



# Perception-Aware Human-Assisted Navigation of Mobile Robots on Persistent Trajectories

Marco Cagnetti, Marco Aggravi, Claudio Pacchierotti, Paolo Salaris, Paolo Robuffo Giordano

## ► To cite this version:

Marco Cagnetti, Marco Aggravi, Claudio Pacchierotti, Paolo Salaris, Paolo Robuffo Giordano. Perception-Aware Human-Assisted Navigation of Mobile Robots on Persistent Trajectories. IEEE Robotics and Automation Letters, 2020, 5 (3), pp.4711-4718. 10.1109/LRA.2020.3003882 . hal-02951947

**HAL Id: hal-02951947**

**<https://inria.hal.science/hal-02951947>**

Submitted on 29 Sep 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Perception-Aware Human-Assisted Navigation of Mobile Robots on Persistent Trajectories

Marco Cognetti<sup>1</sup>, Marco Aggravi<sup>2</sup>, Claudio Pacchierotti<sup>2</sup>, Paolo Salaris<sup>3</sup>, and Paolo Robuffo Giordano<sup>2</sup>

**Abstract**—We propose a novel shared control and active perception framework combining the skills of a human operator in accomplishing complex tasks with the capabilities of a mobile robot in autonomously maximizing the information acquired by the onboard sensors for improving its state estimation. The human operator modifies at runtime some suitable properties of a persistent cyclic path followed by the robot so as to achieve the given task (e.g., explore an environment). At the same time, the path is concurrently adjusted by the robot with the aim of maximizing the collected information. This combined behavior enables the human operator to control the high-level task of the robot while the latter autonomously improves its state estimation. The user's commands are included in a task priority framework together with other relevant constraints, while the quality of the acquired information is measured by the Shatten norm of the Constructibility Gramian. The user is also provided with guidance feedback pointing in the direction that would maximize this information metric. We evaluated the proposed approach in two human subject studies, testing the effectiveness of including the Constructibility Gramian into the task priority framework as well as the viability of providing either visual or haptic feedback to convey this information metric.

**Index Terms**—Human-Centered Robotics; Reactive and Sensor-Based Planning; Optimization and Optimal Control

## I. INTRODUCTION

IN this paper, we consider a shared control framework involving a mobile robot traveling along a desired trajectory for exploration/navigation purposes, with the shape/location of the trajectory being partially controlled by a human operator. As in typical shared control scenarios [1], [2], [3], [4], [5], we envisage a division of roles between the robot and the human operator. The mobile robot is equipped with onboard sensors and has enough autonomy for implementing lower-level control actions for addressing some ‘local’ constraints/requirements that would otherwise be hard to handle by the human operator. For instance, obstacle avoidance or attraction towards regions of interest are typical low-level behaviors, as well as dealing with limited actuation/energy, constrained dynamics or limited sensing [6]. The human operator is, instead, in charge of higher-level behaviors such as steering the mobile robot (either the robot itself or the whole trajectory followed by the robot) towards areas of interest or the next waypoint. The operator can provide commands to the mobile robot by acting on an input device and, when haptics is included (as in the case of this work), she/he can also receive a force feedback informing about what actions the robot would like to execute. The operator is

then left with the choice of whether (and to what degree) follow the feedback suggestions, thus blending her/his higher-level goals with the local needs of the robot.

This general idea is instantiated in this paper by considering a fundamental task for any mobile robot navigating in an environment with onboard sensing: the quality of the information acquired by the onboard sensors, which is needed for properly estimating the robot state. In fact, having a good knowledge about the robot internal state and possibly also about self-calibration and environment parameters is essential in order to safely move in an unstructured environment. The quality of the robot state estimation is highly influenced by the kind and amount of sensor information, especially in case of limited sensing capabilities and/or low cost (and noisy) sensors. Moreover, since any non-trivial robot dynamics are non-linear, the quality of information also depends on the actual trajectory performed by the robot which may be optimized for enhancing the estimation process. For these reasons, the problems of optimal information gathering (see [7] and references therein), aka *active sensing control*, as well as the *optimal sensors placement* ([8], [9]) have been widely studied in the literature.

In this context, in [8] we have recently proposed a trajectory planning framework aimed at solving *online* the active sensing control problem: we proposed a method to determine the optimal control actions for a mobile robot that maximize the amount of information collected by the onboard sensors (and, thus, improve the accuracy and convergence speed of an observer). In this work we illustrate how to embed the active sensing control method developed in [8] into a shared control framework: a mobile robot equipped with onboard sensors travels along a desired trajectory and localizes itself by measuring distances from some landmarks in the environment. The trajectory is continuously adjusted online by the robot autonomy so as to maximize the information acquired by the sensors following the approach presented in [8]. At the same time, the human operator controls some geometric properties of the trajectory: for instance she/he may control (i) the centroid location (in case of a closed trajectory) as a ‘pivot’ for exploring the environment, or (ii) the final point (in case of an open-ended trajectory), in order to guide the robot towards a specific point. The user's commands have a higher priority w.r.t. the active perception actions so as to ensure their correct fulfillment. However, as explained above, the operator is nevertheless provided with a force feedback for informing about where the robot autonomy would like to steer the trajectory along the user controlled degrees-of-freedom. In this way, the operator retains control over the global task, but she/he has the possibility to follow the autonomy suggestions whenever appropriate. We evaluate the proposed shared control framework with two human subjects studies in a virtual environment where a unicycle robot persistently travels along a closed path which

<sup>1</sup> University of Oulu - Center of Ubiquitous Computing, Faculty of Information Technology and Electrical Engineering – Oulu, Finland, e-mail: marco.cognetti@oulu.fi.

<sup>2</sup> CNRS, Univ Rennes, Inria, IRISA – Rennes, France, e-mail: {marco.aggravi, claudio.pacchierotti, prg}@irisa.fr.

<sup>3</sup> Department of Information Engineering and the research Center ‘E Piaggio’, University of Pisa – Pisa, Italy, e-mail: paolo.salaris@unipi.it.

is jointly modified by the local autonomy (for enhancing the robot estimation accuracy) and the human operator actions. The results show the benefits of the approach in typical navigation tasks.

## II. PRELIMINARIES

Let us consider a generic robot with dynamics

$$\dot{\mathbf{q}}(t) = \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t)), \quad \mathbf{q}(t_0) = \mathbf{q}_0 \quad (1)$$

$$\mathbf{z}(t) = \mathbf{h}(\mathbf{q}(t)) + \boldsymbol{\nu} \quad (2)$$

where  $\mathbf{q}(t) \in \mathbb{R}^n$  is the robot state,  $\mathbf{u}(t) \in \mathbb{R}^m$  the control inputs, and  $\mathbf{z}(t) \in \mathbb{R}^p$  the sensor outputs (i.e., the measurements available through the sensors mounted on the robot). We assume that  $\mathbf{f}$  and  $\mathbf{h}$  are analytic functions and that  $\boldsymbol{\nu} \sim \mathcal{N}(0, \mathbf{R}(t))$  is a normally-distributed Gaussian output noise with zero mean covariance matrix  $\mathbf{R}(t)$ . As the onboard sensors are not able to directly provide a measure of the whole state of the robot, we also assume that an observer (an Extended Kalman Filter – EKF – in our case) provides (online) an estimation  $\hat{\mathbf{q}}(t)$  of the true state  $\mathbf{q}(t)$ , together with the associated covariance matrix  $\mathbf{P}$ . Notice that  $\mathbf{q}(t)$  can be extended to also include self-calibration and environment parameters to be estimated.

