



**HAL**  
open science

## Online Optimal Perception-Aware Trajectory Generation

Paolo Salaris, Marco Cognetti, Riccardo Spica, Paolo Robuffo Giordano

► **To cite this version:**

Paolo Salaris, Marco Cognetti, Riccardo Spica, Paolo Robuffo Giordano. Online Optimal Perception-Aware Trajectory Generation. IEEE Transactions on Robotics, 2019, pp.1-16. 10.1109/TRO.2019.2931137 . hal-02278900

**HAL Id: hal-02278900**

**<https://inria.hal.science/hal-02278900v1>**

Submitted on 4 Sep 2019

**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.

# Online Optimal Perception-Aware Trajectory Generation

Paolo Salaris\*, Marco Cognetti†, Riccardo Spica‡ and Paolo Robuffo Giordano†

**Abstract**—This paper proposes an online optimal active perception strategy for differentially flat systems meant to maximize the information collected via the available measurements along the planned trajectory. The goal is to generate *online* a trajectory that minimizes the maximum state estimation uncertainty provided by the employed observer. To quantify the richness of the acquired information about the current state, the smallest eigenvalue of the *Constructibility Gramian* is adopted as a metric. We use B-Splines for parametrizing the trajectory of the flat outputs and we exploit a constrained gradient descent strategy for optimizing *online* the location of the B-Spline control points in order to actively maximize the information gathered over the whole planning horizon. To show the effectiveness of our method in maximizing the estimation accuracy, we consider two case studies involving a unicycle and a quadrotor that need to estimate their poses while measuring two distances w.r.t. two fixed landmarks. Concurrent estimation of calibration/environment parameters is also considered for illustrating how the proposed method copes with instances of active self-calibration and map building.

## I. INTRODUCTION

In humans, action selection is an important decision process that depends on the state of the body and of the environment [1]. Because signals in our sensory and motor systems are affected by variability or noise, our nervous system needs to estimate these states. Evidence from neuroscience shows that humans take into account the quality of sensory feedback when planning their future actions for better solving this estimation problem. This seems to be achieved by coupling feedforward strategies, aimed at reducing the negative effects of noise, with feedback actions, mainly intended to accomplish a given motor task and reduce the effects of control uncertainties [2]. In most cases, a robot also needs to solve a similar estimation problem in order to safely move in unstructured environments. For instance, it has to self-calibrate and self-localize w.r.t. the environment while, at the same time, a map of the surroundings may be built. These possibilities are highly influenced by the quality and amount of sensor information (i.e., available measurements), especially in case of limited sensing capabilities and/or low cost sensors. Moreover, including self-calibration states (or environment states) in the estimator increases the dimensionality of the state vector while the number of measurements typically remains unchanged [3]. As a consequence, it is important to determine inputs/trajectories that render all states and calibration parameters observable and, among them, the ones that can maximize

the information gathered along the trajectory under possible constraints, such as, e.g., limited energy/control effort. This allows reducing the estimation uncertainty and increasing the overall estimation performance of the employed estimator [4].

Given a dynamical system with some outputs (the available measurements), a first step is to establish whether the observation problem, which consists in finding an estimation of the true (but unknown) state of the robot/environment from the knowledge of the inputs and the outputs over a period of time, admits a solution [5], [6]. Differently from linear systems, in the nonlinear case state observability may also depend on the chosen inputs and, in some cases, one can show the existence of *singular inputs* that do not allow at all the reconstruction of the whole state [7]. A relevant problem is hence to consider some level of *active sensing/perception* in the control strategies of autonomous robots [8]. One crucial point in this context is the choice of an appropriate *measure of observability* to be optimized.

Starting from the nonlinear observability analysis in [5], which can only provide a “binary” answer about the local weak observability property of a nonlinear system, in [9] the authors have developed an observability measure based on the (local) Observability Gramian (OG). This can be seen as the OG of the linear time-varying system obtained by linearizing the nonlinear one around a given nominal trajectory. The main issue with this observability measure is the difficulty of obtaining a closed-form expression for several cases of practical interest in robotics. Indeed, the OG depends on the transition matrix associated to the linearized time-varying system, and a closed-form expression for this matrix is, in general, not available apart from some very special cases, as e.g., the first-order nonholonomic system (cf. [10]) with a particular choice of outputs [11] and the unicycle vehicle [12]. For all the other cases, i.e. quadrotor UAVs, manipulators, or humanoids, in which the transition matrix is not available, the Empirical Observability Gramian is a possible, widely used, alternative [9]:

