

# Hierarchical Task-Based Control of Multirobot Systems With Terminal Attractors

Gustavo Arechavaleta, *Member, IEEE*, América Morales-Díaz, Héctor M. Pérez-Villeda, and Mario Castelán

**Abstract**—This brief proposes a hierarchical control scheme based on the definition of a set of multirobot task functions. To deal with the inherent conflicts between tasks, a strict hierarchy is imposed on them. We present a novel scheme that copes with two main difficulties shared in standard task-based controllers: 1) to impose a desired time convergence of tasks and 2) to avoid discontinuous task transitions occurred when a task is inserted or removed in the hierarchical structure. As a result, continuous input references are generated for the low-level control of the group. The validation is achieved in simulation and by performing an experiment with wheeled mobile robots.

**Index Terms**—Multirobot systems, task-based control.

## I. INTRODUCTION

THE motion generation and the control of a group of robots have been widely studied in the robotics and control communities [1], [2]. Several formation control schemes have been developed [3]. These schemes can be classified in distributed and centralized approaches. The former implies that the data of each element are partially shared with a subset of the group for control purposes. The interconnections in linear controllers are managed by Laplacian matrices. This approach is commonly called consensus [4], [5]. These algorithms model the behavior of each robot as a weighted average of its neighbor state. The main objective is to reach a common agreement based on different types of graph topologies.

Recently, the task function approach, introduced in [6], has been used to develop a controller–observer scheme for tracking the formation centroid [7]. Similar to consensus, the behavior of each robot is governed by a single-integrator kinematics, and the information exchange topology is based on the Laplacian matrix. However, different from the consensus algorithms, the control objectives are formulated by means of task functions. An important feature of such approach is the definition of hierarchical tasks to achieve individual and collective behaviors [8]. Other types of hierarchies have been proposed in the context of multirobot systems. In [9], a hierarchical distributed controller deals with some formation constraints, such as velocity and curvature. In [10], a hierarchical biologically inspired model is suggested to deal with the task assignment problem for a distributed group of robots.

Manuscript received September 9, 2015; revised January 14, 2016; accepted March 12, 2016. Manuscript received in final form March 25, 2016. Recommended by Associate Editor A. G. Aghdam.

The authors are with the Robotics and Advanced Manufacturing Group, Centro de Investigación y de Estudios Avanzados del IPN, Saltillo 25903, México (e-mail: garechav@cinvestav.edu.mx; america.morales@cinvestav.edu.mx; hecvilleda@gmail.com; fmmusgo3@gmail.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCST.2016.2549279

In centralized schemes, the data of the whole system are completely handled by a single controller. Some controllers rely on the leader–follower paradigm. In [11], a feedback linearization controller is suggested to exponentially stabilize the relative distance and orientation of the followers. In [12], the information about the group trajectory is handled by the leader. A model-independent coordination strategy is introduced in [13]. Recently, the navigation function method has been extended to consider multiple disk-shaped robots, each one with independent goal positions [14]. Most of the above-mentioned control schemes allow to coordinate particular behaviors of multirobot systems. In this sense, the task-based controller introduced in [8] represents a versatile scheme for achieving generic group behaviors. Some experiments with wheeled mobile robots have been reported in [15]. However, the hierarchy of tasks cannot be arbitrarily modified during the motion execution. Otherwise, discontinuities in the input signals appear [16].

In addition, a variety of multirobot applications are subject to final time constraints, but most of the task-based controllers use exponentially decreasing task functions. This also causes discontinuities in the control law when the task is abruptly activated. Since the signals generated with these controllers are used as input commands to each robot, their continuity and smoothness play important roles.

In this context, this brief proposes a task-based control scheme for multirobot systems relying on terminal attractors for enforcing a desired time convergence of task errors while preserving the continuity of the control signals. This is important since the time at which hierarchical tasks should be completed may determine the success of the mission assigned to multirobot systems.

The first contribution of this brief is to build a time-varying gain for controlling the time convergence of hierarchical tasks independently of initial conditions of each robot and without causing discontinuities in their input signals. The stability analysis provided in this brief considers the case of time-varying gains. As a second contribution, we describe how hierarchical kinematic controllers capable of handling smooth task transitions, such as [17], can be naturally adapted to ensure desired time convergence for all tasks. We also provide the stability analysis during task transitions and verify the success of the proposed control scheme with some experiments using a set of wheeled mobile robots.

This brief is organized as follows. In Section II, the hierarchical task-based control of multirobot systems is briefly recalled. In Section III, a class of terminal attractor is introduced for handling the desired time convergence of tasks without causing discontinuities in the control signals.

In Section IV, the discontinuities produced by task transitions are solved by adapting the intermediate desired value strategy with the proposed terminal attractor. Section V presents the simulation and experimental results with a set of wheeled mobile robots. Finally, concluding remarks are given in Section VI.

## II. HIERARCHICAL TASK-BASED CONTROL FOR MULTIPLE ROBOTS

The task-based control is a centralized motion generation scheme to accomplish team objectives. The current state of the task uses the position variables of all robots. In general, a task function is defined in [6]

$$\mathbf{e}(\mathbf{q}) = \mathbf{x}(\mathbf{q}) - \mathbf{x}^d \quad (1)$$

where  $\mathbf{e}(\mathbf{q}) \in \mathbb{R}^m$  establishes the difference between the current value of  $\mathbf{x}(\mathbf{q}) \in \mathbb{R}^m$  and its reference  $\mathbf{x}^d \in \mathbb{R}^m$ . Note that  $\mathbf{x}(\mathbf{q})$  depends on the system configuration, which is composed by the location of each robot with respect to an inertial reference frame

$$\mathbf{q} = (\mathbf{q}_1^T \quad \mathbf{q}_2^T \quad \dots \quad \mathbf{q}_N^T)^T \in \mathbb{R}^n \quad (2)$$

where  $N$  is the number of robots and  $\mathbf{q}_i = (x_i, y_i) \in \mathbb{R}^2$ ,  $\forall i \in [1, \dots, N]$ . By differentiating (1) with respect to time, we obtain the following linear system:

$$\dot{\mathbf{e}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad (3)$$

where  $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{m \times n}$  is the task Jacobian which relates the system and task velocities. To solve (3), the following quadratic program (QP) is formulated:

$$\begin{aligned} \min_{\dot{\mathbf{q}}} \quad & \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{q}} \\ \text{s.t.} \quad & \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} - \dot{\mathbf{e}}(\mathbf{q}) = 0. \end{aligned} \quad (4)$$

The time behavior of the task error can be defined by imposing an exponential convergence, such that

$$\dot{\mathbf{e}}(\mathbf{q}) = -\alpha \mathbf{e}(\mathbf{q}) \quad (5)$$

where  $\alpha > 0$  is a positive constant. According to (4) and (5), the input velocity reference becomes

$$\dot{\mathbf{q}} = -\alpha \mathbf{J}^+(\mathbf{q})\mathbf{e}(\mathbf{q}) \quad (6)$$

where  $\mathbf{J}^+ = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}$  represents the right pseudoinverse of matrix  $\mathbf{J} \in \mathbb{R}^{m \times n}$ , for  $m < n$ .