Without loss of generality, we also assume that the system (1)–(2) is differentially flat [10]. This property applies to most of the mobile robots of our interest (e.g., unicycles and quadrotors). In practice this assumption is needed for the sake of computational efficiency, since it allows avoiding the numerical integration of (1) along the planned future path [8]. The flat outputs, and hence the whole state trajectory of the robot, is parametrized as a *closed* or *open* B-Spline. B-Splines curves are linear combinations, through a *finite* number  $N$  of control points  $\mathbf{x}_c = (\mathbf{x}_{c,1}^T, \mathbf{x}_{c,2}^T, \dots, \mathbf{x}_{c,N}^T)^T \in \mathbb{R}^{\kappa \cdot N}$ , of basis functions  $B_j^\alpha : S \rightarrow \mathbb{R}$  for  $j = 1, \dots, N$ . Each B-Spline is defined as

$$\gamma(\mathbf{x}_c, \cdot) : S \rightarrow \mathbb{R}^\kappa, \quad s \mapsto \sum_{j=1}^N \mathbf{x}_{c,j} B_j^\alpha(s, s) = \mathbf{B}_s(s) \mathbf{x}_c, \quad (3)$$

where  $S$  is a compact subset of  $\mathbb{R}$  and  $\mathbf{B}_s(s) \in \mathbb{R}^{\kappa \times N}$ . The degree  $\alpha > 0$  and knots  $\mathbf{s} = (s_1, s_2, \dots, s_\ell)$  are constant parameters, with  $\ell = N \geq \alpha$ .  $\mathbf{B}_s(s)$  is the set of basis functions and  $B_j^\alpha$  is the  $j$ -th basis function evaluated at  $s$ , obtained by the classical Cox-De Boor recursion formula in case of open B-Spline, or by a slightly modified version in case of closed B-Spline as shown in [6]. In the following we will then let  $\gamma_\gamma(\mathbf{x}_c, s)$  and  $\mathbf{u}_\gamma(\mathbf{x}_c, s)$  represent the state  $\mathbf{q}$  and inputs  $\mathbf{u}$  obtained (via the flatness) as a function of the B-Spline  $\gamma(\mathbf{x}_c, s)$ .

## III. SHARED ACTIVE SENSING CONTROL – PROBLEM FORMULATION

In this section, we show how the active perception problem defined in [8] can be integrated in a shared control architecture. The solution proposed in [8] is able to generate *online*, in *real-time*, a trajectory for a robotic system over a future time horizon aimed at maximizing the amount of information obtained by the onboard sensors. Having a real-time solution for the active sensing control part is also important in the context of this

work, due of the presence of an operator-in-the-loop that would be far less comfortable with the computational delays of an offline solution. For the reader's convenience, we now briefly summarize the main definition and results presented in [8] which are also relevant here.

### A. Constructibility Gramian

In [8], the *Constructibility Gramian* (CG) has been introduced as a suitable metric for quantifying the amount of information collected along a given trajectory by the onboard sensors of a robot with dynamics (1)–(2). By letting  $\mathbf{q}_f = \mathbf{q}(t_f)$  (where  $t_f$  can be considered as either a fixed final time or as the current running time) and  $\mathbf{P}_0$  the *a priori* information about the state  $\mathbf{q}_0 = \mathbf{q}(t_0)$  collected in the (infinite) time interval  $(-\infty, t_0)$  and available at  $t_0$ , the expression of the CG is

$$\begin{aligned} \mathcal{G}_c(t_0, t_f) &= \Phi^T(t_0, t_f) \mathbf{P}_0^{-1} \Phi(t_0, t_f) + \\ &+ \int_{t_0}^{t_f} \Phi(\tau, t_f)^T \mathbf{C}(\tau)^T \mathbf{W}(\tau) \mathbf{C}(\tau) \Phi(\tau, t_f) d\tau. \end{aligned} \quad (4)$$

where  $\mathbf{C}(\tau) = \frac{\partial \mathbf{h}(\mathbf{q}(\tau))}{\partial \mathbf{q}(\tau)}$ ,  $\mathbf{W}(\tau) \in \mathbb{R}^{p \times p}$  is a symmetric positive definite weight matrix, and  $\Phi(t, t_f) \in \mathbb{R}^{n \times n}$  is the state transition matrix (see [11] for its definition and properties). In [8] we also showed that, in absence of process noise, (4) is the solution of the Continuous Riccati Equation (CRE) and hence maximization of some norm of  $\mathcal{G}_c(t_0, t_f)$  is expected to produce a trajectory that minimizes the state estimation uncertainty.

### B. Optimization Problem

The reactive planning framework introduced in [8] for maximizing online (4) is also exploited in this paper as one of the tasks to be executed by the mobile robot. The overall optimization problem is defined as follows:

**Problem 1 (Online Shared Active Sensing Control)** For all  $t \in [t_0, t_f]$ , find the optimal location of the control points

$$\mathbf{x}_c^*(t) = \arg \max_{\mathbf{x}_c} \|\mathcal{G}_c(s_0, s_f)\|_\mu,$$

s.t.

- 1)  $\mathbf{q}_\gamma(\mathbf{x}_c(t), s_t) - \hat{\mathbf{q}}(t) \equiv \mathbf{0}$ , (state coherency)
- 2)  $\mathbf{f}(\mathbf{x}_c(\tau), s_\tau) \neq \mathbf{0}$ ,  $\forall \tau \in [t, t_f]$ , (flatness regularity)
- 3)  $L(\mathbf{x}_c(t), s_t, s_f) = L_d - L_t$ , (fixed length)
- 4)  $\mathbf{usr}(\mathbf{x}_c(t), s) - \mathbf{usr}_d \equiv \mathbf{0}$ , (user's task)

where

$$L_t = L(s_0, s_t) = \int_{s_0}^{s_t} v(\mathbf{x}_c, \sigma) d\sigma$$

represents the length already traveled by the robot on the previous interval  $[t_0, t]$  (and, analogously,  $L(\mathbf{x}_c(t), s_t, s_f)$  is the length of the trajectory in the future interval  $[s_t, s_f]$ ). Finally,  $v(\mathbf{x}_c, \sigma) = \|\partial \gamma(\mathbf{x}_c, s)/\partial s\|_2$ .

In Problem 1,  $\|\mathbf{A}\|_\mu = \sqrt[\mu]{\sum_{i=1}^n \lambda_i^\mu(\mathbf{A})}$  represents the Shatten norm of a matrix  $\mathbf{A}$ , with  $\mu \ll -1$  and  $\lambda_i(\mathbf{A})$  the  $i$ -th eigenvalue of  $\mathbf{A}$ . The Shatten norm is exploited as an approximation of the smallest eigenvalue of a matrix with the

benefit of always being differentiable even in case of repeated eigenvalues (thus, avoiding possible numerical issues).

