Time-Constrained Whole Body Control With Smooth Task Transitions

Gerardo Jarquín, Gustavo Arechavaleta, Adrien Escande and Eiichi Yoshida

***Abstract*— The aim of this work is to increase the dexterity of humanoid robots by handling both the time axis and the necessary transitions of kinematic tasks during the execution of whole-body motions in real-time. The desired time convergence (DTC) of tasks is commonly achieved by finding the appropriate control gains at hand. Also, it is known that the execution of a sequence of tasks as well as the insertion, removal and swapping of hierarchical tasks or constraints generate undesired discontinuities in robot velocities. Recently, some control schemes solved these problems but they are either computationally expensive for real-time purposes or they need to modify the main structure of the hierarchical task-based solver. In this work, we propose a class of terminal attractors to automatically control the DTC of tasks. In addition, we propose a real-time strategy to handle smooth transitions of hierarchical tasks or constraints by time-varying weights. We validated our whole-body control strategy with the humanoid robots NAO and HRP-2.**

I. INTRODUCTION

Task-based controllers are widely used for generating the whole body motion of humanoid robots. This class of controllers allows the execution of operational tasks given a hierarchical order. An exponential convergence of each hierarchical task is commonly used. To achieve a desired time convergence (DTC), particular time-varying gains have been suggested [1], [2] and [3]. In this context, we propose a general strategy relies on terminal attractors to induce DTC of a family of linear differential equations without dependency on the final time or initial conditions. These equations represent the time derivative of the error function in task-based control (see Sections II and III).

In some scenarios, whole-body controllers need to modify the hierarchy of active tasks but also to insert new constraints or to remove existing ones. These operations are called task transitions and should be computed while the robot is moving. Thus, if the transitions are not correctly performed then discontinuities appear in the control signals (see Section IV). For this, we adopt a weighting method for solving linear least-squares under linear constraints and we extend it to solve real-time smooth task transitions while applying DTC of hierarchical tasks (see Section V).

II. ENFORCING DESIRED TIME CONVERGENCE

Consider a task function of the form:

$$\boldsymbol{e}(\boldsymbol{q},t) = \boldsymbol{x}_d - \boldsymbol{x}(\boldsymbol{q}) \tag{1}$$

G. Jarquín and G. Arechavaleta are with the Robotics and Advanced Manufacturing Group, Centro de investigación y de Estudios Avanzados del IPN, Saltillo, Coah. México {gerardojarquin, garechav}@cinvestav.edu.mx. A. Escande and E. Yoshida are with the CNRS-AIST JRL (Joint Robotics Laboratory), UMI3218/CRT {adrien.escande, e.yoshida}@aist.go.jp

where $\boldsymbol{q} \in \mathbb{R}^n$ (with $n \geq m$), $\boldsymbol{x}(\boldsymbol{q}) \in \mathbb{R}^m$ and $\boldsymbol{x}_d \in \mathbb{R}^m$ are the robot configuration, the current and desired value of the task, respectively. From the time derivative of (1), a control law for $\dot{\boldsymbol{q}}$ is obtained as:

$$\dot{\boldsymbol{q}} = -J(\boldsymbol{q})^{+}\dot{\boldsymbol{e}}(\boldsymbol{q},t) \tag{2}$$

where $J(\boldsymbol{q})^{+}$ is the pseudo-inverse of the task Jacobian $J(\boldsymbol{q}) \in \mathbb{R}^{m \times n}$. An exponential convergence is commonly used such that

$$\dot{\boldsymbol{e}}(\boldsymbol{q},t) = -\alpha \boldsymbol{e}(\boldsymbol{q},t) \tag{3}$$

