Predefined-time robust hierarchical inverse dynamics on torque-controlled redundant manipulators

Jonathan Obregón-Flores[1], Gustavo Arechavaleta[1], *Member, IEEE*, Héctor M. Becerra[2], *Member, IEEE*, and América Morales-Díaz[1]

***Abstract*—We propose a robust hierarchical inverse dynamics control scheme for redundant manipulators that guarantees both, predefined-time convergence of regulation tasks and robustness against model uncertainties/disturbances. Predefined-time convergence of a robotic task means that the task error is regulated to the origin at a desired preset time, independently of the initial state of the robot. The strict hierarchy of tasks is ensured by the so-called dynamic consistency, which avoids conflicts between decoupled task constraints. Thus, it enables the robot with more dexterity for reaching targets in its workspace while overcoming obstacles and joint limits. Predefined-time convergence facilitates the design of a time schedule to execute sequential and simultaneous tasks. All these features are supported by a stability analysis and experiments with two torque-controlled mobile manipulators in pick and place applications.**

***Index Terms*—predefined-time convergence, super-twisting control, hierarchical inverse dynamics, torque-controlled robots.**

I. Introduction

Robots have become essential to perform specific tasks in industrial facilities. Commonly, some tasks require fixed base robotic manipulators while others demand mobile platforms for object transportation. To make progress towards general purpose assignments, mobile manipulators with kinematic redundancy represent a better choice. In addition, the performance and compliance of robot motions can be directly handled with torque-controlled mobile manipulators.

Among the existing robot control schemes that accommodate these properties, the operation space formulation (OSF) is doubtless the most studied robot control framework [1]. By means of OSF it is possible to ensure dynamic consistency between a primary task and secondary robot posture objectives [2], [3]. Its extension to multiple hierarchical tasks has been suggested in [4]. However, model uncertanties and disturbances significantly degrade the performance of OSF in real robotic platforms [5].

In addition, industrial applications demand the execution of robotic tasks in a desired time, which benefits the planning and repeatability of a given mission. For some years now finite and fixed-time stability have been studied to control dynamical systems aiming to accomplish some time constraints, for instance in [6] for single systems and in [7], [8] for multi-agent systems. Such properties only provide upper bounds of the settling time for certain regions of the initial state, and it is often not easy to have a direct relationship between the control gains and the convergence time. There are recent efforts to propose feedback controllers able to guarantee *constant convergence time* independently of initial conditions, for instance, [9], [10] for single systems, [11] for multi-agent systems and [12] for robot manipulators. This property is called predefined-time convergence, and it allows the user to specify *a priori* the settling time of a task error as a parameter of the control law. Different from finite and fixed-time convergence, the predefined-time convergence allows the regulation of the closed-loop state trajectories exactly in a preset time, independently of the initial state of the system.

A. Contributions

In this paper we deal with all the problems referred above, i.e. model uncertainties/disturbances in OSF, a recursive formulation that ensures dynamic consistency of hierarchical tasks, decoupled closed-loop system, the corresponding stability analysis, and experimental validation with torque-controlled mobile manipulators. Furthermore, we introduce a new player of great importance in industrial applications, which is the strict time scheduling of tasks execution.

This work significantly extend our preliminary results in simulation presented in [13]. Here, we provide a formal analysis of the problem to derive a new robust hierarchical inverse dynamics (RHID) capable to ensure the accomplishment of sequential and simultaneous tasks in a predefined time. We use a super-twisting control approach to provide the required robustness against robot model uncertainties and disturbances. From the recursive shared null-space projectors, we deduce some properties to verify the dynamic consistency of the tasks, and we exploit them to simplify the multi-task closed-loop system where terms due to model uncertainty appear. The analysis of projector properties, and the robustness of the super-twisting control, allow us to prove convergence of task errors at a predefined time, independently of the initial robot state and despite of the execution of several non-conflicting tasks. In addition, we present four experiments using kinematic redundant torque-controlled mobile manipulators. In all the cases, the robots are asked to reach a desired pose for their end-effectors in the workspace subject to collision avoidance and joint limits. A particular interesting experiment shows the ability of the proposed scheme to incorporate a desire schedule for executing tasks in a pick-and-place application.

[1] J. Obregón-Flores, G. Arechavaleta and A. Morales-Díaz are with Robotics and Advanced Manufacturing Group, Centro de Investigación y de Estudios Avanzados del IPN, Saltillo, Coah., Mexico. {garechav,america.morales}@cinvestav.edu.mx

[2] H. M. Becerra is with Centro de Investigación en Matemáticas (CIMAT), Guanajuato, Gto., Mexico. hector.becerra@cimat.mx

The experimental scenario is depicted in Fig. 1, where two mobile manipulators perform the collaborative mission of transporting an object, exchanging it at a meeting point, and placing the object in a bin, all in a desired (predefined) time while handling the hierarchy of tasks when required.

B. Related work

1) Hierarchical task space control: Since the OSF was introduced in [1], several improvements guided by the robotics community have been proposed to fully exploit the kinematic redundancy of robotic systems. In [14], the control problem is reformulated in terms of efficient convex quadratic programs to tackle the case of general actuation constraints, i.e. control bounds, underactuation/overactuation. The inclusion of hierarchical inequality constraints in quadratic programs was previously studied in [15]. In [16], the activation of inequalities is carefully handled to ensure the continuity of the input signals.

On the other hand, the undesired effects due to model uncertainties of the robotic system have been studied with a real platform in [5]. Particular attention has been given to the weighting matrix in the generalized inversion. It is verified that the control performance degrades if the inertia matrix is chosen as the weighting matrix. This is more evident in case of fast robot movements due to inaccuracies of the inertia matrix estimation. The analysis is extended in [3], which provides more information on the use of either successive or augmented null-space projectors. Similar to the first-order kinematic case [17], the augmented methods [18] are less restrictive, but more computationally demanding unless recursive algorithms come to play [19]. In this direction, there exist some efforts to cope with model uncertainties/disturbances within the OSF, such as [20] where a time-delay estimation scheme is applied. A different approach based on nonlinear optimization is suggested in [21], where the generalized torque is considered as a random variable to introduce chance constraints. Thus, the uncertainty is encapsulated in the generalized torque. Although this optimization-based method provides robustness to constraint satisfaction, a constrained nonlinear optimization problem has to be solved at each instant of time. It is then suggested to apply sequential quadratic programming, which is clearly more demanding in terms of computation time than solving a quadratic program.

Different from the first-order kinematic case [17], the stability analysis of OSF with multiple hierarchical tasks had not been carefully studied until [22] and [23] for the ideal case (i.e. without model uncertainties/disturbances). The main difficulty relies on the inherent coupling between hierarchical tasks in the closed-loop system. In [22], it is proposed a change of coordinates to shape the operational inertia matrix as a block-diagonal matrix. This allows to perform the stability analysis of a multi-task compliance controller based on passivity and semi-definite Lyapunov functions. Recently, this control scheme has been extended to cope with trajectory tracking and contact impedance in the context of physical interaction tasks [24]. Although a rigorous proof of asymptotic stability is provided, the robustness against model uncertainties/disturbances has not been explicitly considered. Thus, the theoretical results assume the ideal case where the closed-loop system nicely decouples hierarchical tasks by means of dynamic consistency. There exist learning based methods for OSF [25], which also have some robustness since the model is learned by an inverse problem, but, to our knowledge, they focus on a single task, and it is not clear how convergence time constraints can be incorporated. Moreover, learning based approaches are computationally costly.

2) Predefined-time convergence and robust control: There exist other robot manipulator control schemes than OSF, which overcome inaccuracies and external disturbances [26], [27], [28]. In the first work, the design of an observer-based controller to achieve exponential trajectory tracking subject to both uncertain dynamics and kinematics is presented. In [27], [28], a super-twisting controller is used, which is known for being a robust continuous second order sliding mode control [29]. In particular, a nested super-twisting algorithm for uncertain robotic manipulators is proposed in [27], such that robustness against matched and unmatched perturbations, and continuity of the control signals are provided. A super-twisting algorithm with time delay estimation is designed in [28], which is based on input/output feedback linearization for uncertain robot manipulators. To the authors' knowledge, the robustness properties of controllers, like the super-twisting, have not been explored before in the OSF.

Related to the state of the art on predefined-time control, our work extends the applicability of existing approaches [9], [12] by considering several tasks in a hierarchical scheme. Moreover, in contrast to [12], our approach is also valid for mobile manipulators and guarantees accurate regulation under bounded model uncertainties and disturbances. The proposed control law is well-conditioned all the time, which marks a difference from no robust approaches [10] and [11]. A drawback of [10] is that it uses a time-scaling of the state by means of a time function that grows unbounded as reaching the convergence time.

The organization of the remaining sections is as follows. In Section II, we briefly describe related works on finite and predefined-time convergence for second order systems. Then, Section III presents the hierarchical inverse dynamics formulation. In Section IV, we describe the proposed RHID scheme, and we provide the stability analysis of the closed-loop system. In Section V we give a detailed description of the tasks we used in the RHID. The experimental validation with different scenarios are presented in Section VI. Finally, we provide some concluding remarks in Section VII.

II. Predefined-time convergence

Providing guaranties that a controlled system will converge in a certain bounded time has been an intense field of research in the last decade. Such time constraints have been integrated in the control of redundant robots in works like [30], [31], [32], [33], [34]. Finite-time stability means that the trajectories of a system converge to an equilibrium point in finite time. For uncertain robotic manipulators, this has been applied for instance in [30], [31]. In particular, these works propose a class of continuous robust controllers based on a non-singular

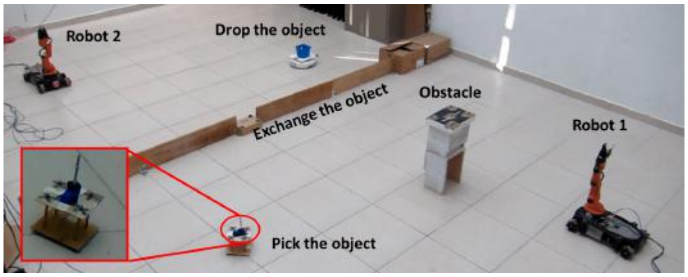


Fig. 1: **Experimental scenario**: Two omnidirectional mobile manipulators perform a collaborative pick and place task.

terminal sliding mode control. In addition, adaptive techniques and integral sliding mode control have been suggested to achieve finite-time control of uncertain robot manipulators [32], [33]. Fixed-time stability demands uniform boundedness of the settling time for a globally finite-time stable system, such that convergence of the closed-loop system is guaranteed before an estimated bound on the settling time, independently of the initial conditions [35]. A fixed-time inverse dynamics control for trajectory tracking of robot manipulators is proposed in [34] by using bi-limit homogeneity technique.

Few works have addressed the problem to guarantee convergence in a predefined time, for instance [9], [10], [12], [11]. As described in the previous section, we have considered the extension of [9] as the best option, since the other referred approaches have practical issues. In [9], continuous time-dependent polynomial functions, called time base generators (TBGs), which reduce to zero in a specified time, are used as reference trajectories in a tracking scheme together with a robust controller. It guarantees convergence of the state in predefined-time, even for bounded matched disturbances. Such robust scheme is exploited in this work to guarantee that hierarchical robotic tasks, controlled at dynamic level, are performed in a desired time.

In the rest of this section, we briefly summarize the definition of TBG functions. Consider the double integrator system $\ddot{e}(t) = u(t)$, which is of interest for dynamic control of robotic tasks. According to [9], to achieve predefined-time convergence, the control input $u(t)$ must be designed to track desired profiles for $e(t)$ and $\dot{e}(t)$. Two TBGs are required to set such profiles, which are defined as follows:

$$h_j(t) = \left\{ \begin{array}{ll} g(t) \cdot c_j & \text{if } t \in [0, t_f] \\ 0 & \text{otherwise} \end{array} \right. \tag{1}$$

where $j \in \{1, 2\}$, $g(t) = [t^r \; t^{r-1} \; \ldots \; t \; 1]$ is the time basis vector with $r \geq 5$ and c_j is a vector of coefficients. The functions $h_j(t)$ are continuous and smooth such that $\dot{h}_j(t)$ and $\ddot{h}_j(t)$ exist. The desired profiles evolution of the state trajectories in terms of the TBGs $h_1(t)$, and $h_2(t)$ are:

$$\begin{aligned} e(t) &= e(0)h_1(t) + \dot{e}(0)h_2(t), \\ \dot{e}(t) &= e(0)\dot{h}_1(t) + \dot{e}(0)\dot{h}_2(t). \end{aligned} \tag{2}$$

Thus, for time $t = 0$, the TBG functions must accomplish $h_1(0) = \dot{h}_2(0) = 1$ from the first equation, and $h_2(0) = \dot{h}_1(0) = 0$ from the second equation in (2). Besides, $\dot{h}_1(t) = \dot{h}_2(t) = 0$ for $t \geq t_f$.

A straightforward method to compute $h_j(t)$ and its first and second time derivatives consists of solving the following linear system to find the coefficients c_j for $r = 5$:

$$\begin{bmatrix} c_1 & c_2 \end{bmatrix} = \begin{bmatrix} G(0) \\ G(t_f) \end{bmatrix}^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{3}$$

where $c_j \in \mathbb{R}^6$ is the vector of coefficients,

$$G(t) = \begin{bmatrix} g(t) \\ \dot{g}(t) \\ \ddot{g}(t) \end{bmatrix}$$

is the time basis matrix, $I \in \mathbb{R}^{3\times2}$ is an identity matrix and $0 \in \mathbb{R}^{3\times2}$ a matrix of zeros. It is, however, possible to allow more flexibility during the computation of coefficients by setting $r > 5$ as it has been suggested in [9]. In that case, pseudoinversion can be used in (3) to compute the coefficients of the TBGs.