Problem 1 contains a set of constraints. The fourth task is one of the novelties of this paper w.r.t. [8] and represents the *user's task*. The human operator is indeed in charge of modifying some geometric characteristics (e.g. a specific point, the centroid, the area, and so on) of the planned path for the robot. Two examples of possible geometric characteristics also used in Sec. V are:

- *Point on the path*: a point on the future path w.r.t. the current robot position can be controlled by the user in order to specify a waypoint in the environment where the robot needs to pass through. In this case, one can set constraint 4) in Problem 1 as

$$\mathbf{q}_\gamma(\mathbf{x}_c(t), s(t) + \tilde{s}) - q_d = 0,$$

with  $\tilde{s}$  being the arc-length distance between the current location of the robot and the point to be controlled by the user, and  $q_d$  the desired position for the forward point imposed by the user's task;

- *the centroid of a closed path*: the centroid may represent a "pivot" whose location can be controlled by the user in order to explore a desired zone of the environment. In this case, one can set constraint 4) in Problem 1 as

$$\frac{\int_{s_0}^{s_f} \mathbf{q}_\gamma(\mathbf{x}_c(t), \sigma) v(\mathbf{x}_c, \sigma) d\sigma}{\int_{s_0}^{s_f} \mathbf{q}_\gamma(\mathbf{x}_c(t), \sigma) d\sigma} - C_d = 0,$$

with  $C_d$  being the desired position for the centroid imposed by the user.

The other constraints of Problem 1, already used in [8], are: *state coherency*, for ensuring that the optimization over the future path is coherent with the current state estimate; *flatness regularity*, for avoiding intrinsic singularities introduced by the flatness transformations (e.g., for the unicycle the intrinsic singularity corresponds to the forward velocity equal to zero); *fixed length*, for guaranteeing well-posedness of Problem 1 since  $\|\mathcal{G}_c\|$  could be unbounded from above if the robot has an unlimited path length<sup>1</sup>.

We chose B-Spline curves (widely used in the literature, e.g., [6], [12]) to avoid an infinite-dimensional optimization problem, which would be intractable at runtime. B-Spline allowed us to formulate a finite-dimensional (thus numerically tractable at runtime) optimization problem, where the control points of the B-Spline become the optimization variables. However, our approach is compatible with any similar parametric path solution, e.g., Nurbs.

#### IV. SHARED ACTIVE SENSING CONTROL – PROPOSED SOLUTION

By letting

$$\dot{\mathbf{x}}_c(t) = \mathbf{u}_c(t), \quad \mathbf{x}_c(t_0) = \mathbf{x}_{c,0}$$

where  $\mathbf{u}_c(t) \in \mathbb{R}^{\kappa \times N}$ , Problem 1 is solved by an online constrained gradient descent action  $\mathbf{u}_c(t)$  affecting the location of the control points  $\mathbf{x}_c(t)$  (starting from an initial path defined

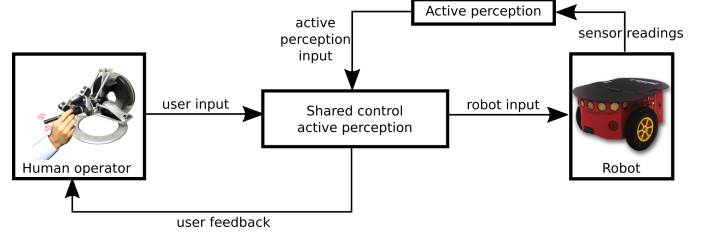


Fig. 1. Proposed framework. The priority-based controller computes the action  $\mathbf{u}_c$  according to the user inputs. Then, the trajectory generator computes the controls points  $\mathbf{x}_c$  of the B-Spline and, exploiting the flatness property, it also evaluates the inputs  $\mathbf{u}$  to the robotic system. The robot moves following these inputs and collects new measurements about the environment. The inputs  $\mathbf{u}$  and measurements  $\mathbf{z}$  feed an EKF that provides the estimation  $\hat{\mathbf{q}}$  of the state and the associated covariance matrix  $\mathbf{P}$ . These become inputs to the priority-based controller that, together with the user input  $\mathbf{u}_h$ , determines the next control action  $\mathbf{u}_c$  for optimizing the positioning of the control points on the future path. Finally, the user receives a feedback, pointing in the direction where the CG is maximized (i.e., where the estimation uncertainty is minimized).

by  $\mathbf{x}_{c,0}$ ), based on a task-priority approach (see, e.g., [13]) that translates all the constraints and the cost function in tasks with different priorities. Moreover, we also introduce a guidance feedback provided to the operator in order to make her/him aware about the possibility of increasing the amount of information collected by the robot along the future trajectory for reducing the estimation uncertainty.

##### A. Prioritized Stack Of Tasks

The overall architecture of our online shared active sensing control is illustrated in Fig. 1. The stack of tasks in the "Prioritized task control" of Fig. 1 is (starting from the task with highest priority): 1) state coherency, 2) flatness regularity, 3) length, 4) user's task and, finally, 5) optimization of the CG. The Appendix reports how  $\mathbf{u}_c(t)$  can be recursively generated. The user's commands have higher priority w.r.t. the CG optimization task since, as explained before, the operator must have full control over the geometric properties of the path and, therefore, the feedback generated by the active sensing represents a suggestion of how the robot should move for improving the estimation performance.

##### B. Feedback to the user

The operator is provided with a feedback information about how to steer the robot in order to maximize the amount of collected information. This feedback indicates the direction towards which the geometric characteristic of the robot trajectory, controlled by the operator, should move for maximizing the Shatten norm of the CG (while also being subject to all the other constraints). Its intensity is proportional to the improvement that could be achieved in that direction, i.e., the gradient of the Shatten norm. Since the CG maximization is the last task in the stack, its gradient is projected into the null space of the other tasks via the projector  ${}^A\mathbf{N}_4$  (see (11) in the Appendix). As a consequence, the component of the gradient of the Shatten norm of the CG filtered out by  ${}^A\mathbf{N}_4$  (and, thus, not implemented by the robot autonomy) is  $(\mathbf{I}_{\kappa N \times \kappa N} - {}^A\mathbf{N}_4) \nabla_{\mathbf{x}_c} \|\mathcal{G}_c(-\infty, s_f)\|_\mu$ . This component, when mapped back onto the user space via the Jacobian  $\mathbf{J}_4$

<sup>1</sup>Notice that, differently from this paper, in [8] we used the energy to guarantee the well-posedness of the optimization problem.

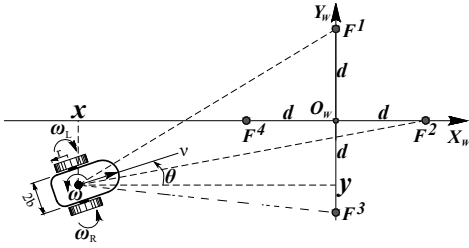


Fig. 2. Unicycle mobile robot with an onboard sensor able to measure the squared distances w.r.t. four markers  $F^i, i \in [1, \dots, 4]$  in the environment.

