Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Coordinated multi-robot trajectory tracking control over sampled communication[☆]

Enrica Rossi [a], Marco Tognon [b], Luca Ballotta [c,*], Ruggero Carli [c], Juan Cortés [d], Antonio Franchi [d,e], Luca Schenato [c]

[a] *Mas Automazioni S.r.l., Torre di Mosto, Italy*
[b] *Inria, Université de Rennes, CNRS, IRISA, Rennes, France*
[c] *Department of Information Engineering, University of Padova, Italy*
[d] *LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France*
[e] *Robotics and Mechatronics lab, University of Twente, Enschede, The Netherlands*

ARTICLE INFO

Article history:
Received 3 December 2021
Received in revised form 29 November 2022
Accepted 15 January 2023
Available online 26 February 2023

Keywords:
Control over sampled communications
Distributed control
Multi-robot systems
Trajectory tracking
UAVs

ABSTRACT

In this paper, we propose an inverse-kinematics controller for a class of multi-robot systems in the scenario of sampled communication. The goal is to make a group of robots perform trajectory tracking in a coordinated way when the sampling time of communications is much larger than the sampling time of low-level controllers, disrupting theoretical convergence guarantees of standard control design in continuous time. Given a desired trajectory in configuration space which is pre-computed offline, the proposed controller receives configuration measurements, possibly via wireless, to re-compute velocity references for the robots, which are tracked by a low-level controller. We propose joint design of a sampled proportional feedback plus a novel continuous-time feedforward that linearizes the dynamics around the reference trajectory: this method is amenable to distributed communication implementation where only one broadcast transmission is needed per sample. Also, we provide closed-form expressions for instability and stability regions and convergence rate in terms of proportional gain k and sampling period T. We test the proposed control strategy via numerical simulations in the scenario of cooperative aerial manipulation of a cable-suspended load using a realistic simulator (*Fly-Crane*). Finally, we compare our proposed controller with centralized approaches that adapt the feedback gain online through smart heuristics, and show that it achieves comparable performance.

1. Introduction

Unmanned Aerial Vehicles (UAVs) are used in the context of mobile robotics to perform surveillance, coverage, exploration, and transportation (Maza, Kondak, Bernard, & Ollero, 2010). Generally speaking, a group of robots allows to improve task performance with respect to (w.r.t.) the single-robot solution. Multiple robots can mitigate problems such as limited payload and time of flight (Arai, Pagello, Parker, et al., 2002). However, they require careful consideration of cooperation or coordination strategies to achieve common goals (Erhart & Hirche, 2015). Consider a group of UAVs that transports a load while avoiding obstacles. In this case, not only each UAV needs to avoid obstacles, but the overall multi-robot system should move in a way such that the load avoids obstacles, as well (Cortés & Egerstedt, 2017).

One of the most direct approaches to control multi-robot systems is the centralized kinematic/dynamic inversion (Antonelli & Chiaverini, 2006; Manubens, Devaurs, Ros, & Cortés, 2013; Masone, Bülthoff, & Stegagno, 2016; Sanalitro, Savino, Tognon, Cortés, & Franchi, 2020; Sreenath & Kumar, 2013). Because of its nature, failure of the central unit may cause the whole task to fail. Distributed and decentralized approaches spread computations across the robots, guaranteeing greater robustness and flexibility w.r.t. a centralized one (Sabattini, Secchi, Chopra, & Gasparri, 2013). Although such approaches are preferable for multi-robot systems, they often lack global information such as load state and parameters, or the total number of robots. This aspect increases the difficulty of the controller design and might even degrade performance. Examples of distributed control methods, where robots explicitly exchange local data, can

[☆] This work was partially funded by: the European Commission project H2020 AERIAL-CORE (EC 871479), the ANR, Project ANR-17- CE33-0007 MuRoPhen, and by University of Padova, Italy grant “Magic” SCHE_SID17_01. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Dimitra Panagou under the direction of Editor Christos G. Cassandras.

[*] Corresponding author.
E-mail addresses: enrica.rossi.1@studenti.unipd.it (E. Rossi), marco.tognon@inria.fr (M. Tognon), ballotta@dei.unipd.it (L. Ballotta), carlirug@dei.unipd.it (R. Carli), juan.cortes@laas.fr (J. Cortés), a.franchi@utwente.nl (A. Franchi), schenato@dei.unipd.it (L. Schenato).

https://doi.org/10.1016/j.automatica.2023.110941
0005-1098/© 2023 Elsevier Ltd. All rights reserved.

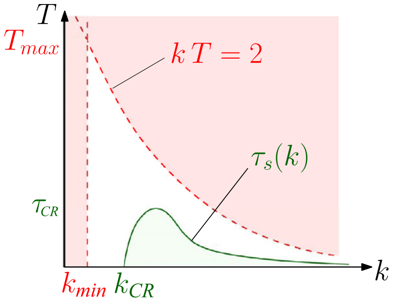


Fig. 1. We propose a controller for trajectory tracking when measurements are transmitted every T seconds, with feedback proportional gain k. We show that, if (k, T) lies outside the dashed red curve, the tracking error is not ρ-monotonically contractive, while zero tracking error is guaranteed if (k, T) lies below the solid green curve $\tau_s(k)$. In particular, there exist sampling time T_{max} such that no gain k can guarantee convergence if $T > T_{max}$ and sampling time τ_{CR} such that a stabilizing gain exists for any $T < \tau_{CR}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

be found for groups of ground (Dimarogonas & Kyriakopoulos, 2006; Franchi, Petitti, & Rizzo, 2019; Pereira, Campos, & Kumar, 2004; Ren, Sosnowski, & Hirche, 2020), underwater (Conti, Meli, Ridolfi, & Allotta, 2015; Simetti & Casalino, 2016), and aerial robots (Mellinger, Shomin, Michael, & Kumar, 2013). Conversely, decentralized control strategies allow for no direct communication among robots, with examples including ground or aerial manipulators (Sieber & Hirche, 2019; Verginis, Nikou, & Dimarogonas, 2018). Also, to reduce communication issues, communication-less approaches relying on a leader–follower paradigm were presented for cooperative transportation and manipulation (Farivarnejad & Berman, 2018; Gabellieri, Tognon, Sanalitro, Palottino, & Franchi, 2020; Tagliabue, Kamel, Siegwart, & Nieto, 2019; Tsiamis, Verginis, Bechlioulis, & Kyriakopoulos, 2015; Wang & Schwager, 2016). In these cases, communication is implicitly given by forces exerted on the load (Tognon, Gabellieri, Pallottino, & Franchi, 2018). However, force feedback may be insufficient for precise tracking, because it lacks pose information. The latter can be retrieved by communication among robots, e.g., making them exchange poses, or installing a sensor on the load to broadcast its pose. This setup can also be extended to formation control problems where a group of robots needs to complete a task (Franchi & Giordano, 2018; Franchi et al., 2012). For example, a common goal is mapping or surveillance of an area while robots keep a certain 3D formation in order to, e.g., minimize overlaps among their fields of view (Schwager, Julian, Angermann, & Rus, 2011). In this case, communication-based approaches let robots exchange group-level measurements, such as relative distances.

In applications, limited bandwidth of wireless channels disrupts the assumption of continuous-measurement feedback. As so, specific strategies are needed to deal with sampled communication (Hayakawat, Matsuzawat, & Harat, 2006; Liu, Chen, Lu, Wang, & Wang, 2016; Stramigioli, Secchi, van der Schaft, & Fantuzzi, 2005). However, little work is currently available for multi-robot manipulation.

1.1. Preview of key results

In this paper, we aim to design a multi-robot distributed communication controller for trajectory tracking when wireless communication induces non-negligible sampling of feedback measurements. In this context, a distributed implementation is preferred because (i) it reduces the overall communication burden and related issues such as packet loss or latency and (ii) it enhances system robustness and scalability. Inspired by previous work (Rossi et al., 2020) where point-stabilization was considered, we propose a novel *Sampled communication-aware Inverse-Kinematic controller for Multi-robot systems* (SIKM) to address the problem of trajectory tracking under sampled communication. Our contributions are summarized as follows.

- We develop a distributed SIKM controller for trajectory tracking that receives sampled measurements and re-computes reference robot velocities along the trajectory, exploiting a novel continuous-time feedforward term that allows exact trajectory tracking even in the presence of sampled communication.
- We show that, differently from Rossi et al. (2020), there are a maximum sampling time T_{max} and a minimum feedback gain k_{min} beyond which trajectory tracking cannot be achieved, as graphically depicted in red in Fig. 1.
- We provide closed-form expressions, whose coefficients can be numerically computed, for the stability region (depicted in green in Fig. 1) and for the (exponential) convergence rate of the trajectory tracking error norm in terms of communication sampling period T and feedback gain k. We consider stability in terms of *ρ-monotonic contractiveness*, which, roughly speaking, ensures that the trajectory monotonically decreases the tracking error overtime in the absence of external disturbances and is bounded away from singularities.
- We validated our strategy by testing the controller on a realistic dynamical simulator which replicates with high accuracy the experimental setup available at LAAS-CNRS Lab, *Fly-Crane* (Sanalitro et al., 2020), including dynamical inertial terms, motor actuators, sensor noise, and real-time embedded software implementation.

1.2. Paper outline

In Section 2 we introduce the class of considered multi-robot systems, provide the kinematic model (Section 2.1), and give an example of real system (Section 2.2). In Section 3 we review control architectures for trajectory tracking, and present our proposed SIKM controller in Section 3.1. In Section 4 we derive fundamental stability limitations in terms of feedback gain and sampling time. In Section 5 we compute an upper bound for the convergence rate, outline a numerical procedure to estimate it from data (Section 5.1), and explicitly find controller parameters that yield the fastest convergence (Section 5.2). In Section 6 we test our controller on a realistic simulator of the *Fly-Crane*, showing that it outperforms standard designs under sampled communication. Final remarks are drawn in Section 7.