III. Task-space inverse dynamics formulation

The robot inverse dynamics problem involves the computation of the robot's equations of motion. For this purpose, we use the d'Alembert-Lagrange formulation:

$$A(q)\ddot{q} + b(q, \dot{q}) = \tau \tag{4}$$

where $\{q, \dot{q}, \ddot{q}\} \in \mathbb{R}^n$ are the joint position, velocity and acceleration, respectively. $A(q) \in \mathbb{R}^{n\times n}$ denotes the symmetric positive definite inertial matrix, the vector of nonlinear terms is:

$$b(q, \dot{q}) \triangleq C(q, \dot{q})\dot{q} + g(q) \in \mathbb{R}^n \tag{5}$$

where $C(q, \dot{q}) \in \mathbb{R}^{n\times n}$ contains the Coriolis and centrifugal terms, $g(q) \in \mathbb{R}^n$ represents the gravity and $\tau \in \mathbb{R}^n$ is the vector of generalized input torques.

A. Task definition

A task is simply an error function in terms of the robot's configuration together with the corresponding differential mapping between the task and configuration coordinates associated to the robotic system [36]:

$$e = x(q) - x^d \in \mathbb{R}^m \tag{6}$$

where $x(q)$ is obtained by means of forward kinematics, x^d is a constant vector of desired values of the task, and therefore $\dot{e} = \dot{x} = J\dot{q}$ with $J = \dfrac{\partial e}{\partial q} \in \mathbb{R}^{m\times n}$ the task Jacobian. We assume that (6) is twice differentiable with respect to time:

$$\ddot{e} = J\ddot{q} + \dot{J}\dot{q}. \tag{7}$$

By solving for $\ddot{q}$ in (4) and plugging it in (7) yields:

$$\ddot{e} = Q\tau + \mu \tag{8}$$

where $Q = JA^{-1} \in \mathbb{R}^{m\times n}$ and $\mu = -Qb + \dot{J}\dot{q} \in \mathbb{R}^m$ is the task's drift. The task-based inverse dynamics is obtained by solving for τ in (8) as follows:

$$\tau = Q^{\#_A}(u - \mu) \in \mathbb{R}^n \tag{9}$$

where

$$Q^{\#_A} = AQ^T \left[QAQ^T\right]^{-1} = J^T \left[JA^{-1}J^T\right]^{-1} \in \mathbb{R}^{n\times m} \tag{10}$$

is the weighted generalized inversion of JA^{-1}, and u is an auxiliary vector of control inputs to be designed.

B. Definition of hierarchical tasks

The hierarchical inverse dynamics is a powerful control framework to exploit the kinematic redundancy of torque-controlled robotic systems [1], which can be subject to rigid contacts and underactuation constraints [4], [15] as well as actuation redundancy [14]. The main idea is to define a set of p tasks to be executed simultaneously. This is possible because the dimension of the task space is less than that of the configuration space, i.e. $m < n$. To overcome possible conflicts among tasks, a hierarchy between them is imposed such that (9) becomes:

$$\begin{aligned} \tau &= \sum_{i=1}^{p} \tau_i \in \mathbb{R}^n, & (11)\\ \tau_i &= \overline{Q}_i^{\#_A} \left(u_i - Q_i \sum_{k=0}^{i-1} \tau_k - \mu_i \right) \in \mathbb{R}^n, & (12)\\ N_i &= N_{i-1} - \overline{Q}_i^{\#_A} \overline{Q}_i \in \mathbb{R}^{n\times n} & (13) \end{aligned}$$

where $\overline{Q}_i = Q_i N_{i-1} \in \mathbb{R}^{m_i \times n}$, $\overline{Q}_i^{\#_A} = A\overline{Q}_i^T \left[\overline{Q}_i A \overline{Q}_i^T \right]^{-1} \in \mathbb{R}^{n\times m_i}$, $\tau_0 = 0$ and $N_0 = I_n$. Notice that N_i is the null-space projector of $\overline{Q}_i$.

The recursion in (12) is the inverse dynamics formulation of the kinematics one that was introduced in [18], where the contribution of higher hierarchical tasks is taken into account to solve the current task. In [18], the shared null-space projector is calculated with the so-called augmented Jacobian, which stacks the associated task Jacobians of higher hierarchy. However, the recursive update of the null-space projector (13), that was proposed in [19], is faster than the augmented Jacobian method.

C. Properties of null-space projectors

Based on the dynamically consistent property of null-space projectors introduced in [2], and recently studied in [3], we deduced the following set of properties that turn to be useful to proof the stability of the proposed control law described in Section IV. In particular, it is possible to verify that

$$N_i N_j = \begin{cases} N_j & \text{if } j \geq i \\ N_i & \text{if } j < i \end{cases} \tag{14}$$

$$\overline{Q}_i N_j = \begin{cases} \overline{Q}_i & \text{if } i > j \\ 0 & \text{if } i \leq j \end{cases} \tag{15}$$

$$N_j \overline{Q}_i^{\#_A} = \begin{cases} \overline{Q}_i^{\#_A} & \text{if } i > j \\ 0 & \text{if } i \leq j \end{cases} \tag{16}$$

where (16) mainly relies on the fact that

$$N_i A N_i^T = A N_i^T \tag{17}$$

It can also be deduced:

$$\overline{Q}_i \overline{Q}_j^{\#_A} = \begin{cases} I & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{18}$$

However, note that

$$Q_i \overline{Q}_j^{\#_A} = \begin{cases} I & \text{if } i = j \\ 0 & \text{if } i < j \\ Q_i \overline{Q}_j^{\#_A} & \text{if } i > j \end{cases} \tag{19}$$

The proof of (14)-(19) is given in Appendix A.

IV. Robust hierarchical inverse dynamics with predefined-time convergence

The objective now is to use the TBG as reference profiles in a tracking control law to drive the task errors of a hierarchical inverse dynamics scheme to converge in a predefined-time. In particular, we take advantage of both the TBG together with a *super-twisting control* (STC), which is capable to cope with matched uncertainties/disturbances [37].

Definition 4.1 (Problem definition): Given a robot model of the form (4) and one or p tasks defined as in (6), design a control law $\tau = \Upsilon(e_1, ..., e_p, t)$ such that the task error functions $e_1, ..., e_p$ converge all of them to zero in a predefined finite time t_f from any initial condition when there are not conflicts between hierarchical tasks. In case of conflicts, the error convergence of lower hierarchical tasks is subject to the error convergence of tasks with higher hierarchy. Moreover, convergence is achieved even in the presence of uncertainty in the parameters of the robot model and external disturbances.

Remark 4.2: We assume, as in the literature on predefined-time convergence, that the specified desired settling time is physically feasible. Thus, the minimum time that the system is able to execute for a given task, in some particular conditions, is not in the scope of the problem definition.

A. Uncertainty bounds in the parameters of the robot model

Since the inertia matrix A is positive definite, it is assumed:

$$A_{min} \leq \|A^{-1}\| \leq A^{max} < \infty, \quad \forall q \in \mathbb{R}^n \tag{20}$$

where A_{min} and A^{max} are two positive constants. Also there exists a constant $0 < \alpha < 1$ such that (see [38]),

$$\|A^{-1}\widetilde{A} - I\| \leq \alpha \tag{21}$$

where model uncertainties of the robotic system are contained in $\widetilde{A} \in \mathbb{R}^{n\times n}$. Hence, in task space we have:

$$\begin{aligned} M &= JA^{-1}J^T \left[J\widetilde{A}^{-1}J^T \right]^{-1} = Q\widetilde{Q}^{\#_A} \\ \|M\| &= \left\| Q\widetilde{Q}^{\#_A} \right\| \leq M^{max} & (22) \end{aligned}$$

where M^{max} is a positive constant upper bound. Notice that in the ideal case in which $\widetilde{A} = A$, we have $M = I_m$.

B. Single task formulation

Let us define a vector of tracking error as

$$\varepsilon = e - e^d(t) \in \mathbb{R}^m, \tag{23}$$

where

$$e^d(t) = h_1(t)e(0) + h_2(t)\dot{e}(0) \in \mathbb{R}^m. \tag{24}$$

with $e = x(q) - x^d$ and $\dot{e} = \dot{x}$ according to (6).

Theorem 4.3: Consider the TBG functions (1) with t_f as the desired convergence time for the dynamics (8) and estimated values of A and μ as $\widetilde{A}$ and $\widetilde{\mu}$, respectively. There exist gains $\{k_1, k_2, k_3\} \in \mathbb{R}$ such that the continuous time-varying STC

$$\tau = \widetilde{Q}^{\#_A}(u - \widetilde{\mu}) \in \mathbb{R}^n, \tag{25}$$

with

$$\begin{aligned} u &= \ddot{e}^d(t) - k_3\dot{\varepsilon} - k_1 S\text{sign}(s) + v \in \mathbb{R}^m, \\ \dot{v} &= -k_2\text{sign}(s) \in \mathbb{R}^m, \end{aligned} \tag{26}$$

and $s = k_3\varepsilon + \dot{\varepsilon} \in \mathbb{R}^m$, $S = diag(|s_1|^{1/2}, ..., |s_m|^{1/2})$, achieves global asymptotic stability of the tracking error (23), and therefore predefined-time convergence for the system (8) to the origin is attained. Thus, the task specified by (6) is achieved at time t_f.

Proof: Given the definition of the tracking error, we have $\ddot{\varepsilon} = \ddot{e} - \ddot{e}^d(t)$, where $\ddot{e}$ is given by (8). The model uncertainties of the robotic system are contained in $\widetilde{A} \in \mathbb{R}^{n\times n}$ and $\widetilde{b} \in \mathbb{R}^n$, with A fulfilling (21). As a consequence, the task-space dynamics is also subject to these model uncertainties, and the control law compensates the uncertain drift of the task in (25), where $\widetilde{\mu}$ is given by:

$$\widetilde{\mu} = -\widetilde{Q}\widetilde{b} + \dot{J}\dot{q}. \tag{27}$$

Therefore, under the presence of parametric uncertainties, we have that $M \neq I_m$, which fulfills the bounding condition (22). Moreover, the closed-loop dynamics of the tracking error has the form

$$\begin{aligned} \ddot{\varepsilon} &= Mu + \gamma, \\ &= -k_1 MS\text{sign}(s) + Mv - k_3 M\dot{\varepsilon} + M\ddot{e}^d(t) + \gamma \end{aligned} \tag{28}$$

where $\gamma = \mu - M\widetilde{\mu} - \ddot{e}^d(t)$. We assume that the model uncertainties are such that the matrix M remains close to diagonal. This is justified since algorithms of parameter identification are accurate enough, given uncertainties lower than 15% even for an inertial matrix only composed by diagonal constant coefficients [20].

In that case, the sliding surface dynamics is given by

$$\dot{s} = k_3\dot{\varepsilon} + \ddot{\varepsilon}, \tag{29}$$

and substituting (28) into (29) we have:

$$\begin{aligned} \dot{s} &= -k_1 MS\text{sign}(s) + Mv + \Gamma, \\ \dot{v} &= -k_2\text{sign}(s), \end{aligned} \tag{30}$$

where

$$\Gamma = k_3\dot{\varepsilon} - k_3 M\dot{\varepsilon} + \mu - M\widetilde{\mu} - \ddot{e}^d(t) + M\ddot{e}^d(t) \tag{31}$$

is a bounded perturbation.

In the ideal case in which $\widetilde{A} = A$ and $\widetilde{\mu} = \mu$, we have that $\widetilde{Q} = Q$ and $\widetilde{\mu} = \mu$. Therefore $M = I_m$ and $\Gamma = 0$. Then, the closed-loop dynamics of the tracking error is given by

$$\ddot{\varepsilon} = u - \ddot{e}^d(t), \tag{32}$$

The sliding surface dynamics is then simplified as

$$\begin{aligned} \dot{s} &= -k_1 S\text{sign}(s) + v \\ \dot{v} &= -k_2\text{sign}(s) \end{aligned} \tag{33}$$

This corresponds to the typical super-twisting dynamics, which is known to achieve convergence in finite-time to the origin [37], i.e. $(s, v) = (0, 0)$, provided that $k_1 > 0$ and $k_2 > 0$. Then, the dynamics of the tracking error (28) is constrained to $s = \dot{s} = 0$, and consequently $\dot{\varepsilon} = -k_3\varepsilon$. It implies the global asymptotic stability of the tracking error (28). Consequently, the state of the system (8) tracks the TBG references and, it is driven to the origin $(e, \dot{e}) = (0, 0)$ in the predefined time t_f.

In the case of considering the parametric uncertainty in the robot model, we take into account the perturbation $\Gamma \neq 0$ and the matrix $M \neq I_m$. Using the change of variable $\varrho = Mv + \Gamma$, we have:

$$\begin{aligned} \dot{s} &= -k_1 MS\text{sign}(s) + \varrho \\ \dot{\varrho} &= -k_2 M\text{sign}(s) + \dot{\Gamma} \end{aligned} \tag{34}$$

The matrix M can be seen as a scaling of the control gains. Moreover, it can be replaced by its upper bound value $M^{max} I_2$ in the worst case. For an appropriate choice of those k_1 and k_2, the global convergence in finite time of (34) to the origin, i.e. $(s, \varrho) = (0, 0)$, is ensured in the presence of continuously and smooth bounded entries of the disturbance vector, i.e. $|\Gamma_i| < \ell$ and $|\dot{\Gamma}_i| < \eta$ for $i = 1, ..., m$ and some constants $\ell > 0$ and $\eta > 0$, as it has been proved in [37], [39]. Particularly, the gains can be set as $k_2 > \eta$ and $k_1 > 1.41\sqrt{k_2 + \eta}$, according to [39], to guarantee global convergence to the origin of (34) in a characterized finite time.

Therefore, the tracking error dynamics is constrained to evolve on the sliding surface, and the same conclusion as in the ideal case holds: the state of the system (8) tracks the TBG references, and it is driven to the origin $(e, \dot{e}) = (0, 0)$ in the predefined time t_f. ■

Remark 4.4: The super-twisting control is well known for its robustness property, and it mitigates the chattering problem, since it does not require high closed-loop frequency, at being a second order sliding mode approach [40].

C. Hierarchical tasks

The convergence in predefined-time of several task functions is also addressed by tracking time-varying references given by TBG signals.