### A. Hierarchical Quadratic Programming Formulation

Since the number of constraints to represent one task is commonly less than the number of degrees of freedom (DoF), it is possible to establish a hierarchy of  $p$  QPs for solving the secondary tasks, as expressed in [18]

$$\begin{aligned} \min_{\dot{\mathbf{q}}_k \in \mathbb{R}^n, \mathbf{w}_k \in \mathbb{R}^{m_k}} \quad & \frac{1}{2} \mathbf{w}_k^T \mathbf{w}_k + \frac{1}{2} \lambda_k \dot{\mathbf{q}}_k^T \dot{\mathbf{q}}_k \\ \text{s.t.} \quad & \bar{\mathbf{J}}_k \dot{\mathbf{q}}_k - (\dot{\mathbf{e}}_k - \mathbf{J}_k \dot{\mathbf{q}}_{k-1}) = \mathbf{w}_k \end{aligned} \quad (7)$$

where  $\bar{\mathbf{J}}_k = \mathbf{J}_k \mathbf{Q}_{k-1}$ . Note that  $\mathbf{w}_k \in \mathbb{R}^{m_k}$  is used to relax the unfeasibility of constraints. In [19], the following recursion computes the solution of (7) for  $k = 1, \dots, p$ :

$$\begin{cases} \dot{\mathbf{q}}_0 = 0 \\ \dot{\mathbf{q}}_k = \dot{\mathbf{q}}_{k-1} + \bar{\mathbf{J}}_k^{+\lambda_k} (\mathbf{w}_k + \dot{\mathbf{e}}_k - \mathbf{J}_k \dot{\mathbf{q}}_{k-1}) \end{cases} \quad (8)$$

where  $\bar{\mathbf{J}}_k^{+\lambda_k} = \bar{\mathbf{J}}_k^T (\bar{\mathbf{J}}_k \bar{\mathbf{J}}_k^T + \lambda_k^2 \mathbf{I}_{m_k})^{-1}$  is a singularity robust pseudoinversion of  $\bar{\mathbf{J}}_k$  and the factor  $\lambda_k$  regulates this operation [20]. The projection operator  $\mathbf{Q}_k$  can be efficiently updated as [21]

$$\mathbf{Q}_k = \mathbf{Q}_{k-1} - \bar{\mathbf{J}}_k^+ \bar{\mathbf{J}}_k, \quad \mathbf{Q}_0 = \mathbf{I}. \quad (9)$$

However, if the system is characterized by a large number of DoF, then the computational cost of evaluating the pseudo-inversion and projector operators can be avoided with the use of a complete orthogonal decomposition [22]. It is also important to note that the stability analysis in the Lyapunov sense carried out in [23] concludes that the asymptotic convergence of task errors is ensured if the tasks are independent and the gains are positive.

### B. Task Functions for Multirobot Systems

In the context of multirobot systems, the task functions can be classified as local or global [15]. The difference relies on the required information to compute the task. Thus, the local task functions only need local information. The task Jacobians of the following task functions are provided in [8].

1) *Local Task Functions*: The task function to maintain the robot  $\mathbf{q}_i$  over a circumference is expressed as

$$e_c(\mathbf{q}_i) = \left( \frac{1}{2} (\mathbf{q}_i - \mathbf{x}_c)^T (\mathbf{q}_i - \mathbf{x}_c) \right) - \frac{r_i^2}{2} \quad (10)$$

where  $\mathbf{x}_c = (x_c \ y_c)^T \in \mathbb{R}^2$  is the center of the circle and  $r$  is the distance that each robot should maintain from  $\mathbf{x}_c$ .

The obstacle avoidance task acts per robot and is activated if the security distance  $d_s$  between the obstacle  $\mathbf{x}_o$  and the robot  $\mathbf{q}_i$  is violated. It is defined as

$$e_o(\mathbf{q}_i) = \|\mathbf{q}_i - \mathbf{x}_o\| - d_s \in \mathbb{R}. \quad (11)$$

The task function for reaching independent goal positions is simply defined as

$$\mathbf{e}_g(\mathbf{q}_i) = \mathbf{q}_i - \mathbf{x}_i^d \in \mathbb{R}^2 \quad (12)$$

where  $\mathbf{x}_i^d \in \mathbb{R}^2$ .

2) *Global Task Functions*: The mean value and the variance with respect to the robots' positions can be used to reach a formation. The task function of the mean value is

$$\mathbf{e}_m(\mathbf{q}) = \mathbf{x}_m(\mathbf{q}) - \mathbf{x}_m^d \quad (13)$$

where the mean value  $\mathbf{x}_m(\mathbf{q}) = (x_m(\mathbf{q}), y_m(\mathbf{q})) \in \mathbb{R}^2$  can be easily obtained with

$$\mathbf{x}_m(\mathbf{q}) = \frac{1}{N} \sum_{i=1}^N \mathbf{q}_i \quad (14)$$

and  $\mathbf{x}_m^d \in \mathbb{R}^2$  is the desired value. The mean value is used to compute the task function of the variance as

$$\mathbf{e}_v(\mathbf{q}) = \mathbf{x}_v(\mathbf{q}) - \mathbf{x}_v^d \quad (15)$$

where  $\mathbf{x}_v^d \in \mathbb{R}^2$  is the desired value and

$$\mathbf{x}_v(\mathbf{q}) = \frac{1}{N} \sum_{i=1}^N \left( \frac{(x_i - x_m(\mathbf{q}))^2}{(y_i - y_m(\mathbf{q}))^2} \right). \quad (16)$$