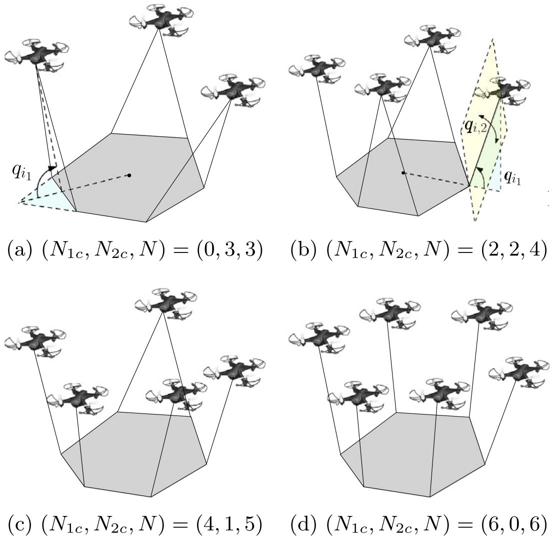
2. System model

2.1. Kinematics of multi-robot systems

In this section, we describe the kinematic model of a multi-robot system composed of N robots that exchange state information with a common *pivot*, labeled as V, to fulfill a task. As so, robots need not communicate among themselves, but they implicitly coordinate their motions by communicating with the pivot. The pivot may be an object to be manipulated, a robot to be escorted, or a vehicle in the space. The kinematic model is given by

$$\begin{bmatrix}\mathbf{p}_1 & \dots & \mathbf{p}_N\end{bmatrix}^\top = \mathbf{h}(\mathbf{q}) = \begin{bmatrix}\mathbf{h}^{(1)}(\mathbf{q}_1, \mathbf{q}_V) & \dots & \mathbf{h}^{(N)}(\mathbf{q}_N, \mathbf{q}_V)\end{bmatrix}^\top .$$

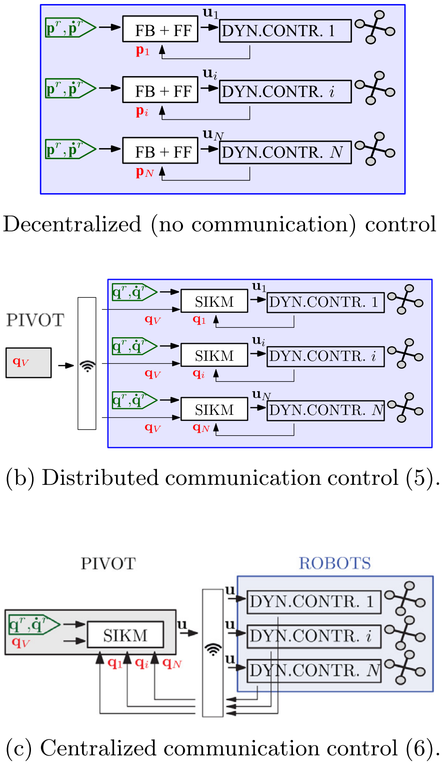
Function $\mathbf{h}(\cdot)$ maps the Lagrangian coordinates of the system (measured or desired) $\mathbf{q} = [\mathbf{q}_1^\top \dots \mathbf{q}_N^\top \; \mathbf{q}_V^\top]^\top \in \mathbb{R}^m$ to the vector $\mathbf{p} = [\mathbf{p}_1^\top \dots \mathbf{p}_N^\top]^\top \in \mathbb{R}^n$ gathering robot configurations. In particular, $\mathbf{p}_i \in \mathbb{R}^{n_i}$ is the position of the ith robot in space, $\mathbf{q}_i \in \mathbb{R}^{m_i}$ gathers angles and/or distances between the pivot and



(a) $(N_{1c}, N_{2c}, N) = (0, 3, 3)$ (b) $(N_{1c}, N_{2c}, N) = (2, 2, 4)$

(c) $(N_{1c}, N_{2c}, N) = (4, 1, 5)$ (d) $(N_{1c}, N_{2c}, N) = (6, 0, 6)$

Fig. 2. Square systems where a common object is manipulated by a group of UAVs. All the possible combinations of (N_{1c}, N_{2c}, N) introduced in Section 2.2 are represented.



(a) Decentralized (no communication) control (4).

(b) Distributed communication control (5).

(c) Centralized communication control (6).

Fig. 3. Controller architectures for trajectory tracking. The pivot is colored in gray, each robot (equipped with a dynamical controller which converts $\mathbf{u}$ to forces) in blue, sensor measurements in red, and the reference trajectory in green. The wireless symbol refers to sampled communication. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the ith robot, and $\mathbf{q}_V \in \mathbb{R}^{m_V}$ represents the pose (position and orientation) of the pivot. Note that $m = \sum_{i=1}^N m_i + m_V$ and $n = \sum_{i=1}^N n_i$. The differential kinematics of the system is

$$\dot{\mathbf{p}} = \mathbf{A_q}\, \dot{\mathbf{q}}, \tag{1}$$

where the Jacobian $\mathbf{A_q} = \frac{\partial \mathbf{h}(\mathbf{q})}{\partial \mathbf{q}} \in \mathbb{R}^{n \times m}$ has structure

$$\mathbf{A_q} = \left[\begin{array}{ccc|c} \mathbf{A}^{(1)}_{\mathbf{q}_1} & & \mathbf{0} & \mathbf{A}^{(1)}_{\mathbf{q}_V} \\ & \ddots & & \vdots \\ \mathbf{0} & & \mathbf{A}^{(N)}_{\mathbf{q}_N} & \mathbf{A}^{(N)}_{\mathbf{q}_V} \end{array}\right], \tag{2}$$

$\mathbf{A}^{(i)}_{\mathbf{q}_i} = \frac{\partial \mathbf{h}^{(i)}(\mathbf{q}_i, \mathbf{q}_V)}{\partial \mathbf{q}_i} \in \mathbb{R}^{n_i \times m_i}$ and $\mathbf{A}^{(i)}_{\mathbf{q}_V} = \frac{\partial \mathbf{h}^{(i)}(\mathbf{q}_i, \mathbf{q}_V)}{\partial \mathbf{q}_V} \in \mathbb{R}^{n_i \times m_V}$. We focus on the case $n = m$, corresponding to *square systems*. Let us make an example to justify this choice. Consider the multi-robot system in Fig. 2(a), where robots are linked to a platform though rigid cables, and assume that $\mathbf{A_q}$ is invertible and that desired robot velocities $\dot{\mathbf{p}}^d$ are assigned *a priori*. In this case, there always exists a vector $\dot{\mathbf{q}} = \mathbf{A}_{\mathbf{q}}^{-1}\dot{\mathbf{p}}^d$ in configuration space that allows the robots to achieve the desired velocity. The next paragraph expands this example more in detail. If $n \neq m$, the Jacobian is not square. In particular, the system is redundant if $n > m$. If $n < m$, there exist trajectories $\dot{\mathbf{q}}$ in the configuration space which are infeasible for any input $\dot{\mathbf{p}}^d$. Such cases require a dedicated analysis which goes beyond the scope of this paper. We refer to the preprint (Rossi, Tognon, Carli, Franchi and Schenato, 2021) for such an analysis, that, with some attention, allows to integrate the control strategy studied here into more general systems.

Furthermore, we consider a cascade control architecture where we design the proposed SIKM controller at the kinematic level. The latter provides the desired motor velocities which are tracked by low-level dynamics controllers (see Fig. 3), assuming that these have larger closed-loop bandwidth than the SIKM. This assumption is indeed validated by extensive simulations of the proposed architecture on a full dynamical model (including low-level dynamics as well as external disturbances) with a realistic simulator of the testbed *Fly-Crane*.

2.2. Example of square systems

In the literature, we can find several applications of square systems. We now provide an example to showcase relevance of such model. We consider the system depicted in Fig. 2(a) (*Fly-Crane* Manubens et al. (2013)), where three UAVs transport a load linked through rigid cables. The generalized coordinates are chosen as $\mathbf{q} = [q_1\ q_2\ q_3\ \mathbf{q}_V^\top]^\top$. The robot positions $\mathbf{p}_i \in \mathbb{R}^3$ are collected in the vector $\mathbf{p} = [\mathbf{p}_1^\top\ \mathbf{p}_2^\top\ \mathbf{p}_3^\top]^\top$. Thus, the velocity vectors $\dot{\mathbf{q}}, \dot{\mathbf{p}} \in \mathbb{R}^9$ have the same dimension and $\mathbf{A_q} \in \mathbb{R}^{9\times 9}$ is a square matrix. However, this is a particular case of a larger class: different square systems can be obtained by simply changing the number of robots transporting the platform or the number of cables linking each robot to it. Notice that if one cable was used instead of two, as shown in Fig. 2(b), then $\mathbf{q}_i = [q_{i_1}\ q_{i_2}]^\top \in \mathbb{R}^2$ because each cable can move in two directions (assuming that movements about the cable axis are not allowed). In this case, the system is no more guaranteed to be square. In particular, it holds $n = 3N$ and $m = 6 + 2N_{1c} + N_{2c}$ where $N_{1c} \geq 0$ is the number of robots linked to the load through one cable and $N_{2c} \geq 0$ indicates the number of robots linked through two cables. We have that $N = N_{1c} + N_{2c}$ and, for a square system $n = m$, it must be $3N = 6 + 2N_{1c} + N_{2c}$. From these relations it turns out that $N_{2c} \leq 3$, $N_{1c} \leq 6$ and $N \leq 6$. In particular, the possible configurations (N_{1c}, N_{2c}, N) are: (0,3,3) in Fig. 2(a); (2,2,4) in Fig. 2(b); (4,1,5) in Fig. 2(c); (6,0,6) in Fig. 2(d).

3. Control architectures

We consider a tracking problem where a multi-robot system is required to follow a sequence of desired configurations assigned *a priori*. We assume that such reference trajectory is generated offline by a high-level planner that takes into account goals such as obstacle avoidance, singular points, and energy minimization.

Also, we assume that robots are equipped with dynamical controllers sufficiently fast w.r.t. the dynamics of the system, such that (s.t.) their velocities are fully controllable,

$$\dot{\mathbf{p}}(t) = \mathbf{u}(t), \tag{3}$$

where $\mathbf{u} = [\mathbf{u}_1^\top \cdots \mathbf{u}_N^\top]^\top \in \mathbb{R}^n$. We now enumerate possible control architectures to achieve trajectory tracking.

Decentralized (no communication) control. A commonly adopted architecture is the following *fully decentralized* controller,

$$\mathbf{u}_i(t) = \kappa_i^{\text{dec}}\left(\mathbf{p}_i(t); \mathbf{p}_i^r(t), \dot{\mathbf{p}}_i^r(t)\right). \tag{4}$$

In this case, each robot needs only local position measurements $\mathbf{p}_i(t)$ (no configuration variables $\mathbf{q}(t)$ are needed) to implement feedback and follow its reference trajectory $(\mathbf{p}_i^r(t), \dot{\mathbf{p}}_i^r(t))$, namely, controller (4) is *communication-less* and the control design reduces to a distributed planning problem (Fig. 3(a)). A typical example is $\mathbf{u}_i(t) = -k(\mathbf{p}_i(t) - \mathbf{p}_i^r(t)) + \dot{\mathbf{p}}_i^r(t)$. However, lacking communication and coordination, strategy (4) is not robust against disturbances or robot failures/biases.

Distributed communication control. A distributed controller reads

$$\mathbf{u}_i(t) = \kappa_i^{\text{distr}}(\mathbf{q}_i(t), \mathbf{q}_V(hT); \mathbf{q}_i^r(t), \mathbf{q}_V^r(t), \dot{\mathbf{q}}_i^r(t), \dot{\mathbf{q}}_V^r(t)), \tag{5}$$

where $t \in [hT, (h+1)T)$, $h \in \mathbb{N}$. Specifically, $\mathbf{u}_i(t)$ depends on the robot's own configuration $\mathbf{q}_i(t)$, which can be measured at all times, and on the load configuration $\mathbf{q}_V(hT)$, which is transmitted via wireless and available at discrete time instants hT, T being the sampling time. Also, reference trajectories of robot $(\mathbf{q}_i^r(t), \dot{\mathbf{q}}_i^r(t))$ and load $(\mathbf{q}_V^r(t), \dot{\mathbf{q}}_V^r(t))$ are continuously available as they are computed offline and pre-stored on robots. Hence, controller (5) implements a *hybrid continuous-sampled* control that includes both continuous-time and discrete-time signals, inducing more challenging design (Fig. 3(b)).

Centralized control. Lastly, the centralized controller

$$\mathbf{u}_i(t) = \kappa_i^{\text{centr}}(\mathbf{q}(hT); \mathbf{q}^r(t), \dot{\mathbf{q}}^r(t)) \tag{6}$$

depends on the full system configuration vector $\mathbf{q}(hT)$. Fig. 3(c) shows a possible implementation where the controller, located on the pivot, receives all measurements $\mathbf{q}(t)$ and broadcasts the control input $\mathbf{u}(t)$ to the robots via wireless. In this case, while performance is theoretically maximized, all-to-all communication burden may cause issues through limitations of wireless communication in terms of bandwidth and reliability.