Let us consider p different tasks specified as in (6) where their second order dynamics are given by (8). The corresponding error to track the TBG references are defined as (23).

In the following theorem, we extend the previous robust control law for predefined-time convergence to the hierarchical task-based scheme (11).

Theorem 4.5: Consider the TBG functions (1) with t_f the desired convergence time for the dynamics (8) and estimated values of A and μ as $\widetilde{A}$ and $\widetilde{\mu}$, respectively. There exist positive gains $\{k_{i_1}, k_{i_2}, k_{i_3}\} \in \mathbb{R}$ such that the hierarchical task-based control law

$$\begin{aligned} \tau &= \sum_{i=1}^{p} \tau_i \in \mathbb{R}^n, && (35) \\ \tau_i &= \widetilde{\overline{Q}}_i^{\#_A} (u_i - \widetilde{\alpha}_i) \in \mathbb{R}^n, && (36) \\ \widetilde{N}_i &= \widetilde{N}_{i-1} - \widetilde{\overline{Q}}_i^{\#_A} \widetilde{\overline{Q}}_i \in \mathbb{R}^{n\times n} \end{aligned}$$

with $\tau_0 = 0$ and $\widetilde{N}_0 = I_n$, where $\widetilde{\overline{Q}}_{\underset{\sim}{i}} = \widetilde{Q}_i \widetilde{N}_{i-1} \in \mathbb{R}^{m_i\times n}$, $\widetilde{\alpha}_i = \widetilde{Q}_i \sum_{k=0}^{i-1} \tau_k - \widetilde{\mu}_i \in \mathbb{R}^{m_i}$, $\widetilde{\mu}_i = \widetilde{Q}_i h - \dot{J}_i \dot{q} \in \mathbb{R}^{m_i}$ and

$$\begin{aligned} u_i &= \ddot{e}_i^d(t) - k_{i_3}\dot{\varepsilon}_i - k_{i_1} S_i \text{sign}(s_i) + v_i \in \mathbb{R}^{m_i}, \\ \dot{v}_i &= -k_{i_2}\text{sign}(s_i) \in \mathbb{R}^{m_i}, && (37) \end{aligned}$$

and $s_i = k_{i_3}\varepsilon_i + \dot{\varepsilon}_i \in \mathbb{R}^{m_i}$, $S_i = diag(|s_{i_1}|^{1/2}, ..., |s_{m_i}|^{1/2})$, achieves global asymptotic stability of each tracking error (23), and therefore predefined-time convergence for each system (8) to the origin is attained. Thus, the tasks specified by (6) are all simultaneously achieved in time t_f.

Proof: The stability proof is first shown for three tasks, then it is generalized for p tasks. In the three tasks case, (35) takes the following form:

$$\tau = \tau_1 + \tau_2 + \tau_3 \in \mathbb{R}^n \tag{38}$$

where

$$\begin{aligned} \tau_1 &= \widetilde{\overline{Q}}_1^{\#_A} (u_1 - \widetilde{\alpha}_1) \\ \tau_2 &= \widetilde{\overline{Q}}_2^{\#_A} (u_2 - \widetilde{\alpha}_2) \\ \tau_3 &= \widetilde{\overline{Q}}_3^{\#_A} (u_3 - \widetilde{\alpha}_3) \end{aligned}$$

with

$$\begin{aligned} \widetilde{\alpha}_1 &= -\widetilde{\mu}_1 \\ \widetilde{\alpha}_2 &= \widetilde{Q}_2\tau_1 - \widetilde{\mu}_2 \\ \widetilde{\alpha}_3 &= \widetilde{Q}_3(\tau_1 + \tau_2) - \widetilde{\mu}_3 \end{aligned}$$

The second order dynamics of the tracking errors (23) can be written as:

$$\begin{aligned} \ddot{\varepsilon}_1 &= Q_1(\tau_1 + \tau_2 + \tau_3) + \alpha_1 - \ddot{e}_1^d(t) \in \mathbb{R}^{m_1} && (39) \\ \ddot{\varepsilon}_2 &= Q_2(\tau_2 + \tau_3) + \alpha_2 - \ddot{e}_2^d(t) \in \mathbb{R}^{m_2} && (40) \\ \ddot{\varepsilon}_3 &= Q_3\tau_3 + \alpha_3 - \ddot{e}_3^d(t) \in \mathbb{R}^{m_3} && (41) \end{aligned}$$

where

$$\begin{aligned} \alpha_1 &= -\mu_1 \\ \alpha_2 &= Q_2\tau_1 - \mu_2 \\ \alpha_3 &= Q_3(\tau_1 + \tau_2) - \mu_3 \end{aligned}$$

The computation of (39), (40) and (41) can be expressed in matrix form as:

$$\begin{bmatrix} \ddot{\varepsilon}_1 \\ \ddot{\varepsilon}_2 \\ \ddot{\varepsilon}_3 \end{bmatrix} = \begin{bmatrix} M_1 & C_{12} & C_{13} \\ 0 & M_2 & C_{23} \\ 0 & 0 & M_3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix} \tag{42}$$

where M_1, M_2 and M_3 are defined as in (22). The upper triangular terms in (42) stands for $C_{12} = Q_1\widetilde{\overline{Q}}_2^{\#_A}$, $C_{13} = Q_1\widetilde{\overline{Q}}_3^{\#_A}$ and $C_{23} = Q_2\widetilde{\overline{Q}}_3^{\#_A}$ with

$$\begin{aligned} \gamma_1 &= \alpha_1 - M_1\widetilde{\alpha}_1 - C_{12}\widetilde{\alpha}_2 - C_{13}\widetilde{\alpha}_3 - \ddot{e}_1^d(t) \\ \gamma_2 &= \alpha_2 - M_2\widetilde{\alpha}_2 - C_{23}\widetilde{\alpha}_3 - \ddot{e}_2^d(t) \\ \gamma_3 &= \alpha_3 - M_3\widetilde{\alpha}_3 - \ddot{e}_3^d(t) && (43) \end{aligned}$$

In the ideal case, in which $\widetilde{A} = A$ and $\widetilde{\mu} = \mu$, we have that $\widetilde{Q}_i = Q_i$ and $\widetilde{\alpha}_i = \alpha_i$. On the one hand, the matrices M_i:

$$M_i = Q_i\overline{Q}_i^{\#_A} = I_{m_i} \text{ for } i = 1, \ldots, 3. \tag{44}$$

Note that from (19), the upper triangular terms become:

$$C_{ij} = Q_i\overline{Q}_j^{\#_A} = 0_{m_i\times m_j} \text{ for } i < j \text{ and } \{i, j\} \in \{1, 2, 3\} \tag{45}$$

Consequently, the tracking error dynamics is reduced to the following three decoupled second order systems:

$$\begin{bmatrix} \ddot{\varepsilon}_1 \\ \ddot{\varepsilon}_2 \\ \ddot{\varepsilon}_3 \end{bmatrix} = \begin{bmatrix} I_{m_1} & 0_{m_1\times m_2} & 0_{m_1\times m_3} \\ 0_{m_2\times m_1} & I_{m_2} & 0_{m_2\times m_3} \\ 0_{m_3\times m_1} & 0_{m_3\times m_2} & I_{m_3} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} - \begin{bmatrix} \ddot{e}_1^d(t) \\ \ddot{e}_2^d(t) \\ \ddot{e}_3^d(t) \end{bmatrix} \tag{46}$$

The second order dynamics of each tracking error is then similar to (32), which has been proved in Theorem 1 to converge globally asymptotically to the origin. Therefore, by using the control inputs defined in (37), each decoupled tracking error dynamics (46) achieves global asymptotic stability and predefined-time convergence for each system (8) to the origin is attained.

In the non-ideal case, in which we consider parametric uncertainty in the robot model, the second order dynamics in (42) must be analyzed with all its terms. At this stage, the second order dynamics is generalized to p hierarchical tasks since its structure is preserved:

$$\begin{bmatrix} \ddot{\varepsilon}_1 \\ \ddot{\varepsilon}_2 \\ \vdots \\ \ddot{\varepsilon}_p \end{bmatrix} = \begin{bmatrix} M_1 & C_{12} & \cdots & C_{1p} \\ 0 & M_2 & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & M_p \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_p \end{bmatrix} + \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_p \end{bmatrix} \tag{47}$$

with $\gamma_i = \alpha_i - M_i\widetilde{\alpha}_i - \left(\sum_{j=i+1}^{p} C_{ij}\widetilde{\alpha}_j\right) - \ddot{e}_i^d(t)$.

First, we need to prove that as matrix $\widetilde{A}$ tends to A, then

$$\begin{aligned} M_i = Q_i\widetilde{\overline{Q}}_i^{\#_A} &\approx I_{m_i\times m_i}, \text{ for } i = 1, \ldots, p && (48) \\ C_{ij} = Q_i\widetilde{\overline{Q}}_j^{\#_A} &\approx 0_{m_i\times m_j}, \text{ if } i < j. && (49) \end{aligned}$$

Recall the definitions $Q_i := J_iA^{-1}$, $\overline{Q}_i := Q_iN_{i-1}$ and $Q_i^{\#_A} := AQ_i^T(Q_iAQ_i^T)^{-1}$. Now, let us prove (48) as follows:

$$\begin{aligned} Q_i\widetilde{\overline{Q}}_i^{\#_A} &= J_iA^{-1}\widetilde{A}\widetilde{\overline{Q}}_i^T \left[\widetilde{\overline{Q}}_i\widetilde{A}\widetilde{\overline{Q}}_i^T\right]^{-1} \\ &= J_iA^{-1}\widetilde{A}\left[\widetilde{Q}_i\widetilde{N}_{i-1}\right]^T \left[\widetilde{Q}_i\widetilde{N}_{i-1}\widetilde{A}\left[\widetilde{Q}_i\widetilde{N}_{i-1}\right]^T\right]^{-1} \\ &= J_iA^{-1}\widetilde{A}\widetilde{N}_{i-1}^T\widetilde{A}^{-T}J_i^T \left[J_i\widetilde{A}^{-1}\widetilde{N}_{i-1}\widetilde{A}\widetilde{N}_{i-1}^T\widetilde{A}^{-T}J_i^T\right]^{-1} \end{aligned} \tag{50}$$

By applying the property $\widetilde{N}_{i-1}\widetilde{A}\widetilde{N}_{i-1}^T = \widetilde{A}\widetilde{N}_{i-1}^T$ given in (17), it is deduced that:

$$\begin{aligned} Q_i\widetilde{\overline{Q}}_i^{\#_A} &= J_iA^{-1}\widetilde{A}\widetilde{N}_{i-1}^T\widetilde{A}^{-T}J_i^T\left[J_i\widetilde{A}^{-1}\widetilde{A}\widetilde{N}_{i-1}^T\widetilde{A}^{-T}J_i^T\right]^{-1} \\ &= J_i\underline{A^{-1}\widetilde{A}}\widetilde{N}_{i-1}^T\widetilde{A}^{-T}J_i^T\left[J_i\widetilde{N}_{i-1}^T\widetilde{A}^{-T}J_i^T\right]^{-1} := M_i \end{aligned} \tag{51}$$

It can be seen in the underlined terms that as $\widetilde{A}$ tends to A, $M_i = Q_i\widetilde{\overline{Q}}_i^{\#_A}$ tends to $I_{m_i\times m_i}$, since $A^{-1}\widetilde{A} \approx I_n$.

To prove (49), we proceed as follows:

$$\begin{aligned} Q_i\widetilde{\overline{Q}}_j^{\#_A} &= Q_i\widetilde{A}\widetilde{\overline{Q}}_j^T\left[\overline{\widetilde{Q}}_j\widetilde{A}\widetilde{\overline{Q}}_j^T\right]^{-1} \\ &= Q_i\widetilde{A}\left[\widetilde{Q}_j\widetilde{N}_{j-1}\right]^T\left[\widetilde{Q}_j\widetilde{N}_{j-1}\widetilde{A}\left[\widetilde{Q}_j\widetilde{N}_{j-1}\right]^T\right]^{-1} \\ &= Q_i\widetilde{A}\widetilde{N}_{j-1}^T\widetilde{Q}_j^T\left[\widetilde{Q}_j\widetilde{N}_{j-1}\widetilde{A}\widetilde{N}_{j-1}^T\widetilde{Q}_j^T\right]^{-1} \end{aligned}$$

By applying the property (17) and $\widetilde{A}\widetilde{N}_{j-1}^T = \widetilde{N}_{j-1}\widetilde{A}$, we have:

$$Q_i\widetilde{\overline{Q}}_j^{\#_A} = \underline{Q_i\widetilde{N}_{j-1}}\widetilde{A}\widetilde{Q}_j^T\left[\widetilde{Q}_j\widetilde{N}_{j-1}\widetilde{A}\widetilde{Q}_j^T\right]^{-1} \tag{52}$$

From (103), proved in Appendix A, $Q_iN_i = 0$ holds in the ideal case. Therefore, the underlined product $Q_i\widetilde{N}_{j-1}$ is the one that could nullify (52), and it is analyzed. Since $i < j$, then given the property (14), and more generally (101), we rewrite

$$Q_i\widetilde{N}_{j-1} = Q_i\widetilde{N}_i\widetilde{N}_{i+1}...\widetilde{N}_{j-2}\widetilde{N}_{j-1} \tag{53}$$

$$\begin{aligned} &= Q_i\left[\widetilde{N}_{i-1} - \widetilde{\overline{Q}}_i^{\#_A}\overline{\widetilde{Q}}_i\right]\widetilde{N}_{i+1}...\widetilde{N}_{j-2}\widetilde{N}_{j-1} \\ &= \left[Q_i\widetilde{N}_{i-1} - Q_i\widetilde{\overline{Q}}_i^{\#_A}\overline{\widetilde{Q}}_i\right]\widetilde{N}_{i+1}...\widetilde{N}_{j-2}\widetilde{N}_{j-1} \end{aligned} \tag{54}$$

Using (51), i.e., $Q_i\widetilde{\overline{Q}}_i^{\#_A} = M_i$, and knowing that M_i tends to $I_{m_i\times m_i}$ as well as $Q_i\widetilde{N}_{i-1}$ tends to $\overline{\widetilde{Q}}_i$ as $\widetilde{A}$ tends to A, we have that $Q_i\widetilde{\overline{Q}}_j^{\#_A}$ tends to a null matrix as $\widetilde{A}$ tends to A.