Some time-varying functions $\alpha(t)$ have been proposed in [1], [2] and [3] to avoid the discontinuity when the task becomes active. In this work we propose the following terminal attractor:

$$\alpha(t,t_0,t_f) = \frac{\dot{\xi}(t,t_0,t_f)}{1-\xi(t,t_0,t_f)+\delta} \tag{4}$$

where $0 < \delta < 0.1$, t_0 and $t_f > t_0$ are the initial and final times, respectively. Hereafter, the dependence on t_0 and t_f will be only used when necessary. The function $\xi(t)$ is called time base generator (TBG) [4]. The gain δ is useful to avoid the indetermination at $t \geq t_f$ when $\alpha(t \geq t_f) = 0/\delta$, [5]. Note that $\xi(t)$, $\dot{\xi}(t)$ and $\alpha(t)$ reach their final values at $t = t_f$ (see Fig. 1). The function $\alpha(t)$ is called TBG gain, its value is null at the initial time and after the final time.

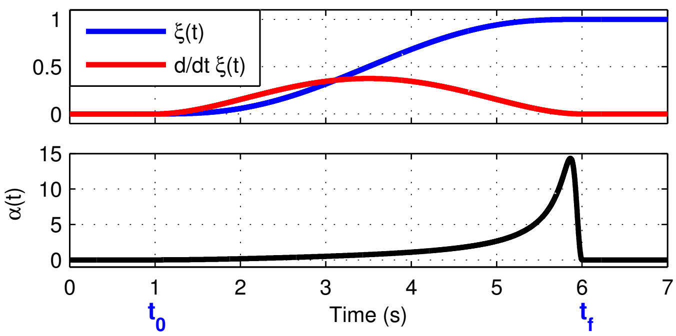

Fig. 1. **From the top row to the bottom**: Profiles of the TBG function $\xi(t)$, its time derivative $\dot{\xi}(t)$ and the TBG gain $\alpha(t)$. All functions are continuous and their final values are reached at the desired final time $t_f = 6$.

The solution of each equation in (3) and its derivative using the TBG gain in (4) become:

$$e(t) = e(t_0)\left(1-\frac{\xi(t)}{1+\delta}\right) \tag{5}$$

$$\dot{e}(t) = -e(t_0)\left(\frac{\dot{\xi}(t)}{1+\delta}\right) \tag{6}$$

The shapes of $e(t)$ and $\dot{e}(t)$ closely follow the profiles of $1-\xi(t)$ and -$\dot{\xi}(t)$, respectively. This means that $e(t)$ is a smooth function that asymptotically converges to its final value at $t=t_f$ regardless of $e(0)$.

III. DTC for hierarchical tasks [6]

Hierarchical inverse kinematics (HIK) [7] is widely used to efficiently exploit the robot redundancy by performing each task at best without affecting tasks with higher levels of hierarchy. This is achieved by solving each task in the null space of all tasks with higher hierarchy. The control law for p equality tasks is [8]:

$$\begin{aligned} \dot{\boldsymbol{q}}_1 &= -J_1(\boldsymbol{q})^{+}\dot{\boldsymbol{e}}_1(\boldsymbol{q},t) \\ \dot{\boldsymbol{q}}_k &= \dot{\boldsymbol{q}}_{k-1} + Q_k^{+}\left(-\dot{\boldsymbol{e}}_k(\boldsymbol{q},t) - J_k(\boldsymbol{q})\dot{\boldsymbol{q}}_{k-1}\right) \end{aligned} \quad (7)$$

where $k \in [2,3,\ldots,p]$ is the task index ordered with decreasing hierarchy, $Q_k = J_k(\boldsymbol{q})P_{k-1} \in \mathbb{R}^{m_k \times n}$ projects $J_k(\boldsymbol{q})$ onto the shared null space of all tasks with higher hierarchy. Algorithmic singularities, occurred when Q_k loses rank, may be avoided by using the classical damping [9].
To induce DTC, we introduce the TBG gain in (7):

$$\begin{aligned} \dot{\boldsymbol{q}}_1 &= -J_1(\boldsymbol{q})^{+}\dot{\boldsymbol{e}}_1(\boldsymbol{q},\alpha,t) \\ \dot{\boldsymbol{q}}_k &= \dot{\boldsymbol{q}}_{k-1} + Q_k^{+}\left(-\dot{\boldsymbol{e}}_k(\boldsymbol{q},\alpha,t) - J_k(\boldsymbol{q})\dot{\boldsymbol{q}}_{k-1}\right) \end{aligned} \quad (8)$$

Note that the stability analysis in [10] holds due to the positiveness of $\alpha(t)$ for all time.