3.1. Proposed SIKM architectures

In this paper, we focus on the two communication-based approaches. While these are attractive by virtue of robustness properties, their design gets challenging when shifting from continuous-time (typically assumed in the literature) to hybrid continuous- and discrete-time dynamics. In particular, the presence of sampled measurements in (5)–(6) makes both controller design and stability analysis nontrivial.

Indeed, for continuous-time systems, the controller (Siciliano, Sciavicco, Villani, & Oriolo, 2009)

$$\mathbf{u}(t) = \underbrace{-k\mathbf{A}_{\mathbf{q}(t)}(\mathbf{q}(t) - \mathbf{q}^r)}_{:=\mathbf{u}_k(t)} + \underbrace{\mathbf{A}_{\mathbf{q}(t)}\dot{\mathbf{q}}^r(t)}_{:=\mathbf{u}_{\text{ff}}(t)} \tag{7}$$

drives the system configuration $\mathbf{q}(t)$ to the desired reference trajectory $\mathbf{q}^r(t)$ exponentially fast, whereby the tracking error $\mathbf{e}(t) \overset{\text{def}}{=} \mathbf{q}(t) - \mathbf{q}^r(t)$ obeys dynamics

$$\dot{\mathbf{e}}(t) = \dot{\mathbf{q}}(t) - \dot{\mathbf{q}}^r(t) = -k\big(\mathbf{q}(t) - \mathbf{q}^r(t)\big) = -k\mathbf{e}(t).$$

Moreover, in light of (2), the centralized controller (7) can be implemented with a distributed architecture,

$$\begin{aligned}\mathbf{u}_i(t) = &- k\mathbf{A}^{(i)}_{\mathbf{q}_i(t)}\left(\mathbf{q}_i(t) - \mathbf{q}_i^r(t)\right) + \mathbf{A}^{(i)}_{\mathbf{q}_i(t)}\dot{\mathbf{q}}_i^r(t) \\ &- k\mathbf{A}^{(i)}_{\mathbf{q}_V(t)}\left(\mathbf{q}_V(t) - \mathbf{q}_V^r(t)\right) + \mathbf{A}^{(i)}_{\mathbf{q}_V(t)}\dot{\mathbf{q}}_V^r(t).\end{aligned} \tag{8}$$

When measurements are sampled, suitably modifying (7) is nontrivial. In Rossi et al. (2020), the authors proposed the following feedback controller for point-stabilization,

$$\mathbf{u}(t) = \mathbf{u}_k(hT) = -k\mathbf{A}_{\mathbf{q}(hT)}(\mathbf{q}(hT) - \mathbf{q}^r), \tag{9}$$

for $t \in [hT, (h+1)T)$. A common strategy for trajectory tracking in robotic applications is sampling (9),

$$\mathbf{u}(t) = \mathbf{u}_k(hT) = -k\mathbf{A}_{\mathbf{q}(hT)}(\mathbf{q}(hT) - \mathbf{q}^r(hT)), \tag{10}$$

which however cannot guarantee perfect tracking under nominal conditions because it lacks feedforward correction (see technical report Rossi, Tognon, Ballotta et al. (2021, Appendix B) for details).

A naive attempt to improve (10) is sampling (7),

$$\begin{aligned}\mathbf{u}(hT+\tau) = &- k\mathbf{A}_{\mathbf{q}(hT)}(\mathbf{q}(hT) - \mathbf{q}^r(hT)) \\ &+ \mathbf{A}_{\mathbf{q}(hT)}\dot{\mathbf{q}}^r(hT+\tau), \quad \tau \in [0, T),\end{aligned} \tag{11}$$

which results in the following error dynamics,

$$\begin{aligned}&\dot{\mathbf{e}}(hT+\tau) = -k\mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{e}(hT) + \\ &+ \left(\mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)} - \mathbf{I}\right)\dot{\mathbf{q}}^r(hT+\tau).\end{aligned}$$

It is indeed easy to show that, under controller (11), reference $\mathbf{q}^r(t)$ is not an equilibrium trajectory, making also the latter attempt not suitable for trajectory tracking.

Finally, we modify controller (11) as follows,

$$\begin{aligned}\mathbf{u}(hT+\tau) = &- k\mathbf{A}_{\mathbf{q}(hT)}(\mathbf{q}(hT) - \mathbf{q}^r(hT)) \\ &+ \mathbf{A}_{\mathbf{q}^r(hT+\tau)}\dot{\mathbf{q}}^r(hT+\tau), \quad \tau \in [0, T).\end{aligned} \tag{12}$$

Differently from (11), the Jacobian of the feedforward term in (12) is computed at $\mathbf{q}^r(hT+\tau)$ instead of $\mathbf{q}(hT)$. Accordingly, the error dynamics become

$$\begin{aligned}\dot{\mathbf{e}}(hT+\tau) = &-k\mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{e}(hT) + \\ &+ \left(\mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{q}^r(hT+\tau)} - \mathbf{I}\right)\dot{\mathbf{q}}^r(hT+\tau) \\ &\overset{\text{def}}{=} \mathbf{f}(\mathbf{e}(hT+\tau), \mathbf{q}^r(hT+\tau), \dot{\mathbf{q}}^r(hT+\tau)),\end{aligned} \tag{13}$$

whereby the reference can be shown to be an equilibrium trajectory (see proofs of this and previous claim in Rossi, Tognon, Ballotta et al. (2021)). Also, by virtue of the assumed structure (2) of the Jacobian, controller (12) can be decoupled and is thus amenable of a distributed communication implementation of the form (5),

$$\begin{aligned}\mathbf{u}_i(hT+\tau) = &- k\mathbf{A}^{(i)}_{\mathbf{q}_i(hT)}\left(\mathbf{q}_i(hT) - \mathbf{q}_i^r(hT)\right) \\ &- k\mathbf{A}^{(i)}_{\mathbf{q}_V(hT)}\left(\mathbf{q}_V(hT) - \mathbf{q}_V^r(hT)\right) \\ &+ \mathbf{A}^{(i)}_{\mathbf{q}_i^r(hT+\tau)}\dot{\mathbf{q}}_i^r(hT+\tau) \\ &+ \mathbf{A}^{(i)}_{\mathbf{q}_V^r(hT+\tau)}\dot{\mathbf{q}}_V^r(hT+\tau).\end{aligned} \tag{14}$$

In the following, we analyze system dynamics under controller (12) to evaluate stability and performance. In view of (14), all results straightly carry over to a distributed architecture. Furthermore, in simulation we will test the centralized controller where the feedback gain k is adaptive: this cannot be decoupled, because computing k online requires the full system configuration.

4. Stability limitations of SIKM

In this section we prove a negative result: perfect nominal tracking cannot be ensured for some values of (k, T). In particular, the gain k needs to be sufficiently large *and* the sampling time T sufficiently small to avoid instability. To this aim, we state some preliminary assumptions.

Assumption 1. The following relations hold.

(i) The reference trajectory $\mathbf{q}^r(t) \in \mathcal{Q}$ is twice continuously differentiable and $\mathcal{Q}$ is a compact set. Moreover, velocities and accelerations are uniformly bounded, i.e., $\|\dot{\mathbf{q}}^r(t)\| \leq v_{max}$ and $\|\ddot{\mathbf{q}}^r(t)\| \leq a_{max}$.

(ii) $\mathbf{A}_{\mathbf{q}(0)}$ is twice continuously differentiable and invertible.

(iii) There exists $d > 0$ such that, for any configuration $\mathbf{q}$ at distance smaller than d from the reference trajectory, i.e., $\exists t : \|\mathbf{q} - \mathbf{q}^r(t)\| < d$, $\mathbf{A}_{\mathbf{q}(0)}$ is twice continuously differentiable and invertible.

Assumption 1-(i) are smoothness properties of the reference trajectory needed to derive error bounds. Assumption 1-(ii) is required to avoid that the initial condition is a singular point. Assumption 1-(iii) further guarantees that all configurations $\mathbf{q}^r(t)$ belonging to the reference trajectory are distant enough from singular points, ensuring robustness to, e.g., external disturbances, and can be accommodated through an offline high-level planner. We will later show that our proposed control strategies, under such assumptions, also guarantees that the actual trajectory is always bounded away from singular points.

The error flow $\mathbf{f}(\cdot, \cdot, \cdot)$ in (14) is discontinuous because the feedback term depends on $\mathbf{e}(hT)$ and resets at every sampling instant $t = hT$. Thus, existence of a global solution based on standard Lipschitz continuity cannot be invoked, in general. However, in view of Assumption 1, the flow is Lipschitz continuous for $\tau \in [0, T)$. Hence, if we can show that a solution $\mathbf{e}(hT + \tau)$ exists for any $\tau \in [0, T)$, and that the limit $\lim_{\tau \to T} \mathbf{e}(hT + \tau)$ exists finite starting from any $\mathbf{e}(hT)$ satisfying Assumption 1, then global existence is guaranteed by patching together those intervals. As so, we study the flow $\mathbf{f}(\mathbf{e}(t), \mathbf{q}^r(t), \dot{\mathbf{q}}^r(t); \mathbf{e}(hT))$ for $t = hT + \tau$, $\tau \in [0, T)$, where we make the dependence on the "initial condition" $\mathbf{e}(hT)$ explicit. To prove asymptotic stability, we will use the following notion of contraction.

Definition 1. Given fixed T, k, and under Assumption 1, the flow $\mathbf{f}(\mathbf{e}(t), \mathbf{q}^r(t), \dot{\mathbf{q}}^r(t); \mathbf{e}(hT))$ is ρ-monotonically contractive if, for any $\|\mathbf{e}(hT)\| < d$, it holds

(1) $\|e(hT + \tau)\| \leq \|e(hT)\|, \quad \tau \in [0, T)$,

(2) $\lim_{\tau \to T} \|e(hT + \tau)\| \leq \rho \|e(hT)\|, \quad \rho \in [0, 1)$.

Given Definition 1, the following lemma easily follows.

Lemma 2. *If the error flow defined in (13) is ρ-monotonically contractive and under Assumption 1, then*

$$\|\mathbf{q}(t) - \mathbf{q}^r(t)\| \leq \rho^{\frac{t}{T}-1} \|\mathbf{q}(0) - \mathbf{q}^r(0)\|, \quad t \geq 0.$$

Lemma 2 ensures exponential convergence to the reference trajectory. The property of ρ-monotonically contractiveness might appear rather strong since it must hold for any segment of the trajectory of length T, however, it is necessary to guarantee that the system avoids singularity configurations at all times. The next result encodes necessary conditions for stability.

Proposition 3. *Under Assumption 1, there exists $k_{min} > 0$ such that, if one of the following conditions is satisfied,*

(1) $k < k_{min}$,

(2) $k > \frac{2}{T}$,

the error flow (13) is not ρ-monotonically contractive.

Intuitively, if the gain k is too small, there exist reference trajectories such that the feedforward term "pushes" the system too much without being suitably balanced by the feedback term. Conversely, if k is too large, the feedback causes the trajectory to overshoot, possibly amplifying the error of the initial condition.