Thus, for bounded uncertainties in matrix A, we will have bounded no-null terms C_{ij}, and (47) can be written as

$$\ddot{\varepsilon}_i = M_iu_i + \Gamma_i \quad \text{for} \quad i = 1, \dots, p \tag{55}$$

with

$$\Gamma_i = \left(\sum_{j=i+1}^{p} C_{ij}u_j\right) + \gamma_i \tag{56}$$

Each dynamics in (55) is similar to (28) in the perturbed case. Therefore, similarly as in Theorem 4.1, it is concluded that the state of the systems (8) tracks the TBG references, and they are driven to the origin $(e_i, \dot{e}_i) = (0, 0)$ in the predefined time t_f.

■

V. The structure of hierarchical tasks

In hierarchical inverse dynamics schemes, the relation between the expected motion behavior of the robot and the corresponding number of hierarchical tasks heavily relies on user experience. However, there exist some tasks that are commonly considered regardless of the type of kinematically redundant robot. For instance, positioning tasks to regulate the robot's end-effector to a desired position and orientation. Also, joint limits satisfaction and obstacle avoidance are important. Although these constraints are naturally defined as a set of inequalities, the proposed scheme handles them by means of smooth transition functions as it is explained below.

A. Joint limits task

The task for the upper limit of the j-th joint coordinate is:

$$e_{l_j} \quad = \quad q_j^* - q_j \in \mathbb{R} \tag{57}$$

where $q_j^* = \bar{q}_j - \beta$ is the difference between the upper joint limit $\bar{q}_j$ and an activation buffer β. In this case, the second order task dynamics is:

$$\ddot{e}_{l_j} \quad = \quad Q_{l_j}\tau_{l_j} + \mu_{l_j} \tag{58}$$

where $Q_{l_j} = J_{l_j}A^{-1}$ and $\mu_{l_j} = -Q_{l_j}b$ with

$$J_{l_j} = \begin{bmatrix}\alpha_1 & \cdots & \alpha_j & \cdots & \alpha_n\end{bmatrix} \in \mathbb{R}^n, \tag{59}$$

the element $\alpha_j = 1$ if $q_j \geq q_j^*$, i.e. the joint limit is an active constraint, otherwise $\alpha_j = 0$. The remaining row elements $\alpha_i = 0$ for $i = 1, \dots, n$ and $i \neq j$. The control law for a single joint limit task j is computed as follows:

$$\tau_{l_j} \quad = \quad \overline{Q}_{l_j}^{\#_A}\left(u'_{l_j} - \mu_{l_j}\right) \tag{60}$$

where $\overline{Q}_{l_j} = Q_{l_j}N_0$ with $N_0 = I_n$, and u'_{l_j} corresponds to the intermediate value of the j-th joint limit, as suggested in the smooth task transition strategies described in [41] to overcome abrupt activation and deactivation of joint limits. In particular, the intermediate value is computed as:

$$u'_{l_j} = f(q)u_{l_j} + (1 - f(q))Q_{l_j}\tau_{l_{[\setminus j]}} \tag{61}$$

with the auxiliary control input u_{l_j} obtained from (37), where no predefined-time convergence trajectory needs to be specified, i.e., $e_{l_j}^d(t) = \dot{e}_{l_j}^d(t) = \ddot{e}_{l_j}^d(t) = 0$. $f(q) \in [0, 1]$ is a generic smooth transition function and $\tau_{l_{[\setminus j]}}$ denotes a control law similar to (60) for all active tasks different from the jth-task. It is computed as:

$$\tau_{l_{[\setminus j]}} \quad = \quad \overline{Q}_{l_{[\setminus j]}}^{\#_A}\left(u_{l_{[\setminus j]}} - \mu_{l_{[\setminus j]}}\right)$$

where $\overline{Q}_{l_{[\setminus j]}} = Q_{l_{[\setminus j]}}N_0$,

$$Q_{l_{[\setminus j]}} = \begin{bmatrix} Q_{l_1} \\ \vdots \\ Q_{l_{j-1}} \\ Q_{l_{j+1}} \\ \vdots \\ Q_{l_n} \end{bmatrix}, \; u_{l_{[\setminus j]}} = \begin{bmatrix} u_{l_j} \\ \vdots \\ u_{l_{j-1}} \\ u_{l_{j+1}} \\ \vdots \\ u_{l_n} \end{bmatrix}, \text{ and } \mu_{l_{[\setminus j]}} = \begin{bmatrix} \mu_{l_j} \\ \vdots \\ \mu_{l_{j-1}} \\ \mu_{l_{j+1}} \\ \vdots \\ \mu_{l_n} \end{bmatrix}$$

B. Obstacle avoidance task

Similar to joint limits, if the robot approaches the obstacles, the corresponding equality constraint is activated. In particular, each obstacle is embedded within a spherical shell of radii d_m and d_M, with $d_m < d_M$. Thus, the obstacle avoidance task is defined as

$$e_{c_i} = \begin{bmatrix} d_{i,j} - d_M \\ d_{i,j+1} - d_M \\ \vdots \\ d_{i,r} - d_M \end{bmatrix} \in \mathbb{R}^r \tag{62}$$

with $d_{i,j} = \|c_i - c_j\|$ the distance from a control point attached to the robot $c_i(q) \in \mathbb{R}^3$, and the nearest point over the j-th obstacle $c_j \in \mathbb{R}^3$. Assuming c_j is constant, the time derivative of (62) becomes

$$\dot{e}_{c_i} \quad = \quad L_i^T J_{c_i} \dot{q} \in \mathbb{R}^r \tag{63}$$

where $J_{c_i} = \dfrac{\partial c_i}{\partial q} \in \mathbb{R}^{3 \times n}$ is the linear velocity Jacobian and $L_i = \begin{bmatrix} l_{i,j} & l_{i,j+1} & \dots & l_{i,r} \end{bmatrix} \in \mathbb{R}^{3 \times r}$ with

$$l_{i,j} = \frac{c_i - c_j}{\|c_i - c_j\|} \in \mathbb{R}^3 \tag{64}$$

and its time derivative:

$$\dot{l}_{i,j} = \left(l_{ij}^T J_{c_i} \dot{q}\right) \frac{c_i - c_j}{\|c_i - c_j\|^2} - J_{c_i} \dot{q} \frac{\|c_i - c_j\|}{\|c_i - c_j\|^2} \in \mathbb{R}^3 \tag{65}$$

is needed to construct $\dot{L}_i = \begin{bmatrix} \dot{l}_{i,j} & \dot{l}_{i,j+1} & \dots & \dot{l}_{i,r} \end{bmatrix} \in \mathbb{R}^{3 \times r}$ in order to compute the time derivative of (63):

$$\ddot{e}_{c_i} \quad = \quad L_i^T J_{c_i} \ddot{q} + \left[L_i^T \dot{J}_{c_i} + \dot{L}_i^T J_{c_i} \right] \dot{q} \in \mathbb{R}^r \tag{66}$$

By solving for $\ddot{q}$ in (4) and plugging it in (66) gives the error dynamics of the obstacle avoidance task:

$$\ddot{e}_{c_i} \quad = \quad Q_{c_i} \tau_{c_i} + \mu_{c_i} \in \mathbb{R}^r \tag{67}$$

where $Q_{c_i} = L_i^T J_{c_i} A^{-1} \in \mathbb{R}^{r \times n}$, and $\mu_{c_i} = -Q_{c_i} h + \left[L_i^T \dot{J}_{c_i} + \dot{L}_i^T J_{c_i} \right] \dot{q} \in \mathbb{R}^r$. Thus, the control law becomes:

$$\tau_{c_i} \quad = \quad \overline{Q}_{c_i}^{\#_A} \left(u'_{c_i} - \mu_{c_i} \right) \tag{68}$$

where $\overline{Q}_{c_i} = Q_{c_i} N_l$ with N_l as the null-space projector of joint limits task. Given that the rank of $\overline{Q}_{c_i}$ depends on how many obstacle avoidance tasks are activated, i.e. r, the auxiliary control $u'_{c_i} \in \mathbb{R}^r$ contains the intermediate values $u'_{c_{i,j}} \in \mathbb{R}$ which are computed as

$$u'_{c_{i,j}} = f(d_{i,j}) u_{c_{i,j}} + (1 - f(d_{i,j})) Q_{c_{i,j}} \tau_{c_{i,[\backslash j]}} \tag{69}$$

where $Q_{c_{i,j}} = l_{i,j}^T J_{c_i} A^{-1} \in \mathbb{R}^n$, $f(d_{i,j}) \in [0, 1]$ is a smooth transition function, and

$$\tau_{c_{i,[\backslash j]}} = \overline{Q}_{c_{i,[\backslash j]}}^{\#_A} (u_{c_{i,[\backslash j]}} - \mu_{c_{i,[\backslash j]}}) \tag{70}$$

where $\overline{Q}_{c_{i,[\backslash j]}} = L_{i,[\backslash j]}^T J_{c_i} A^{-1} N_l \in \mathbb{R}^{(r-1) \times n}$, $u_{c_{i,[\backslash j]}} \in \mathbb{R}^{r-1}$ and $\mu_{c_{i,[\backslash j]}} \in \mathbb{R}^{r-1}$. Similar to the joint limits task, the auxiliary control $u_{c_{i,j}}$ is computed as (37) without specifying a predefined-time convergence trajectory.

C. Position and orientation tasks

Regarding the orientation task, we parametrized the end-effector's orientation with unit quaternions to avoid singularities. Thus, the task error with respect to the orientation of the robot's end-effector is:

$$e_\xi \quad = \quad \xi_{d_1} \xi_{[2:4]} - \xi_1 \xi_{d_{[2:4]}} + \widehat{\xi_{d_{[2:4]}}} \xi_{[2:4]} \tag{71}$$

where $\xi_d = [\xi_{d_1}\ \xi_{d_2}\ \xi_{d_3}\ \xi_{d_4}]^T \in \mathbb{S}^3$ is the desired robot's end-effector orientation expressed in terms of a unit quaternion, $\xi(q) = [\xi_1\ \xi_2\ \xi_3\ \xi_4]^T \in \mathbb{S}^3$ is the current robot's end-effector orientation, and

$$\widehat{\xi_{d_{[2:4]}}} \quad = \quad \begin{bmatrix} 0 & -\xi_{d_4} & \xi_{d_3} \\ \xi_{d_4} & 0 & -\xi_{d_2} \\ -\xi_{d_3} & \xi_{d_2} & 0 \end{bmatrix} \tag{72}$$

is the 3 by 3 skew-symmetric matrix operator. The second-order system for the orientation task becomes:

$$\ddot{e}_\xi = Q_\xi \tau_\xi + \mu_\xi \tag{73}$$

and thus, the orientation control is:

$$\tau_\xi = \overline{Q}_\xi^{\#_A} (u_\xi - \mu_\xi) \tag{74}$$

where u_ξ is obtained from (37), and $Q_\xi = J_\xi A^{-1}$, $\overline{Q}_\xi = Q_\xi N_c$ with N_c as the null-space projector of the obstacle avoidance task, $\mu_\xi = -Q_\xi b + \dot{J}_\xi \dot{q}$ with J_ξ as the end-effector's orientation Jacobian, and μ_ξ is the drift of angular accelerations at the end-effector.

The position task is defined by the error between the current p and desired p_d end-effector positions:

$$e_p = p(q) - p_d \in \mathbb{R}^3 \tag{75}$$

The second order system of this task becomes:

$$\ddot{e}_p = Q_p \tau_p + \mu_p \tag{76}$$

and thus, the position control is:

$$\tau_p = \overline{Q}_p^{\#_A} (u_p - \mu_p) \tag{77}$$

where u_p is obtained from (37), and $Q_p = J_p A^{-1}$, $\overline{Q}_p = Q_p N_\xi$ with N_ξ as the null-space projector of the orientation task, $\mu_p = -Q_p b + \dot{J}_p \dot{q}$, with J_p the end-effector's position Jacobian, and μ_p the drift of linear accelerations at the robot's end-effector. It can be now observed that the hierarchical structure is imposed as $e_l \succ e_c \succ e_\xi \succ e_p$ where $a \succ b$ means that a is at a higher hierarchical level than b.

Remark 5.1: Notice that we propose a reactive method to avoid collisions of the robot for unknown environments; the method does not require a path planning stage. Due to the hierarchical structure of the tasks, it could happen that tasks with higher hierarchy may prevent the full accomplishment of secondary tasks, meaning that a task error may converge to a vicinity of the origin, respecting the desired predefined time as its settling time.