IV. DTC with task transitions [11]

This section shows how to enforce DTC under transitions by extending recent strategies that allow to perform the insertion, removal and swapping of tasks smoothly.

A. Activation and removal of constraints

Let us redefine the task error in terms of an activation matrix $H_{\hat{\xi}}(t) \triangleq \text{diag}\{\hat{\xi}_1(t),\ldots,\hat{\xi}_m(t)\}$

$$\boldsymbol{e}(\boldsymbol{q},\hat{\xi},t) = H_{\hat{\xi}}(t)\boldsymbol{e}(\boldsymbol{q},t) \quad (9)$$

where

$$\hat{\xi}(t) = \xi_{in}(t,t_0^{in},t_f^{in})(1-\xi_{rm}(t,t_0^{rm},t_f^{rm})) \quad (10)$$

with t_0^{in}, t_f^{in}, t_0^{rm} and t_f^{rm} as the initial and final insertion and removal times, respectively. By deriving w.r.t. time (9) and by including the TBG gain (4), the control law is:

$$\dot{\boldsymbol{q}} = J(\boldsymbol{q})^{\oplus^{H_{\hat{\xi}}}}\boldsymbol{e}(\boldsymbol{q},\alpha,H_{\hat{\xi}},t) \quad (11)$$

where $\boldsymbol{e}(\boldsymbol{q},\alpha,H_{\hat{\xi}},t) = \left(\alpha(t)H_{\hat{\xi}}(t) + \dot{H}_{\hat{\xi}}(t)\right)\boldsymbol{e}(\boldsymbol{q},t)$. The operator $J(\boldsymbol{q})^{\oplus^{H}} \in \mathbb{R}^{n\times m}$ preserves the continuity during the transition [12].

B. Time-constrained hierarchical task transitions

1) Intermediate desired value approach (IDVA) with TBG: The IDVA [13] can be extended to handle time constraints together with task transitions by defining the operational reference velocities as:

$$\begin{aligned} \dot{\boldsymbol{e}}_k'(\boldsymbol{q},\alpha,H_{\hat{\xi}},t) &= \hat{\xi}_k(t)\dot{\boldsymbol{e}}_k(\boldsymbol{q},\alpha,H_{\hat{\xi}},t) \\ &+ (1-\hat{\xi}_k(t))J_k(\boldsymbol{q})\dot{\boldsymbol{q}}_{\{\hat{\xi}_1,\ldots,\hat{\xi}_j,\ldots,\hat{\xi}_p\}} \end{aligned} \quad (12)$$

where $j \neq k$ and $\dot{\boldsymbol{q}}_{\{\hat{\xi}_1,\ldots,\hat{\xi}_j,\ldots,\hat{\xi}_p\}}$ is the current solution without task k.

2) Linear interpolation (LI) with TBG: Let i and j be the indexes of two consecutive tasks where their hierarchy should be interchanged. The swap is performed by a LI between two partial control laws [14]. This operation can be expressed in two forms:

$$\dot{\boldsymbol{q}} = {}^{A}\dot{\boldsymbol{q}}_{i-1}^{1} + \hat{\xi}(t)\ {}^{A}\dot{\boldsymbol{q}}_p^{i} + (1-\hat{\xi}(t))\ {}^{B}\dot{\boldsymbol{q}}_p^{j} \quad (13)$$

$$\dot{\boldsymbol{q}} = {}^{A}\dot{\boldsymbol{q}}_{i-1}^{1} + \hat{\xi}(t)\ {}^{A}\dot{\boldsymbol{q}}_j^{i} + (1-\hat{\xi}(t))\ {}^{B}\dot{\boldsymbol{q}}_i^{j} + {}^{A}\dot{\boldsymbol{q}}_p^{j+1} \quad (14)$$

where the left upper script denotes the arrange of tasks in the hierarchical structure before and after the swap (A and B respectively). Note that method (13) computes twice the remaining levels while method (14) only doubles the computation at levels i and j at the cost of modifying the HIK solver.

