IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

On the Convergence of a Closed-Loop Inverse Kinematics Solver with Time-Varying Task Functions

Mario D. Fiore[1], Ciro Natale[1]

***Abstract*—Many control algorithms devised to allow redundant robots to execute complex multiple tasks with priorities require a numerical inverse kinematics (IK) solver. The present letter investigates the conditions that, if satisfied, guarantee that a specific module of closed-loop numerical IK solvers, which is at the kernel of some of the aforementioned algorithms, converges to a feasible solution. The investigation has the objective to prove the convergence in those cases when the task function is time-varying. The conditions found to ensure convergence include not only the initial task error and the loop gain - as it happens for stationary task functions - but also the maximum sampling time to be used in the computation of the solution.**

***Index Terms*—Inverse kinematics, Redundant Robots, Convergence Analysis.**

I. Introduction

EXPLOITING the redundant Degrees of Freedom (DOFs) of a robotic system is one of the most addressed challenges in robot motion generation. Several methods in the field of local optimization have been proposed over the last decades that instantaneously solve redundancy according to various costs and constraints. They are based on the definition of a so-called *task function* [1], which allows for the mathematical specification of the task objectives in the form of a regulation problem characterized by a vector function of the joint positions and, possibly,f time. Building on this approach, several constraint-based programming frameworks [2], [3], [4], [5], [6] have been developed for the specification of tasks as minimal set of constraints, and the generation of constrained-optimization problems for motion control. Such problems, are solved by numerical methods based on the pseudoinverse of the task Jacobian matrix and null space projections [7]. An essential element in these methods is the presence of a feedback term in the task reference [8], [9], which allows to recover from initial errors and cope with unavoidable drift of the solution due to numerical integration, whatever method is used. Closed-loop methods have been continuously extended, focusing on several aspects including singularity handling [10], joint limit avoidance [11], [12], the ability of (dynamically) achieving prioritized goals [13], and arbitrary inequality constraints handling [14], [15], [16].

Although in practice all the corresponding algorithms run on digital controllers, almost all the solid results consolidated over the years have been found in the continuous-time domain. Indeed, not so many papers address the study of redundancy resolution considering the discrete-time nature of the dynamic system at hand. In fact, this aspect is often overlooked in real applications. Yet, specific choices in the selection of the closed-loop gains or the integration method, can significantly affect the behavior of the redundancy resolution algorithm in terms of performance and/or stability [17]. For the basic inverse kinematics problem, some stability aspects have been addressed in [18], [19] proposing Lyapunov-based arguments or in [20] based on standard nonlinear solvers. Sufficient conditions for the convergence of an inverse kinematics solver based on Jacobian pseudo-inverse for the (sensor-based) control of redundant robots are instead proposed in [21], [22], leading to useful guidelines for gain selection in relation to the sampling time and initial task error as well as in relation to the bandwidth of inner motion control loops in the joint space.

Building on the methodology in [21], this paper proposes a convergence analysis for generic time-varying task function, as this aspect has not yet been investigated in the literature. As a matter of fact, the task function is often depending on time not only due to the time-varying reference trajectory of the task variables, but also due to the intrinsic time-varying characteristics [6] (see also Sect. IV). It is therefore reasonable to study this case, to see if and how the sufficient conditions found in [21] can be extended, with the aim to aid the design of such closed-loop solvers in any of the modern sophisticated algorithms dealing with redundancy resolution.

The paper is organized as follows: Sect. II recalls the task function approach and states the problem formulation; Sect. III streamlines the results of the presented convergence analysis; Sect. IV supports the theoretical findings with simulations carried out on a 7-DOFs manipulator; conclusion and possible future works are found in Sect. V.

II. Mathematical background

A. Task function approach

Let $\boldsymbol{e} \in \mathcal{E}$ be the vector of task error variables for a given robot, with $\mathcal{E}$ being a domain of $\mathbb{R}^m$. Moreover, let $\boldsymbol{q} \in \mathcal{Q} \subseteq \mathbb{R}^n$, with $m \leq n$, be the vector of the variables describing the robot joint configuration, and $t \in \mathcal{T} \subset \mathbb{R}_0^+$ indicate the time in an interval limited by the task execution time. The task error function is then expressed as

$$\boldsymbol{e} : (\boldsymbol{q}, t) \in \mathcal{Q} \times \mathcal{T} \rightarrow \boldsymbol{e}(\boldsymbol{q}, t) \in \mathcal{E}. \tag{1}$$

[1]Authors are with the Dipartimento di Ingegneria, Università della Campania "Luigi Vanvitelli", Via Roma 29, 81031 Aversa, Italy.
mariodaniele.fiore@unicampania.it, ciro.natale@unicampania.it

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

As an example, in a positioning task, $\boldsymbol{e}$ might represent the position error of the end effector. The function $\boldsymbol{e}(\boldsymbol{q}, t)$ is assumed of class $\mathcal{C}^1$, and the corresponding task is considered executed if $\boldsymbol{e} = \boldsymbol{0}$. Closed-loop Inverse Kinematics (CLIK) algorithms rely on the inversion of this constraint equation at the first-order differential level

$$\dot{\boldsymbol{e}}(\boldsymbol{q}, t) = \boldsymbol{0}, \tag{2}$$

where $\dot{\boldsymbol{e}}$ can be expressed as

$$\dot{\boldsymbol{e}}(\boldsymbol{q}, t) = \boldsymbol{J}(\boldsymbol{q}, t)\dot{\boldsymbol{q}} + \boldsymbol{j}(\boldsymbol{q}, t),$$

with

$$\boldsymbol{J}(\boldsymbol{q}, t) = \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}} \in \mathbb{R}^{m \times n}, \qquad \boldsymbol{j}(\boldsymbol{q}, t) = \frac{\partial \boldsymbol{e}}{\partial t} \in \mathbb{R}^m.$$