Proof. *(1)* We will prove the first condition showing that the error norm initially increases for some initial conditions $\mathbf{e}(hT)$ and choice of reference trajectory $\mathbf{q}^r(t)$. Consider the following Lyapunov function

$$V(\tau) = \frac{1}{2} \|\mathbf{e}(hT + \tau)\|^2, \tag{15}$$

and its time derivative at $\tau = 0$ (cf. (13)):

$$\begin{aligned}\dot{V}(0) &= \mathbf{e}(hT)^\top \dot{\mathbf{e}}(hT) \\ &= -k\|\mathbf{e}(hT)\|^2 + \\ &\quad \underbrace{\mathbf{e}^\top(hT)(\mathbf{A}^{-1}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)} \mathbf{A}_{\mathbf{q}^r(hT)} - \mathbf{I})\dot{\mathbf{q}}^r(hT)}_{\overset{\text{def}}{=} \phi(\mathbf{e}(hT), \mathbf{q}^r(hT), \dot{\mathbf{q}}^r(hT))}.\end{aligned}$$

Let us define

$$\begin{aligned} c_{max} \overset{\text{def}}{=} \quad & \max \ c \\ \text{s.t.} \quad & \|\mathbf{e}(hT)\| \leq d, \ \mathbf{q}^r(hT) \in \mathcal{Q}, \\ & \phi(\mathbf{e}(hT), \mathbf{q}^r(hT), \dot{\mathbf{q}}^r(hT)) \geq c v_{max} \|\mathbf{e}(hT)\|^2. \end{aligned}$$

The parameter c_{max} is surely strictly positive because (i) all arguments of $\phi(\cdot)$ are defined on a compact set (see also Assumption 1-(i)) and (ii) $\phi(\cdot)$ is continuously differentiable. The only case for which $c_{max} = 0$ is when $\mathbf{A}_\mathbf{q}$ is a constant matrix for all $\mathbf{q}$. This scenario is not admissible since it would imply $\dot{\mathbf{p}} = \mathbf{A}\dot{\mathbf{q}}$, which is not the case for the problem at hand. This implies that there exist tuples $(\mathbf{e}(hT), \mathbf{q}^r(hT), \dot{\mathbf{q}}^r(hT))$ such that

$$\dot{V}(0) \geq -(k - c_{max} v_{max}) \|\mathbf{e}(hT)\|^2$$

Let $k_{min} \overset{\text{def}}{=} c_{max} v_{max}$, then $\dot{V}(0) > 0$ for any $k < k_{min}$, hence there exists $0 < \bar{\tau} < T$ s.t. $\|\mathbf{e}(hT + \tau)\| > \|\mathbf{e}(hT)\| \, \forall \tau \in (0, \bar{\tau})$ and (13) it is not ρ-monotonically contractive.

(2) In order to prove the second condition we first choose $\mathbf{q}^r(t) \equiv \mathbf{q}^r, \ t \geq 0$, which satisfies Assumption 1. As so, the error dynamics reduce to

$$\dot{\mathbf{e}}(hT + \tau) = -k \underbrace{\mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r} \mathbf{A}_{\mathbf{e}(hT)+\mathbf{q}^r} \mathbf{e}(hT)}_{\overset{\text{def}}{=} \mathbf{g}(\mathbf{e}(hT+\tau))}.$$

The error trajectory can be written in the form

$$\mathbf{e}(hT + \tau) = \mathbf{e}(hT) + k \int_{hT}^{hT+\tau} \mathbf{g}(\mathbf{e}(hT + \tau'))d\tau'.$$

By using Taylor's theorem for multivariate functions with integral form of the remainder, it becomes

$$\begin{aligned} &\mathbf{e}(hT + \tau) = \mathbf{e}(hT) + k\tau \, \mathbf{g}(\mathbf{e}(hT)) + \\ &+ k^2\tau^2 \int_0^1 (1 - \varepsilon) \frac{\partial \mathbf{g}(\mathbf{e}(hT + \epsilon\tau))}{\partial \mathbf{e}} \, \mathbf{g}(\mathbf{e}(hT + \varepsilon\tau))d\varepsilon \\ &= (1 - k\tau)\mathbf{e}(hT) + k^2\tau^2 \mathbf{r}(\mathbf{e}(hT + \tau); \mathbf{e}(hT)), \end{aligned} \tag{16}$$

where $\mathbf{r}(\cdot)$ is the second-order reminder where we made explicit the dependence on $\mathbf{e}(hT)$. Under Assumption 1, the functions $\mathbf{g}$ and $\frac{\partial \mathbf{g}}{\partial \mathbf{e}}$ are continuously differentiable. Moreover their arguments are defined in a compact set and have the additional properties that $\mathbf{g}(\mathbf{e}(hT + \tau)) = 0$, $\frac{\partial \mathbf{g}(hT+\tau)}{\partial \mathbf{e}} = \mathbf{0}$, $\forall \tau \in [0, T]$ if $\mathbf{e}(hT) = \mathbf{0}$.

Therefore, by applying Lemma A.3 in Appendix, there must exist $\delta > 0$ such that

$$\|\mathbf{r}(\mathbf{e}(hT+\tau);\mathbf{e}(hT))\| \le \delta\|\mathbf{e}(hT)\|^2, \quad \forall\tau\in[0,T), \forall\mathbf{e}(hT).$$

We now use the reverse triangle inequality and the previous inequality to get:

$$\begin{aligned}&\|\mathbf{e}(hT+\tau)\| \ge \\ &\ge \big| \|(1-k\tau)\mathbf{e}(hT)\| - k^2\tau^2\|\mathbf{r}(\mathbf{e}(hT+\tau);\mathbf{e}(hT))\| \big| \\ &\ge \max\{0\,,\,|1-k\tau| - k^2\tau^2\delta\|\mathbf{e}(hT)\|\}\|\mathbf{e}(hT)\|.\end{aligned}$$

If $kT > 2$, there exist $\epsilon > 0$ and $\bar\tau \in (0,T)$ s.t. $k\bar\tau = 2+\epsilon$. Then, we can choose $\mathbf{e}(hT)$ s.t. $\|\mathbf{e}(hT)\| = \frac{\epsilon}{2k\bar\tau^2\delta}$, hence

$$\|\mathbf{e}(hT+\bar\tau)\| \ge \left(1+\epsilon-\frac{\epsilon}{2}\right)\|\mathbf{e}(hT)\| \ge \|\mathbf{e}(hT)\|. \quad \square$$

Remark 4 (*Stability Limitations on T and k*)**.** Proposition 3 states that there are choices of T and k for which the error flow (13) is not ρ-monotonically contractive, implying that the robots may not track the reference trajectory. In particular, there exists a feedback gain k_{min} below which the system cannot be ρ-monotonically contractive. Also, there exists a sampling time $T_{max} = \frac{2}{k_{min}}$ above which the same instability consideration applies, as graphically shown in Fig. 1. This is in stark contrast with the result for position stabilization in Rossi et al. (2020), where (doubly) exponential stability could be achieved without sharp limitation on sampling time T or control gain k. This fundamental difference is mainly due to the novel feedforward term considered in this work.

In the next section, we find sufficient conditions for stability in the form of a bounded area in the (k,T)-space that guarantees ρ-monotonic contractiveness of (13), and hence perfect asymptotic tracking under nominal conditions.

5. Stability and convergence rate for SIKM

In this section, we compute explicit bounds for parameters k and T that ensure decrease of the tracking error. In particular, we show that there exist a continuous function $\tau_s(k)$ and a scalar $k_{CR} > 0$ such that, for any pair $(k,T) \in \mathcal{U}$, with $\mathcal{U} := \{k_{CR} < k < +\infty, 0 < \tau < \tau_s(k)\}$, flow (13) is ρ-monotonically contractive (pictorially portrayed by the green area in Fig. 1).

The next Proposition bounds the convergence rate through a parametric function $z(\cdot,\cdot)$ of k and T, with coefficients depending on system dynamics. A numerical procedure to estimate such coefficients is described in Section 5.1.

Proposition 5. *For system (13), it is possible to upper bound the decrease rate of the error norm $\|\mathbf{e}(\cdot)\|$,*

$$\|\mathbf{e}(hT+\tau)\| \le z\big(k,\tau;\mu,\alpha,\gamma_1,\gamma_2\big)\cdot\|\mathbf{e}(hT)\|, \tag{17}$$

for $h\in\mathbb{N}$, $\tau\in(0,\tau_s(k))$, and

$$z\big(k,\tau;\mu,\alpha,\gamma_1,\gamma_2\big) = |1-k\tau| + \tau\alpha + \tau^2(k^2\mu + k\gamma_1 + \gamma_2),$$

where constants $\alpha,\mu,\gamma_1,\gamma_2$ depend on the system dynamics and are defined in (23) in the proof.

Proof. The flow defined by (13) can be written as

$$\mathbf{e}(hT+\tau) = \mathbf{e}(hT) + \int_{hT}^{hT+\tau} \mathbf{f}(\mathbf{e}(\cdot),\mathbf{q}^r(\cdot),\dot{\mathbf{q}}^r(\cdot))\Big|_{hT+\tau'} d\tau'.$$

By using Taylor's theorem for multivariate functions with integral form of the remainder, we get

$$\begin{aligned}\mathbf{e}(hT+\tau) &= \mathbf{e}(hT) + \tau\,\mathbf{f}(\mathbf{e}(hT),\mathbf{q}^r(hT),\dot{\mathbf{q}}^r(hT)) + \\ &+ \tau^2\int_0^1 (1-\varepsilon)\cdot\Bigg[\frac{\partial\mathbf{f}(\mathbf{e}(\cdot),\mathbf{q}^r(\cdot),\dot{\mathbf{q}}^r(\cdot))}{\partial\mathbf{e}(\cdot)}\,\mathbf{f}(\mathbf{e}(\cdot),\mathbf{q}^r(\cdot),\dot{\mathbf{q}}^r(\cdot)) + \\ &+ \frac{\partial\mathbf{f}(\mathbf{e}(\cdot),\mathbf{q}^r(\cdot),\dot{\mathbf{q}}^r(\cdot))}{\partial\mathbf{q}^r(\cdot)}\,\dot{\mathbf{q}}^r(\cdot) + \\ &+ \frac{\partial\mathbf{f}(\mathbf{e}(\cdot),\mathbf{q}^r(\cdot),\dot{\mathbf{q}}^r(\cdot))}{\partial\dot{\mathbf{q}}^r(\cdot)}\,\ddot{\mathbf{q}}^r(\cdot)\Bigg]_{hT+\varepsilon\tau} d\varepsilon.\end{aligned} \tag{18}$$

The above expression allows to find an upper bound of $\|\mathbf{e}(hT+\tau)\|$, $\tau\in(0,\tau_s(k))$, by acting on the single terms of (18). Hence, we can compute a more precise estimate of the convergence rate, i.e., how quickly $\|\mathbf{e}(hT+\tau)\|$, with $\tau\in(0,\tau_s(k))$, decreases w.r.t. $\|\mathbf{e}(hT)\|$. Moreover, the following computations will be useful to find values (k,T) s.t. the system is stable and the tracking error converges to zero. First, we observe that

$$\begin{aligned}&\mathbf{f}(\mathbf{e}(hT),\mathbf{q}^r(hT),\dot{\mathbf{q}}^r(hT)) = -k\mathbf{e}(hT) + \\ &+\big(\mathbf{A}^{-1}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{A}_{\mathbf{q}^r(hT)} - \mathbf{I}\big)\,\dot{\mathbf{q}}^r(hT).\end{aligned} \tag{19}$$