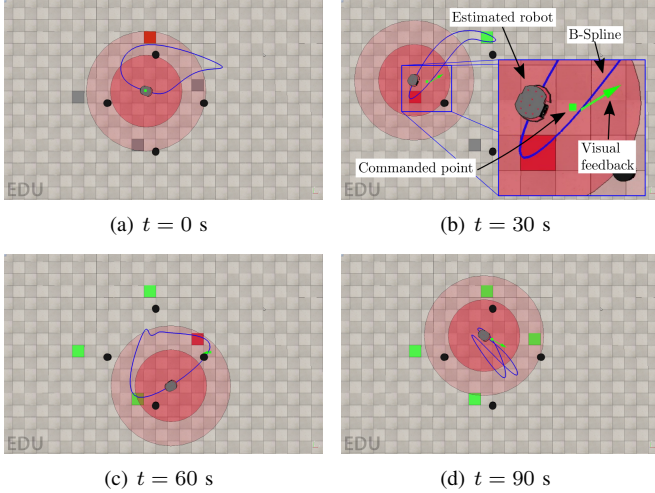


Fig. 3. Scenario #1: power grid activation. The users have to activate four switches in a certain order. The next switch to activate is red; when activated, it turns green. A switch activates when the real robot passes over it, but users only see the estimated robot position. Black dots are the markers from which the robot sensors compute the distances to estimate the robot position, the green dot is the point on the B-Spline controlled by the user, and the green arrow is the visual guidance provided to the user in condition CG-V.

(see (10) in the Appendix), can then be used for generating a feedback

$$\mathbf{f}_{\text{usr}} = \beta \mathbf{J}_4 (\mathbf{I}_{\kappa N \times \kappa N} - {}^A \mathbf{N}_4) \nabla_{\mathbf{x}_c} \|\mathcal{G}_c(-\infty, s_f)\|_{\mu}. \quad (5)$$

with  $\beta > 0$  a tunable gain.  $\mathbf{f}_{\text{usr}}$  will then guide the user in following the gradient of the Shatten norm of the CG along those directions that are not implemented by the robot autonomy because of the projection action of  ${}^A \mathbf{N}_4$ .

The feedback information  $\mathbf{f}_{\text{usr}}$  could be conveyed in different ways, e.g., as an arrow on a screen or as a kinesthetic force provided by a grounded haptic interface. Both of these options are tested in the experimental evaluation reported in the next Section.

## V. EXPERIMENTAL RESULTS

In order to evaluate the proposed approach, we consider two scenarios where the human subjects control the motion of a mobile robot by acting on particular geometric properties of the followed path through a kinesthetic haptic interface. The subjects also receive a visual or a haptic feedback about the quality and quantity of the information currently collected by the onboard sensors. A video of the experiments is available at <https://youtu.be/yGL7i48ZisA>.

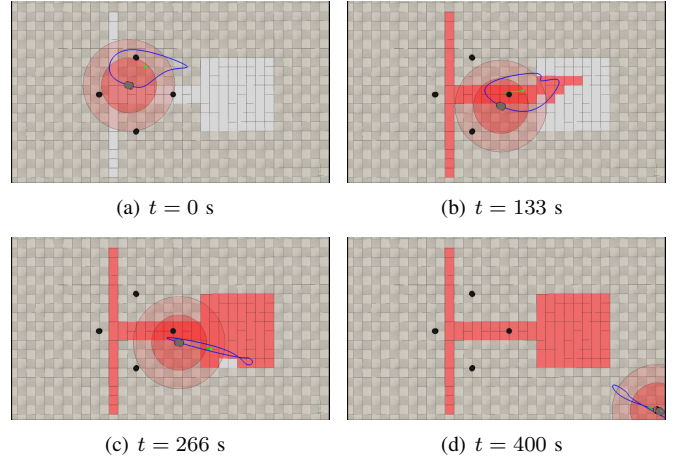


Fig. 4. Scenario #2: infrastructure monitoring. Users have to make the real robot visit 100 tiles on a factory floor. However, they can only see the action of the estimated robot. If the quality of the robot position estimation is low, the real robot will act differently from what users see, leading to a different set of visited tiles (see also Fig. 5). The red tiles are those visited by the estimated robot, the black dots are the markers from which the robot sensors compute the distances to estimate its state, the green dot is the centroid of the B-Spline (controlled by the user), and the green arrow is the visual guidance provided in condition CG-V.

**Mobile robot:** As case study, we consider a unicycle vehicle (see Fig. 2) moving on a plane  $\mathbf{X}_W \times \mathbf{Y}_W$ . The state of the robot is  $\mathbf{q}(t) = (x(t), y(t), \theta(t))^T$ , where the first two components are the position of a reference point attached to the robot on  $\mathbf{X}_W \times \mathbf{Y}_W$  and  $\theta(t)$  is the robot heading w.r.t.  $\mathbf{X}_W$ . The unicycle kinematic model is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (6)$$

where  $v$  and  $\omega$  are the linear and angular velocity of the robot, respectively. The flat outputs are  $\boldsymbol{\zeta} = [\zeta_1, \zeta_2]^T = [x, y]^T$  and hence  $\theta = \arctan(\dot{\zeta}_2/\dot{\zeta}_1)$ ,  $v = \sqrt{\dot{\zeta}_1^2 + \dot{\zeta}_2^2}$  and  $\omega = (\ddot{\zeta}_2\dot{\zeta}_1 - \dot{\zeta}_1\ddot{\zeta}_2)/(\dot{\zeta}_1^2 + \dot{\zeta}_2^2)$ . Moreover, we assume that the onboard sensor is able to provide measurements of the squared distances w.r.t. four markers  $F^i, i \in [1, \dots, 4]$ :

$$\mathbf{z} = \begin{bmatrix} x^2 + (y-d)^2 \\ (x-d)^2 + y^2 \\ x^2 + (y+d)^2 \\ (x+d)^2 + y^2 \end{bmatrix} + \boldsymbol{\nu}, \quad (7)$$

where  $d = 2$  m is the distance of each marker from the origin of the global reference frame. Moreover, in order to emulate the behavior of a real sensor, we consider a measurement noise that increases with the distance to the markers. This is obtained by weighting the measurement covariance matrix  $\mathbf{R}^{-1}$  with a weight matrix  $\mathbf{W}$  in such a way that  $\mathbf{R}_W^{-1} = \mathbf{W}^T \mathbf{R}^{-1} \mathbf{W}$ , where  $\mathbf{W} = \text{diag}(w_1, \dots, w_m)$ , with  $m$  the number of measurements ( $m = 4$  in (7)). Weight  $w_i$  is a function equal to one when the distance between the robot and the  $i$ -th marker is below a given threshold  $D_1$ , and it monotonically reaches zero when the distance is greater than another threshold  $D_2 > D_1$ . This mechanism ensures an infinite covariance matrix as soon as

the  $i$ -th distance is greater than  $D_2$  and hence, the measurement from the  $i$ -th marker can be considered no longer available. In our experiments, we set  $D_1 = 1.5$  m and  $D_2 = 2.5$  m.

*Master interface:* The subjects work in synergy with the active perception algorithm by acting on a grounded Omega.6 haptic interface (as shown in Fig. 1). By moving its end-effector, subjects control a geometric property of the closed B-Spline defining the trajectory of our mobile robot (see Sec. III-B). A screen in front of the subjects shows the considered virtual scenario.