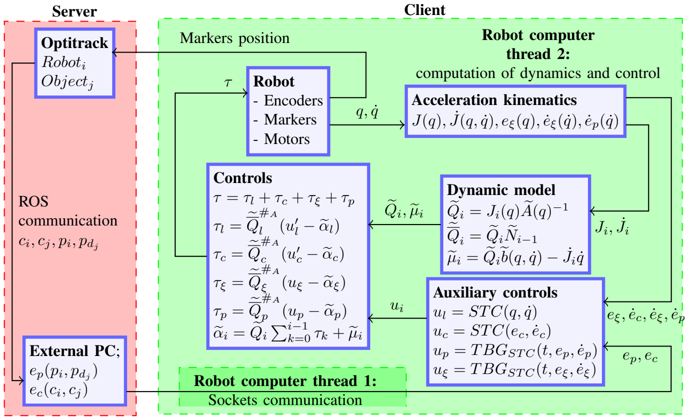

Server
Client
Optitrack
$Robot_i$
$Object_j$
Markers position
Robot computer thread 2:
computation of dynamics and control
τ
Robot
- Encoders
- Markers
- Motors
$q, \dot{q}$
Acceleration kinematics
$J(q), \dot{J}(q,\dot{q}), e_\xi(q), \dot{e}_\xi(\dot{q}), \dot{e}_p(\dot{q})$
ROS communication
c_i, c_j, p_i, p_{d_j}
Controls
$\tau = \tau_l + \tau_c + \tau_\xi + \tau_p$
$\tau_l = \overset{\sim}{\widetilde{Q}}{}_l^{\#_A}(u_l' - \widetilde{\alpha}_l)$
$\tau_c = \overset{\sim}{\widetilde{Q}}{}_c^{\#_A}(u_c' - \widetilde{\alpha}_c)$
$\tau_\xi = \overset{\sim}{\widetilde{Q}}{}_\xi^{\#_A}(u_\xi - \widetilde{\alpha}_\xi)$
$\tau_p = \overset{\sim}{\widetilde{Q}}{}_p^{\#_A}(u_p - \widetilde{\alpha}_p)$
$\widetilde{\alpha}_i = \widetilde{Q}_i \sum_{k=0}^{i-1} \tau_k + \widetilde{\mu}_i$
$\widetilde{Q}_i, \widetilde{\mu}_i$
Dynamic model
$\widetilde{Q}_i = J_i(q)\widetilde{A}(q)^{-1}$
$\overset{\sim}{\widetilde{Q}}_i = \widetilde{Q}_i \widetilde{N}_{i-1}$
$\widetilde{\mu}_i = \widetilde{Q}_i \widetilde{b}(q,\dot{q}) - \dot{J}_i \dot{q}$
$J_i, \dot{J}_i$
u_i
Auxiliary controls
$u_l = STC(q, \dot{q})$
$u_c = STC(e_c, \dot{e}_c)$
$u_p = TBG_{STC}(t, e_p, \dot{e}_p)$
$u_\xi = TBG_{STC}(t, e_\xi, \dot{e}_\xi)$
$e_\xi, \dot{e}_c, \dot{e}_\xi, \dot{e}_p$
e_p, e_c
External PC;
$e_p(p_i, p_{d_j})$
$e_c(c_i, c_j)$
Robot computer thread 1:
Sockets communication

Fig. 2: **Communication scheme:** the green box represents the robot's onboard computation over two threads. The red box represents the external computer where the task errors are evaluated.

VI. Experiments

We implemented the torque-mode hierarchical control law (35), using the robust predefined-time controller (37) for regulation tasks, on mobile manipulators KUKA youBot composed by eight degrees of freedom. The computation was carried out on the robots onboard computer Mini-ITX, with an Intel AtomTM Dual Core D510 (1M Cache, 2 x 1.66 GHz), RAM Memory 2 GB single-channel DDR2 667 MHz. Its operating system is Ubuntu 12.04 LTS. Each arm has a payload of 0.5 Kg. Communication with actuators and encoders is via EtherCAT at 1 ms cycle, through which we can send torque commands and read joint positions and velocities. The whole control scheme was implemented in ANSI C++, and the routines related to numerical linear algebra used Eigen 3.1.1.

We used an optical tracking system (Optitrack) with twelve cameras placed around the workspace to get the position coordinates of the robot end-effector, its mobile base as well as the target and obstacles positions with respect to the same workspace reference frame. The Optitrack worked at 120 fps. which allowed to compute the task errors (62), (75) and (71).

We created a centralized communication network using an external computer with Ubuntu 14.04 LTS (Trusty) as the central hub, which communicated with the Optitrack using ROS Indigo Igloo. We used socket programming in a client/server model to communicate the robots built-in computers (clients) with the central hub (server). As depicted in Fig. 2, the server received spatial coordinates from the Optitrack, and the task errors were sent to the robots. Each robot computed its own control law (35) to be sent to the motors as shown in Fig. 3.

Note that the control law is model-based, which implies the computation of the robot's equations of motion (4). In particular, we applied the spatial Newton-Euler algorithm to efficiently evaluate the non-linear terms encoded in $b(q, \dot{q})$, and the inertia matrix $A(q)$ is obtained with the Composite-Rigid-Body algorithm [42]. We used the manufacturer's physical parameters, which are found in youbot website[1]. These parameters are typically an estimation of the real ones. Thus, the model considered for control design is uncertain.

A. Mobile manipulator model

The configuration of the experimental platform is given by

$$q \;=\; \begin{bmatrix} q_b \\ q_m \end{bmatrix} \in SE(2) \times \mathbb{T}^5 \tag{78}$$

where $q_b \in SE(2)$ corresponds to the position and orientation coordinates of the vehicle, and $q_m \in \mathbb{T}^5$ is the manipulator's joint coordinates. The KUKA youBot is an omni-directional mobile manipulator, with a mobile base actuated by four mecanum-wheels. It has indeed nine actuators. Thus, the following assumptions have been considered to determine the relationship between actuators/torques and generalized forces/torques:

- A horizontal flat surface is considered as the floor.
- The wheels are always in contact with the floor, and they do not slip.
- The contact forces at the wheels match the gravity force acting on the mobile base.
- The driving and sliding torques of the wheels, and the contact forces at the wheels, are the only acting forces.

The generalized torques τ are related with the actuated torques τ_a, containing the wheel's and manipulator torques $\tau_b \in \mathbb{R}^4$ and $\tau_m \in \mathbb{R}^5$, respectively, by means of the following torque distribution transformation:

$$\tau_a = \begin{bmatrix} \tau_b \\ \tau_m \end{bmatrix} \;=\; \begin{bmatrix} J_b^+ & 0 \\ 0 & I_5 \end{bmatrix} \tau \in \mathbb{R}^9 \tag{79}$$

where

$$J_b^+ = \begin{bmatrix} \frac{r}{4} & -\frac{r}{4} & -\frac{r}{4(\ell+d)} \\ \frac{r}{4} & \frac{r}{4} & \frac{r}{4(\ell+d)} \\ \frac{r}{4} & -\frac{r}{4} & \frac{r}{4(\ell+d)} \\ \frac{r}{4} & \frac{r}{4} & -\frac{r}{4(\ell+d)} \end{bmatrix} \tag{80}$$

depends on the vehicle's parameters as depicted in Fig. 3.

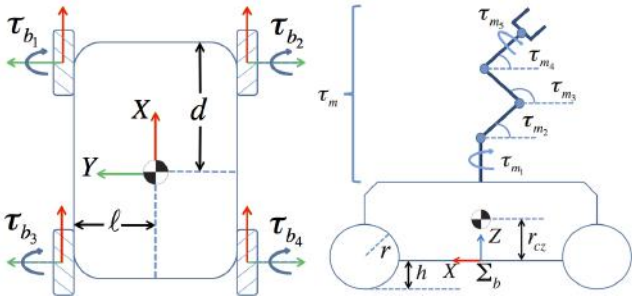
Proposition 6.1: The equations of motion in (4) as well as the control laws (25) and (35) hold for mobile manipulators under the given assumptions together with (79).

Proof: See Appendix B. ■

B. Experimental Results

The proposed robust control scheme with predefined-time convergence was evaluated through five experiments to test its performance under different operation regimes. Four tasks described in Section V have been considered in the experimental evaluation according to the following hierarchy: $e_l \succ e_c \succ e_\xi \succ e_p$.

[1] http://www.youbot-store.com/developers/kuka-youbot-kinematics-dynamics-and-3d-model-81



(a) Top view

(b) Side view

Fig. 3: **The mobile manipulator**. **Left**: torque distribution of the mobile platform. **Right**: kinematic structure of the mobile manipulator.

1) Experiment 1: Fast regulation task: The robustness of the controller was verified in three different cases of regulation tasks for the end-effector, which included short and long robot displacements to be completed in 5 s. The three initial conditions, in distance to the goal, are 1.6 m, 3.1 m and 2.2 m. The control gains were $k_{p_1} = k_{\xi_1} = 8$, $k_{p_2} = k_{\xi_2} = 4$, $k_{p_3} = 30$ and $k_{\xi_3} = 35$. It is important to note that these control gains did not vary regardless of the initial conditions. Also, note that joint limits did not require to be accomplished at a given time (i.e. they are handled as inequality constraints). The gains for the joint limits task were $k_{l_{j1}} = 1$, $k_{l_{j2}} = 0.1$ and $k_{l_{j3}} = 25$ for each joint j and, they were the same for the rest of experiments.

In Figs. 4.(a) - 4.(f) are shown the errors convergence of the end-effector pose coordinates at the predefined time of 5 s in the three different cases. It is observed that the position and orientation errors converged with good precision, even in the case of large displacements and fast motions.

Figs. 4.(g) - 4.(i) show the sliding surfaces associated to the end-effector pose coordinates, which reached values close to zero around the predefined-time. Figs. 4.(j) - 4.(l) show the smooth activation of joint limits.

2) Experiment 2: Regulation task with two static obstacles: We verified the controller performance for executing a regulation task that consists of reaching an object. The robot had to navigate through a longer distance than in *Experiment 1* while avoiding static obstacles. The reaching task must be completed at a predefined time of 20 s. In this case, the control gains were $k_{p_1} = k_{\xi_1} = 8$, $k_{p_2} = k_{\xi_2} = 4$, $k_{p_3} = 25$ and $k_{\xi_3} = 30$. Also, note that the obstacle avoidance did not require to be accomplished at a given time. The gains for the obstacle avoidance task were $k_{c_1} = 5$, $k_{c_2} = 0.5$ and $k_{c_3} = 15$ for the end effector, and for the mobile base were $k_{c_1} = 3$, $k_{c_2} = 0.2$ and $k_{c_3} = 10$. Obstacle avoidance control gains were the same for the rest of the experiments.

Figs. 5.(a)-5.(b) show the position and orientation errors of the end-effector, which converged at the desired time, despite having to avoid the obstacles. As depicted in Fig. 5.(c), evasion control signals of the form (68) become active when the robot approached obstacles, which appeared at 6 s and 12 s, respectively.

3) Experiment 3: Regulation task with one moving obstacle: In this experiment the robot was asked to reach the same target position and orientation for its end-effector than in the previous experiment, but an intrusive moving obstacle deliberately disturbed the robot. Thus, repulsive control signals handled such situation. It is important to point out that the control gains and the predefined time to accomplish the reaching task were the same as those in *Experiment 2*.

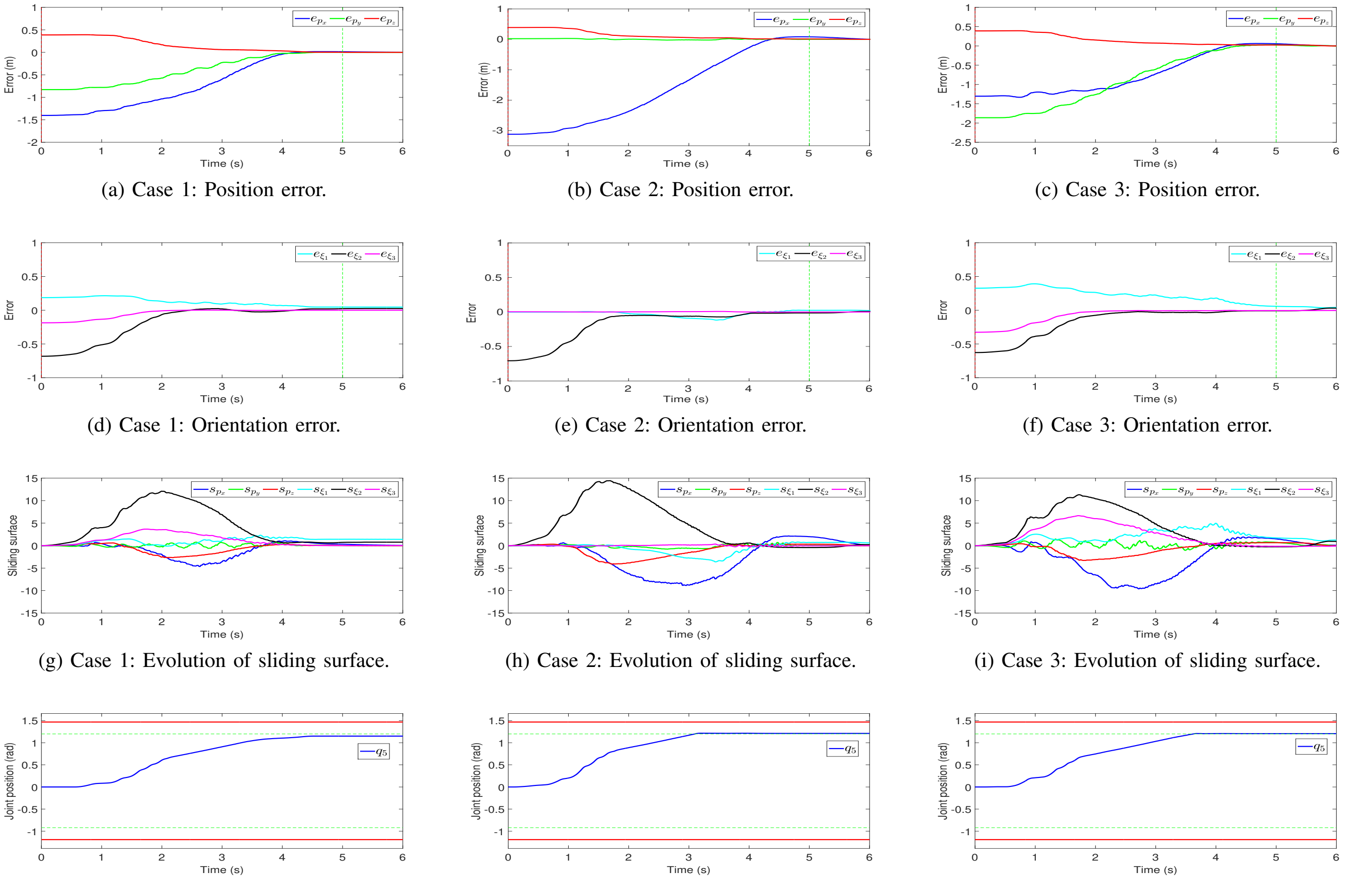
Despite the moving obstacle, the robot was able to reach its desired set-point in the predefined time of 20 s as depicted in Figs. 6.(a)-6.(b). It can be observed in Fig. 6.(c) that the evasion auxiliary control signals generated by the obstacle disturbed the robot twice.