By using Lemma A.2 on the second addend, it holds

$$\|\big(\mathbf{A}^{-1}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{A}_{\mathbf{q}^r(hT)} - \mathbf{I}\big)\,\dot{\mathbf{q}}^r(hT)\| \le \alpha\|\mathbf{e}(hT)\|,$$

where $\alpha \overset{\text{def}}{=} av_{max}$ for some $a > 0$ that depends on system dynamics. Moreover, from (13),

$$\begin{aligned}&\|\mathbf{f}(\mathbf{e}(hT+\tau),\mathbf{q}^r(hT+\tau),\dot{\mathbf{q}}^r(hT+\tau))\| \le \\ &\le \|-k\mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{e}(hT)\| + \\ &+ \|\big(\mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{q}^r(hT+\tau)} - \mathbf{I}\big)\dot{\mathbf{q}}^r(hT+\tau)\|.\end{aligned} \tag{20}$$

Both addends in (20) are continuously differentiable functions on a compact set. In addition, they are equal to zero if $\mathbf{e}(hT) = \mathbf{0}$ and $\mathbf{e}(hT+\tau) = \mathbf{0}$, respectively. Then, we can apply Lemmas A.1–A.2 to get $\|k\mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{e}(hT)\| \le kb\|\mathbf{e}(hT)\|$ and $\|\big(\mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{q}^r(hT+\tau)} - \mathbf{I}\big)\dot{\mathbf{q}}^r(hT+\tau)\| \le d\|\mathbf{e}(hT)\|v_{max}$, where we used Definition 1 and constants $b,d > 0$ depend on system dynamics. Thus, we rewrite (20) as

$$\|\mathbf{f}(\mathbf{e}(\cdot),\mathbf{q}^r(\cdot),\dot{\mathbf{q}}^r(\cdot))\big|_{hT+\tau}\| \le (kb + cv_{max})\|\mathbf{e}(hT)\|. \tag{21}$$

Also, being $\mathbf{f}(\mathbf{e}(\cdot),\mathbf{q}^r(\cdot),\dot{\mathbf{q}}^r(\cdot))$ differentiable w.r.t. all its variables on the compact set $\mathcal{B}_r(\mathbf{e}(\cdot))$, we can derive the following bounds based on (19),

$$\begin{aligned}&\left\|\frac{\partial\mathbf{f}(\mathbf{e}(\cdot),\mathbf{q}^r(\cdot),\dot{\mathbf{q}}^r(\cdot))}{\partial\mathbf{e}}\bigg|_{hT+\tau}\right\| \le (k + gv_{max})\|\mathbf{e}(hT)\| \\ &\left\|\frac{\partial\mathbf{f}(\mathbf{e}(\cdot),\mathbf{q}^r(\cdot),\dot{\mathbf{q}}^r(\cdot))}{\partial\mathbf{q}^r}\bigg|_{hT+\tau}\right\| \le mv_{max}\|\mathbf{e}(hT)\| \\ &\left\|\frac{\partial\mathbf{f}(\mathbf{e}(\cdot),\mathbf{q}^r(\cdot),\dot{\mathbf{q}}^r(\cdot))}{\partial\dot{\mathbf{q}}^r}\bigg|_{hT+\tau}\right\| \le kn\|\mathbf{e}(hT)\|,\end{aligned} \tag{22}$$

where again $g,m,n > 0$ depend on system dynamics. In light of (19), (21), (22), we then derive the bound in (17),

$$\|\mathbf{e}(hT+\tau)\| \le \big(|1-k\tau| + \alpha\tau + \tau^2(k^2\mu + k\gamma_1 + \gamma_2)\big)\|\mathbf{e}(hT)\|,$$

where

$$\begin{aligned}\alpha &\overset{\text{def}}{=} av_{max} \\ \mu &\overset{\text{def}}{=} b\|\mathbf{e}(hT)\| \\ \gamma_1 &\overset{\text{def}}{=} (d+gb)v_{max}\|\mathbf{e}(hT)\| + na_{max} \\ \gamma_2 &\overset{\text{def}}{=} gdv_{max}\|\mathbf{e}(hT)\| + mv_{max}. \quad \square\end{aligned} \tag{23}$$

Remark 6 (*Constant Reference*)**.** Consider (17) and suppose a constant reference is assigned, i.e. $\frac{d^{(n)}\mathbf{q}^r(t)}{dt^{(n)}} \equiv \mathbf{0}, n \ge 1$. Then, $\alpha = \gamma_1 = \gamma_2 = 0$ and $z(k,\tau;\mu)$ coincides with the function $g(\tau;\mu)\big|_{\tau=k\tau}$ found in Rossi et al. (2020).

5.1. Estimating parameters $\mu, \alpha, \gamma_1, \gamma_2$

We would like to find an estimate of the function $z(k, \tau; \mu, \alpha, \gamma_1, \gamma_2)$ that bounds the convergence rate of the tracking error, and choose the values (k, T) which yield the fastest convergence. To this aim, we provide a numerical procedure to estimate $\theta \overset{\text{def}}{=} (\mu, \alpha, \gamma_1, \gamma_2)$. Proposition 5 implies that the following set is nonempty,

$$\Theta \overset{\text{def}}{=} \{\theta > 0 \,|\, \|\mathbf{e}(hT+\tau)\| \leq z\big(k, \tau; \theta\big)\|\mathbf{e}(hT)\|,\\ \forall \mathbf{q}^r(\cdot), \mathbf{q}(0)\},$$

where the inequality is componentwise. Ideally, we would like to pick the smallest possible values for the parameters in θ in order to get the largest set of pairs (k, T) that induce stability. One possibility is choosing

$$\vartheta \overset{\text{def}}{=} \operatorname*{argmin}_{\theta \in \Theta} \|\theta\|.$$

Such ϑ surely exists because $\mathbf{q}^r(\cdot)$ and $\mathbf{q}(0)$ belong to a compact set. However, it cannot be computed numerically because one would need to check all pairs (k, T) and points $\mathbf{q}^r(t)$ and $\mathbf{q}(0)$. We propose a strategy to estimate ϑ by sampling $\mathbf{q}^r(t)$ and $\mathbf{q}(0)$ from their domains for different values of (k, T) and run simulations to get a set of samples $\{(k_i, T_i, \mathbf{e}^i_{h+1}, \mathbf{e}^i_h)\}_{i=1}^S$, S being the number of samples. Let us define the following quantities,

$$y_i \overset{\text{def}}{=} \|\mathbf{e}^i_{h+1}\|,$$
$$s_i^\top \overset{\text{def}}{=} [T_i^2 \ k_i^2 \ T_i \ T_i^2 k_i \ T_i^2]\|\mathbf{e}^i_h\|$$
$$b_i \overset{\text{def}}{=} |1 - k_i T_i| \ \|\mathbf{e}^i_h\|,$$

then the inequality $\|\mathbf{e}^i_{h+1}\| \leq z\big(k_i, T_i; \theta\big)\|\mathbf{e}^i_h\|$ can be written as $y_i \leq s_i^\top \theta + b_i$. Based on such sampled trajectories, we solve the following quadratic programming:

$$\begin{aligned} \widehat{\theta}_S \overset{\text{def}}{=} \operatorname*{argmin}_{\theta} \quad & \|\theta\|^2 \\ \text{s. t.} \quad & \theta \geq 0, \\ & y_i \leq s_i^\top \theta + b_i, \quad i = 1, \ldots, S. \end{aligned} \tag{24}$$

Unfortunately, it is possible that $\widehat{\theta}_S \notin \Theta$ since we are checking the inequality $\|\mathbf{e}(hT+\tau)\| \leq z\big(k, \tau; \theta\big)\|\mathbf{e}(hT)\|$ over a finite number of points. We expect that $\lim_{S\to\infty} \widehat{\theta}_S = \vartheta$ if the sampling procedure covers domains of $\mathbf{q}^r(\cdot)$ and $\mathbf{q}(0)$ widely enough. However, formally proving this claim is nontrivial. There might be alternative numerical strategies to compute better estimates of ϑ, or other parameter choices in the set Θ. Such comparison goes beyond the scope of this work, however, we will show effectiveness of our proposed strategy through simulations.

5.2. Analysis of the function $\big(k, \tau; \mu, \alpha, \gamma_1, \gamma_2\big)$

In this section, we analyze the function $z\,(k, \tau)$ in order to compute the bound $\tau_s(k)$ of the stability region (green area in Fig. 1), defined as[1]

$$\mathcal{U} \overset{\text{def}}{=} \{(k, \tau) : z(k, \tau) < 1\}, \tag{25}$$

and the convergence rate for each point in such region. We also compute the optimal controller gain $k_o(T)$ for fixed sampling time T and the optimal sampling time $T_s(k)$ for fixed gain k.[2]

In the interest of space, proofs of the following propositions are deferred to the technical report (Rossi, Tognon, Ballotta et al., 2021, Appendix C).

Proposition 7. *The guaranteed stability set of $z(k, \tau)$ is*

$$\mathcal{U} = \{k_{CR} < k < +\infty, 0 < \tau < \tau_s(k)\},$$

where $k_{CR} = \alpha$ and

$$\tau_s(k) = \begin{cases} \tau_{s_1}(k) & & \text{if } \mu > 1 \\ \tau_{s_1}(k) & \text{for } \alpha < k < \bar{k} & \text{if } \mu < 1\,, \\ \tau_{s_2}(k) & \text{for } k > \bar{k} & \text{if } \mu < 1 \end{cases}$$

where $\tau_{s_1} \overset{\text{def}}{=} \frac{k-\alpha}{k^2\mu+k\gamma_1+\gamma_2}$, $\bar{k} \overset{\text{def}}{=} \frac{\alpha+\gamma_1+\sqrt{(\alpha+\gamma_1)^2+4\gamma_2(1-\mu)}}{2(1-\mu)}$, $\tau_{s_2} \overset{\text{def}}{=} \frac{-(\alpha+k)+\sqrt{(\alpha+k)^2+8(k^2\mu+k\gamma_1+\gamma_2)}}{2(k^2\mu+k\gamma_1+\gamma_2)}$, and all the constant parameters were defined in Proposition 5.

With fixed k, we can define the optimal sampling time and convergence rate as functions of the feedback gain,

$$\tau_o(k) \overset{\text{def}}{=} \operatorname*{argmin}_{\tau} z(k, \tau) \tag{26}$$

$$\rho_o(k) \overset{\text{def}}{=} z(k, \tau_o(k)) = \min_{\tau} z(k, \tau), \tag{27}$$

where τ_o is the time when $\|\mathbf{e}(hT+\tau)\|$, $0 \leq \tau \leq T$, is closest to the origin, hence it corresponds to the fastest convergence rate of the error ρ_o.