In particular, the solution to (2) is computed in a closed-loop fashion as [9]

$$\dot{\boldsymbol{q}} = -\boldsymbol{J}^{\#}(\boldsymbol{q}, t)(\boldsymbol{j}(\boldsymbol{q}, t) + \gamma \boldsymbol{e}(\boldsymbol{q}, t)), \tag{3}$$

where $\#$ denotes the generic right pseudo-inverse operator and γ is a positive scalar gain. The feedback term in (3) ensures an exponential convergence of $\boldsymbol{e}$ to zero, with a convergence rate depending on the value of γ. Starting from (3), the evolution of the joint positions $\boldsymbol{q}$ can be obtained by integration as

$$\boldsymbol{q}(t) = \boldsymbol{q}(0) - \int_0^t \boldsymbol{J}^{\#}(\boldsymbol{q}(\tau), \tau)(\boldsymbol{j}(\boldsymbol{q}(\tau), \tau) + \gamma \boldsymbol{e}(\boldsymbol{q}(\tau), \tau))d\tau. \tag{4}$$

B. Assumptions

The convergence analysis presented in this paper is based on the following assumptions:

1) $\exists\, \delta \in \mathbb{R}^+ : \|\boldsymbol{J}^{\#}(\boldsymbol{q}, t)\| \leq \delta, \quad \forall\, \boldsymbol{q} \in \mathcal{Q},\ t \in \mathcal{T}$
2) $\exists\, \mu \in \mathbb{R}^+ : \|\boldsymbol{H}(e_i(\boldsymbol{\theta}))\| \leq \mu, \forall \boldsymbol{\theta} \in \mathcal{Q} \times \mathcal{T}, i = 1, \ldots, m.$
3) $\exists\, \omega \in \mathbb{R}_0^+ : \|\boldsymbol{j}(\boldsymbol{q}, t)\| \leq \omega, \quad \forall\, \boldsymbol{q} \in \mathcal{Q},\ t \in \mathcal{T}$

Here, the spectral norm, i.e., the largest singular value of a matrix, is assumed as matrix norm. Moreover, e_i denotes the ith component of $\boldsymbol{e}(\cdot)$, whereas $\boldsymbol{\theta}$ groups the joint coordinates and the time variable in a single vector, namely $\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{q}^T & t \end{bmatrix}^T$. Assumption 1) aims at quantifying the remoteness from singularity of $\boldsymbol{J}$; assumption 2) imposes a smoothness constraint on the task function, as it assumes a bounded norm on $\mathcal{Q} \times \mathcal{T}$ of all the Hessian matrices $\boldsymbol{H}(e_i(\boldsymbol{\theta}))$ of the components of $\boldsymbol{e}(\boldsymbol{\theta})$. This smoothness assumption holds for the direct kinematics of revolute-joint manipulators, and for some typical spatial relations employed in constraint-based programming (see also Sect. IV). Finally, assumption 3) imposes a bound to the rate of variation of the task function, e.g., to the rate of variation of the task reference trajectory, which appears in the error function (1). For example, if the task consists in following a certain end-effector position with a given velocity profile, ω is an upper bound to the end-effector velocity norm.

The derivation of the task error dynamics uses Taylor's theorem with explicit second-order Lagrange remainders. The Taylor expansion of $\boldsymbol{e}(\boldsymbol{q} + \tilde{\boldsymbol{q}}, t + \tilde{t})$ around $(\boldsymbol{q}, t)$ for some $(\tilde{\boldsymbol{q}}, \tilde{t}) \in \mathbb{R}^{n+1}$ is given by

$$\boldsymbol{e}(\boldsymbol{q}+\tilde{\boldsymbol{q}}, t+\tilde{t}) = \boldsymbol{e}(\boldsymbol{q}, t)+\boldsymbol{J}(\boldsymbol{q}, t)\tilde{\boldsymbol{q}}+\boldsymbol{j}(\boldsymbol{q}, t)\tilde{t}+\boldsymbol{r}(\boldsymbol{q}, t, \tilde{\boldsymbol{q}}, \tilde{t}), \tag{5}$$

with the Lagrange remainder $\boldsymbol{r}$ being

$$\boldsymbol{r}(\boldsymbol{q}, t, \tilde{\boldsymbol{q}}, \tilde{t}) = \frac{1}{2} \begin{bmatrix} \tilde{\boldsymbol{\theta}}^T \boldsymbol{H}(e_1(\boldsymbol{\xi}))\tilde{\boldsymbol{\theta}} \\ \vdots \\ \tilde{\boldsymbol{\theta}}^T \boldsymbol{H}(e_m(\boldsymbol{\xi}))\tilde{\boldsymbol{\theta}} \end{bmatrix},$$

having denoted with $\tilde{\boldsymbol{\theta}}$ the vector $(\tilde{\boldsymbol{q}}, \tilde{t}) \in \mathbb{R}^{n+1}$ and being $\boldsymbol{\xi}$ a point of the line connecting $\boldsymbol{\theta}$ and $\boldsymbol{\theta} + \tilde{\boldsymbol{\theta}}$. As shown in Lemma 1 of [21], assumption 2) implies that $\boldsymbol{r}$ is bounded. In particular, it is

$$\|\boldsymbol{r}\| \leq \frac{\sqrt{m}}{2}\mu\|\tilde{\boldsymbol{\theta}}\|^2 \triangleq \nu \left(\|\tilde{\boldsymbol{q}}\|^2 + \tilde{t}^2\right). \tag{6}$$