The task function to minimize the sum of distances among adjacent robots up to a desired value is

$$e_d(\mathbf{q}) = x_d(\mathbf{q}) - x_d^d \quad (17)$$

where

$$\begin{aligned} x_d(\mathbf{q}) &= \frac{1}{2} (\mathbf{q}_1 - \mathbf{q}_N)^T (\mathbf{q}_1 - \mathbf{q}_N) \\ &+ \frac{1}{2} \sum_{i=2}^N (\mathbf{q}_i - \mathbf{q}_{i-1})^T (\mathbf{q}_i - \mathbf{q}_{i-1}). \end{aligned} \quad (18)$$

### III. TIME PARAMETERIZATION OF MULTIROBOT TASKS

Commonly, an exponential convergence is used to regulate a task function of the form (1). As a consequence, a discontinuity occurs when the task becomes active. To avoid this undesired effect, we propose to adopt a terminal attractor that ensures desired time convergence of tasks while avoiding discontinuities at tasks activation. It is of the form

$$\alpha(t, t_0, t_f) = \alpha_0 + \frac{\dot{\zeta}(t, t_0, t_f)}{1 - \zeta(t, t_0, t_f) + \delta} \quad (19)$$

where  $0 < \delta < 0.1$ , and  $t_0$  and  $t_f > t_0$  are the initial and final times, respectively. The minimum value of  $\alpha(t, t_0, t_f)$  is reached at  $\alpha_0$ , which is a lower bound below the physical resolution of encoders and necessary for the stability proof. The function  $\zeta(t)$  is called time-base generator (TBG) [24]. The gain  $\delta$  is useful to avoid the indetermination at  $t \geq t_f$  when  $\alpha(t \geq t_f) = 0/\delta$ . The profiles of  $\zeta(t)$ ,  $\dot{\zeta}(t)$ , and  $\alpha(t)$  reach their final values at  $t = t_f$  [see Fig. 1(c)]. The function  $\alpha(t)$  is called TBG gain, and its value is negligible at the initial time and after the final time.

*Lemma 1:* The control law for a time parameterized task

$$\dot{\mathbf{q}} = -\alpha(t) \mathbf{J}^+(\mathbf{q}) \mathbf{e}(\mathbf{q}) \quad (20)$$

where  $\alpha(t)$  is a time-varying gain defined in (19), which guarantees finite-time convergence of (3) to the origin  $\mathbf{e} = \mathbf{0}$ .

*Proof:* An important aspect of (20) is that  $\alpha(t)$  causes that the closed-loop system becomes time-dependent

$$\dot{\mathbf{e}} = -\alpha(t) \mathbf{J}(\mathbf{q}) \mathbf{J}^+(\mathbf{q}) \mathbf{e}. \quad (21)$$

Consequently, the stability analysis should consider that (21) is a nonautonomous system with equilibrium point  $\mathbf{e} = \mathbf{0}$ . Let us define the following Lyapunov candidate function:

$$V = \frac{1}{2} \mathbf{e}^T \mathbf{e} \quad (22)$$

where  $V$  is a continuously differentiable, positive definite, and decrescent function, such that

$$W_1(\mathbf{e}) \leq V \leq W_2(\mathbf{e}) \quad (23)$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial \mathbf{e}} \dot{\mathbf{e}} \leq -W_3(\mathbf{e}) \quad (24)$$

are satisfied for all  $\mathbf{e} \in \mathbb{R}^m$ , where  $W_1(\mathbf{e})$ ,  $W_2(\mathbf{e})$ , and  $W_3(\mathbf{e})$  are the positive definite functions [25]. Since  $V$  does not explicitly depend on time, the first inequality becomes  $W_1(\mathbf{e}) = W_2(\mathbf{e}) = V$ . The term  $\partial V / \partial t$  is zero in (24) because  $V$  is not a function of time. Thus,  $W_3(\mathbf{e})$  should be chosen to show that  $\mathbf{e} = \mathbf{0}$  is asymptotically stable. We know that the time derivative of (22) is

$$\dot{V} = -\alpha(t) \mathbf{e}^T \mathbf{J} \mathbf{J}^+ \mathbf{e}.$$

In addition, we have that

$$\frac{\partial V}{\partial \mathbf{e}} \dot{\mathbf{e}} = \dot{V} = -\alpha(t) \mathbf{e}^T \mathbf{J} \mathbf{J}^+ \mathbf{e}. \quad (25)$$

Given that  $\alpha(t)$  satisfies a lower bound condition

$$\alpha(t) \geq \alpha_0 > 0. \quad (26)$$

From (26), we can set  $W_3(\mathbf{e}) = \alpha_0 \mathbf{e}^T \mathbf{J} \mathbf{J}^+ \mathbf{e}$ , such that

$$\frac{\partial V}{\partial \mathbf{e}} \dot{\mathbf{e}} = -\alpha(t) \mathbf{e}^T \mathbf{J} \mathbf{J}^+ \mathbf{e} \leq -\alpha_0 \mathbf{e}^T \mathbf{J} \mathbf{J}^+ \mathbf{e}. \quad (27)$$

Therefore, asymptotic stability is guaranteed if and only if  $\mathbf{J} \mathbf{J}^+ > 0$ , and in this case, the condition  $\alpha(t) \geq \alpha_0 > 0$  must be satisfied. Now, let us express the individual dynamics

$$\dot{e}_i(t) = -\alpha(t) e_i(t) \text{ for } i = 1, \dots, m.$$

By the separation of variables and integration, the solution of each error dynamics using (19) becomes

$$e_i(t) = c_0 e^{-\alpha_0(t-t_0)} (1 - \zeta(t) + \delta) \quad (28)$$

where  $c_0$  is found out evaluating the previous equation for  $t = t_0$ . Thus,  $c_0 = e(t_0) / (1 + \delta)$ . Replacing  $c_0$  in (28), we obtain

$$e_i(t) = e_i(t_0) e^{-\alpha_0(t-t_0)} \left( 1 - \frac{\zeta(t)}{1 + \delta} \right). \quad (29)$$

The exponential term represents a slow dynamics because  $\alpha_0$  is very small. However,  $1 - (\zeta(t) / (1 + \delta)) \rightarrow 0$  as  $t \rightarrow t_f$ . Thus, the shape of  $e(t)$  closely follows the profile of  $1 - \zeta(t)$ , and the convergence time is controlled by  $\alpha(t)$ . This means that  $e(t)$  is a smooth function that asymptotically converges to zero at  $t = t_f$  regardless of the initial condition  $e(t_0)$ , and the finite-time convergence of the error is ensured. ■

A simulation example is shown in Fig. 1. It can be noticed how the task of reaching a target position while avoiding an obstacle is achieved in a time interval of 2 s. Although the reaching task is able to be performed in a given time without an abrupt change in velocity at  $t_0 = 0$ , discontinuities in velocity still occur along the trajectory of the robot caused by the activation of the obstacle avoidance with higher hierarchical level than the reaching task.