Proposition 8. *With fixed k, the optimal sampling time $\tau_o(k)$ is*

$$\tau_o(k) = \begin{cases} \tau_{o_1}(k) & & \text{if } \mu > \frac{1}{2} \\ \tau_{o_1}(k) & \text{for } \alpha < k < \bar{\bar{k}} & \text{if } \mu < \frac{1}{2}\,, \\ \frac{1}{k} & \text{for } k > \bar{\bar{k}} & \text{if } \mu < \frac{1}{2} \end{cases}$$

$\tau_{o_1}(k) \overset{\text{def}}{=} \frac{k-\alpha}{2(k^2\mu+k\gamma_1+\gamma_2)}$, $\bar{\bar{k}} \overset{\text{def}}{=} \frac{\alpha+2\gamma_1+\sqrt{(\alpha+2\gamma_1)^2+8\gamma_2(1-2\mu)}}{2(1-2\mu)}$. The corresponding convergence rate $\rho_o(k)$ is

$$\rho_o(k) = \begin{cases} \rho_{k_1}(k) & & \text{if } \mu > \frac{1}{2} \\ \rho_{k_1}(k) & \text{for } \alpha < k < \bar{\bar{k}} & \text{if } \mu < \frac{1}{2}\,, \\ \rho_{k_2}(k) & \text{for } k > \bar{\bar{k}} & \text{if } \mu < \frac{1}{2} \end{cases}$$

where $\rho_{k_1}(k) \overset{\text{def}}{=} 1 - \frac{(\alpha-k)^2}{4(k^2\mu+k\gamma_1+\gamma_2)}$ and $\rho_{k_2}(k) \overset{\text{def}}{=} \mu + \frac{1}{k}(\alpha+\gamma_1) + \frac{\gamma_2}{k^2}$.

The same quantities can be found as functions of τ,

$$k_o(\tau) \overset{\text{def}}{=} \operatorname*{argmin}_{k} z(k, \tau) \tag{28}$$

$$\rho_o(\tau) \overset{\text{def}}{=} z(k_o(\tau), \tau) = \min_{k} z(k, \tau). \tag{29}$$

Proposition 9. *With fixed τ, the optimal gain $k_o(\tau)$ is*

$$k_o(\tau) = \begin{cases} \frac{1-\tau\gamma_1}{2\tau\mu} & \text{for } 0 < \tau < \tau_{M_\tau} & \text{if } \mu > \frac{1}{2} \\ \frac{1-\tau\gamma_1}{2\tau\mu} & \text{for } \tau_{m_k} < \tau < \tau_{M_\tau} & \text{if } \mu < \frac{1}{2}\,, \\ \frac{1}{\tau} & \text{for } 0 < \tau < \tau_{m_k} & \text{if } \mu < \frac{1}{2} \end{cases}$$

where $\tau_{m_k} \overset{\text{def}}{=} \frac{1-2\mu}{\gamma_1}$, $\tau_{M_\tau} \overset{\text{def}}{=} \min\{\frac{1}{\gamma_1}, \tau^-_{v_2}\}$, and $\tau^-_{v_2} = \frac{-(\gamma_1+\alpha\mu)+\sqrt{(\gamma_1+\alpha\mu)^2+(-\gamma_1^2+4\gamma_2\mu)}}{-\gamma_1^2+4\gamma_2\mu}$. The corresponding convergence rate

[1] In the following, we use the shorthand notation $z\big(k, \tau\big) \overset{\text{def}}{=} z\big(k, \tau; \mu, \alpha, \gamma_1, \gamma_2\big)$ for the sake of readability.

[2] Note that, while $T_s(k)$ is the *optimal* sampling time for given k, $\tau_s(k)$ is the stability *bound*, therefore $T_s(k) < \tau_s(k)$.

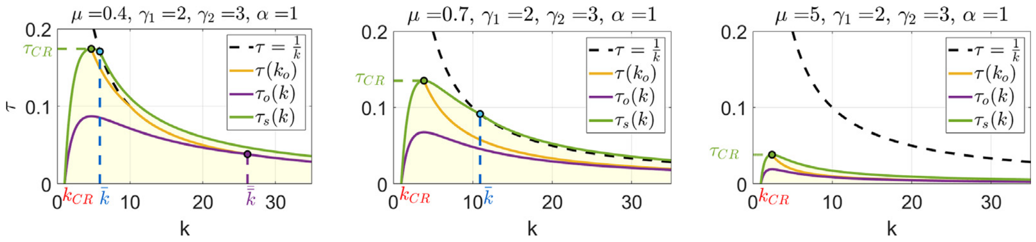


Fig. 4. Representation of the quantities $\tau_s(k)$, $\tau_o(k)$, and $\tau(k_o)$ defined in Propositions 7–9.

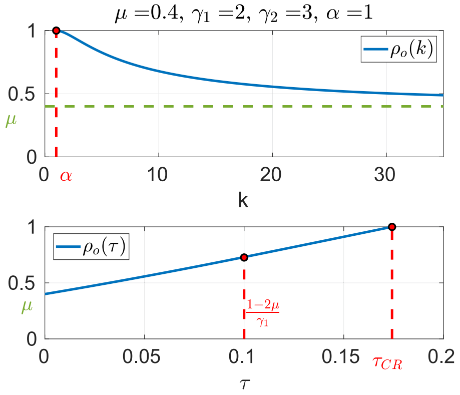


Fig. 5. Estimated convergence rate ρ_0 as a function of the gain k and of the sampling time τ.

$\rho_o(\tau)$ is

$$\rho_o(\tau) = \begin{cases} \rho_{\tau_1}(\tau) & \text{for } 0 < \tau < \tau_{M_\tau} \quad \text{if } \mu > \frac{1}{2} \\ \rho_{\tau_1}(\tau) & \text{for } \tau_{m_k} < \tau < \tau_{M_\tau} \quad \text{if } \mu < \frac{1}{2} \\ \rho_{\tau_2}(\tau) & \text{for } 0 < \tau < \tau_{m_k} \quad \text{if } \mu < \frac{1}{2} \end{cases},$$

where $\rho_{\tau_1}(\tau) \overset{def}{=} \frac{(-\gamma_1^2+4\gamma_2\mu)\tau^2+2(\gamma_1+2\alpha\mu)\tau+4\mu-1}{4\mu}$ *and* $\rho_{\tau_2}(\tau) \overset{def}{=} \gamma_2\tau^2 + (\alpha+\gamma_1)\tau + \mu$.

Fig. 4 depicts the quantities defined above, where $\tau(k_o)$ is the inverse function of $k_o(\tau)$ and represents the sampling time for which k_o is the optimal gain. We chose three cases corresponding to $\mu < 1/2$ (left box), $1/2 < \mu < 1$ (middle box), and $\mu > 1$ (right box), in order to span all cases defined in the propositions. Fig. 5 represents the convergence rate ρ_o as a function of k (top box) and τ (bottom box). The convergence rate is always smaller than one, i.e. $\|\mathbf{e}(hT+\tau)\| \leq \|\mathbf{e}(hT)\|,\ \tau \in [0, T]$.

Remark 10 (*Sampling Time and Stabilizability*). Proposition 3 shows that there exists a maximum sampling time $T_{\max}$ s.t. no feedback gain k can ensure stability if $T > T_{\max}$. Conversely, Proposition 7 implies that there exists a threshold τ_{CR}, which can be analytically found by setting $\frac{d\tau_{s_1}(k)}{dk} = 0$, s.t. a stabilizing gain always exists if $T < \tau_{CR}$ (see Fig. 1). Such values may help to evaluate the communication hardware to be used.

6. Simulation results

In this section, we implement and compare four techniques for trajectory tracking, which are summarized in Table 1. The first one, named SIKM-D, is the distributed controller (14) proposed in this work, where the gain $k = k_{\text{off}}$ is computed offline following the procedure in Section 5. The last three, referred to as PS, FF, and SIKM-C, respectively, are inspired by the online control strategy proposed in Rossi et al. (2020). In particular, they all adopt a centralized communication architecture and differ only in the feedforward term design. Specifically, at the hth step, k_{on} is the solution to the following optimization problem with initial condition $\mathbf{e}_h = \mathbf{e}(hT)$,

Table 1
Description of the four strategies used in simulation.

Control strategy	$\mathbf{u}_k(hT)$	$\mathbf{u}_{\text{ff}}(hT+\tau)$
SIKM-D (Eq. (14))	$-k_{\text{off}}\mathbf{A}_{\mathbf{q}_h}(\mathbf{q}_h - \mathbf{q}_h^r)$	$\mathbf{A}_{\mathbf{q}_{hT+\tau}^r}\dot{\mathbf{q}}_{hT+\tau}^r$
PS (Eq. (10))	$-k_{\text{on}}\mathbf{A}_{\mathbf{q}_h}(\mathbf{q}_h - \mathbf{q}_h^r)$	$\mathbf{0}$
FF (Eq. (11))	$-k_{\text{on}}\mathbf{A}_{\mathbf{q}_h}(\mathbf{q}_h - \mathbf{q}_h^r)$	$\mathbf{A}_{\mathbf{q}_{hT}}\dot{\mathbf{q}}_{hT+\tau}^r$
SIKM-C (Eq. (12))	$-k_{\text{on}}\mathbf{A}_{\mathbf{q}_h}(\mathbf{q}_h - \mathbf{q}_h^r)$	$\mathbf{A}_{\mathbf{q}_{hT+\tau}^r}\dot{\mathbf{q}}_{hT+\tau}^r$

$$\begin{aligned} k_{\text{on}}(hT) = \ & \underset{k}{\text{argmin}}\ \|\mathbf{e}(hT+T)\| \\ \text{s. t.} \quad & \mathbf{u}(\tau) = \mathbf{u}_k(hT) + \mathbf{u}_{\text{ff}}(hT+\tau), \end{aligned} \tag{30}$$

where $\tau \in [hT, hT+T)$. The PS technique corresponds to a simple point-stabilization with no feedforward correction, therefore it is expected to always lag behind the desired trajectory. The FF technique implements the naive controller (11) obtained by discretizing the standard continuous-time feedforward term which evaluates the Jacobian at $\mathbf{q}_{hT}$, that was shown insufficient for asymptotic tracking. Even this strategy is expected to perform worse than SIKM-D despite the potential benefit coming from the online design of the feedback gain. Finally, SIKM-C has the same control structure as SIKM-D, but it re-computes the optimal feedback gain k_{on} at each sampling time. This strategy should provide the best possible performance as opposed to SIKM-D, which computes its gain offline solving a worst-case minmax problem as shown in Section 5. However, SIKM-C requires solving an optimization problem and receiving the state of all robots at each sampling time.

Remark 11 (*Distributed vs. Centralized Control*). In contrast with the centralized scheme, a distributed communication control architecture avoids limitations due to all-to-all wireless communication, such as larger latency or packet loss, and is more robust, cheaper and easier to maintain since it does not depend on the number of robots. Specifically, the distributed communication controller requires only one broadcast communication packet from the load/pivot to the robots, while the centralized communication requires in addition to the broadcast packet also the (possibly synchronized) transmission of N packets from robots to load.

We tested the four techniques on the *Fly-Crane* system (Sanalitro et al., 2020), whose simulated environment is depicted in Fig. 6. The dynamical model of the system has been developed in a physics-based simulator, simulating cables dynamics as well.

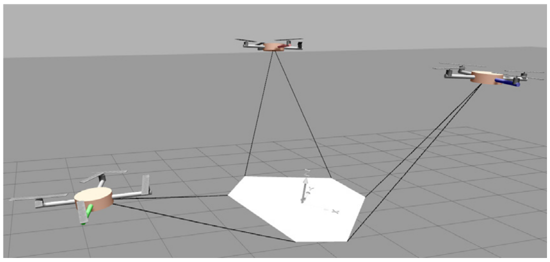


Fig. 6. The *Fly-Crane* system in the *Gazebo* simulator.