*Experimental modalities:* We consider

- (N) The closed B-Spline trajectory of the robot is calculated by solving Problem 1 (see Secs. III-B, IV-A and the Appendix), where the CG maximization task is removed from the stack of tasks shown in Fig. 1. In other words, the quality of the robot state estimation is not maximized by the autonomy and it completely depends on the trajectory chosen by the human operator. The user receives no feedback on how to improve the estimation of the robot state;
- (CG-N) The closed B-Spline trajectory of the robot is generated by solving Problem 1, including the CG maximization task. However, the user receives no feedback on how to improve the estimation of the robot state.
- (CG-V) The closed B-Spline is generated by solving Problem 1, including the CG maximization task. The user receives visual guidance on how to move the considered trajectory point by means of an arrow defined by (5);
- (CG-H) The closed B-Spline is generated by solving Problem 1, including the CG maximization task. But this time the user receives haptic guidance on how to move the considered trajectory point, by means of a kinesthetic force, defined by (5) and provided via the Omega.6 haptic interface.

In all conditions, the subjects use the haptic interface to control the geometric property of the robot path, as explained in Sec. III-B.

#### A. Experimental Scenario #1: power grid activation

We consider a situation where the operator teleoperates the mobile robot for activating four electrical switches in a given order by controlling a point on the closed B-Spline defining the trajectory of the mobile robot, as seen in Sec. III-B.

The virtual scene, shown in Fig. 3, is simulated using V-REP. It consists of the four switches (dark grey, red, or green in Fig. 3), four markers (black), the estimated position of the robot (solid robot model), its closed B-Spline trajectory (blue, see also Sec. II), and the point of this trajectory controlled by the operator (green). The real robot estimates its position by measuring the squared distances w.r.t. the black markers, with a measurement noise increasing with the distance from the markers. As described in Secs. II and IV-A, the robot continuously moves along a closed B-Spline following the prioritized stack of tasks algorithm.

The task of this experiment consists in activating the four switches in the given order in no more than 90 seconds. The next switch to be activated is indicated in red while the already activated switches are marked in green. A switch is activated when the *real* robot passes through the center of its

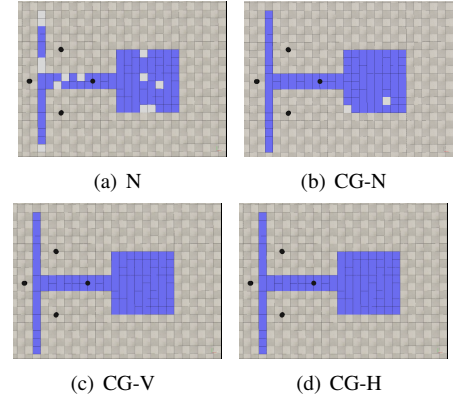


Fig. 5. Scenario #2: infrastructure monitoring. Representative example of the set of tiles visited by the *real* robot for one user. In condition N, where the quality of the robot position estimation is low, users think to have completed the task while some tiles still need to be visited by the real robot. The high quality of the estimation in the other conditions prevent this issue, proving the importance of including the CG maximization task as well that of providing an effective feedback to the user.

tile. However, the screen only shows the *estimated* robot state to the human operator (see Fig. 3(b)). It is therefore clear that, as the error in the estimation of the robot position increases, it becomes increasingly difficult for the user to complete the task, as she/he is only aware of the robot *estimated pose* while the switches are activated by the robot *real pose*. This situation is representative of many maintenance tasks carried out in remote or dangerous areas, where the operator needs to physically act on the environment without having an external reliable measure on the current robot status with, thus, the only available information being the robot estimation of its own pose.

For this scenario, the desired path length is set to  $L_d = 10$  m. We assume that a zero-mean Gaussian noise is acting on the measurements with covariance  $R = I$  and we used an EKF as observer. At the beginning of the experiment, the real robot is located at  $q(t_0) = (1.0 \text{ m}, 1.0 \text{ m}, 0.0 \text{ rad})^T$ , while the estimated one is at  $\hat{q}(t_0) = (-0.4 \text{ m}, 0.5 \text{ m}, -0.3 \text{ rad})^T$ .

1) *Subjects:* Ten participants took part to our two experiments, including 1 woman and 9 men (age 24–37 years old). The experimenter explained the context, the observability concept, and the meaning of the feedback. Then, he explained the task and adjusted the setup to be as comfortable as possible. Users performed one randomized repetition of the task per experimental condition, yielding 40 trials for this Scenario. Operators were asked to complete the task as fast as possible, taking however into account the received feedback.

2) *Results:* As a measure of performance, we registered the task completion time, the quality of the robot state estimation, and the perceived effectiveness as registered by the users.

Fig. 6(a) shows the average normalized task completion time, which is calculated as the time needed by the robot to complete the task, divided by the time limit (90 s). We ran a one-way repeated-measures ANOVA test ( $\alpha = 0.05$ ). The experimental condition (N vs. CG-N vs. CG-V vs. CG-H) was considered as the within-subject factor. Data were transformed using an arcsin transformation before running the statistical analysis. The ANOVA test revealed a statistically significant change in

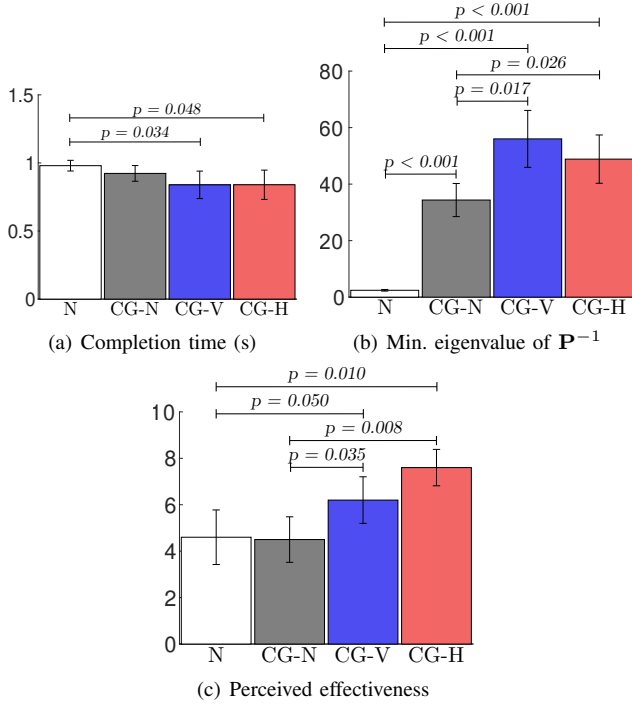


Fig. 6. Scenario #1: results. Mean and 95% confidence interval of (a) completion time, (b) minimum eigenvalue of  $\mathbf{P}^{-1}$  (from the EKF), and (c) perceived effectiveness for the four experimental conditions.