### IV. SMOOTH TRANSITIONS OF MULTIROBOT TASKS

The main purpose of this scheme is twofold. First, the tasks must be accomplished at a predefined desired time without disturbing the hierarchy between them. Second, the activation and the removal of hierarchical tasks during motion execution do not have to produce discontinuities in the joint velocity profiles.

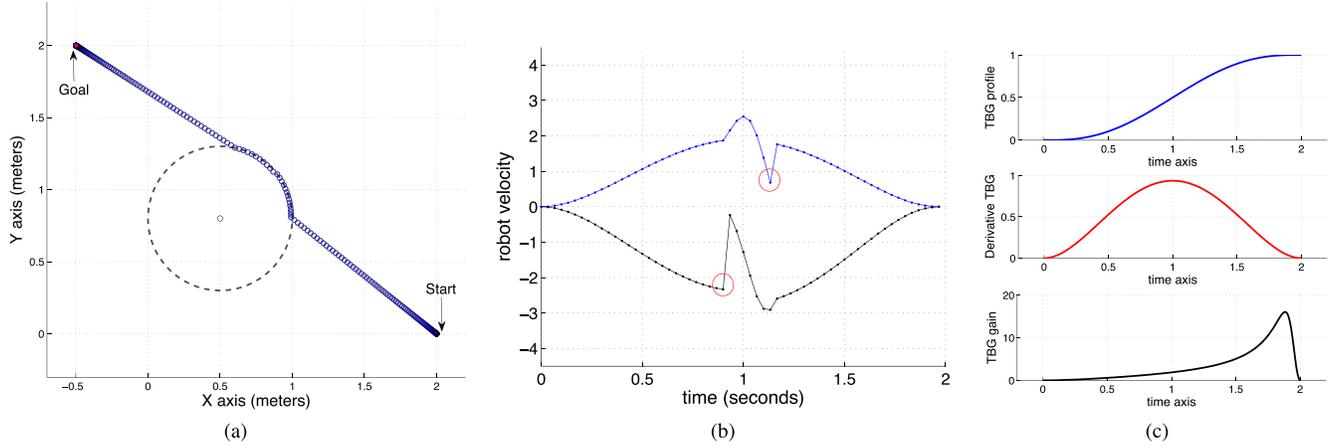


Fig. 1. Solution for two tasks with time parameterization. The two tasks must be completed within a hierarchical scheme, and the time to reach the target is of 2 s. Note how instabilities in the velocity profiles are still caused by the activation of the task corresponding to obstacle avoidance. (a) Bird-eye view of task. (b) Velocity profile. (c) TBG functions. The rows of (c) show  $\zeta(t)$ ,  $\dot{\zeta}(t)$ , and  $a(t)$  from top to bottom, respectively.

Let us adapt the control law reported in [17] to enforce desired time convergence of two hierarchical tasks

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}'_1 + \dot{\mathbf{q}}'_2 \quad (30)$$

where

$$\dot{\mathbf{q}}'_1 = \mathbf{J}_1^+ \dot{\mathbf{e}}'_1 \quad (31)$$

$$\dot{\mathbf{q}}'_2 = \overline{\mathbf{J}}_2^+ (\dot{\mathbf{e}}'_2 - \mathbf{J}_2 \dot{\mathbf{q}}'_1) \quad (32)$$

such that

$$\begin{aligned} \dot{\mathbf{e}}'_1 &= -\alpha_1(t)\zeta_1(t)\mathbf{e}_1 \\ &\quad - (1 - \zeta_1(t))\mathbf{J}_1\mathbf{J}_2^+\alpha_2(t)\zeta_2(t)\mathbf{e}_2 \end{aligned} \quad (33)$$

$$\begin{aligned} \dot{\mathbf{e}}'_2 &= -\alpha_2(t)\zeta_2(t)\mathbf{e}_2 \\ &\quad - (1 - \zeta_2(t))\mathbf{J}_2\mathbf{J}_1^+\alpha_1(t)\zeta_1(t)\mathbf{e}_1 \end{aligned} \quad (34)$$

encode the intermediate desired values that can be interpreted as the contribution of active tasks to perform the current task  $k$  when  $\zeta_k(t) = 0$ . During a transition interval  $0 < \zeta_k(t) < 1$ , we propose a transition function of the form

$$\zeta_k(t) = \frac{1}{2} \left( 1 - \cos \left( \frac{\pi(t - t_{0k})}{t_{fk} - t_{0k}} \right) \right), \quad t_{0k} \leq t \leq t_{fk} \quad (35)$$

where  $t_{0k}$  and  $t_{fk}$  correspond to the initial and final time intervals, such that  $\zeta_k$  evolves from  $\zeta_k(t_{0k}) = 0$  to  $\zeta_k(t_{fk}) = 1$ . Note that the transition (35) can be defined by other functions with similar shape, such as polynomials. The extension to  $p$  hierarchical tasks in transition involves the following computation of the control law:

$$\dot{\mathbf{q}} = \sum_{k=1}^p \dot{\mathbf{q}}'_k \quad (36)$$

with

$$\dot{\mathbf{q}}'_0 = 0 \quad (37)$$

$$\dot{\mathbf{q}}'_k = \overline{\mathbf{J}}_k^{+\lambda_k} (\dot{\mathbf{e}}'_k - \mathbf{J}_k \dot{\mathbf{q}}'_{k-1}) \quad (37)$$

$$\dot{\mathbf{e}}'_k = -\alpha_k(t)\zeta_k(t)\mathbf{e}_k - (1 - \zeta_k(t))\mathbf{J}_k \dot{\mathbf{q}}'_{[p \setminus k]} \quad (38)$$

where  $\dot{\mathbf{q}}'_{[p \setminus k]}$  is the solution of  $p$  hierarchical tasks with the corresponding activation parameters but without including the

current task  $k$ . The stability analysis for two tasks in transition is established by Theorem 1.