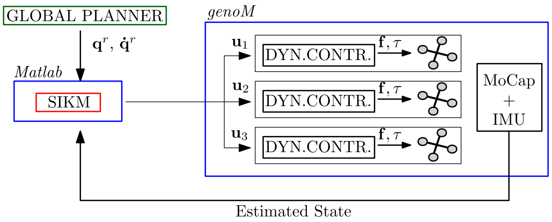


Fig. 7. Architecture used to perform simulations: a global planner generates the desired trajectory $\mathbf{q}^r, \dot{\mathbf{q}}^r$ and sends it to the local planner which generates the desired robot velocities. The blue rectangle on the right represents a realistic environment where the robotic system is simulated. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Simulations have been performed with *software in the loop*, including measurement noise and communication latency. Communication across the system, planning, and sensing are implemented using the middle-ware *Pocolibs* and the software framework *genoM*, enabling realistic simulations. The simulator of *Fly-Crane* has been used as preliminary validation step for several experiments, such as the ones in Sanalitro et al. (2020) and Jiménez-Cano, Sanalitro, Tognon, Franchi, and Cortés (2022), proving excellent adherence to the real testbed. Technical details about the simulation software and realistic numerical experiments can be found in Petitti et al. (2020).

The used system architecture is represented in Fig. 7: a global planner generates offline the desired trajectory $(\mathbf{q}^r, \dot{\mathbf{q}}^r)$ and this information is made available to the SIKM controller, implemented in Matlab-Simulink. The latter generates the desired robot velocities $\mathbf{u}_i$ which are sent to the robots every T seconds via wireless. Then, the low-level dynamical controller (geometric position controller (Lee, Leoky, & McClamroch, 2010)) of each robot converts these velocities into thrust and torque for the quadrotors. An unscented Kalman filter, running at 1 [kHz], fuses the Motion Capture (MoCap) system measurements (at 120 [Hz]) with the IMU measurements (at 1 [kHz]). The estimated system state is then sent to the SIKM controller.

We first compare the tracking error norm for the four control strategies when $T = 1.5$ [s] (see Figs. 8–9 and Table 2). Note that the tracking error does not converge to zero because of non-idealities implemented in simulation, such as sensor noise. The desired trajectory was generated in order to stress all components of $\mathbf{q}$, except for y-translation because of the system symmetry. From Fig. 9, one can see the benefits of the feedforward term w.r.t. to point-stabilization (PS), which is slower in tracking the desired trajectory as emphasized in the zoom plot of the component z_L. This is because the desired velocity $\mathbf{u}$ is updated only when a new measurement arrives. As for the three feedback-feedforward techniques, SIKM-C and SIKM-D exhibit the best performance. This is highlighted in the zoomed plot of z_L, where FF causes an overshoot.

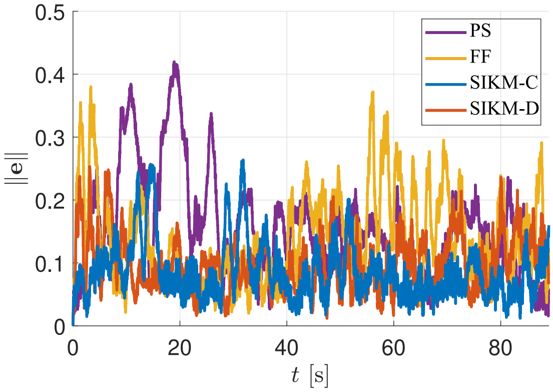


Fig. 8. Tracking error norm $\|\mathbf{e}(t)\| = \|\mathbf{q}(t) - \mathbf{q}^r(t)\|$ obtained with the tracking strategies described in Table 1 ($T = 1.5$ [s]).

Table 2
Mean tracking error norms obtained in simulation (best two results for each sampling time highlighted in bold font).

Sampling time T [s]	Mean tracking error norm			
	SIKM-D	SIKM-C	FF	PS
0.5	0.06	**0.05**	**0.05**	0.07
0.75	**0.06**	0.07	**0.06**	0.10
1.5	**0.10**	**0.09**	0.15	0.16

More interesting and general is the comparison of the four strategies with different sampling times. To compute the feedback gain k_{off} used in the SIKM-D, we first estimated the parameters $[\mu, \alpha, \gamma_1, \gamma_2] = [0.02,\ 0.13,\ 0.1,\ 0.2]$ along the desired trajectory depicted in Fig. 9 for different couples of (k, T) as described in Section 5.1, obtaining the stability region depicted in Fig. 10. Then, for each value of T, we chose the gain k which ensures the fastest highest convergence rate while keeping the system stable, by choosing the x coordinate corresponding to T on the curve $\tau(k_o)$ in Fig. 10. Table 2 reports the observed mean tracking error norms for each strategy. The resulting feedback gain–sampling time pairs (k_{off}, T) were $(2, 0.5)$, $(1.28, 0.75)$ and $(0.67, 1.5)$. Point-stabilization PS yields the largest error which increases quickly with the sampling time. All feedback-feedforward strategies are comparable for short values of T, while SIKM-C and SIKM-D yield the lowest errors for large T. In particular, the good performance of FF may be explained by the fact that the naive feedforward term in (11) is a good approximation of the one in (12) for small T. However, we stress that *SIKM-D is distributed*. Hence, the results in Table 2 shall be intended in even stronger way: the distributed version of our proposed controller not only outperforms standard centralized techniques (PS and FF), but is even comparable with its centralized version with online gain adaptation.

To further validate our method under realistic conditions, we also performed simulations with high position measurement noise, comparable to a GPS-based positioning system. As expected, the tracking error remains bounded, proving our design robust. Results of such additional simulations are provided in Rossi, Tognon, Ballotta et al. (2021, Appendix E).

7. Conclusions and future work

In this paper, we proposed a controller for multi-robot systems where feedback measurements are transmitted via wireless. We introduced a novel feedforward term and proposed a strategy to compute the feedback gain with convergence guarantees. The more multi-robot systems will be employed for applications, the

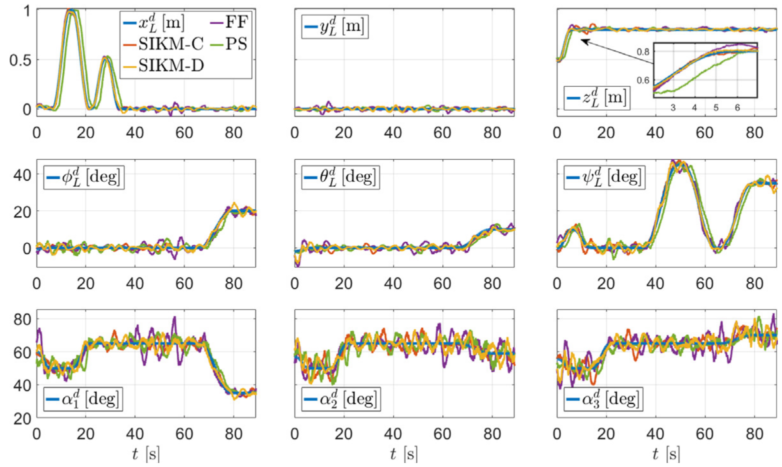


Fig. 9. Comparison of the variables $\mathbf{q}(t)$ in four different simulations, where the tracking strategies described in Table 1 are used. The sampling time is $T = 1.5$ [s]. The first two rows represent respectively the position x_l, y_L, z_L and orientation ϕ_L, θ_L, ψ_L (roll, pitch and yaw) of the load. On the bottom, row the angles $\alpha_i \overset{\text{def}}{=} q_i,\ i = 1, 2, 3$ between cables and load are depicted.

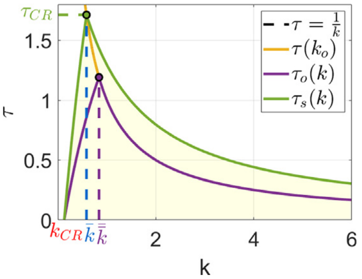


Fig. 10. Stability region, optimal gain k_o, and sampling time τ_o obtained by estimating the parameters $\mu, \alpha, \gamma_1, \gamma_2$ along the trajectory depicted in Fig. 9 for different couples (k, T).

more sampled-based control strategies will be needed: this work paves the way to future developments and implementations.

A natural evolution of this work is performing experiments on a real system. Further, a formal analysis including packets loss and latency, which are typical issues of wireless communication, should be developed.

Appendix. Useful lemmas

Lemma A.1. *Given* $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$, $\mathcal{X}$ *compact, and given a continuously differentiable function* $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$ *s.t.* $\mathbf{f}(\mathbf{0}) = \mathbf{0}$, *there exists* $\alpha > 0$ *s.t.:*

$$\|\mathbf{f}(\mathbf{x})\| \le \alpha \|\mathbf{x}\| \quad \forall \mathbf{x} \in \mathcal{X}.$$

Lemma A.2. *Given* $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$, $\mathbf{y} \in \mathcal{Y} \subset \mathbb{R}^p$, $\mathcal{X}$ *and* $\mathcal{Y}$ *compact, and given a continuously differentiable function in the second argument* $\mathbf{f}(\mathbf{x}, \mathbf{y}) : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^m$ *s.t.* $\mathbf{f}(\mathbf{x}, \mathbf{0}) = \mathbf{0}\, \forall \mathbf{x} \in \mathcal{X}$, *there exists* $\alpha > 0$ *s.t.:*

$$\|\mathbf{f}(\mathbf{x}, \mathbf{y})\| \le \alpha \|\mathbf{y}\| \quad \forall (\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}.$$

Lemma A.3. *Given* $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$, $\mathcal{X}$ *compact, and given two continuously differentiable functions* $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$, $\mathbf{g}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$ *s.t.* $\mathbf{f}(\mathbf{0}) = \mathbf{g}(\mathbf{0}) = \mathbf{0}$, *there exists* $\alpha > 0$ *s.t.:*

$$\|\mathbf{f}(\mathbf{x}) \cdot \mathbf{g}(\mathbf{x})\| \le \alpha \|\mathbf{x}\|^2 \quad \forall \mathbf{x} \in \mathcal{X}.$$

References

Antonelli, G., & Chiaverini, S. (2006). Kinematic control of platoons of autonomous vehicles. *IEEE Transactions on Robotics*, *22*(6), 1285–1292.

Arai, T., Pagello, E., Parker, L. E., et al. (2002). Advances in multi-robot systems. *IEEE Transactions on Robotics and Automation*, *18*(5), 655–661.

Conti, R., Meli, E., Ridolfi, A., & Allotta, B. (2015). An innovative decentralized strategy for I-AUVs cooperative manipulation tasks. *Robotics and Autonomous Systems*, *72*, 261–276.

Cortés, J., & Egerstedt, M. (2017). Coordinated control of multi-robot systems: A survey. *SICE Journal of Control, Measurement, and System Integration*, *10*(6), 495–503.

Dimarogonas, D. V., & Kyriakopoulos, K. J. (2006). Distributed cooperative control and collision avoidance for multiple kinematic agents. In *IEEE Conference on Decision and Control* (pp. 721–726). IEEE.

Erhart, S., & Hirche, S. (2015). Internal force analysis and load distribution for cooperative multi-robot manipulation. *IEEE Transactions on Robotics*, *31*(5), 1238–1243.

Farivarnejad, H., & Berman, S. (2018). Stability and convergence analysis of a decentralized proportional-integral control strategy for collective transport. In *American Control Conference* (pp. 2794–2801). IEEE.