C. Problem statement

When implementing the CLIK algorithm in discrete time, e.g., by resorting to the (forward) Euler integration method with sampling time T, (4) takes the form

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k - T\boldsymbol{J}_k^{\#}(\boldsymbol{j}_k + \gamma \boldsymbol{e}_k), \tag{7}$$

being $k \in \mathbb{Z}_0^+$ the discrete time variable, i.e., $\boldsymbol{q}_k = \boldsymbol{q}(kT)$. Moreover, it is $\boldsymbol{J}_k = \boldsymbol{J}(\boldsymbol{q}_k, kT)$, $\boldsymbol{j}_k = \boldsymbol{j}(\boldsymbol{q}_k, kT)$, and $\boldsymbol{e}_k = \boldsymbol{e}(\boldsymbol{q}_k, kT)$. Here, the closed-loop action additionally aims at avoiding the drift of the error function that comes as a result of the numerical integration. Different integration methods can be used as well, and preliminary results on the comparison between forward Euler and Runge-Kutta 4 can be found in [17]. This work instead aims at finding sufficient conditions for the convergence of the algorithm (7), i.e., at determining if there exist a positive feedback gain γ and an admissible initial condition $\boldsymbol{q}_0$ such that

$$\lim_{k \to \infty} \|\boldsymbol{e}_k\| = 0$$

III. Convergence Analysis

This section introduces the convergence of the discrete-time CLIK algorithm for time varying task error functions. Resorting again to the Euler integration method, eqs. (5) and (7) can be used to derive an expression of the dynamics of the task error function

$$\begin{aligned} \boldsymbol{e}_{k+1} &= \boldsymbol{e}_k - T\boldsymbol{J}_k\boldsymbol{J}_k^{\#}(\boldsymbol{j}_k + \gamma \boldsymbol{e}_k) + T\boldsymbol{j}_k + \boldsymbol{r}_k \\ &= \boldsymbol{e}_k - T\boldsymbol{j}_k - \gamma T\boldsymbol{e}_k + T\boldsymbol{j}_k + \boldsymbol{r}_k \\ &= (1 - \gamma T)\boldsymbol{e}_k + \boldsymbol{r}_k \end{aligned} \tag{8}$$

with $\boldsymbol{r}_k = \boldsymbol{r}(\boldsymbol{q}_k, kT, \boldsymbol{\zeta})$. To prove the convergence of the error, consider the following inequalities:

$$\begin{aligned} \|\boldsymbol{e}_{k+1}\| &\leq |1 - \gamma T|\|\boldsymbol{e}_k\| + \|\boldsymbol{r}_k\| \\ &\leq |1 - \gamma T|\|\boldsymbol{e}_k\| + \nu\|T\boldsymbol{J}_k^{\#}(\boldsymbol{j}_k + \gamma \boldsymbol{e}_k)\|^2 + \nu T^2 \\ &\leq |1 - \gamma T|\|\boldsymbol{e}_k\| + T^2\nu\delta^2\|\boldsymbol{j}_k + \gamma \boldsymbol{e}_k\|^2 + \nu T^2 \\ &\leq |1 - \gamma T|\|\boldsymbol{e}_k\| + \\ &\qquad T^2\nu\delta^2(\omega^2 + \gamma^2\|\boldsymbol{e}_k\|^2 + 2\gamma\omega\|\boldsymbol{e}_k\|) + \nu T^2 \\ &= (|1 - \gamma T| + 2\gamma T^2\nu\delta^2\omega + \gamma^2 T^2\nu\delta^2\|\boldsymbol{e}_k\|)\|\boldsymbol{e}_k\| \\ &\qquad + T^2\nu\delta^2\omega^2 + \nu T^2 \end{aligned} \tag{9}$$

where the bounds in assumptions 1) and 3), and in (6) have been exploited, together with standard norm properties.

Before presenting the main result showing that the error is bounded by an exponential sequence, a short note on the relation to the case of time-invariant task error functions is opportune. In (9), the case of a time-invariant task error function is easily recovered by letting not only $\omega = 0$ but also removing the last term νT^2 deriving from the Taylor series remainder in (6). Therefore, to better highlight how the time-invariant case is a special case of the more general dissertation here, let denote the last two terms in (9) with $T^2\nu\mu_t^2$, having defined

$$\mu_t^2 = 1 + \delta^2\omega^2, \tag{10}$$

which should be set to zero in the time-invariant case.

Finally, the following lemma is conveniently introduced to streamline the proof of the main theorem.