4) Experiment 4: Comparison with hierarchical-quadratic-programming based controller: We have performed two experiments to show the difference between hierarchical-quadratic-programming (HQP), and the proposed control scheme. In both cases, we asked the robot to reach a desired position with its end-effector while avoiding joint limits. The formulation of the HQP follows the method suggested in [15]. The first level in the hierarchical structure has been assigned to the robot's equations of motion and joint limits as equality and inequality constraints, respectively. In the second hierarchical level, the reaching task is performed. Each quadratic program has been solved on the robot's computer with the qpOASES solver.

The desired end-effector position in the perpendicular axis with respect to the floor implied that joint q_5 reached the security threshold of its mechanical limit. According to Figs 7.(a)-7.(b), it is observed that HQP is not robust to model uncertainties since the position error does not converge to zero. It can also be observed that joint q_5 reached the security threshold towards the limit very quickly due to the exponential behavior of the error convergence. On the other hand, as it is shown in Figs. 7.(c)-7.(d), with the proposed control scheme the end-effector reached the desired position in a predefined time of 6 s by performing smooth displacements with its mobile base while its arm is reconfigured. It is also observed that the same joint q_5 reached the security threshold. Although both controllers were asked to regulate the same position task for the end-effector, the error dynamics differ since HQP imposed an exponential convergence instead of the TBGs in (2).

5) Experiment 5: Collaborative pick and place mission: This is a more complex experiment with two robots involved. A finite state machine was designed to coordinate the stack of hierarchical tasks that each robot executed to fulfill a pick and place mission. The workspace was separated in two cells. Each robot was enclosed in different cells by static obstacles. The main challenge was that the pick and place mission required to grasp an object placed in one cell and drop it over the adjacent cell as it is depicted in Figs.1 and 8. Thus, a meeting point was defined for the robots to exchange a common object. Each robot performed its own finite state machine. The collaborative mission followed a strict time schedule depicted in Fig. 9. Each of the tasks was a regulation one with predefined time convergence. In between tasks there is an idle time wait for the grippers to be opened and closed.

Figs. 10.(a)-10.(b) and 10.(d)-10.(e) show end-effector er-



(a) Case 1: Position error.

(b) Case 2: Position error.

(c) Case 3: Position error.

(d) Case 1: Orientation error.

(e) Case 2: Orientation error.

(f) Case 3: Orientation error.

(g) Case 1: Evolution of sliding surface.

(h) Case 2: Evolution of sliding surface.

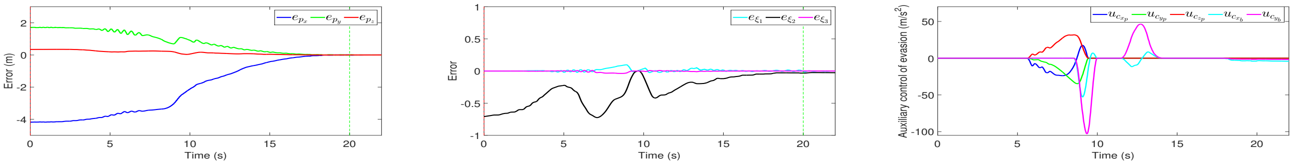
(i) Case 3: Evolution of sliding surface.

(j) Case 1: Active joint limit.

(k) Case 2: Active joint limit.

(l) Case 3: Active joint limit.

Fig. 4: **Comparison of three cases, each with different initial conditions, of regulation tasks for a predefined time of 5 s.** (a), (b) and (c) show the end-effector position errors, (d), (e) and (f), show the end-effector orientation error, both with a convergence at 5 s (dotted green vertical line). Despite model uncertainties, fast motion and active joint limits, the task is achieved with good precision at predefined time. (g), (h) and (i) show the sliding surfaces of the controlled coordinates. (j), (k) and (l) show the active joint limits.



(a) End-effector position errors.

(b) End-effector orientation errors.

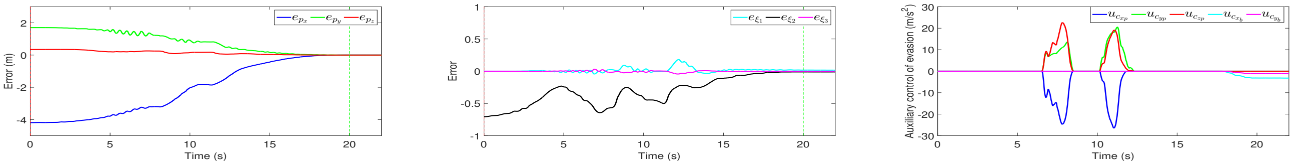
(c) Evasion auxiliary control signals.

Fig. 5: **Regulation task at a predefined time of 20 s with two static obstacles.**

rors of both robots, the vertical dotted red and green lines denote the beginning of a task and its convergence time, respectively. From the convergence time to the new task elapse the idle times. Figs. 10.(c) and 10.(f) show the evasion auxiliary control signals affecting both robots along their tasks execution. Notice that at the end of the last task, the controls counteracted the repulsive evasion control signals, such that they remained in steady state. In addition, Figs. 10.(g) and 10.(h) show the sliding surfaces, where most of the pose coordinates converged at predefined time, excepting an orientation coordinate whose error was diminished with more effort due to task conflicts with position coordinates, and the presence of a near obstacle at the end.

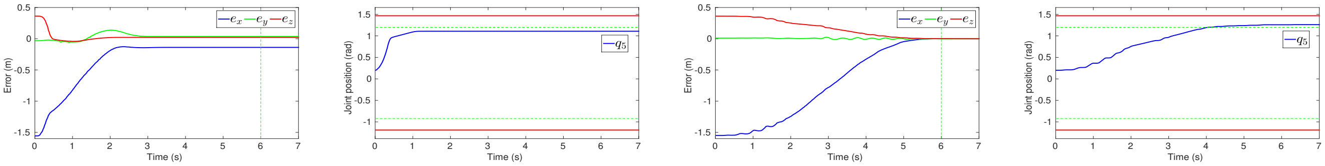
VII. Conclusions

We have proposed a robust hierarchical inverse dynamics (RHID) scheme that achieves predefined-time convergence of reaching tasks. The model dependency of the RHID requires robustness, which, in our proposal, is provided by a super-twisting control approach, as the controller unavoidably deals with robot model uncertainties and disturbances. We derived the stability proof for the multi-task closed-loop system includ-



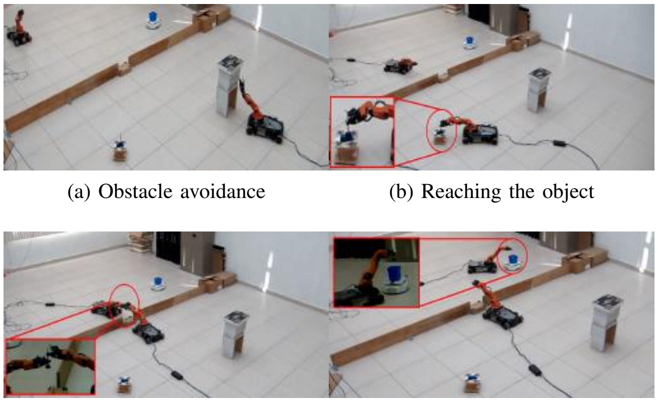
(a) End-effector position errors.
(b) End-effector orientation errors.
(c) Evasion auxiliary control signals.

Fig. 6: **Regulation task at a predefined time of 20 s while avoiding a moving obstacle.**



(a) End-effector position errors.
(b) Joint limits.
(c) End-effector position errors.
(d) Joint limits.

Fig. 7: **Comparison between hierarchical-quadratic-programming (HQP) based control and the proposed controller**. (a) and (b) profiles belong to the HQP while (c) and (d) profiles belong to the proposed control scheme.



(a) Obstacle avoidance
(b) Reaching the object
(c) Exchanging the object
(d) Droping the object

Fig. 8: **Main execution snapshots of the experiment**: (a) Robot 1 evaded an obstacle, (b) Robot 1 reached the object, (c) Robot 1 exchanged the object with Robot 2, (d) Robot 2 dropped the object in the container

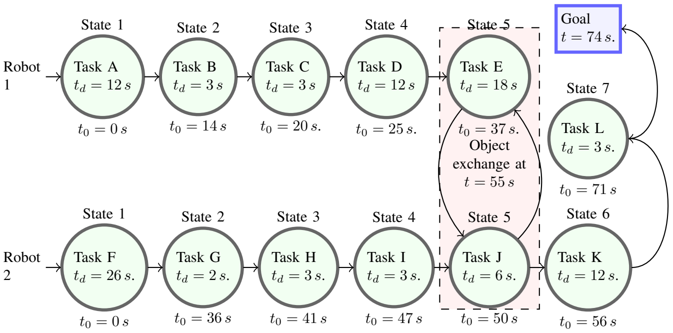
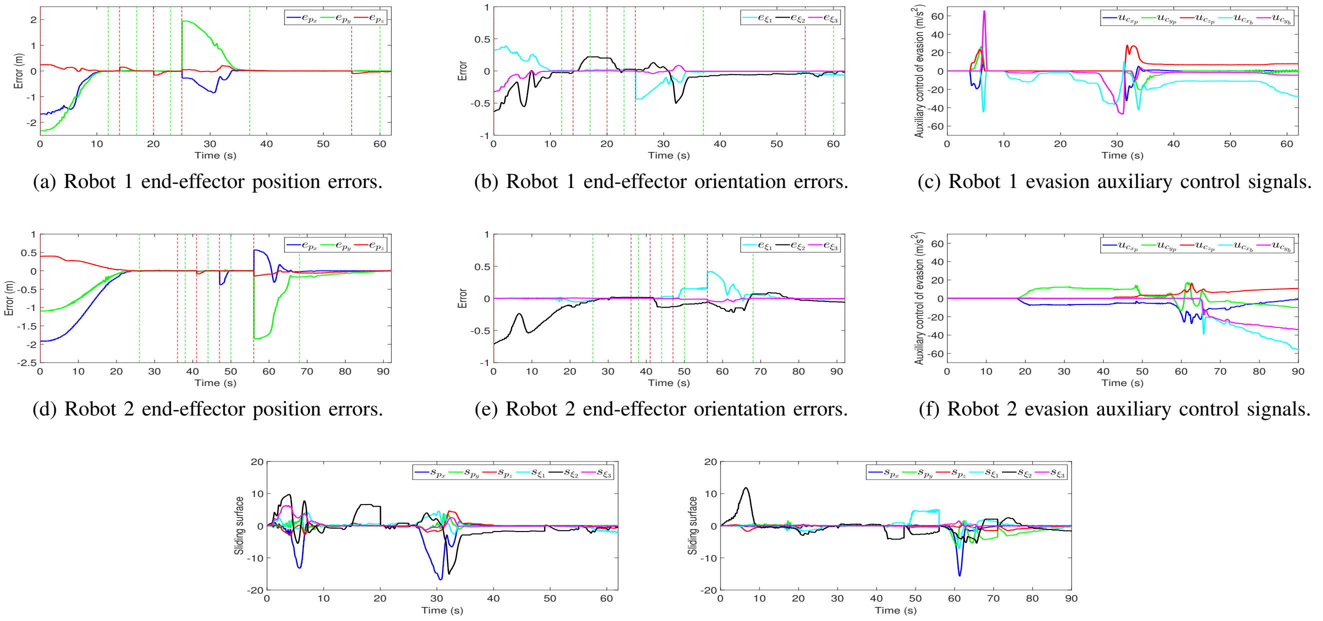


Fig. 9: **Finite state machine with a time parametrized task schedule for each robot.** Robot 1: in A the robot approached the object to be grasped, in B the end-effector reached the object, in C the object was lifted, in D the object is transported towards the meeting point, in E the grippers released the object while Robot 2 was holding the object. Robot 2: in F the robot moved close to the meeting point, in G the end-effector pose was corrected if necessary, in H the end-effector aligned with Robot 1, in I the robot reached the object, in J the object is grasped before being released by Robot 1, in K the object is transported to the container. Finally in L the end-effector pose was corrected, and the object is dropped into the container. The total execution time was $t = 74\,s$. The desired time for the tasks is given by t_d, which denotes the convergence time for all task in the stack, excepting E, J and L, where the grippers were activated at some t_d without a motion task. The initial time for each task is given by t_0, which considers the times of previous tasks and idle times t_w.

ing the super-twisting controller, and considering uncertainty in the robot parameters and disturbances. Some properties of dynamically-consistent null-space projectors were deduced and exploited along the proof, such that it became possible to simplify the multi-task closed-loop system in the non-ideal case (i.e. under model uncertainties), which, written in matrix form, leaves a predictable upper-triangular form for any number of hierarchical tasks. This also allowed us to guarantee convergence of task errors at a predefined time, defined as a parameter of the control scheme, independently of the initial robot state and despite of the execution of several non-conflicting tasks. The effectiveness of the proposed RHID scheme has been evaluated by means of experiments using torque-controlled 8-DoF mobile manipulators. We assigned the robots several tasks that demanded high control performance. In particular, for picking and placing objects in the environment. Regulation tasks were defined to reach desired positions and orientations for the robot's end-effector while avoiding obstacles and joint limits. The obstacle avoidance task was formulated at acceleration level, such that we obtained repulsive accelerations that served as auxiliary control signals. Currently,



(a) Robot 1 end-effector position errors. (b) Robot 1 end-effector orientation errors. (c) Robot 1 evasion auxiliary control signals.

(d) Robot 2 end-effector position errors. (e) Robot 2 end-effector orientation errors. (f) Robot 2 evasion auxiliary control signals.

(g) Robot 1 evolution of sliding surface. (h) Robot 2 evolution of sliding surface.

Fig. 10: **Collaborative robots to pick and place a common object with strict time schedule.**

we are improving the computational time to obtain the control law by making use of Cholesky and QR factorizations. We are also extending the RHID to handle contact tasks.

Appendix A

We provide the proof of the set of properties stated in (14)-(19), that follow the dynamic consistency of null-space projectors introduced in [2].