*Theorem 1:* Consider the kinematics of two stacked tasks

$$\dot{\mathbf{e}}' = \begin{pmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \end{pmatrix} \dot{\mathbf{q}} \quad (39)$$

where  $\mathbf{e}' = (\mathbf{e}'_1 \mathbf{e}'_2)^T$ . The control law (30) guarantees finite-time convergence to the origin  $\mathbf{e}' = \mathbf{0}$  while ensuring continuity of the input signals within task transition intervals.

*Proof:* Let us propose the Lyapunov candidate function

$$V = \frac{1}{2} \mathbf{e}'^T \mathbf{e}' \quad (40)$$

with time derivative as

$$\dot{V} = \mathbf{e}'^T \dot{\mathbf{e}}' = \mathbf{e}'^T \begin{pmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \end{pmatrix} \dot{\mathbf{q}}. \quad (41)$$

By substituting (30) in (41), we obtain

$$\dot{V} = -\mathbf{e}'^T \mathbf{M} \mathbf{e}' \quad (42)$$

where

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_a & \mathbf{M}_b \\ \mathbf{M}_c & \mathbf{M}_d \end{pmatrix} \quad (43)$$

with

$$\begin{aligned} \mathbf{M}_a &= c_1 \mathbf{J}_1 [\mathbf{J}_1^+ + c_2 (\overline{\mathbf{J}}_2^+ \mathbf{J}_2 \mathbf{J}_1^+ - \overline{\mathbf{J}}_2^+ \mathbf{J}_2 \mathbf{J}_1^+)] \\ \mathbf{M}_b &= c_3 \mathbf{J}_1 [\overline{\mathbf{J}}_2^+ + c_4 (\mathbf{J}_1^+ \mathbf{J}_1 \mathbf{J}_2^+ - \overline{\mathbf{J}}_2^+ \mathbf{J}_2 \mathbf{J}_1^+ \mathbf{J}_1 \mathbf{J}_2^+)] \\ \mathbf{M}_c &= c_1 \mathbf{J}_2 [\mathbf{J}_1^+ + c_2 (\overline{\mathbf{J}}_2^+ \mathbf{J}_2 \mathbf{J}_1^+ - \overline{\mathbf{J}}_2^+ \mathbf{J}_2 \mathbf{J}_1^+)] \\ \mathbf{M}_d &= c_3 \mathbf{J}_2 [\overline{\mathbf{J}}_2^+ + c_4 (\mathbf{J}_1^+ \mathbf{J}_1 \mathbf{J}_2^+ - \overline{\mathbf{J}}_2^+ \mathbf{J}_2 \mathbf{J}_1^+ \mathbf{J}_1 \mathbf{J}_2^+)] \end{aligned}$$

where  $c_1 = \alpha_1(t)\zeta_1(t)$ ,  $c_2 = 1 - \zeta_2(t)$ ,  $c_3 = \alpha_2(t)\zeta_2(t)$ , and  $c_4 = 1 - \zeta_1(t)$ . The preceding terms can be reduced by considering the definition of the pseudoinverse,  $\mathbf{J}_1 \mathbf{J}_1^+ = \mathbf{I}_{m_1}$  and  $\mathbf{J}_2 \overline{\mathbf{J}}_2^+ = \mathbf{J}_2 (\mathbf{J}_2 \mathbf{Q}_1)^+ = \mathbf{I}_{m_2}$ . In addition, the orthogonality property,  $\mathbf{J}_1 \mathbf{Q}_1 = \mathbf{0}$ , is used. Therefore

$$\mathbf{M} = \begin{pmatrix} c_1 \mathbf{I} & c_3 c_4 \mathbf{J}_1 \mathbf{J}_2^+ \\ c_1 c_2 \mathbf{J}_2 \mathbf{J}_1^+ & c_3 \mathbf{I} \end{pmatrix}. \quad (44)$$