Lemma 1. *Let γ, T, ν, δ, and ω be positive real numbers, with $\gamma T < 1$. Moreover, let α be the unknown variable of the second order equation*

$$\alpha^2 + b\alpha + c = 0, \tag{11}$$

where $b = \gamma T - 2 - 2\gamma T^2\nu\delta^2\omega$, $c = 1 - \gamma T + 2\gamma T^2\nu\delta^2\omega + \gamma^2T^4\nu^2\delta^2\mu_t^2$, and $\mu_t^2 = \delta^2\omega^2 + 1$. Equation (11) admits two real positive solutions smaller than 1 iff

$$\gamma < \min\left(\frac{1}{T}, \frac{1 - 2T\nu\delta^2\omega}{T^3\nu^2\delta^2\mu_t^2}\right) \quad \textit{and} \quad T \leq \frac{1}{2\nu\delta(\delta\omega + \mu_t)}.$$

Proof. Since $\gamma T < 1$ and $\omega > 0$, it is $c > 0$ and $b < 0$, hence eq. (11) always admits two positive solutions if they are real, i.e., iff

$$b^2 \geq 4c \Leftrightarrow \gamma^2T^2(1 - 2T\nu\delta^2\omega)^2 \geq 4\gamma^2T^4\nu^2\delta^2\mu_t^2$$
$$1 - 2T\nu\delta^2\omega \geq 2T\nu\delta\mu_t \Leftrightarrow T \leq \frac{1}{2\nu\delta(\delta\omega + \mu_t)},$$

that is ensured by the Lemma hypothesis. These solutions α are smaller than one iff (Jury stability criterion)

$$0 < c < 1 \tag{12a}$$
$$(c + 1 - b)(c + 1 + b) > 0. \tag{12b}$$

Since it is already $c > 0$, the condition (12a) is satisfied by imposing $c < 1$. This yields

$$1 - \gamma T + 2\gamma T^2\nu\delta^2\omega + \gamma^2T^4\nu^2\delta^2\mu_t^2 < 1$$
$$\gamma T(-1 + 2T\nu\delta^2\omega + \gamma T^3\nu^2\delta^2\mu_t^2) < 0$$
$$\Leftrightarrow \gamma < \frac{1 - 2T\nu\delta^2\omega}{T^3\nu^2\delta^2\mu_t^2}.$$

Moreover, the condition (12b) is always verified since $b < 0$, and thus $c + 1 - b > 0$, and $c + 1 + b$ is also positive, in fact

$$c + 1 + b = 1 - \gamma T + 2\gamma T^2\nu\delta^2\omega + \gamma^2T^4\nu^2\delta^2\mu_t^2$$
$$+ 1 + \gamma T - 2 - 2\gamma T^2\nu\delta^2\omega = \gamma^2T^4\nu^2\delta^2\mu_t^2 > 0.$$

□

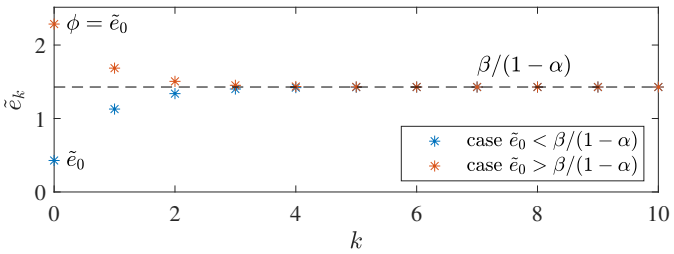

Fig. 1. Expected task error qualitative trends depending on the initial error.

Theorem 1. *Under the assumptions 1), 2), and 3), if the gain γ, the sampling time T and the initial task space error $\boldsymbol{e}_0$ are such that*

$$0 < \gamma < \min\left(\frac{1}{T}, \frac{1 - 2T\nu\delta^2\omega}{T^3\nu^2\delta^2\mu_t^2}\right) \quad \textit{and}$$
$$\|\boldsymbol{e}_0\| < \tilde{e}_{0,l} \quad \textit{and} \tag{13}$$
$$T \leq \frac{1}{2\nu\delta(\delta\omega + \mu_t)},$$

or

$$0 < \gamma < \min\left(\frac{1}{T}, \frac{1 - 2T\nu\delta^2\omega}{T\nu\delta^2\|\boldsymbol{e}_0\|}\right) \quad \textit{and}$$
$$\tilde{e}_{0,l} < \|\boldsymbol{e}_0\| < \tilde{e}_{0,u} \quad \textit{and} \tag{14}$$
$$T \leq \frac{1}{2\nu\delta(\delta\omega + \mu_t)},$$

where $\tilde{e}_{0,l}$ and $\tilde{e}_{0,u}$ are the initial lower and upper error bounds, respectively expressed as

$$\frac{(1 - 2T\nu\delta^2\omega) \mp \sqrt{(1 - 2T\nu\delta^2\omega)^2 - 4T^2\nu^2\delta^2\mu_t^2}}{2\gamma T\nu\delta^2}, \tag{15}$$

then the CLIK algorithm in (7) ensures the exponential convergence of the task space error dynamics, i.e.,

$$\exists\, \alpha \in (0,1), \tilde{e}_s > 0, \tilde{e}_t \in \mathbb{R} : \|\boldsymbol{e}_k\| \leq \tilde{e}_t\alpha^k + \tilde{e}_s \quad \forall k \geq 0. \tag{16}$$