Proof: It is straightforward to prove that

$$\overline{Q}_i\overline{Q}_i^{\#_A} = \overline{Q}_i A\overline{Q}_i^T\left[\overline{Q}_i A\overline{Q}_i^T\right]^{-1} = I \qquad (81)$$

always holds for any $i = \{1,\dots,p\}$, which corresponds to (18) when $i = j$. Also, it is trivial to proof that N_i is idempotent when $i = 1$:

$$\begin{aligned} N_1N_1 &= \left(N_0 - \overline{Q}_1^{\#_A}\overline{Q}_1\right)\left(N_0 - \overline{Q}_1^{\#_A}\overline{Q}_1\right) \\ &= N_0 - \overline{Q}_1^{\#_A}\overline{Q}_1 \qquad (82) \end{aligned}$$

where $N_0 = I$. Therefore, it can be verified that $\overline{Q}_1N_0 = \overline{Q}_1$, $N_0\overline{Q}_1^{\#_A} = \overline{Q}_1^{\#_A}$ and $N_1N_0 = N_0N_1 = N_1$. By using (81), it is easy to prove that N_1 is dynamically consistent with respect to $\overline{Q}_1$ and $\overline{Q}_1^{\#_A}$:

$$\begin{aligned} \overline{Q}_1N_1 &= \overline{Q}_1N_0 - \overline{Q}_1\overline{Q}_1^{\#_A}\overline{Q}_1 = \overline{Q}_1 - \overline{Q}_1 = 0 \qquad (83) \\ N_1\overline{Q}_1^{\#_A} &= N_0\overline{Q}_1^{\#_A} - \overline{Q}_1^{\#_A}\overline{Q}_1\overline{Q}_1^{\#_A} \\ &= \overline{Q}_1^{\#_A} - \overline{Q}_1^{\#_A} = 0 \qquad (84) \end{aligned}$$

Knowing that N_1 is idempotent, it can be verified that $N_1AN_1^T = AN_1^T$ as follows:

$$\begin{aligned} N_1AN_1^T &= \left(N_0 - \overline{Q}_1^{\#_A}\overline{Q}_1\right)A\left(N_0^T - \overline{Q}_1^T\overline{Q}_1^{\#_A^T}\right) \\ &= \left(AN_0^T - A\overline{Q}_1^T\overline{Q}_1^{\#_A^T}\right)\left(N_0^T - \overline{Q}_1^T\overline{Q}_1^{\#_A^T}\right) \\ &= AN_1^TN_1^T = AN_1^T \end{aligned} \qquad (85)$$

where $AN_1^T = N_1A$. From (82) and (85), it is deduced $\overline{Q}_2N_1 = \overline{Q}_2$ and $N_1\overline{Q}_2^{\#_A} = \overline{Q}_2^{\#_A}$:

$$\begin{aligned} \overline{Q}_2N_1 &= Q_2N_1N_1 = Q_2N_1 = \overline{Q}_2 \qquad (86) \\ N_1\overline{Q}_2^{\#_A} &= N_1A\overline{Q}_2^T\left[\overline{Q}_2A\overline{Q}_2^T\right]^{-1} = \overline{Q}_2^{\#_A} \qquad (87) \end{aligned}$$

where $N_1A\overline{Q}_2^T = N_1AN_1^TQ_2^T = AN_1^T\overline{Q}_2^T = A\overline{Q}_2^T$. From (81), (82), (86) and (87), it is then verified that N_2 is dynamically consistent with respect to $\overline{Q}_2$ and $\overline{Q}_2^{\#_A}$

$$\begin{aligned} \overline{Q}_2N_2 &= \overline{Q}_2N_1 - \overline{Q}_2\overline{Q}_2^{\#_A}\overline{Q}_2 = \overline{Q}_2 - \overline{Q}_2 = 0 \qquad (88) \\ N_2\overline{Q}_2^{\#_A} &= N_1\overline{Q}_2^{\#_A} - \overline{Q}_2^{\#_A}\overline{Q}_2\overline{Q}_2^{\#_A} \\ &= \overline{Q}_2^{\#_A} - \overline{Q}_2^{\#_A} = 0 \qquad (89) \end{aligned}$$

Also, it can be verified that N_2 is idempotent:

$$\begin{aligned} N_2N_2 &= N_1N_1 - \overline{Q}_2^{\#_A}\overline{Q}_2N_1 - N_1\overline{Q}_2^{\#_A}\overline{Q}_2 \qquad (90) \\ &\quad + \overline{Q}_2^{\#_A}\overline{Q}_2N_1\overline{Q}_2^{\#_A}\overline{Q}_2 \\ &= N_1 - \overline{Q}_2^{\#_A}\overline{Q}_2 = N_2 \end{aligned}$$

Then, the following holds

$$\begin{aligned} N_1N_2 &= N_0N_1 - \overline{Q}_1^{\#_A}\overline{Q}_1N_1 - N_0\overline{Q}_2^{\#_A}\overline{Q}_2 \qquad (91) \\ &\quad + \overline{Q}_1^{\#_A}\overline{Q}_1\overline{Q}_2^{\#_A}\overline{Q}_2 \\ &= N_1 - \overline{Q}_2^{\#_A}\overline{Q}_2 = N_2 \end{aligned}$$

and

$$\begin{aligned} N_2 N_1 &= N_1 N_0 - N_1 \overline{Q}_1^{\#_A} \overline{Q}_1 - \overline{Q}_2^{\#_A} \overline{Q}_2 N_0 \\ &\quad + \overline{Q}_2^{\#_A} \overline{Q}_2 \overline{Q}_1^{\#_A} \overline{Q}_1 \\ &= N_1 - \overline{Q}_2^{\#_A} \overline{Q}_2 = N_2 \end{aligned} \tag{92}$$

where $\overline{Q}_1 \overline{Q}_2^{\#_A} = \overline{Q}_1 N_1 \overline{Q}_2^{\#_A}$ and $\overline{Q}_2 \overline{Q}_1^{\#_A} = \overline{Q}_2 N_1 \overline{Q}_1^{\#_A} = 0$, which obeys (18).

Note that in order to prove that both N_2 is idempotent and its dynamic consistency, it was necessary to first prove that N_1 is idempotent, which implies that properties (86) and (87) hold. Consequently (88), (89), (90), (91) and (92) also hold. Similarly, for proving that N_3 is idempotent and its dynamic consistency, it must be first verified that N_2 is idempotent. Then, it holds that $\overline{Q}_3 N_2 = Q_3 N_2 N_2 = Q_3 N_2 = \overline{Q}_3$ and $N_2 \overline{Q}_3^{\#_A} = \overline{Q}_3^{\#_A}$, since $N_2 A N_2^T = A N_2^T$. Also, the dynamic consistency of N_3 with respect to $\overline{Q}_3$ is verified as in (88) and (89). From (90) it is verified that N_3 is idempotent. By applying the same derivation in (91) and (92), it is deduced:

$$N_2 N_3 = N_3 \quad \text{and} \quad N_3 N_2 = N_3$$

Therefore, by applying the following equivalences:

$$N_3 N_1 = (N_3 N_2) N_1 = N_3 (N_2 N_1) = N_3 N_2 = N_3 \tag{93}$$

$$N_1 N_3 = N_1 (N_2 N_3) = (N_1 N_2) N_3 = N_2 N_3 = N_3 \tag{94}$$

a recursive procedure can be implemented to verify that null-space projectors are idempotent together with their dynamic consistency. Without loss of generality, we set (82), (83), (84), and (85) as our base case, with $i = 1$ and $N_0 = I$ such that $N_1 N_1 = N_1$ is verified directly. Then, the recursion is as follows:

$$\overline{Q}_i N_{i-1} = \overline{Q}_i \tag{95}$$

$$N_{i-1} A N_{i-1}^T = A N_{i-1}^T \tag{96}$$

$$N_{i-1} \overline{Q}_i^{\#_A} = \overline{Q}_i^{\#_A} \tag{97}$$

$$\overline{Q}_i N_i = 0 \tag{98}$$

$$N_i \overline{Q}_i^{\#_A} = 0 \tag{99}$$

$$N_i N_i = N_i \tag{100}$$

$$\left(\prod_{j<i}^{i-1} N_j \right) N_i \left(\prod_{j<i}^{i-1} N_j \right) = N_i \tag{101}$$

for $i = 2, 3, \ldots, p$. To state that (95), (96), (97) and (100) hold for any $i > 1$, it is required to prove that N_{i-1} is idempotent, which is not a problem when starting from the base case. By considering (93) and (94), it can be noted that both (100) and (101) satisfy (14) for $i = j$ and $i \neq j$, also that (98) and (99) satisfy (15) and (16) respectively for $i \leq j$. Consequently both (95) and (98) are together analogous to (15), while both (97) and (99) are together analogous to (16). In fact, it can be noted that (15) and (16) are a consequence of (14), whereas (18) is a consequence of (15) and (16). On the other hand, the particular property (19) also corresponds to (81) when $i = j$ because of the following equivalence:

$$Q_i \overline{Q}_i^{\#_A} = Q_i \left(N_{i-1} \overline{Q}_i^{\#_A} \right) = (Q_i N_{i-1}) \overline{Q}_i^{\#_A} = \overline{Q}_i \overline{Q}_i^{\#_A} \tag{102}$$

Also note that (19) is also a consequence of (15) and (16) when $i < j$, because of (102)

$$Q_i N_i = Q_i N_{i-1} - Q_i \overline{Q}_i^{\#_A} \overline{Q}_i = \overline{Q}_i - \overline{Q}_i = 0 \tag{103}$$

However, the term $Q_i \overline{Q}_j^{\#_A}$ cannot be reduced for $i > j$ as $Q_{i+1} \overline{Q}_i^{\#_A} = Q_{i+1} \left(N_{i-1} \overline{Q}_i^{\#_A} \right)$ since $Q_{i+1} N_{i-1} \neq \overline{Q}_{i+1}$. ■

Appendix B

Under the assumptions stated in Proposition 6.1, it is proved that a well defined J_b^+ exists. Next, it is proved that the equations of motion of an omni-directional mobile manipulator can be expressed as those of inertial robots in generalized coordinates. As a consequence, the proposed control scheme can be applied to inertial and omni-directional mobile manipulator.

Proof: Let us consider the operator that maps coordinates from actuation to generalized spaces $J_b : \mathbb{R}^a \to \mathbb{R}^n$ as stated in (80) of Proposition 6.1, such that

$$\overline{F}_b = J_b \tau_b \tag{104}$$

$$\begin{bmatrix} f_x \\ f_y \\ n_z \end{bmatrix} = J_b \begin{bmatrix} \tau_{b_1} \\ \tau_{b_2} \\ \tau_{b_3} \\ \tau_{b_4} \end{bmatrix} \tag{105}$$

where τ_b are the torques of the wheels and $\overline{F}_b$ represents the generalized torques of the mobile base. On the other hand, there are three acting forces for each wheel: the contact forces f_{c_i}, the driven-torques produced by the motors $\tau_{x_i} = \tau_{b_i}$, and the free-sliding-torques due to the roller $\tau_{y_i} = \tau_{b_i} \tan\gamma$, where γ is the angle of the free-sliding-motion direction. Therefore, the acting wrench of each wheel becomes:

$$F_{w_i} = \begin{bmatrix} f_{x_i} & f_{y_i} & f_{z_i} & n_{x_i} & n_{y_i} & n_{z_i} \end{bmatrix}^T \tag{106}$$

$$= \begin{bmatrix} \frac{1}{r}\tau_{x_i} & \frac{1}{r}\tau_{y_i} & f_{c_i} & 0 & 0 & 0 \end{bmatrix}^T \tag{107}$$

where $i \in \{1, 2, 3, 4\}$. Now, from equations (106) and (107) we define the following operators:

$$\begin{bmatrix} J_{x_1} & J_{x_2} & J_{x_3} & J_{x_4} \end{bmatrix} \begin{bmatrix} \tau_{x_1} \\ \tau_{x_2} \\ \tau_{x_3} \\ \tau_{x_4} \end{bmatrix} = J_x \tau_x \tag{108}$$

where J_{x_i} contains the coordinate transformations to express τ_{x_i}, given in (107), in the mobile-base frame. The same is performed with J_{y_i} and τ_{y_i}, also with C_i and f_{c_i} to obtain the mobile-base wrench as

$$F_b = J_x \tau_x + J_y \tau_y + C f_c \in \mathbb{R}^6 \tag{109}$$

where $J_x, J_y, C \in \mathbb{R}^{6\times 4}$. Knowing that $\tau_x = \tau_b$ and $\tau_y = \tau_b \tan\gamma$, we rearrange (109) as follows

$$\begin{aligned} F_b &= J_x \tau_b + J_y \tan\gamma \tau_b + C f_c = (J_x + J_y \tan\gamma) \tau_b + C f_c \\ &= J \tau_b + C f_c \end{aligned} \tag{110}$$

Notice that $J\tau_b$ produces motion, and $C f_c$ produces contact forces only. Then, $C^T \nu_b = 0$, where $\nu_b \in \mathbb{R}^6$ is the twist of the mobile-base. The equations of motion of the mobile-base are given by

$$M \dot{\nu}_b + h_b = J \tau_b + C f_c \tag{111}$$

with $h_b \triangleq \text{ad}^*_{\nu_b}(M\nu_b + g)$, where $M \in \mathbb{R}^{6\times 6}$ is the body inertia matrix, $\dot{\nu}_b \in \mathbb{R}^6$ is the body acceleration, $g \in \mathbb{R}^6$ is the gravity wrench which is counteracted by Cf_c, and $\text{ad}^*_\nu \colon se(3) \times se^*(3) \to se^*(3)$ is the dual linear mapping:

$$\text{ad}^*_{\nu_b} = \begin{bmatrix} -\widehat{\omega_b} & 0 \\ -\widehat{v_b} & -\widehat{\omega_b} \end{bmatrix} \tag{112}$$

where $\widehat{a} \in so(3)$, $\forall a \in \mathbb{R}^3$, is the skew-symmetric matrix. The reduced free part of (111) can be obtained as follows

$$Y = \text{span}(I - CC^+) \qquad Z = \text{span}\left(CC^+\right) \tag{113}$$

where the rank of C, expressed as $\rho(C)$, represents the dimension of constrained motions, and the number of available degrees of freedom is $n_b = 6 - \rho(C)$. Now, we define $X = [Y\,Z]$ where $X \in \mathbb{R}^{6\times 6}$ and $\rho(X) = 6$. It can be verified that $XX^T = I_6$ and $Y \in \mathbb{R}^{6\times n_b}$, $Z \in \mathbb{R}^{6\times\rho(C)}$ span the column-space of the unconstrained and constrained spaces, respectively. By using X in (111), the equations of motion become

$$\begin{aligned} X^T(M\dot{\nu}_b + h_b) &= X^T(J\tau_b + Cf_c) \\ X^TMXX^T\dot{\nu}_b + X^Th_b &= X^TJ\tau_b + X^TCf_c. \end{aligned} \tag{114}$$