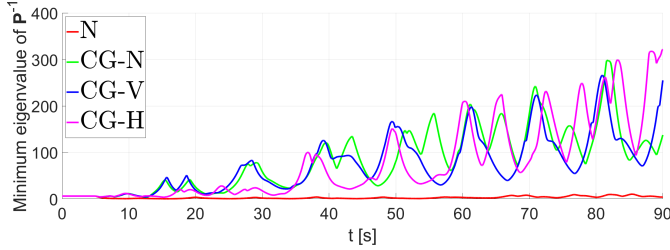


Fig. 7. Scenario #1. Evolution of the minimum eigenvalue of  $\mathbf{P}^{-1}$  (i.e., the maximum estimation uncertainty) for one representative subject in the four experimental conditions.

the task completion time ( $F(3,27) = 8.143$ ,  $p = 0.001$ ). Post hoc analysis with Bonferroni adjustments revealed a statistically significant difference between N vs. CG-V and N vs. CG-H. Statistically significant  $p$  values of this and following comparisons are reported in Figs. 6 and 8. Fig. 6(b) shows the average minimum eigenvalue of  $\mathbf{P}^{-1}$ . It was calculated as the mean value of the minimum eigenvalue of  $\mathbf{P}^{-1}$  throughout the task. This metric is interesting because it is inversely proportional to the maximum estimation uncertainty [8], making it a measure of the quality of the estimation of the robot state. We ran again a one-way repeated-measures ANOVA test ( $\alpha = 0.05$ ), revealing a statistically significant change in the value of this eigenvalue across experimental conditions ( $F(3,27) = 51.989$ ,  $p < 0.001$ ). A representative example of how this metrics evolves during the task is reported in Fig. 7. As expected, this metric reaches higher values in CG-V and CG-H conditions w.r.t. N and CG-N, confirming the results of Fig. 6(b). At the same time, the positive effect of the Gramian can be also seen when no feedback is available (CG-N) w.r.t. N.

At the end of the experiment, we asked the participants to rate the perceived effectiveness of the three experimental

conditions. The responses were given using bipolar Likert-type scales that ranged from 0 to 10, where a score of 0 meant “very low” and a score of 10 meant “very high” [14], [15]. To compare this metrics, we ran a one-way repeated-measures ANOVA test ( $\alpha = 0.05$ ) as before. Data were transformed using a square-root transformation. Fig. 6(c) shows the perceived effectiveness of the three experimental conditions. The ANOVA test revealed a statistically significant change in the perceived effectiveness ( $F(3,27) = 13.410$ ,  $p < 0.001$ ).

Finally, four subjects out of ten found conditions CG-H and CG-V to be the most effective at completing the task, followed by N (two subjects).

### B. Experimental Scenario #2: infrastructure monitoring

We also carried out a second experiment, considering a scenario where users teleoperate a mobile robot to inspect a factory floor by controlling the centroid of the closed B-Spline defining the mobile robot trajectory, as seen in Sec. III.

The virtual scene, shown in Fig. 4(a), is again simulated using V-REP and consists of a large rectangular area in which we define 100 tiles that need to be inspected (light grey in Fig. 4), four markers (black), the estimated position of the robot (solid robot model), its closed B-Spline trajectory (blue), and the centroid of this trajectory (green). As in Sec. V-A, the real robot estimates its position by measuring the squared distances w.r.t. black markers, with a measurement noise increasing with the distance from the markers. Again, the robot continuously moves along a closed B-Spline.

The task consists in visiting all 100 tiles in no more than 400 seconds. A tile is considered visited when the robot passes through its center. As before, the screen only shows the *estimated* robot to the human operator (see Fig. 4). However, differently from the previous Scenario #1, here the user does not have a direct feedback about the tiles actually visited by the real robot as she/he can only see the tiles visited by the *estimated* one.

Therefore, as the error in the estimation of the robot position increases, the probability that the *real* robot leaves some tile unchecked is likely to rise. This situation is representative of many inspection applications carried out in remote or dangerous areas, where the operator needs to check the status of a certain infrastructure without having an external reliable measure on the current robot status.

For this scenario, we considered the same setup, modalities, and subjects of the previous experiment of Sec. V-A. The users again performed one randomized repetition of the task per experimental condition, yielding 40 additional trials for this Scenario. Moreover, the desired path length is set to  $L_d = 10$  m. Again, we assume that a zero-mean Gaussian noise is acting on the measurements with covariance  $\mathbf{R} = \mathbf{I}$  and we used an EKF as observer. At the beginning, the real robot is located at  $\mathbf{q}(t_0) = (0.0 \text{ m}, 1.0 \text{ m}, 0.0 \text{ rad})^T$ , while the estimated one is at  $\hat{\mathbf{q}}(t_0) = (-0.4 \text{ m}, 0.5 \text{ m}, -0.3 \text{ rad})^T$ .

1) *Results*: Also in this case, as a measure of performance, we registered the average task completion time, the quality of the estimation of the robot state, and the perceived effectiveness as registered by the human users.

The average normalized task completion time was calculated as the time needed by the robot to complete the task, divided

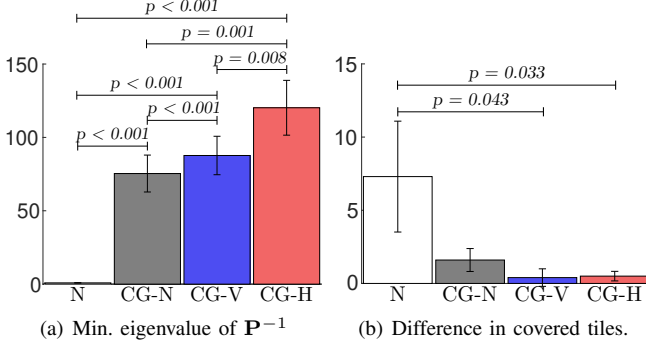


Fig. 8. Scenario #2: results. Mean and 95% confidence interval of (a) minimum eigenvalue of  $P^{-1}$  and (b) difference in the covered tiles.

by the time limit (400 s). We ran a one-way repeated-measures ANOVA test ( $\alpha = 0.05$ ). The experimental condition (N vs. CG-N vs. CG-V vs. CG-H) was considered as the within-subject factor. Data were transformed using an arcsin transformation before running the statistical analysis. The ANOVA test revealed no statistically significant change in the task completion time ( $F(3,27) = 1.657, p > 0.05$ ). Fig. 8(a) shows the average minimum eigenvalue of  $P^{-1}$ . We ran again a one-way repeated-measures ANOVA test ( $\alpha = 0.05$ ), revealing a statistically significant change in the value of this eigenvalue across experimental conditions ( $F(3,27) = 110.361, p < 0.001$ ). Fig. 8(b) shows the average difference between the number of tiles visited by the estimated robot (shown to the operator) vs. those actually covered by the real robot. The ANOVA test revealed a statistically significant change in the number of visited tiles ( $F(1,080,9.722) = 9.974, p = 0.010$ ).

As in Sec. V-A, at the end of the experiment, we asked the participants to rate the perceived effectiveness of the three experimental conditions using 11-points bipolar Likert-type scales. This time, the ANOVA test revealed no statistically significant change in the perceived effectiveness ( $F(3,27) = 1.048, p > 0.05$ ). Finally, seven subjects out of ten found condition CG-H to be the most effective at completing the task, followed by N (two subjects) and CG-V (one subject).