Eventually, the configuration variables $\boldsymbol{q}_k$ are such that their increment remains bounded, i.e.,

$$\|\boldsymbol{q}_{k+1} - \boldsymbol{q}_k\| \leq T\delta(\omega + \gamma\tilde{e}_t\alpha^k + \gamma\tilde{e}_s),\ \forall k \geq 0. \tag{17}$$

Proof. In view of (9), assume that the task error norm is bounded, i.e.,

$$\|\boldsymbol{e}_k\| \leq \phi \quad \forall k \geq 0, \tag{18}$$

and consider the case $\gamma T < 1$. Then, eq. (9) can be rewritten as

$$\|\boldsymbol{e}_{k+1}\| \leq \alpha\|\boldsymbol{e}_k\| + \beta \quad \forall k \geq 0,$$

with

$$\alpha = 1 - \gamma T + 2\gamma T^2\nu\delta^2\omega + \gamma^2T^2\nu\delta^2\phi,$$
$$\beta = T^2\nu\mu_t^2. \tag{19}$$

It will soon be shown that the condition (18) is guaranteed by the hypothesis (13) and (14). To this purpose, consider the scalar linear system

$$\tilde{e}_{k+1} = \alpha\tilde{e}_k + \beta,$$

which asymptotically converges to a constant for any $\alpha < 1$, and its response with initial condition $\tilde{e}_0 = \|\boldsymbol{e}_0\|$ is given by

$$\begin{aligned}\tilde{e}_k &= \tilde{e}_0\alpha^k + \beta\frac{1-\alpha^k}{1-\alpha} \\ &= \left(\tilde{e}_0 - \frac{\beta}{1-\alpha}\right)\alpha^k + \frac{\beta}{1-\alpha} \triangleq \tilde{e}_t\alpha^k + \tilde{e}_s\end{aligned}.$$

Two cases may happen, both illustrated in Fig. 1. In the case

$$\tilde{e}_0 < \frac{\beta}{1-\alpha}, \tag{20}$$

the choice $\phi = \beta/(1-\alpha)$ can be considered. This leads to an implicitly defined α, namely

$$\alpha = 1 - \gamma T + 2\gamma T^2\nu\delta^2\omega + \gamma^2 T^2\nu\delta^2\frac{\beta}{1-\alpha}$$

that is equation (11) in Lemma 1. Under the conditions (13), such equation admits in view of the result of Lemma 1 two positive solutions α' and α'', with $\alpha' < \alpha'' < 1$. Thus, two different bounds are possible for $\tilde{e}_0$ according to (20). The most restrictive limit is obtained as

$$\begin{aligned}\frac{\beta}{1-\alpha'} &= \frac{T\nu\mu_t^2/\gamma}{1 - 2T\nu\delta^2\omega + \sqrt{(1-2T\nu\delta^2\omega)^2 - 4T^2\nu^2\delta^2\mu_t^2}} \\ &= \frac{1 - 2T\nu\delta^2\omega - \sqrt{(1-2T\nu\delta^2\omega)^2 - 4T^2\nu^2\delta^2\mu_t^2}}{2\gamma T\nu\delta^2},\end{aligned}$$

which is $\tilde{e}_{0,l}$ from (15). Then, from Lemma 2 in [21], it results

$$\|\boldsymbol{e}_k\| \leq \tilde{e}_t\alpha^h + \tilde{e}_s \quad \forall k \geq 0,$$

which proves (16) and ensures that (18) is verified.

In the case $\tilde{e}_0 \geq \beta/(1-\alpha)$, ϕ can be taken from the system initial value, i.e., $\phi = \tilde{e}_0$. To have $\alpha < 1$, the following inequality must hold

$$1 - \alpha = \gamma T - 2\gamma T^2\nu\delta^2\omega - \gamma^2 T^2\nu\delta^2\tilde{e}_0 > 0,$$

which is ensured by the condition on γ from (14). Then, Lemma 2 in [21] ensures that the result (16) is obtained. The proof is completed by noting that the condition on $\tilde{e}_0 \geq \beta/(1-\alpha)$ generates the following inequality

$$\tilde{e}_0 \geq \frac{\beta}{\gamma T - 2\gamma T^2\nu\delta^2\omega - \gamma^2 T^2\nu\delta^2\tilde{e}_0}$$

that, in view of (19), is

$$(\gamma^2 T\nu\delta^2)\tilde{e}_0^2 - \gamma(1 - 2T\nu\delta^2\omega)\tilde{e}_0 + \nu T\mu_t^2 \leq 0, \tag{21}$$

which, under the condition on T from (14) is equivalent to $\tilde{e}_{0,l} \leq \tilde{e}_0 \leq \tilde{e}_{0,u}$, with $\tilde{e}_{0,l}$ and $\tilde{e}_{0,u}$ defined in (15). The last statement (17) immediately follows from (7) in view of assumptions 1 and 3 and of (16). □

Remark 1. *The main theorem considers only the case $\gamma < 1/T$, while convergence may be obtained with higher values of the gain. In fact, with an analogous procedure, it is possible to derive conditions on the sampling step and the initial error for gains γ up to $2/T$, as for the case of stationary task functions [21]. Nevertheless, these conditions are left out of this paper as they do not add any interesting insight. Indeed, the conclusion is always that convergence is facilitated by low sampling times, low initial errors and low gains.*