V. Real-time task transitions with DTC [15]

We propose to build a transition phase where the tasks in transition are merged in the same hierarchical level with time-varying weights to invert the hierarchies. The advantages of this method are that the HIK solver is not modified and the computational cost is not increased. Thus, it enables the possibility of being used in real-time applications.

According to [16], the constrained least-squares problem:

$$\begin{aligned} &\min_{\mathrm{x}} \frac{1}{2}\left\|G\mathrm{x} - \boldsymbol{g}\right\|^2 \\ &s.t. \quad H\mathrm{x} - \boldsymbol{h} = 0 \end{aligned} \quad (15)$$

is equivalent to the following unconstrained one:

$$\min_{\mathrm{x}} \frac{1}{2}\left\|\begin{bmatrix} G \\ \beta H \end{bmatrix}\mathrm{x} - \begin{bmatrix} \boldsymbol{g} \\ \beta\boldsymbol{h} \end{bmatrix}\right\|^2 \quad (16)$$

where $\mathrm{x} \in \mathbb{R}^n$, $G \in \mathbb{R}^{m\times n}$, $\boldsymbol{g} \in \mathbb{R}^m$, $H \in \mathbb{R}^{p\times n}$ and $\boldsymbol{h} \in \mathbb{R}^p$. β is a positive finite gain which preserves the equivalence between two problems under the following assumptions: rank$(H) = p$ and $N(G) \cap N(H) = \{0\}$. The goal is to transit smoothly from

$$\begin{aligned} &\min_{\dot{\boldsymbol{q}}_k,\boldsymbol{w}_k} \frac{1}{2}\left||\boldsymbol{w}_k|\right|^2 \\ s.t.\quad & {}^{B}J_k(\boldsymbol{q})\dot{\boldsymbol{q}}_k - {}^{B}\dot{\boldsymbol{e}}_{d_k}(\boldsymbol{q},\alpha,H_{\hat{\xi}},t) = {}^{B}\boldsymbol{w}_k \\ & {}^{A}J_{k-1}(\boldsymbol{q})\dot{\boldsymbol{q}}_k - {}^{A}\dot{\boldsymbol{e}}_{d_{k-1}}(\boldsymbol{q},\alpha,H_{\hat{\xi}},t) = {}^{A}\boldsymbol{w}_{k-1}^{*} \\ & \bar{J}_{k-2}(\boldsymbol{q})\dot{\boldsymbol{q}}_k - \dot{\bar{\boldsymbol{e}}}_{d_{k-2}}(\boldsymbol{q},\alpha,H_{\hat{\xi}},t) = \bar{\boldsymbol{w}}_{k-2}^{*} \end{aligned} \quad (17)$$

to

$$\min_{\dot{\boldsymbol{q}}_k,\boldsymbol{w}_k} \frac{1}{2}||\boldsymbol{w}_k||^2 \tag{18}$$

$$\begin{aligned} s.t. \quad {}^AJ_k(\boldsymbol{q})\dot{\boldsymbol{q}}_k - {}^A\dot{\boldsymbol{e}}_{d_k}(\boldsymbol{q},\alpha,H_{\hat{\xi}},t) &= {}^A\boldsymbol{w}_k \\ {}^BJ_{k-1}(\boldsymbol{q})\dot{\boldsymbol{q}}_k - {}^B\dot{\boldsymbol{e}}_{d_{k-1}}(\boldsymbol{q},\alpha,H_{\hat{\xi}},t) &= {}^B\boldsymbol{w}^*_{k-1} \\ \bar{J}_{k-2}(\boldsymbol{q})\dot{\boldsymbol{q}}_k - \dot{\bar{\boldsymbol{e}}}_{d_{k-2}}(\boldsymbol{q},\alpha,H_{\hat{\xi}},t) &= \bar{\boldsymbol{w}}^*_{k-2} \end{aligned}$$