## VI. DISCUSSION

The reported results show the effectiveness and viability of the proposed shared control active perception technique. Using the active perception routine (CG-N, CG-V, CG-H) to maximize the information acquired by the robot significantly improves the performance of both tasks w.r.t. not considering the optimization of CG (N). Of course, this difference is most evident in the value of the minimum eigenvalue of  $P^{-1}$  (quality of the estimate), but also in the completion time (Scenario #1), perceived effectiveness (Scenario #1), and number of covered tiles (Scenario #2). When the user is provided with feedback regarding where to move to maximize this metric (CG-V, CG-H), the positive effect is even stronger with respect to when no feedback is provided (CG-N). On the other hand, the difference between the two types of feedback (visual vs. haptic, CG-V vs. CG-H) is more subtle. Haptic guidance outperforms the visual one only in the value of the minimum eigenvalue of

$P^{-1}$  for Scenario #2, although it was preferred by users in both Scenarios.

As the two Scenarios target different tasks, the role of the active sensing and the feedback provided affected the user's performance in different ways. In the first Scenario, the switches were activated only when the *real* robot touched them, but the users only saw the *estimated* robot position on their screen. For this reason, in condition N, where the estimate quality was poor (i.e., the positions of the estimated and real robots were significantly different), the users had to perform multiple passes on the switch until the real robot finally activated it. This behavior resulted in longer completion times as well as in users becoming quickly frustrated. In the second Scenario, we had no feedback at runtime about the tiles actually visited by the real robot. Only at the end, we compared the difference in the tiles covered by the real and the estimated robots. For this reason, during the task, it was less evident the role and importance of being provided with guidance information to maximize the quality of the estimate. Unfortunately, this situation resulted in two users out of ten judging both CG-V and CG-H “not useful” and “distracting,” which prevented us from finding a significant difference in the perceived effectiveness metrics of this Scenario. This results might be due to the limited understanding that these users had of the metrics and the observability concept, despite our effort in instructing them before starting the experiment. Of course, we expect operators to more easily comprehend the meaning of the feedback as well as better appreciate its role in this task. Despite this result, it is clear from the other metrics that providing feedback from the active perception routine still significantly improved the task performance. Finally, we registered two main subjective responses/behaviors regarding the use of visual vs. haptic feedback. One group of users appreciated the capability of the visual arrow in providing the guiding information without forcing/pushing their motion. On the other hand, another group of users appreciated the capability of force feedback in gently pushing them toward the right direction, without the need of additional thinking.

Despite the positive effects of the proposed approach, it is clear that the robot moves quite significantly to complete the task. To address this issue, we recall that the length of the B-spline can be adjusted according to the task at hand. Moreover, by interacting with our operators, we realized that most of them were unable to effectively use *all* the B-spline to carry out the task (e.g., visit the tiles). Conversely, they tended to position only a small part of the robot path on areas of interest (e.g., unvisited tiles). This is mostly due to their difficulty in understanding how the path will change when moving the centroid. For this reason, in the future, we want to employ more experienced users, as it will be the case in real-world industrial scenarios. Moreover, we want to study how to employ *open* B-Splines, enabling the operator to control the end of a continuously-moving path while solving Problem 1.

## VII. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a shared control active perception algorithm aimed at combining the high-level capabilities of a human operator with an autonomous controller whose goal is to minimize the estimation uncertainty along a persistent trajectory

defined as closed B-Spline. The operator is in charge of modifying some geometric properties of this curve. Moreover, visual or haptic feedback is provided to the users, guiding them toward the direction where the estimation uncertainty is minimized. We evaluated our framework in two different scenarios, performing a comparison with a framework that does not consider the active perception task and/or does not provide any feedback to the user. This comparison confirms the effectiveness of our methodology and the usefulness of the providing haptic feedback.

Future works will be dedicated to test our framework in complete experiments with real and more complex robots (including quadrotor UAVs and multi-robot systems). Moreover, different type of feedback will be also considered, such as audio, bracelets [16], and other wearable devices.

## APPENDIX

In this section, we briefly report how the gradient descent control action acting on the location of the control points can be recursively built. Let  ${}^1\mathbf{o}(t) = \mathbf{q}_\gamma(\mathbf{x}_c(t), s(t)) - \hat{\mathbf{q}}(t)$  represent the state coherency task, so that  ${}^1\dot{\mathbf{o}}(t) = \mathbf{J}_1 {}^1\mathbf{u}_c(t) + \mathbf{J}_s \dot{s} - \dot{\hat{\mathbf{q}}}(t)$  where  $\mathbf{J}_s = \frac{\partial \mathbf{q}_\gamma}{\partial s}$ , the Jacobian  $\mathbf{J}_1 = \frac{\partial \mathbf{q}_\gamma}{\partial \mathbf{x}_c} = \frac{\partial \mathbf{q}_\gamma}{\partial \Gamma} \frac{\partial \Gamma}{\partial \mathbf{x}_c}$ , and matrix  $\Gamma = \left[ \gamma(\mathbf{x}_c(t), s_t), \frac{\partial \gamma(\mathbf{x}_c(t), s_t)}{\partial s}, \dots, \frac{\partial^{(k)} \gamma(\mathbf{x}_c(t), s_t)}{\partial s^{(k)}} \right]$  for a suitable  $k \in \mathbb{N}$ . By choosing

$${}^1\mathbf{u}_c = -\mathbf{J}_1^\dagger (k_1 {}^1\mathbf{o}(t) - \dot{\hat{\mathbf{q}}}(t) + \mathbf{J}_s \dot{s}), \quad (8)$$

one obtains *exact* exponential regulation of the highest priority task  ${}^1\mathbf{o}(t)$  with rate  $k_1$ . The projector into the null space of this (first) task is just  ${}^A\mathbf{N}_1 = {}^A\mathbf{N}_0 - (\mathbf{J}_1 {}^A\mathbf{N}_0)^\dagger (\mathbf{J}_1 {}^A\mathbf{N}_0)$  with  ${}^A\mathbf{N}_0 = \mathbf{I}_{\kappa N \times \kappa N}$ .

The flatness regularity task was solved by defining a repulsive potential function  ${}^2\mathbf{o}(t) = U(\mathbf{x}_c, s(t)) = \sum_i \int_{S_i^*} \mathcal{U}_i(\delta_i(\mathbf{x}_c, \sigma)) d\sigma$  acting on the control points when  $\delta_i(\mathbf{x}_c, s)$  is close to zero over some intervals. The task consists in minimizing the potential function. By choosing

$${}^2\mathbf{u}_c = {}^1\mathbf{u}_c - (\mathbf{J}_2 {}^A\mathbf{N}_1)^\dagger (k_2 {}^2\mathbf{o}(t) + \mathbf{J}_2 {}^1\mathbf{u}_c), \quad (9)$$

with  $\mathbf{J}_2 = \partial U / \partial \mathbf{x}_c$ , one obtains *exact* exponential regulation of task  ${}^2\mathbf{o}(t)$  with rate  $k_2$  while still guaranteeing the accomplishment of the highest task  ${}^1\mathbf{o}(t)$ . The projector into the null space of both previous objectives can be computed (recursively) as  ${}^A\mathbf{N}_2 = {}^A\mathbf{N}_1 - (\mathbf{J}_2 {}^A\mathbf{N}_1)^\dagger (\mathbf{J}_2 {}^A\mathbf{N}_1)$ .