Remark 2. *It is important to underline that the main theorem ensures that the task error remains always bounded under the considered assumptions and conditions on the gain, the sampling time, and the initial error. Nevertheless, in a real application, the task reference usually becomes constant after a given execution time, and thus the result of [21] can be used to ensure that the task error goes to zero and the configuration variables achieve a constant value. It is not surprising how that result is a special case of Theorem 1. This can be easily recognized by imposing $\omega = 0$ and $\mu_t = 0$ (hence $\beta = 0$) in the conditions* (14) *of Theorem 1. In such a case there is no upperbound on the sampling time, and all the convergence conditions refer to the product γT only. Whereas, in case of a time-varying error function, there is a limit on the maximum sampling time, independently from the gain γ. Such a limit depends on the three parameters δ, ω and ν characterizing the distance from singularities, the rate of variability in time of the error function and its degree of nonlinearity, respectively. This result might be of great help in those cases where the convergence of the inverse kinematics algorithm is problematic with the minimum sampling time achievable with the available hardware. Indeed, the theorem suggests to stay further from the singularities, if possible, or, in case the task requirements are strict, to slow down the task itself. If the task cannot be scaled down, there is no other way than reducing the sampling time by using a more performant hardware.*

Remark 3. *Even though the conditions of Theorem 1 are actually quantitative, the three key parameters δ, ω and ν might be difficult to estimate or might lead to very conservative conditions. The latter aspect is especially critical for the ν parameter, even if this is not surprising, because eq.* (6) *summarizes the nonlinear character of the error function into a single number, that is a really sharp description of a generic, even though smooth, nonlinear function.*

IV. Simulations

This sections presents two simulations carried out on a KUKA LBR iiwa 7-DOFs manipulator (Fig. 2) to numerically support the theoretical findings of this paper. Simulations are executed in MATLAB® environment.

A. Distance constraint

In the first simulation, the task consists in keeping the tip of the robot tool (the orange cone attached to the flange) at a certain time-varying distance from a moving reference point (Fig. 2). The tool tip is indicated by $P(x, y, z)$, whereas $P_d(x_d, y_d, z_d)$ indicates the reference point. Moreover, $d_d \in \mathbb{R}^+$ is the desired Euclidean distance between the two points. Thus, the task can be expressed using the error function

$$e(\boldsymbol{q}, t) = \|P(\boldsymbol{q}) - P_d(t)\|_2 - d_d(t),$$

which needs to be regulated to zero as illustrated in Sect. II-A. The position of the reference point P_d varies on a circular path (center $P_{d,0}(0.4, 0.25, 0.3)$ m, radius 0.15 m) with sinusoidal velocity profile having time period 1.5 s. The desired distance d_d ranges between 0.06 m and 0.1 m, following a sinusoidal

Fig. 2. Distance constraint: initial setup and motion sequence. The blue sphere allows to visually evaluate the position of the reference point P_d and the evolution of the desired distance d_d.

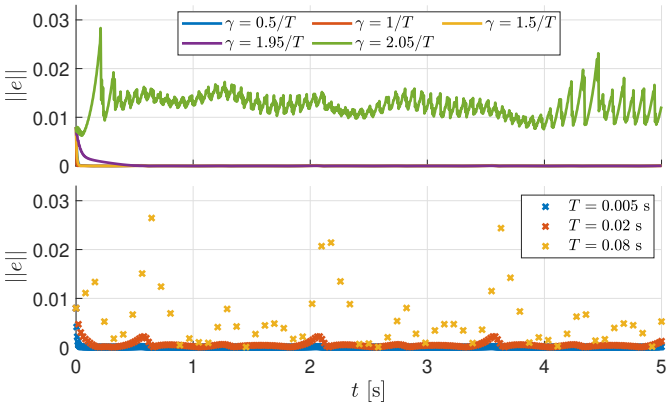

Fig. 3. Distance constraint: norm of the task function under variation of the gain γ (top), for e_0 and T fulfilling the conditions (13); norm of the task function under variation of the sampling time T (bottom), for e_0 and γ fulfilling the conditions (13).

curve of time period 1.5 s. The total simulation time is 5 s. In order to perform the simulation, a subset of $\mathcal{Q}$ has been considered, in which the Jacobian matrix $\boldsymbol{J}$ is full rank. In this subset, it was possible to estimate the constants $\delta = 5.09$, $\omega = 0.71$, and $\mu = 4.1$ from the assumptions 1), 2), and 3), respectively. In detail, parameters μ, δ and ω are estimated by sampling the configuration space $\mathcal{Q}$ as well as the time interval $\mathcal{T}$, and by numerically computing the second order derivatives of the error task function, singular values of the task Jacobian, and the partial time derivative of the error task function, respectively. Then, from Theorem 1 it must be $T < 0.0065$ s. Setting $T = 0.005$ s, Theorem 1 additionally returns $\tilde{e}_{0,l} = 0.013$ m, $\tilde{e}_{0,u} = 0.10$ m, and $\gamma < 1/T$ for both (13) and (14). For an initial error $e_0 = 0.008$ m (thus fulfilling conditions (13)), Fig. 3 presents the trend of the task function norm under variation of the gain γ (top plot), showing that it is possible to obtain convergence also for higher values than $1/T$, as highlighted in Remark 1. In particular, Fig. 3 shows that convergence is lost only for $\gamma > 2/T$, which is the practical limit also found in [21], [17]. The trend of the task function norm under variation of the sampling time T is also reported in Fig. 3 for $\gamma = 0.5/T$ (bottom plot). Although the resulting error might be considered unsatisfactory