Franchi, A., & Giordano, P. R. (2018). Online leader selection for improved collective tracking and formation maintenance. *IEEE Transactions on Control of Network Systems*, *5*(1), 3–13.

Franchi, A., Masone, C., Grabe, V., Ryll, M., Bülthoff, H. H., & Robuffo Giordano, P. (2012). Modeling and control of UAV bearing formations with bilateral high-level steering. *The International Journal of Robotics Research*, *31*(12), 1504–1525.

Franchi, A., Petitti, A., & Rizzo, A. (2019). Distributed estimation of state and parameters in multi-agent cooperative load manipulation. *IEEE Transactions on Control of Network Systems*, *6*(2), 690–701.

Gabellieri, C., Tognon, M., Sanalitro, D., Palottino, L., & Franchi, A. (2020). A study on force-based collaboration in swarms. *Swarm Intelligence*, *14*, 57–82.

Hayakawat, T., Matsuzawat, T., & Harat, S. (2006). Formation control of multi-agent systems with sampled information. In *IEEE Conference on Decision and Control* (pp. 4333–4338).

Jiménez-Cano, A., Sanalitro, D., Tognon, M., Franchi, A., & Cortés, J. (2022). Precise cable-suspended pick-and-place with an aerial multi-robot system. *Journal of Intelligent and Robotic Systems*, *105*(3), 1–13.

~~Lee, T., Leoky, M., & McClamroch, N. H. (2010). Geometric tracking control of a quadrotor UAV on SE(3). In *IEEE Conference on Decision and Control* (pp. 5420–5425). Atlanta, GA.~~

Liu, Z., Chen, W., Lu, J., Wang, H., & Wang, J. (2016). Formation control of mobile robots using distributed controller with sampled-data and communication delays. *IEEE Transactions on Control Systems Technology*, *24*(6), 2125–2132.

Manubens, M., Devaurs, D., Ros, L., & Cortés, J. (2013). Motion planning for 6-D manipulation with aerial towed-cable systems. In *Robotics: Science and Systems*. Berlin, Germany.

Masone, C., Bülthoff, H. H., & Stegagno, P. (2016). Cooperative transportation of a payload using quadrotors: A reconfigurable cable-driven parallel robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1623–1630).

Maza, I., Kondak, K., Bernard, M., & Ollero, A. (2010). Multi-UAV cooperation and control for load transportation and deployment. *Journal of Intelligent and Robotic Systems*, *57*(1–4), 417–449.

Mellinger, D., Shomin, M., Michael, N., & Kumar, V. (2013). Cooperative grasping and transport using multiple quadrotors. In *Distributed Autonomous Robotic Systems* (pp. 545–558). Springer.

Pereira, G. A., Campos, M. F., & Kumar, V. (2004). Decentralized algorithms for multi-robot manipulation via caging. *International Journal of Robotics Research*, *23*(7–8), 783–795.

Petitti, A., Sanalitro, D., Tognon, M., Milella, A., Cortés, J., & Franchi, A. (2020). Inertial estimation and energy-efficient control of a cable-suspended load with a team of UAVs. In *International Conference on Unmanned Aircraft Systems* (pp. 158–165).

Ren, Y., Sosnowski, S., & Hirche, S. (2020). Fully distributed cooperation for networked uncertain mobile manipulators. *IEEE Transactions on Robotics*, *36*(4), 984–1003.

Rossi, E., Tognon, M., Ballotta, L., Carli, R., Cortés, J., Franchi, A., et al. (2021). Coordinated multi-robot trajectory tracking over sampled communication. arXiv e-Prints, arXiv:2112.00165.

Rossi, E., Tognon, M., Carli, R., Franchi, A., & Schenato, L. (2021). Control of over-redundant cooperative manipulation via sampled communication. arXiv e-Prints, arXiv:2112.01107.

Rossi, E., Tognon, M., Carli, R., Schenato, L., Cortés, J., & Franchi, A. (2020). Cooperative aerial load transportation via sampled communication. *IEEE Control Systems Letters*, *4*(2), 277–282.

Sabattini, L., Secchi, C., Chopra, N., & Gasparri, A. (2013). Distributed control of multirobot systems with global connectivity maintenance. *IEEE Transactions on Robotics*, *29*(5), 1326–1332.

Sanalitro, D., Savino, H. J., Tognon, M., Cortés, J., & Franchi, A. (2020). Full-pose manipulation control of a cable-suspended load with multiple UAVs under uncertainties. *IEEE Robotics and Automation Letters*, *5*, 2185–2191.

Schwager, M., Julian, B. J., Angermann, M., & Rus, D. (2011). Eyes in the sky: Decentralized control for the deployment of robotic camera networks. *Proceedings of the IEEE*, *99*(9), 1541–1561.

Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). *Robotics: modelling, planning and control*. Springer.

Sieber, D., & Hirche, S. (2019). Human-guided multirobot cooperative manipulation. *IEEE Transactions on Control Systems Technology*, *27*(4), 1492–1509.

Simetti, E., & Casalino, G. (2016). Manipulation and transportation with cooperative underwater vehicle manipulator systems. *IEEE Journal of Oceanic Engineering*, *42*(4), 782–799.

Sreenath, K., & Kumar, V. (2013). Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots. In *Robotics: Science and Systems*. Berlin, Germany.

Stramigioli, S., Secchi, C., van der Schaft, A. J., & Fantuzzi, C. (2005). Sampled data systems passivity and discrete port-hamiltonian systems. *IEEE Transactions on Robotics*, *21*(4), 574–587.

Tagliabue, A., Kamel, M., Siegwart, R., & Nieto, J. (2019). Robust collaborative object transportation using multiple MAVs. *International Journal of Robotics Research*, *38*(9), 1020–1044.

Tognon, M., Gabellieri, C., Pallottino, L., & Franchi, A. (2018). Aerial co-manipulation with cables: The role of internal force for equilibria, stability, and passivity. *IEEE Robotics and Automation Letters Special Issue on Aerial Manipulation*, *3*(3), 2577–2583.

Tsiamis, A., Verginis, C. K., Bechlioulis, C. P., & Kyriakopoulos, K. J. (2015). Cooperative manipulation exploiting only implicit communication. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 864–869). IEEE.

Verginis, C. K., Nikou, A., & Dimarogonas, D. V. (2018). Communication-based decentralized cooperative object transportation using nonlinear model predictive control. In *European Control Conference* (pp. 733–738). IEEE.

Wang, Z., & Schwager, M. (2016). Force-amplifying n-robot transport system (force-ants) for cooperative planar manipulation without communication. *International Journal of Robotics Research*, *35*(13), 1564–1586.

Enrica Rossi received her Ph.D. Degree in Information Engineering and Master's Degree in Automation Engineering from the University of Padova in 2020 and 2016, respectively.

During the Ph.D. program, she held a visiting position at LAAS CNRS in Toulouse, France.

She currently is Software Developer at Mas Automazioni S.r.l.

Her research interests include control of single-robot systems and multi-robot systems with focus on wireless communication.

Dr. Marco Tognon is a tenured researcher at INRIA Rennes, France, in the Rainbow team. Before he was a postdoctoral researcher in the Autonomous System Lab (ASL) at ETH Zurich, Switzerland.

He received the Ph.D. degree in Robotics in 2018, from INSA Toulouse, France, developing his thesis at LAAS-CNRS.

His thesis has been awarded with three prizes.

He received the M.Sc. degree in automation engineering in 2014, from the University of Padua, Italy, with a master thesis carried out at MPI for Biological Cybernetics, Tübingen, Germany.

From 2018 to 2020 he has been postdoctoral researcher at LAAS-CNRS, Toulouse, France.

His current research interests include robotics, aerial physical interaction, multi-robot systems, and human–robot physical interaction.

Luca Ballotta received his Master's Degree in Automation Engineering in 2019 from the University of Padova, where he is currently pursuing the Ph.D. degree in Information Engineering.

He was Visiting Student at the Massachusetts Institute of Technology in 2020 and 2022.

His research interests include networked control systems subject to resource constraints, resilient distributed control, and learning-based safe control.

He was awarded with the Young Author Prize at the 2020 IFAC World Congress.

Ruggero Carli received the Laurea degree in computer engineering and the Ph.D. degree in information engineering from the University of Padova, Padua, Italy, in 2004 and 2007, respectively. From 2008 to 2010, he was a Postdoctoral Fellow with the Department of Mechanical Engineering, University of California at Santa Barbara, Santa Barbara, CA, USA. He is currently an Associate Professor with the Department of Information Engineering, University of Padova. His research interests include distributed algorithms for optimization, estimation and control over networks, nonparametric estimation and learning for robotics.

Juan Cortés received the engineering degree in control and robotics from the Universidad de Zaragoza (Spain) in 2000. In 2003, he received the Ph.D. degree in automated systems/robotics from the Institut National Polytechnique de Toulouse (France). In 2004, he made a postdoctoral stay in the Biotechnology, Biocatalysis and Bioregulation Unit (CNRS-Université de Nantes), before being hired as a permanent CNRS researcher at LAAS (Toulouse, France). In 2009, he was visiting professor at the Universitat Politècnica de Catalunya (Spain). He obtained a CNRS Research Director position in 2015.

His research interest is focused on the development of algorithms for computing and analyzing the motion of complex systems. Applications of these algorithms go beyond robotics. Indeed, he is strongly involved in interdisciplinary research in the areas of structural biology, biotechnology and materials science. Juan Cortś has participated in numerous European and national projects in all these areas. Between 2009 and 2015, he was co-chair of the IEEE-RAS TC on Algorithms for Planning and Control of Robot Motion. He has participated in the organization of several international workshops, and he coordinates the winter schools Algorithms in Structural Bioinformatics (AlgoSB) since 2012.

Antonio Franchi is a Full Professor in aerial robotics control at the University of Twente, Enschede, The Netherlands. He is an IEEE Fellow, class 2023, and an associate researcher at LAAS-CNRS, Toulouse, France. He received the Ph.D. degree in system engineering from the Sapienza University of Rome, Rome, Italy in 2010, and the HDR degree in Science, from the National Polytechnic Institute of Toulouse in 2016. From 2014 to 2019 he was a tenured CNRS researcher in the RIS group at LAAS-CNRS, Toulouse, France. From 2019 to 2021 he was an Associate Professor at the University of Twente.

He co-authored more than 160 publications in peer-reviewed international journals, books, and conferences on design, control, and estimation for robotic systems applied to multi-robot systems and aerial robots.

Luca Schenato received the Dr. Eng. degree in electrical engineering from the University of Padova in 1999 and the Ph.D. degree in Electrical Engineering and Computer Sciences from the UC Berkeley, in 2003. He held a post-doctoral position in 2004 and a visiting professor position in 2013–2014 at U.C. Berkeley. Currently he is Full Professor at the Information Engineering Department at the University of Padova. His interests include networked control systems, multi-agent systems, wireless sensor networks, distributed optimization and synthetic biology. Luca Schenato has been awarded the 2004 Researchers Mobility Fellowship by the Italian Ministry of Education, University and Research (MIUR), the 2006 Eli Jury Award in U.C. Berkeley and the EUCA European Control Award in 2014, and IEEE Fellow in 2017. He served as Associate Editor for IEEE Trans. on Automatic Control from 2010 to 2014 and he is currently Senior Editor for IEEE Trans. on Control of Network Systems and Associate Editor for Automatica.