Let  ${}^3\mathbf{o}(\mathbf{x}_c(t), s_t, s_f) = L(\mathbf{x}_c(t), s_t, s_f) - L_d(t)$  represent the length task, so that  ${}^3\dot{\mathbf{o}}(t) = \mathbf{J}_3 {}^3\mathbf{u}_c(t) - \dot{L}_d(t)$ , with  $\mathbf{J}_3 = \int_{s_t}^{s_f} \frac{\partial}{\partial \mathbf{x}_c} v(\mathbf{x}_c, \sigma) d\sigma$  and  $\dot{L}_d(t)$  the desired task velocity. By choosing

$${}^3\mathbf{u}_c = {}^2\mathbf{u}_c + (\mathbf{J}_3 {}^A\mathbf{N}_2)^\dagger (-\lambda_3 {}^3\mathbf{o}(t) + \dot{L}_d(t) - \mathbf{J}_3 {}^2\mathbf{u}_c),$$

one obtains *exact* exponential regulation of task  ${}^3\mathbf{o}(t)$  with rate  $k_3$  while still guaranteeing the accomplishment of the previous higher priority tasks. The projector into the null space of all previous tasks is  ${}^A\mathbf{N}_3 = {}^A\mathbf{N}_2 - (\mathbf{J}_3 {}^A\mathbf{N}_2)^\dagger (\mathbf{J}_3 {}^A\mathbf{N}_2)$ .

Let  ${}^4\mathbf{o}(\mathbf{x}_c(t), \tilde{s}) = \mathbf{usr}(\mathbf{x}_c(t), s) - \mathbf{usr}_d(t)$  represent the user's command task, so that  ${}^4\dot{\mathbf{o}}(\mathbf{x}_c(t), \tilde{s}) = \mathbf{J}_4 {}^4\mathbf{u}_c -$

$\mathbf{usr}_d(t) = \mathbf{J}_4 {}^4\mathbf{u}_c - \mathbf{u}_h(t)$ , with  $\mathbf{J}_4 = \frac{\partial}{\partial \mathbf{x}_c} \mathbf{usr}(\mathbf{x}_c, \sigma)$  and  $\mathbf{u}_h(t)$  the user input (see Fig. 1). By choosing

$${}^4\mathbf{u}_c = {}^3\mathbf{u}_c + (\mathbf{J}_4 {}^A\mathbf{N}_3)^\dagger \left( \mathbf{usr}_d(t) - \lambda_4 \int {}^4\dot{\mathbf{o}}(\mathbf{x}_c(\tau), \tilde{s}) d\tau - \mathbf{J}_4 {}^3\mathbf{u}_c \right). \quad (10)$$

one obtains *exact* exponential regulation of task  ${}^4\mathbf{o}(t)$  with rate  $k_4$  while still guaranteeing the accomplishment of the previous higher priority tasks. The projector into the null space of all previous tasks is  ${}^A\mathbf{N}_4 = {}^A\mathbf{N}_3 - (\mathbf{J}_4 {}^A\mathbf{N}_3)^\dagger (\mathbf{J}_4 {}^A\mathbf{N}_3)$ .

Finally, we consider the maximization of the Schatten norm of the CG in the null-space of the previous tasks. The final gradient descent strategy becomes

$${}^5\mathbf{u}_c = {}^4\mathbf{u}_c + {}^A\mathbf{N}_4 \nabla_{\mathbf{x}_c} \|\mathcal{G}_c(-\infty, s_f)\|_\mu \quad (11)$$

The interested reader is referred to [8] for additional details on how to compute  $\nabla_{\mathbf{x}_c} \|\mathcal{G}_c(-\infty, s_f)\|_\mu$ .

## REFERENCES

- [1] D. A. Abbink, M. Mulder, and E. R. Boer, "Haptic shared control: smoothly shifting control authority?" *Cognition, Technology & Work*, vol. 14, no. 1, pp. 19–28, 2012.
- [2] H. Boessenkool, D. A. Abbink, C. J. Heemskerk, F. C. van der Helm, and J. G. Wildenbeest, "A task-specific analysis of the benefit of haptic shared control during telemanipulation," *IEEE Trans. Haptics*, vol. 6, no. 1, pp. 2–12, 2013.
- [3] M. Selvaggio, F. Abi-Farraj, C. Pacchierotti, P. Robuffo Giordano, and B. Siciliano, "Haptic-based shared-control methods for a dual-arm system," *IEEE Robot Autom Lett*, vol. 3, no. 4, pp. 4249–4256, 2018.
- [4] F. Abi-Farraj, C. Pacchierotti, O. Arenz, G. Neumann, and P. Robuffo Giordano, "A haptic shared-control architecture for guided multi-target robotic grasping," *IEEE Trans. Haptics*, 2019.
- [5] R. Rahal, G. Matarese, M. Gabiccini, A. Artoni, D. Prattichizzo, P. Robuffo Giordano, and C. Pacchierotti, "Caring about the human operator: haptic shared control for enhanced user comfort in robotic telemanipulation," *IEEE Trans. Haptics*, vol. 13, no. 1, pp. 197–203, 2020.
- [6] C. Masone, M. Mohammadi, P. Robuffo Giordano, and A. Franchi, "Shared planning and control for mobile robots with integral haptic feedback," *Int J Robot Res*, vol. 37, no. 11, pp. 1395–1420, 2018.
- [7] R. Bajcsy, Y. Aloimonos, and J. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, no. 2, pp. 177–196, Feb. 2018.
- [8] P. Salaris, M. Cognetti, R. Spica, and P. Robuffo Giordano, "Online optimal perception-aware trajectory generation," *IEEE Trans. Robotics*, vol. 35, no. 6, pp. 1307–1322, Dec. 2019.
- [9] M. Bianchi, P. Salaris, and A. Bicchi, "Synergy-based hand pose sensing: Optimal glove design," *Int J Robot Res*, vol. 32, no. 4, pp. 407–424, 2013.
- [10] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of nonlinear systems: Introductory theory and examples," *Int. J. Control*, vol. 61, no. 6, pp. 1327–1361, 1995.
- [11] P. J. Antsaklis and A. N. Michel, *Linear systems*. Springer Science & Business Media, 2006.
- [12] F. Abi-Farraj, N. Pedemonte, and P. Robuffo Giordano, "A visual-based shared control architecture for remote telemanipulation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016, pp. 4266–4273.
- [13] B. Siciliano and J. J. E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Proc. Intern. Conf. Advanced Robotics (ICAR)*, 1991, pp. 1211–1216.
- [14] M. Maisto, C. Pacchierotti, F. Chinello, G. Salvietti, A. De Luca, and D. Prattichizzo, "Evaluation of wearable haptic systems for the fingers in augmented reality applications," *IEEE Trans. Haptics*, vol. 10, no. 4, pp. 511–522, 2017.
- [15] L. Meli, C. Pacchierotti, G. Salvietti, F. Chinello, M. Maisto, A. De Luca, and D. Prattichizzo, "Combining wearable finger haptics and augmented reality: User evaluation using an external camera and the microsoft hololens," *IEEE Robot Autom Lett*, vol. 3, no. 4, pp. 4297–4304, 2018.
- [16] M. Aggravi, F. Pausé, P. Robuffo Giordano, and C. Pacchierotti, "Design and evaluation of a wearable haptic device for skin stretch, pressure, and vibrotactile stimuli," *IEEE Robot Autom Lett*, vol. 3, no. 3, pp. 2166–2173, 2018.