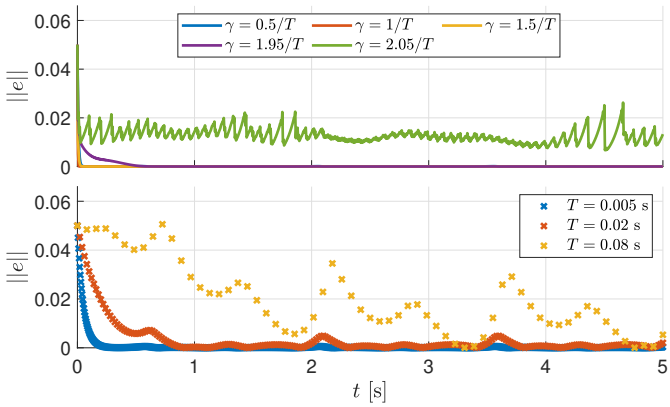

Fig. 4. Distance constraint: norm of the task function under variation of the gain γ (top), for e_0 and T fulfilling the conditions (14); norm of the task function under variation of the sampling time T (bottom), for e_0 and γ fulfilling the conditions (14).

in practical applications due to large oscillations, it does not present a divergent behavior for sampling times greater than 0.0065 s, indicating that the bound on T in (13)-(14) might be conservative. A similar outcome is obtained in Fig. 4 for an initial error $e_0 = 0.05$ fulfilling the conditions (14).

B. Laser tracing

The second simulation showcases a laser tracing application in which the robot has to trace two moving reference points with two different laser devices (Fig. 5). Indicating the reference points as $P_{d,1}$ and $P_{d,2}$, the task can be expressed by the following error function

$$\boldsymbol{e}(\boldsymbol{q}, t) = \begin{bmatrix} \text{null}(\boldsymbol{d}_1^T(\boldsymbol{q})) \cdot (P_1(\boldsymbol{q}) - P_{d,1}) \\ \text{null}(\boldsymbol{d}_2^T(\boldsymbol{q})) \cdot (P_2(\boldsymbol{q}) - P_{d,2}) \end{bmatrix}, \tag{22}$$

where P_i and $\boldsymbol{d}_i$ represent the point and the direction vector describing the line passing through the ith laser beam. Both reference points move on linear segments with trapezoidal velocity profiles, and reach their final positions at $t = 1.2$ s. Thus, from this time on, the task function is time invariant. For the considered subset of $\mathcal{Q}$, it is $\delta = 5.09$, $\omega = 0.45$, and $\mu = 4.20$. Thus, from Theorem 1 it is $T \leq 0.0049$ s, $\tilde{e}_{0,l} = 0.00029$ and $\tilde{e}_{0,u} = 0.082$. Figure 6 reports the trend of the task error norm for two different values of $\|\boldsymbol{e}_0\|$, one smaller than $\tilde{e}_{0,l}$ and one in the range $[\tilde{e}_{0,l}, \tilde{e}_{0,u}]$, respectively. The plots are generated for $T = 0.001$ s and $\gamma = 0.1/T$. In the initial phase, it is possible to recognize the two possible behaviors analyzed in Theorem 1, with the norm of the error function decreasing or increasing depending on the value of the initial error with respect to $\beta/(1-\alpha)$. Furthermore, it is also possible to notice how the norm of the error decreases in the parts of the trajectory where a constant velocity is imposed to the reference points, as opposed to the parts with constant accelerations. The simulation is intentionally stopped at 3.2 s to study the convergence properties also after all time-varying terms in the task function have become stationary. It is possible to observe that at $t = 1.2$ s the error norm starts to exponentially converge to zero. As highlighted in Remark 2, the conditions (13)-(14) fall into the ones from [21], which

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

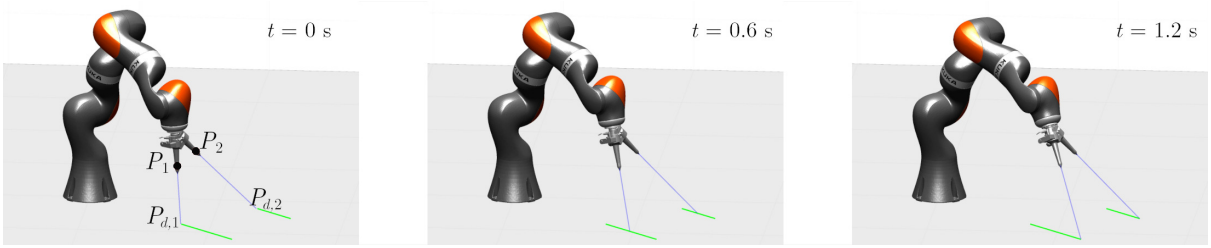

Fig. 5. Initial setup and motion sequence of the laser tracing application.