where A and B identify the tasks in transition and the bar represents a stack. $\boldsymbol{w}_k \in \mathbb{R}^m$ is a vector of slack variables used to relax the infeasible constraints with hierarchy k and $\boldsymbol{w}^*_{k-1}$ represents the optimum value of $\boldsymbol{w}_{k-1}$.The proposed transition phase is:

$$\min_{\dot{\boldsymbol{q}}_k,{}^A\boldsymbol{w}_k,{}^B\boldsymbol{w}_{k-1}} \frac{1}{2}\left\|\begin{bmatrix}\beta_A({}^A\boldsymbol{w}_k)\\ \beta_B({}^B\boldsymbol{w}_{k-1})\end{bmatrix}\right\|^2 \tag{19}$$

$$\begin{aligned} s.t. \quad {}^AJ_k(\boldsymbol{q})\dot{\boldsymbol{q}}_k - {}^A\dot{\boldsymbol{e}}_{d_k}(\boldsymbol{q},\alpha,H_{\hat{\xi}},t) &= {}^A\boldsymbol{w}_k \\ {}^BJ_{k-1}(\boldsymbol{q})\dot{\boldsymbol{q}}_k - {}^B\dot{\boldsymbol{e}}_{d_{k-1}}(\boldsymbol{q},\alpha,H_{\hat{\xi}},t) &= {}^B\boldsymbol{w}_{k-1} \\ \bar{J}_{k-2}(\boldsymbol{q})\dot{\boldsymbol{q}}_k - \dot{\bar{\boldsymbol{e}}}_{d_{k-2}}(\boldsymbol{q},\alpha,H_{\hat{\xi}},t) &= \bar{\boldsymbol{w}}^*_{k-2} \end{aligned}$$

where β_B smoothly evolves from 1 to a big value and β_A evolves from a big value to 1 (see Fig. 2). Then, the idea is to switch from (17) to (19) at the beginning of the transition period tr^0. The initial value of β_A should preserve the hierarchies of (17) to avoid a discontinuity. After this switch, β_A should decrease its value until reach $\beta_A = 1$. At that moment both tasks have the same priority since $\beta_B = 1$. After β_A has reached its final value, β_B begins its variation toward its final value reached at the end of the transition period t^f_{tr}. Then, at $t = t^f_{tr}$ the problem (19) is replaced by (18) to end the swap process. Note that β_B in (19) should preserve the hierachical order of (18) to void a discontinuity.

This strategy requires to perform a set of sequential swaps to insert a task. The new task is merged with the task with lowest hierarchy by solving:

$$\min_{\dot{\boldsymbol{q}}_k,{}^A\boldsymbol{w}_p,{}^B\boldsymbol{w}_{p+1}} \frac{1}{2}\left\|\begin{bmatrix}{}^A\boldsymbol{w}_p\\ {}^B\boldsymbol{w}_{p+1}\beta_{ins}\end{bmatrix}\right\|^2 \tag{20}$$

$$\begin{aligned} s.t. \quad {}^AJ_{p+1}(\boldsymbol{q})\dot{\boldsymbol{q}}_{p+1} - {}^A\dot{\boldsymbol{e}}_{d_{p+1}}(\boldsymbol{q},\alpha,H_{\hat{\xi}},t) &= {}^A\boldsymbol{w}_{p+1} \\ {}^BJ_p(\boldsymbol{q})\dot{\boldsymbol{q}}_{p+1} - {}^B\dot{\boldsymbol{e}}_{d_p}(\boldsymbol{q},\alpha,H_{\hat{\xi}},t) &= {}^B\boldsymbol{w}_p \\ \bar{J}_{p-1}(\boldsymbol{q})\dot{\boldsymbol{q}}_{p+1} - \dot{\bar{\boldsymbol{e}}}_{d_{p-1}}(\boldsymbol{q},\alpha,H_{\hat{\xi}},t) &= \bar{\boldsymbol{w}}^*_{p-1} \end{aligned}$$

Note that $\beta_{ins} = 0$ at t^0_{tr} because this task was not in the hierarchy. Then, a set of swaps brings the new task to the desired hierarchical level. In order to remove a task, the reverse process must be performed.

