hypothesis allows to conclude UGAS rather than the much weaker ULAS. This is possible by utilizing global coordinates via multiple charts. Another benefit of the new formulation is that being derived in the spirit of [16], it could easily lead to robust/adaptive extensions of the approach. A topic not yet extensively considered within the context of hierarchical control. These and the following aspects will be considered in future works. The effects of external forces and passivity of the system will need to be discussed. Finally, the effect of numerical approximations in a neighborhood of a singular configuration suggest to evaluate the effects of the damping in the inversion and other considerations, similarly to [12].

APPENDIX

The proofs and results reported in this appendix are valid for any choice of the weights W_1 and W_2 . Nevertheless, for simplicity, $(\cdot)^+$ will still be used to denote the WMPI.

A. Proofs of Propositions

1) *Proof of Proposition 1:* The first two properties are shown by induction. Both are trivially satisfied for $i = 0$ as base case, since $P_0 = E$. Assuming that the properties hold for $i - 1$, it is shown that they hold also for i .

a) $P_i P_i = P_i$:

$$\begin{aligned} P_i P_i &= (P_{i-1} - P_{i-1} \bar{J}_i^+ \bar{J}_i) (P_{i-1} - P_{i-1} \bar{J}_i^+ \bar{J}_i) \\ &= P_{i-1} - 2 P_{i-1} \bar{J}_i^+ \bar{J}_i + P_{i-1} \bar{J}_i^+ \bar{J}_i = P_i \end{aligned}$$

b) $P_i^\top M = M P_i$: Using (1d)

$$\begin{aligned} P_i^\top &= (M M^{-1} - M \bar{J}_i^+ \bar{J}_i M^{-1}) P_{i-1}^\top \\ &= M (E - \bar{J}_i^+ \bar{J}_i) P_{i-1} M^{-1} . \end{aligned}$$

The proof is concluded using the following Corollary 2.

Lemma 1: $\bar{J}_i^+ = P_{i-1} \bar{J}_i^+$.

Proof: Due to the uniqueness of the solution of (1), it is enough to verify that also $P_{i-1} \bar{J}_i^+$ satisfies (1). ■

Corollary 2: $(E - \bar{J}_i^+ \bar{J}_i) P_{i-1} = P_{i-1} (E - \bar{J}_i^+ \bar{J}_i)$.

c) $P_i P_j = P_j P_i = P_i$ for all $j < i$: The first step is to verify that it holds for $j = i - 1$. Once again this could be done by induction, but it becomes obvious using Corollary 2. To show that it holds for $j = i - 2$, one can always expand the expression as

$$P_i P_{i-2} = \underbrace{P_i P_{i-1}}_{P_i} P_{i-2} = P_i \underbrace{P_{i-1} P_{i-2}}_{P_{i-1}} = P_i .$$

The other cases follow immediately.

2) *Proof of Proposition 2:* The properties are valid for $j \neq i$, but showing them for $j < i$ is enough due to the symmetry of the matrices $\bar{J} M^{-1} \bar{J}^\top$, $\bar{J}^{+\top} M \bar{J}^+$ and $\bar{J} \bar{J}^+$. Therefore, in what follows $j < i$ always.

a) $\bar{J}_j M^{-1} \bar{J}_i^\top = 0$: Using (7) and then (8), (9)

$$\bar{J}_j P_{j-1} M^{-1} P_{i-1}^\top \bar{J}_i^\top = \bar{J}_j P_{i-1} M^{-1} \bar{J}_i^\top = 0 .$$

b) $\bar{J}_i^{+\top} M \bar{J}_j^+ = 0$: Using (1) and then Lemma 1

$$\begin{aligned} \bar{J}_i^{+\top} M \bar{J}_j^+ \bar{J}_j \bar{J}_j^+ &= \bar{J}_i^{+\top} P_{i-1}^\top \bar{J}_j^\top \bar{J}_j^+ M \bar{J}_j^+ \\ &= \bar{J}_i^{+\top} (\bar{J}_j P_{i-1})^\top \bar{J}_j^+ M \bar{J}_j^+ = 0 . \end{aligned}$$

c) $\bar{J}_j \bar{J}_i^+ = 0$: Using (1) and then (12)

$$\bar{J}_j \bar{J}_i^+ \bar{J}_i \bar{J}_i^+ = \bar{J}_j M^{-1} \bar{J}_i^\top \bar{J}_i^{+\top} M \bar{J}_i^+ = 0 .$$

3) *Proof of Proposition 4:* For completeness, it is worth to recall the basic result that if $A \in \mathbb{R}^{m \times n}$, with $m \geq n$ has $\text{rank}(A) = n$, then the invertible matrix $A^+ A$ is the identity. This is shown considering

$$A^+ A = A^+ A A^+ A (A^+ A)^{-1} = A^+ A (A^+ A)^{-1} = E .$$

a) $\dot{q} = \bar{J}^+ v$: Due to Assumption 2, then

$$\dot{q} = E \dot{q} = \bar{J}^+ \bar{J} \dot{q} = \bar{J}^+ v .$$

b) $\dot{x} = T v$: The structure of T will be proved indirectly by giving the expression of \dot{x} in terms of v . Lemma 1 and (9) will be needed to this end.