From (114), the reduced free part of (111) is

$$Y^TMYY^T\dot{\nu}_b + Y^T\left(MZZ^T\dot{\nu}_b + h_b\right) = Y^T\left(J\tau_b + Cf_c\right) \tag{115}$$

From (113), $Z^T\dot{\nu}_b = 0$ and $Y^TC = 0$ hold. Therefore, (115) can be rewritten as

$$\begin{aligned} Y^TMYY^T\dot{\nu}_b + Y^Th_b &= Y^TJ\tau_b \\ \overline{M}\dot{\overline{\nu}}_b + \overline{h}_b &= J_b\tau_b \\ \overline{F}_b &= J_b\tau_b \end{aligned} \tag{116}$$

where $Y^TMY \triangleq \overline{M} \in \mathbb{R}^{n_b\times n_b}$, $Y^T\dot{\nu}_b \triangleq \dot{\overline{\nu}}_b \in \mathbb{R}^{n_b}$, $Y^Th_b \triangleq \overline{h}_b \in \mathbb{R}^{n_b}$ and $Y^TJ \triangleq J_b \in \mathbb{R}^{n_b\times 4}$ where $\rho(J_b) = n_b$, then J_b is full rank, such that

$$\tau_b = J_b^+\overline{F}_b \tag{117}$$

is well-defined. Notice that Y reduced (111) to the unconstrained space of the mobile-base, and J_b is then full-rank in the unconstrained space.

Let us now consider the mobile-manipulator system such that the acceleration mappings between the generalized and actuation spaces are given by

$$\begin{aligned} \ddot{q}_a &= \begin{bmatrix} J_b^T & 0 \\ 0 & I \end{bmatrix}\ddot{q} \\ &= S^T\ddot{q}. \end{aligned} \tag{118}$$

It holds that $\tau = S\tau_a$, $\tau_a = S^+\tau$, and $\ddot{q} = \left[S^T\right]^+\ddot{q}_a$. By applying the previous transformations in (4), it is deduced the expression of the actuated torques as

$$\tau_a = \left[S^TA^{-1}S\right]^{-1}\left(\ddot{q}_a + S^TA^{-1}b\right). \tag{119}$$

Note that (119) represents the equations of motion in the actuation space, where $\left[S^TA^{-1}S\right]^{-1} \in \mathbb{R}^{a\times a}$ is the inertia matrix in such space.

Since torques and accelerations can be mapped between generalized and actuation spaces, we transform (119) back to the generalized space as

$$\begin{aligned} S^+\tau &= \left[S^TA^{-1}S\right]^{-1}\left(S^T\ddot{q} + S^TA^{-1}b\right) \\ \tau &= S\left[S^TA^{-1}S\right]^{-1}S^T\left(\ddot{q} + A^{-1}b\right). \end{aligned}$$

It happens that $S\left[S^TA^{-1}S\right]^{-1}S^T = A$, which allows to get

$$\tau = A\ddot{q} + b \tag{120}$$

■

Notice that operators S and S^+ make (119) and (120) equivalent. The proposed control scheme works in generalized coordinates with inertial and mobile manipulators. The generalized torques are then transformed to the actuation space to move the robot in torque-mode as stated in Section VI.A.

References

[1] 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.

[2] ——, "Inertial properties in robotic manipulation: An object-level framework," *The international journal of robotics research*, vol. 14, no. 1, pp. 19–36, 1995.

[3] A. Dietrich, C. Ott, and A. Albu-Schäffer, "An overview of null space projections for redundant, torque-controlled robots," *The International Journal of Robotics Research*, vol. 34, no. 11, pp. 1385–1400, 2015.

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

[5] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Operational space control: A theoretical and empirical comparison," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 737–757, 2008.

[6] A. Polyakov, D. Efimov, and W. Perruquetti, "Finite-time and fixed-time stabilization: Implicit lyapunov function approach," *Automatica*, vol. 51, pp. 332–340, 2015.

[7] Z. Zuo and L. Tie, "Distributed robust finite-time nonlinear consensus protocols for multi-agent systems," *Int. J. of Syst. Science*, vol. 47, no. 6, pp. 1366–1375, 2016.

[8] D. Meng and Z. Zuo, "Signed-average consensus for networks of agents: a nonlinear fixed-time convergence protocol," *Nonlinear Dyn.*, vol. 85, pp. 155–165, 2016.

[9] H. M. Becerra, C. R. Vázquez, G. Arechavaleta, and J. Delfin, "Predefined-time convergence control for high-order integrator systems using time base generators," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1866–1873, 2018.

[10] Y. Song, Y. Wang, J. Holloway, and M. Krstic, "Time-varying feedback for regulation of normal-form nonlinear systems in prescribed finite time," *Automatica*, 2017.

[11] Y. Zhao, Y. Liu, G. Wen, W. Ren, and G. Chen, "Designing distributed specified-time consensus protocols for linear multiagent systems over directed graphs," *IEEE Trans. on Automatic Control*, vol. 64, no. 7, pp. 2945–2952, 2019.

[12] A. J. Muñoz-Vazquez, J. D. Sanchez-Torres, E. Jimenez-Rodriguez, and A. G. Loukianov, "Predefined-time robust stabilization of robotic manipulators," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 3, pp. 1033–1040, 2019.

[13] G. Arechavaleta, J. Obregón, H. M. Becerra, and A. Morales-Díaz, "Predefined-time convergence in task-based inverse dynamics using time base generators," *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 443–449, 2018.

[14] D. J. Braun, Y. Chen, and L. Li, "Operational space control under actuation constraints using strictly convex optimization," *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 302–309, 2020.

[15] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soueres, and J.-Y. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 346–362, 2013.

[16] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 670–685, 2009.

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

[18] B. Siciliano and J.-J. Slotine, “A general framework for managing multiple tasks in highly redundant robotic systems,” in *IEEE International Conference on Advanced Robot*, Pisa, Italy, June 1991, pp. 1211–1216.

[19] P. Baerlocher and R. Boulic, “An inverse kinematic architecture enforcing an arbitrary number of strict priority levels,” *The Visual Computer: International Journal of Computer Graphics*, vol. 20, no. 6, pp. 402–417, August 2004.

[20] P. H. Chang and J. W. Jeong, “Enhanced operational space formulation for multiple tasks by using time-delay estimation,” *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 773–786, 2012.

[21] A. Del Prete and N. Mansard, “Robustness to joint-torque-tracking errors in task-space inverse dynamics,” *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1091–1105, 2016.

[22] C. Ott, A. Dietrich, and A. Albu-Schäffer, “Prioritized multi-task compliance control of redundant manipulators,” *Automatica*, vol. 53, pp. 416–423, 2015.

[23] A. Dietrich, C. Ott, and J. Park, “The hierarchical operational space formulation: Stability analysis for the regulation case,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1120–1127, April 2018.

[24] A. Dietrich and C. Ott, “Hierarchical impedance-based tracking control of kinematically redundant robots,” *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 204–221, 2020.

[25] J. Peters and S. Schaal, “Learning to control in operational space,” *The Int. Journal of Robotics Research*, vol. 27, no. 2, pp. 197–212, 2008.

[26] B. Xiao and S. Yin, “Exponential tracking control of robotic manipulators with uncertain dynamics and kinematics,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 689–698, Feb 2019.

[27] L. E. Gonzlez-Jimnez, A. Loukianov, and E. Bayro-Corrochano, “Fully nested super-twisting algorithm for uncertain robotic manipulators,” in *2011 IEEE Int. Conf. on Robotics and Automation*, May 2011, pp. 5807–5812.

[28] Y. Kali, M. Saad, and K. Benjelloun, “Optimal super-twisting algorithm with time delay estimation for robot manipulators based on feedback linearization,” *Robotics and Autonomous Systems*, vol. 108, pp. 87 – 99, 2018.

[29] V. Utkin, “On convergence time and disturbance rejection of super-twisting control,” *IEEE Trans. Autom. Control*, vol. 58, no. 8, pp. 2013–2017, 2013.

[30] M. Galicki, “Finite-time control of robotic manipulators,” *Automatica*, vol. 51, pp. 49 – 54, 2015.

[31] ——, “Constraint finite-time control of redundant manipulators,” *International Journal of Robust and Nonlinear Control*, vol. 27, no. 4, pp. 639–660, 2017.

[32] C. Yang, Y. Jiang, W. He, J. Na, Z. Li, and B. Xu, “Adaptive parameter estimation and control design for robot manipulators with finite-time convergence,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 8112–8123, Oct 2018.

[33] L. Zhang, L. Liu, Z. Wang, and Y. Xia, “Continuous finite-time control for uncertain robot manipulators with integral sliding mode,” *IET Control Theory Applications*, vol. 12, no. 11, pp. 1621–1627, 2018.

[34] Y. Su and C. Zheng, “Fixed-time inverse dynamics control for robot manipulators,” *ASME. J. Dyn. Sys., Meas., Control*, 2019.

[35] A. Polyakov, “Nonlinear feedback design for fixed-time stabilization of linear control systems,” *IEEE Trans. Autom. Control*, vol. 57, no. 8, pp. 2106–2110, 2012.

[36] C. Samson, M. L. Borgne, and B. Espiau, *Robot Control: The Task Function Approach*, 1st ed., ser. Oxford Engineering Science Series. New York, USA: Oxford University Press, 1991, vol. 22.

[37] J. A. Moreno and M. Osorio, “Strict Lyapunov functions for the super-twisting algorithm,” *IEEE Trans. on Automatic Control*, vol. 57, no. 4, pp. 1035–1040, April 2012.

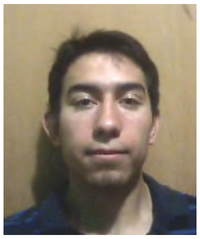
[38] M. W. Spong and M. Vidyasagar, “Robust linear compensator design of nonlinear robotic control,” *IEEE Journal on Robotics and Automation*, vol. RA-3, no. 4, pp. 345–351, 1987.

[39] A. Chalanga and F. Plestan, “Finite time stabilization of an uncertain chain of integrators by integral sliding mode approach,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9613–9618, 2017.

[40] L. Fridman and A. Levant, “Higher order sliding modes,” in *Sliding Mode Control in Engineering*, W. Perruqueti and J. Barbot, Eds. Marcel Dekker, 2002, ch. 3, pp. 53–101.

[41] J. Lee, N. Mansard, and J. Park, “Intermediate desired value approach for task transition of robots in kinematic control,” *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1260–1277, December 2012.

[42] R. Featherstone, “A beginner’s guide to 6-d vectors (part 1),” *IEEE robotics & automation magazine*, vol. 17, no. 3, pp. 83–94, 2010.



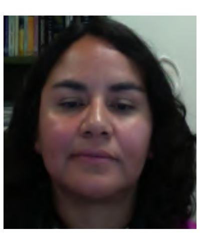
Jonathan Obregón-Flores received a B.Sc. degree in mechatronics engineering from Tecnológico Nacional de Mxico, campus Saltillo, and a M.Sc. degree in robotics and advanced manufacturing from CINVESTAV, Saltillo, Mexico, in 2013, 2017, respectively. He is currently a PhD student at CINVESTAV, Saltillo, Mexico.

Gustavo Arechavaleta received his M.S. degree from Tecnolgico de Monterrey (ITESM), Campus Estado de Mexico, in 2003, and his Ph.D. degree from Institut National des Sciences Appliquées de Toulouse, France, for his work on the computational principles of human walking via optimal control within the Gepetto team at LAAS-CNRS, Toulouse, France, in 2007.

In Fall 2008 he joined the Robotics and Advanced Manufacturing Group at CINVESTAV, Saltillo, Mexico, where he is a Research Scientist. In Fall 2015 he spent a year in the Coordinated Science Laboratory at the University of Illinois at Urbana-Champaign as a visiting researcher. His research interests are mainly focused on robot trajectory optimization and vision-based robot navigation.

Héctor M. Becerra (M’08) received a B.Sc. degree in electronics engineering from the Tecnológico Nacional de México, campus Ciudad Guzmán, a M.Sc. degree in Automatic Control from CINVESTAV-Guadalajara, Mexico, and a Ph.D. degree in systems engineering and computer science from the Universidad de Zaragoza, Spain, in 2003, 2005 and 2011, respectively.

He is currently a full researcher at Centro de Investigación en Matemáticas, CIMAT-Guanajuato, Mexico. He is also member of the National System of Researchers, CONACyT, Mexico. His research interests include applications of control theory to robotics, particularly the use of computer vision as main sensory modality for feedback control of wheeled, humanoid and aerial robots, as well as control of multi-agent systems.



América Morales-Díaz received the B.Sc. degree in chemical-petroleum engineering from Instituto politécnico Nacional, México in 1994, the M.Sc. and Ph.D. degrees in chemical engineering from Universidad Autónoma Metropolitana, México, in 1998 and 2001, respectively.

Since 2005 she has been working as a researcher in the Robotics and Advanced Manufacturing program at CINVESTAV, México. She has published 28 papers in specialized magazines. Her interests include nonlinear control systems with applications to industrial process and mobile robots cooperation.