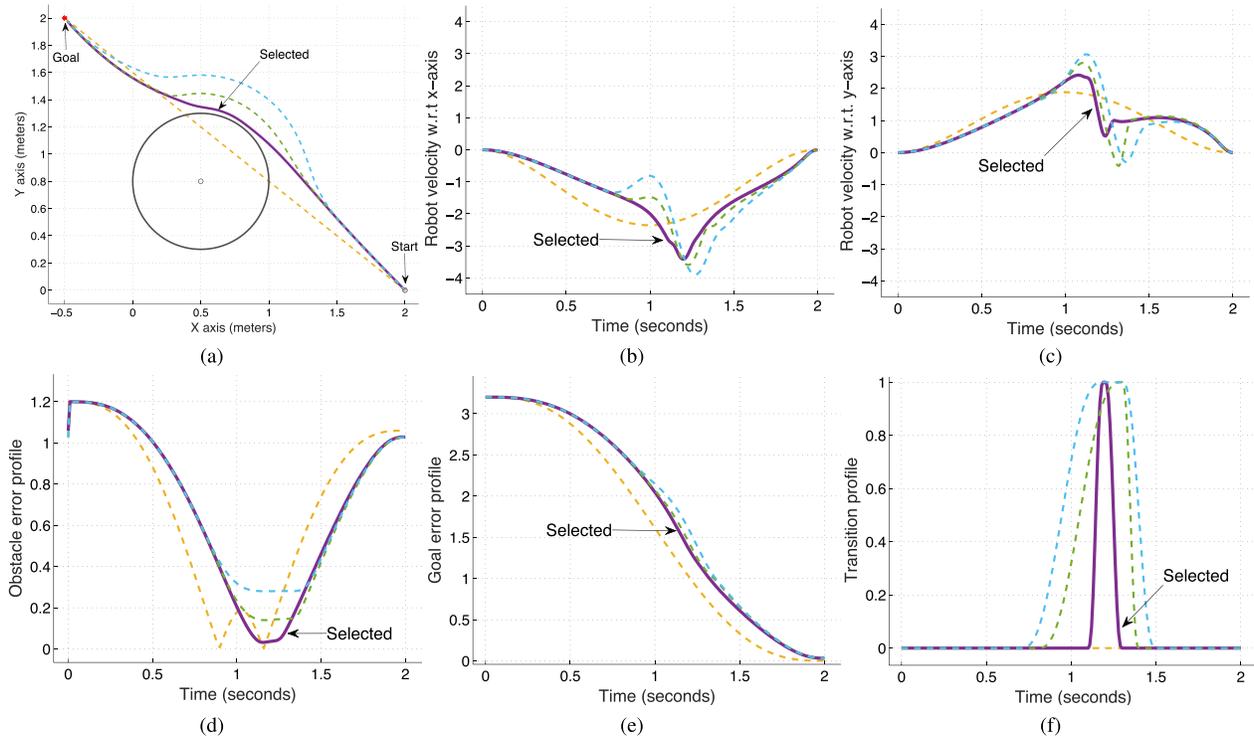


Fig. 2. Solution for two tasks with time parameterization, considering smooth transitions. The effect of adding smooth transitions within the proposed time-parameterized hierarchical task-based control is shown, i.e., abrupt changes along the performed trajectory are eliminated. (a) Four possible trajectories. The initial trajectory (yellow dotted line) does not consider the obstacle avoidance task. The selected trajectory (purple continuous line) avoids the obstacle without performing unnecessary deformation. The remaining trajectories (green and blue dotted lines) are generated by modifying the duration of task transitions. (b) and (c) Velocity profiles along  $X$  and  $Y$  axes, respectively, following the same color patterns. (d) and (e) Obstacle avoidance and reaching task errors. (f) Profiles observed are the transition functions for activating and deactivating the obstacle avoidance task.

Since the eigenvalues of matrix  $\mathbf{M}$  only depend on  $\alpha_1(t)$ ,  $\alpha_2(t)$ ,  $\zeta_1(t)$ , and  $\zeta_2(t)$ , where  $0 < \zeta_k(t) < 1$  for  $k \in [1, 2]$ , from Lemma 1 we conclude that the error  $e'$  converges to the origin in finite time. Hence, the convergence of time-constrained hierarchical tasks in transition is ensured. ■

It is important to note that in the general case, the controller is called  $2^{n_{tr}} - 1$  times for  $n_{tr}$  tasks in transition, i.e., the computational cost increases exponentially with respect to the number of tasks in transition. However, to show the benefit of imposing continuity in the proposed hierarchical task-based control, we present how the problem shown in Fig. 1 is solved now considering the smoothness behavior. The visual analysis of Fig. 2 reveals how it is possible to hierarchically execute two tasks within a period of time while maintaining continuity along the velocity signals.

## V. EXPERIMENTAL RESULTS

This section is devoted to demonstrate the usability of our scheme for local and global tasks performed by several wheeled robots. The experimental setup consists of two top-view cameras connected to a personal computer for localizing the robots (iRobot Create) on the planar environment. This localization is obtained placing distinctive landmarks on each robot. As the purpose of these landmarks is to provide the position  $(x, y)$  and orientation  $\theta$  at 30 frames/s, we used triangular-like black shapes over a white background

(the planar surfaces of the robots). As a consequence, thresholding operations allow the separation and classification of the robots involved in the experiments.

The working environment covered by a single top-view camera is  $2.3 \times 1.5 \text{ m}^2$ . This area turns to be sufficient for performing experiments with a single robot. Otherwise, the pair of adjacent top-view cameras covers  $3 \times 2.66 \text{ m}^2$ .

The personal computer generates the reference trajectories for the low-level controller of each robot and processes the camera images to get the current robot position and orientation. The computation cost for obtaining the input references is  $< 1 \text{ ms}$  per iteration. The output is transmitted via Bluetooth to each robot for moving the right and left wheels. The wheel's controller is a PID. However, the iRobot Create is a differential-drive robot and its nonholonomic constraint must be considered for computing the heading velocity. In particular, we applied a classical input-output linearization by selecting the coordinates of a reference point located in front of the robot [26].

This section is divided into two parts. The first of these considers several tasks, time parameterization and smoothness for a challenging scenario with three robots. The second part presents a different scenario with six robots and global tasks.

### A. Several Local Tasks for Three-Wheeled Robots

Here, we explore the outcome of our scheme in a challenging scenario. This test consists of two main phases. In the

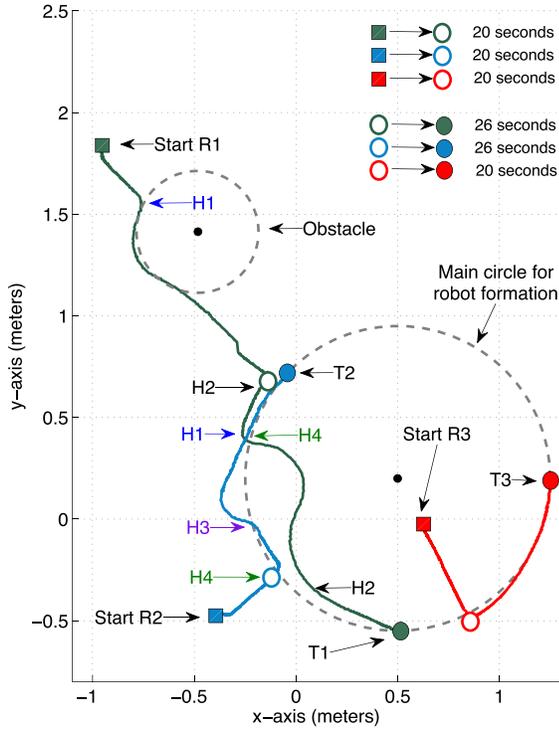


Fig. 3. Bird-eye view of Experiment 1. Three robots are used here, referred to as R1, R2, and R3. Each robot starts in different positions, marked with colored squares. The task consists of arriving at a formation circle to later reach three final targets (T1, T2, and T3, marked with colored circles) located along that formation circle. Moments along trajectories when priorities are activated are marked with H1 and H2 for R1, and H3 and H4 for R2.

first phase, three robots have to reach the nearest point along a formation circle. This phase (Phase 1) has to be carried out within 20 s. For simplicity, the three robots are named R1, R2, and R3. During Phase 1, R1 faces an obstacle and must evade it. The second phase of the experiment requires the robots to reach different target positions (T1, T2, and T3, respectively, for each robot) along the formation circle. This is called Phase 2 and has to be accomplished within 26 s for R1 and R2, while 20 s is set for R3. Phase 2 presents a particularly challenging scenario, since R1 and R2 have to move in opposite directions along the formation circle in order to reach their corresponding targets, and as a consequence, R2 becomes a moving obstacle for R1 and viceversa. Fig. 3 shows the overall behavior of the experiment, where the starting position of the robots, the formation circle, the obstacles, and the performed trajectories (as captured by the vision system) are shown. In Fig. 3, H1 and H2 correspond to the moment along the trajectory of R1 when a task priority is activated. The same holds for R2 with H3 and H4. The priority triggered by H1 and H3 is obstacle avoidance, while H2 and H4 represent the robots keeping the main circle formation.