$$\begin{aligned} \dot{x}_i &= J_i \dot{q} = J_i \sum_{j=1}^r \bar{J}_j^+ v_j \\ &= J_i \sum_{j=1}^{i-1} \bar{J}_j^+ v_j + J_i \bar{J}_i^+ \bar{J}_i \dot{q} + J_i \sum_{j=i+1}^r \bar{J}_j^+ v_j \\ &= J_i \sum_{j=1}^{i-1} \bar{J}_j^+ v_j + J_i P_{i-1} \bar{J}_i^+ \bar{J}_i \dot{q} = J_i \sum_{j=1}^{i-1} \bar{J}_j^+ v_j + v_i . \end{aligned}$$

B. Equivalence to the augmented method

Proposition 6: The projection matrix P_i is the null space projector of the matrices

$$J_i^A = [J_1^\top \ \dots \ J_i^\top]^\top \quad \bar{J}_i^A = [\bar{J}_1^\top \ \dots \ \bar{J}_i^\top]^\top .$$

Proof: This can be shown by induction, where $J_1^A P_1 = J_1 P_1 = 0$ is used as base case and $\bar{J}_i^A P_i = 0$ as induction hypothesis. From the induction hypothesis, follows

$$\begin{aligned} J_{i+1}^A P_{i+1} &= [J_i^{A\top} \ J_{i+1}^\top]^\top P_i P_{i+1} \\ &= [0 \ \bar{J}_{i+1}^\top]^\top P_{i+1} = 0 , \end{aligned}$$

where (9) has been used. The same holds for \bar{J}_i^A . ■

REFERENCES

- [1] O. Kanoun, F. Lamiroux, and P.-B. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Trans. on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.
- [2] A. Herzog, N. Rotella, S. Mason, F. Grimmering, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 473–491, 2016.
- [3] D. J. Agravante, G. Claudio, F. Spindler, and F. Chaumette, "Visual servoing in an optimization framework for the whole-body control of humanoid robots," *IEEE Robotics and Automation Letters (RA-L)*, vol. 2, no. 2, pp. 608–615, 2017.
- [4] J. Salini, V. Padois, and P. Bidaud, "Synthesis of complex humanoid whole-body behavior: A focus on sequencing and tasks transitions," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011, pp. 1283–1290.
- [5] N. Dehio, R. F. Reinhart, and J. J. Steil, "Multiple task optimization with a mixture of controllers for motion generation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, Sept. 2015, pp. 6416–6421.
- [6] K. Bouyarmane and A. Kheddar, "On weight-prioritized multitask control of humanoid robots," *IEEE Trans. on Automatic Control*, vol. 63, no. 6, pp. 1632–1647, 2018.
- [7] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.
- [8] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, vol. 2, no. 4, pp. 505–518, 2005.
- [9] L. Righetti, J. Buchli, M. Mistry, and S. Schaal, "Inverse dynamics control of floating-base robots with external constraints: A unified view," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011, pp. 1085–1090.
- [10] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano, "Task-space control of robot manipulators with null-space compliance," *IEEE Trans. on Robotics*, vol. 30, no. 2, pp. 493–506, 2014.
- [11] C. Ott, A. Dietrich, and A. Albu-Schäffer, "Prioritized multi-task compliance control of redundant manipulators," *Automatica*, vol. 53, pp. 416–423, 2015.
- [12] N. Dehio and J. J. Steil, "Dynamically-consistent generalized hierarchical control," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Montreal, Canada, May 2019, pp. 1141–1147.
- [13] A. Dietrich and C. Ott, "Hierarchical impedance-based tracking control of kinematically redundant robots," *IEEE Trans. on Robotics*, vol. 36, no. 1, pp. 204–221, 2020.
- [14] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.
- [15] B. Siciliano and J.-J. E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Fifth International Conference on Advanced Robotics (ICAR)*, Pisa, Italy, June 1991, pp. 1211–1216.
- [16] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *Int. Journal of Robotics Research*, vol. 6, pp. 49–59, 1987.
- [17] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*. New York: Springer-Verlag, 2003.
- [18] K. L. Doty, C. Melchiorri, and C. Bonivento, "A theory of generalized inverses applied to robotics," *Int. Journal of Robotics Research*, vol. 12, no. 1, pp. 1–19, 1993.
- [19] T. Frankel, *The Geometry of Physics: An Introduction*, III ed. New York: Cambridge University Press, 2004.
- [20] H. A. Edwards, Y. Lin, and Y. Wang, "On input-to-state stability for time varying nonlinear systems," in *IEEE Conf. on Decision and Control*, vol. 4, Sydney, Australia, Dec. 2000, pp. 3501–3506.
- [21] Z. Wang, Y. Tan, G. Wang, and D. Nešić, "On stability properties of nonlinear time-varying systems by semi-definite time-varying Lyapunov candidates," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 1123–1128, 2008.