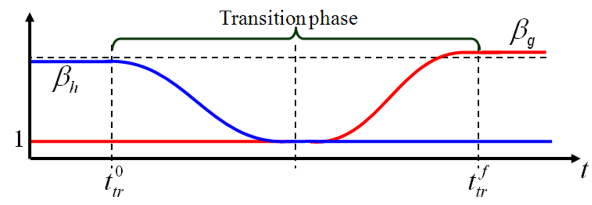

Fig. 2. Profiles of the weights during the transition phase. The decreasing weight β_h leads the set of constraints to lose its hierarchy when $\beta_h = 1$. Then, the weight β_g progressively increases to a high value directly affecting the other set of constraints G. Note that the highest value of weights β_g and β_h is not necessarily the same for both sets.

References

[1] P. Soueres, V. Cadenat, and M. Djeddou, "Dynamical sequence of multi-sensor based tasks for mobile robots navigation," in *7th IFAC Symposium on Robot Control*, Wroclaw, Poland, September 2003, pp. 423–428.

[2] N. Mansard and F. Chaumette, "Task sequencing for high-level sensor-based control," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 60–72, February 2007.

[3] F. Keith, N. Mansard, S. Miossec, and A. Kheddar, "Optimization of tasks warping and scheduling for smooth sequencing of robotic actions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, October 2009, pp. 1609–1614.

[4] P. Morasso, V. Sanguineti, and G. Spada, "A computational theory of targeting movements based on force fields and topology representing networks," *Neurocomputing*, vol. 15, no. 3-4, pp. 411–434, June 1997.

[5] V. Parra-Vega, "Second order sliding mode control for robot arms with time base generators for finite-time tracking," *International Journal of Dynamics and Control*, vol. 11, no. 2, pp. 175–186, April 2001.

[6] G. Jarquin, G. Arechavaleta, and V. Parra-Vega, "Time parametrization of prioritized inverse kinematics based on terminal attractors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, USA, September 2011, pp. 1612–1617.

[7] H. Hanafusa, T. Yoshikawa, and Y. Nakamura, "Analysis and control of articulated robot arms with redundancy," in *IFAC 8th. Triennal World Congres*, Kyoto, Japan, August 1981, pp. 1927–1932.

[8] 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.

[9] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 108, no. 3, pp. 163–171, September 1986.

[10] 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, October 2009.

[11] G. Jarquin, G. Arechavaleta, and V. Parra-Vega, "Continuous kinematic control with terminal attractors for handling task transitions of redundant robots," in *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, May 2013, pp. 1976–1981.

[12] N. Mansard, A. Remazeilles, and F. Chaumette, "Continuity of varying-feature-set control laws," *IEEE Transanctions on Automatic Control*, vol. 54, no. 11, pp. 2493–2505, November 2009.

[13] 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.

[14] F. Keith, P.-B. Wieber, N. Mansard, and A. Kheddar, "Analysis of the discontinuities in prioritized task-space control under discreet task scheduling operations," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, USA, September 2011, pp. 3887–3892.

[15] G. Jarquín, A. Escande, G. Arechavaleta, T. Moulard, E. Yoshida, and V. Parra-Vega, "Real-time smooth task transitions for hierarchical inverse kinematics," in *IEEE Int. Conf. on Humanoid Robots*, Atlanta, Georgia, October 2013.

[16] C. V. Loan, "On the method of weighting for equality-constrained least-squares problems," *SIAM Journal of Numeric Analysis*, vol. 22, no. 6, pp. 851–864, October 1985.