The experiment is designed to analyze two main features. First, obstacle avoidance, which has a major priority over keeping the main circular formation, is emphasized for R1 during both phases of the experiment, and for R2 during the second phase. Here, we seek to confirm the occurrence of smooth transitions between task priorities. Second, time

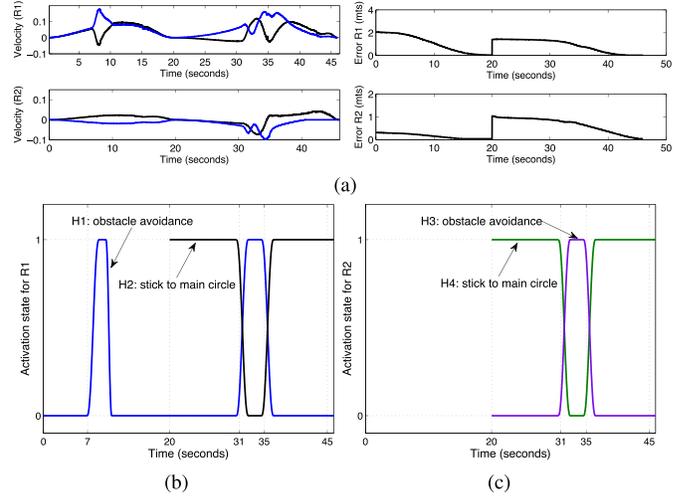


Fig. 4. Overall performance of Experiment 1. In the left column of (a), velocity profiles corresponding to the  $x$  and  $y$  directions are shown for each robot in black and blue lines, respectively. Similarly, the right column of (a) presents convergence behavior for both robots, corresponding to Phases 1 and 2 of the experiment. The smooth transition between two task priorities is provided for R1 in (b) and R2 in (c).

parameterization is also important and is evaluated in both phases for the three robots, as the main goal is to reach both the main circular formation during Phase 1 and to reach the final target during Phase 2.

A further analysis of the experiment is shown in Fig. 4, where quantitative data are shown for R1 and R2. In Fig. 4, the first row (left) presents velocity profiles for each robot along both phases of the experiment. Note how only R1 exhibits velocity changes during the first 20 s of trajectory, while both robots show variations between 30 and 40 s. Such variations correspond to obstacle avoidance events, and the observed continuity along velocity profiles is a result of the smooth transitioning between task priorities imposed by our method. In addition, it is noticeable how, even when sudden obstacles appear, both robots arrive to the formation in exactly 20 s with the same terminal attractor. This experimentally validates that the terminal attractor does not depend on the initial conditions. The convergence of the task error related with reaching the formation circle (Phase 1) and arriving at the desired target along the circle (Phase 2) is shown in the right plots of Fig. 4(a). Here, the diagrams corroborate how the proposed time parameterization provides a reliable and smooth way to successfully achieve tasks within a required period of time.

To finish this analysis, Fig. 4(b) and (c) shows the plots of tasks priority activations for R1 and R2, respectively. The hierarchy in this experiment is  $e_o < e_c < e_g$ . The moment of task activation is consistently marked in Fig. 3 as H1 (blue) and H2 (black) for R1, and H3 (purple), and H4 (green) for R2. Apart from the smoothness of transition between activated tasks, it is to note the synchronicity of tasks during Phase 2, i.e., both robots start avoiding collision against each other during the same moments (around 31 s) just to seek for keeping the main circle formation once their path becomes unobstructed (around 35 s). Finally, it is worth highlighting the smooth transition between the tasks, which is

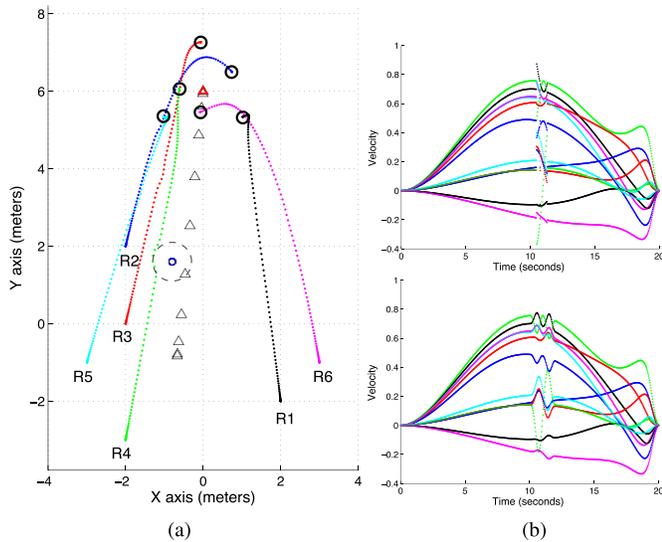


Fig. 5. Results on Experiment 2. Through a simulation, six robots (R1–R6) attempt to converge to a given formation (dark circles), while the centroid of the formation (red triangle) reaches a final position. During the mission, an obstacle is encountered by R4. (a) Bird-eye view of simulation. This is shown for the case when smooth task priority activation is observed. (b) Velocity profiles for each robot, corresponding with colored trajectories in (a), are provided for no smooth (top) and smooth (bottom) task priority transitioning.

a main responsible to maintain appropriate velocity references for the low-level control of the system.

### B. Global Tasks for a Group of Robots

Here, we describe a simulation with six robots that consists in converging to a given formation, while the centroid of the group is regulated to reach a desired position. This can be generated with global hierarchical tasks as follows. First, the desired formation is achieved by computing both the mean value and variance of the robots' positions using (13) and (15), respectively. We applied (17) to minimize the distance among adjacent robots. The hierarchy imposed in this simulation is  $e_o < e_v < e_d < e_m$ . The last task in the hierarchy moves the centroid of the formation  $x_m(q)$  to a desired position. At the highest hierarchical level, the obstacle avoidance task should be activated when a given robot approaches an obstacle.