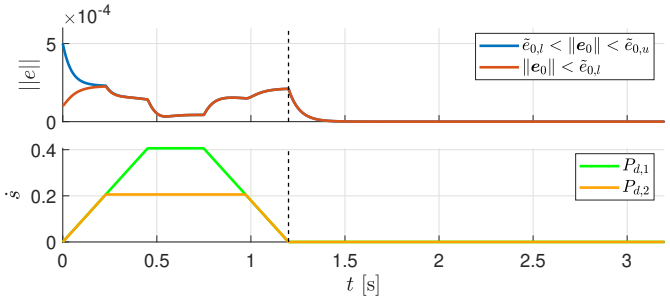

Fig. 6. Laser tracing: the vertical dashed line indicates the time from which the task function is time invariant. Top: norm of the task function for $\|\boldsymbol{e}_0\| < \tilde{e}_{0,l}$ and $\|\boldsymbol{e}_0\|$ in the range $[\tilde{e}_{0,l}, \tilde{e}_{0,u}]$. Bottom: path parameter time derivative, $\dot{s}$, for the trajectory planning of the two reference points $P_{d,1}$ and $P_{d,2}$.

guarantees convergence to zero imposing only a limitation to the product γT, and a larger upper bound for $\|\boldsymbol{e}_0\|$.

V. Conclusion

Many methods for motion generation of robots executing multiple tasks rely on closed-loop inverse kinematics solvers. The convergence properties of such solvers are almost always studied for continuous-time algorithms. However, it is obvious that such algorithms are always implemented in discrete time. Moreover, modern motion generation algorithms based on constraint-based programming offer time-varying task functions to execute complex tasks, which often involve multiple robots that need to be synchronized. Therefore, this letter tackled the relevant problem of finding sufficient conditions that ensure convergence of the closed-loop solver depending on not only the initial error and the loop gain but also on the minimum sampling time required to achieve convergence. The letter presented a couple of case studies to highlight the relevance of the investigation, which also explains some typical situations of critical convergence of such algorithms. Possible ways to obtain less conservative conditions might be including higher order terms of the Taylor expansion in the error dynamics in (8), or, by sacrificing generality, searching for conditions valid for specific error functions only.

References

[1] C. Samson, B. Espiau, and M. L. Borgne, *Robot control: the task function approach*. Oxford University Press, Inc., 1991.

[2] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *The International Journal of Robotics Research*, vol. 26, no. 5, pp. 433–455, 2007.

[3] E. Aertbeliën and J. De Schutter, "etasl/etc: A constraint-based task specification language and robot controller using expression graphs," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 1540–1546.

[4] I. Kresse and M. Beetz, "Movement-aware action control—integrating symbolic and control-theoretic action execution," in *2012 IEEE Int. Conf. on Robotics and Automation*. IEEE, 2012, pp. 3245–3251.

[5] N. Somani, A. Gaschler, M. Rickert, A. Perzylo, and A. Knoll, "Constraint-based task programming with cad semantics: From intuitive specification to real-time control," in *2015 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2854–2859.

[6] M. D. Fiore, "Intuitive programming of redundant robots leveraging constraint-based techniques: Execution of industrially relevant tasks," Ph.D. dissertation, Università della Campania L. Vanvitelli, https://www.ingegneria.unicampania.it/roboticslab/publications#phd-theses, 2023.

[7] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Fifth International Conference on Advanced Robotics*. IEEE, 1991, pp. 1211–1216.

[8] A. Balestrino, G. De Maria, and L. Sciavicco, "Robust control of robotic manipulators," *IFAC Proc. Vol.*, vol. 17, no. 2, pp. 2435–2440, 1984.

[9] L. Sciavicco and B. Siciliano, "Coordinate transformation: A solution algorithm for one class of robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 4, pp. 550–559, 1986.

[10] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, 1997.

[11] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.

[12] F. Flacco, A. De Luca, and O. Khatib, "Control of redundant robots under hard joint constraints: Saturation in the null space," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 637–654, 2015.

[13] 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, 2019.

[14] 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.

[15] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen, and J. Schrimpf, "Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results," *Frontiers in Robotics and AI*, vol. 3, 2016.

[16] M. D. Fiore, G. Meli, A. Ziese, B. Siciliano, and C. Natale, "A general framework for hierarchical redundancy resolution under arbitrary constraints," *IEEE Trans. on Rob.*, vol. 39, no. 3, pp. 2468–2487, 2023.

[17] M. D. Fiore and C. Natale, "Discrete-time closed-loop inverse kinematics: A comparison between euler and rk4 methods," in *2021 29th Medit. Conf. on Control and Automation*. IEEE, 2021, pp. 584–589.

[18] G. De Maria and R. Marino, "A discrete algorithm for solving the inverse kinematic problem for robotic manipulators," in *2nd International Conference on Advanced Robotics*, 1985, pp. 275–282.

[19] H. Das, J. E. Slotine, and T. B. Sheridan, "Inverse kinematic algorithms for redundant systems," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*. IEEE, 1988, pp. 43–48.

[20] T. Sugihara, "Solvability-unconcerned inverse kinematics by the Levenberg–Marquardt method," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 984–991, 2011.

[21] P. Falco and C. Natale, "On the stability of closed-loop inverse kinematics algorithms for redundant robots," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 780–784, 2011.

[22] M. Bjerkeng, P. Falco, C. Natale, and K. Y. Pettersen, "Stability analysis of a hierarchical architecture for discrete-time sensor-based control of robotic systems," *IEEE Trans. on Rob*, vol. 30, no. 3, pp. 745–753, 2014.