Fig. 5(a) shows a bird-eye view of the simulation, where smooth task transitioning was imposed. The starting points of the robots are represented using the letters from R1–R6. Each robot's trajectory is differently colored and the final formation to be achieved is shown with dark circles. Black triangles represent the centroid of the group at different times and the red triangle shows the desired centroid of formation. During the task, R4 encounters an obstacle and has to evade it. The velocity profiles of this simulation are shown in the bottom diagram of Fig. 5(b). Note how, as a consequence of applying smooth priority task transitions, changes in velocity appear continuously, specially around 10 s where R4 (green) faced the obstacle. This is not the case for the diagram at the top, corresponding to the case where no smooth transitioning was considered, generating a great velocity disturbance in comparison with its smooth counterpart.

## VI. CONCLUSION

This brief introduced a robust regulation scheme based on terminal attractors for hierarchical task-based controllers of multirobot systems. The proposed control scheme provides online the reference velocity signals to be tracked by the low-level controller of each robot. Our controller is centralized in the sense that the task errors are defined in terms of the robots' translational coordinates. The control law is able to regulate hierarchical task functions to accomplish the formation requirements assigned to the group of robots while avoiding collisions with static or moving artifacts in the environment. In particular, a time-varying gain called TBG gain, which is a class of terminal attractor that is independent of the robot's initial positions, handles the desired time convergence of reaching tasks. In addition, the TBG has been successfully applied within transition intervals, where tasks are smoothly activated, removed, or swapped according to the hierarchical structure.

In summary, this brief copes with two important concerns for controlling the behavior of multirobot systems: the time parameterization of tasks together with smooth task transitions. We demonstrated that both aspects can be solved by means of terminal attractors. We verified the robustness and versatility of the proposed scheme with global tasks in simulation and one experiment considering local tasks with a set of wheeled robots.

Currently, we are working on the definition of linear inequality constraints to be satisfied at different hierarchical levels to maintain the robots inside a given geometric shape, to avoid obstacles and to cope with velocity limits. All these issues should be considered without causing discontinuities in the reference velocity profiles.

## REFERENCES

- [1] L. E. Parker, "Multiple mobile robot systems," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer Science & Business Media, 2008.
- [2] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Trans. Ind. Inform.*, vol. 9, no. 1, pp. 427–438, Feb. 2013.
- [3] Y. Chen and Z. Wang, "Formation control: A review and a new consideration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 3181–3186.
- [4] W. Ren and R. W. Beard, "Decentralized scheme for spacecraft formation flying via the virtual structure approach," *J. Guid., Control, Dyn.*, vol. 27, no. 1, pp. 73–82, 2004.
- [5] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1465–1476, Sep. 2004.
- [6] C. Samson, M. Le Borgne, and B. Espiau, *Robot Control: The Task Function Approach* (Oxford Engineering Science Series), vol. 22, 1st ed. New York, NY, USA: Oxford Univ. Press, 1991.
- [7] G. Antonelli, F. Arrichiello, F. Caccavale, and A. Marino, "Decentralized time-varying formation control for multi-robot systems," *Int. J. Robot. Res.*, vol. 33, no. 7, pp. 1029–1043, 2014.
- [8] G. Antonelli and S. Chiaverini, "Kinematic control of platoons of autonomous vehicles," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1285–1292, Dec. 2006.
- [9] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "Stabilization of a hierarchical formation of unicycle robots with velocity and curvature constraints," *IEEE Trans. Robot.*, vol. 25, no. 5, pp. 1176–1184, Oct. 2009.
- [10] D. Zhang, G. Xie, J. Yu, and L. Wang, "Adaptive task assignment for multiple mobile robots via swarm intelligence approach," *Robot. Auto. Syst.*, vol. 55, no. 7, pp. 572–588, 2007.

- [11] J. P. Desai, J. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1998, pp. 2864–2869.
- [12] P. K. C. Wang, F. Y. Hadaegh, and K. Lau, "Synchronized formation rotation and attitude control of multiple free-flying spacecraft," *J. Guid., Control, Dyn.*, vol. 22, no. 1, pp. 28–35, 1999.
- [13] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 947–951, Dec. 2001.
- [14] C. S. Karagöz, H. I. Bozma, and D. E. Koditschek, "Coordinated navigation of multiple independent disk-shaped robots," *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 1289–1304, Dec. 2014.
- [15] G. Antonelli, E. Arrichiello, and S. Chiaverini, "Experiments of formation control with multirobot systems using the null-space-based behavioral control," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 5, pp. 1173–1182, Sep. 2009.
- [16] F. Keith, P.-B. Wieber, N. Mansard, and A. Kheddar, "Analysis of the discontinuities in prioritized tasks-space control under discreet task scheduling operations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, CA, USA, Sep. 2011, pp. 3887–3892.
- [17] J. Lee, N. Mansard, and J. Park, "Intermediate desired value approach for task transition of robots in kinematic control," *IEEE Trans. Robot.*, vol. 28, no. 6, pp. 1260–1277, Dec. 2012.
- [18] O. Kanoun, F. Lamiroux, and P.-B. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 785–792, Aug. 2011.
- [19] B. Siciliano and J.-J. E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Proc. IEEE 5th Int. Conf. Adv. Robot.*, Pisa, Italy, Jun. 1991, pp. 1211–1216.
- [20] T. Sugihara, "Solvability-unconcerned inverse kinematics by the Levenberg–Marquardt method," *IEEE Trans. Robot.*, vol. 27, no. 5, pp. 984–991, Oct. 2011.
- [21] P. Baerlocher and R. Boulic, "An inverse kinematics architecture enforcing an arbitrary number of strict priority levels," *Vis. Comput., Int. J. Comput. Graph.*, vol. 20, no. 6, pp. 402–417, Aug. 2004.
- [22] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *Int. J. Robot. Res.*, vol. 33, no. 7, pp. 1006–1028, Jun. 2014.
- [23] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems," *IEEE Trans. Robot.*, vol. 25, no. 5, pp. 985–994, Oct. 2009.
- [24] G. Jarquín, G. Arechavaleta, and V. Parra-Vega, "Time parametrization of prioritized inverse kinematics based on terminal attractors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, CA, USA, Sep. 2011, pp. 1612–1617.
- [25] H. K. Khalil and J. W. Grizzle, *Nonlinear Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
- [26] G. Oriolo, A. De Luca, and M. Vendittelli, "WMR control via dynamic feedback linearization: Design, implementation, and experimental validation," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 6, pp. 835–852, Nov. 2002.