$$W_o = \frac{1}{4\epsilon^2} \int_0^T \begin{bmatrix} \Delta z_1^T(t) \\ \vdots \\ \Delta z_n^T(t) \end{bmatrix} [\Delta z_1(t) \quad \dots \quad \Delta z_n(t)] dt$$

where  $\Delta z_i = z^{+i} - z^{-i}$  and  $z^{\pm i}$  is the simulated measurement when the state  $x^i$  is perturbed by a small value  $\pm\epsilon$ . As reported in [9], by letting  $\epsilon \rightarrow 0$  this approximation converges to the true observability Gramian. However, from a practical point of view  $\epsilon$  cannot be too small in order to avoid numerical issues. As a consequence, depending on the dynamics, the Empirical Observability Gramian may be a rough approximation. In [3] an improved approximation, named Expanded empirical observability Gramian, is introduced by incorporating higher

\* is with INRIA Sophia-Antipolis Méditerranée, 2004 Route des Lucioles, 06902 Sophia Antipolis, France, e-mail: paolo.salaris@inria.fr.

† are with CNRS, Univ Rennes, Inria, IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France, e-mail: {marco.cognetti, prg}@irisa.fr.

‡ is with Multi-Robot Systems Lab, Dep. of Aeronautics and Astronautics, Stanford University, Durand Building, 496 Lomita Mall, Stanford, CA, USA, e-mail: rspica@stanford.edu.

This project was partly funded by EU project CROWDBOT (H2020-ICT-2017-779942).

order Lie derivatives that are included in the observability matrix [5]. This makes it possible to capture input-output dependencies that do not directly appear in the sensor model. Despite the clear improvement, this measure still remains an approximation of the real OG.

By definition, the OG measures the level of observability of the *initial state* and hence, its maximization actually improves the performances in estimating (observing) the initial state of the robot. However, when the objective is to estimate the current/future state of the robot (which is implicitly the goal of most of the previous literature on this subject, and of this paper too), the OG is *not* the right metric. One main contribution of this paper is then to show that the right metric is in this case the *Constructibility Gramian* (CG) that indeed quantifies the level of constructibility of the current/future state (see Section II), which is obviously the state of interest for the sake of motion control/task execution. As another contribution, in our formulation we do not resort to any approximation of the CG even for those cases in which the transition matrix is not available in closed-form (which cover the vast majority of non-trivial robotic systems). We then propose an *online* optimal sensing control problem whose objective is to determine at *runtime* the future trajectory that maximizes the smallest eigenvalue of the CG, which corresponds to the maximum estimation uncertainty. The need for an *online* solution is motivated by the fact that, for a nonlinear system, the CG is a function of the state trajectories, that, in a real scenario, are not assumed directly measurable. By resorting to an offline optimization method that relies on an initial estimation of the state (as done in most prior literature, e.g., all the above-mentioned works), the resulting optimized trajectory would most likely be sub-optimal – for example, in a worst-case scenario of a system admitting singular inputs, the optimal trajectory from the estimated initial state could be very close to a singular one. On the other hand, by exploiting as initial guess an offline optimization based on the information available at the starting time, the use of an *online* strategy can mitigate the above-mentioned shortcoming since the optimal path can be continuously refined by relying on the current state estimation that converges over time towards the true state.

In order to make the online optimization problem tractable, i.e. to be performed in real-time, we restrict our attention to the case of non-linear differentially flat systems [13], which allows representing the flat outputs with a family of parametric curves (B-Splines in our case) function of a finite number of parameters (which become our optimization variables). Finally, we detail an *online* constrained gradient-descent optimization strategy able to consider different levels of priorities for the several optimization constraints, and we couple it with a concurrent estimation scheme (an Extended Kalman Filter in our case) for recovering an estimation of the true (but unknown) state during motion.

In [14], a preliminary version of this work has been proposed. However, the maximization of the smallest eigenvalue

of the OG was considered rather than the one of the CG<sup>1</sup>. Furthermore, in [14] the transition matrix was assumed to be known in closed-form, which is only possible for very simple dynamics (e.g., linear time-invariant systems, or specific cases such as the unicycle). Finally, in order to demonstrate the effectiveness of the proposed method, in this paper we consider two case studies involving a unicycle vehicle and a planar quadrotor (for which a closed-form solution of the transition matrix is not available) and a much larger number of tests and scenarios for a comprehensive validation of the method.

The rest of the paper is structured as follows. In Sect. II, the Constructibility Gramian (CG) is introduced and its link with the EKF is shown. Section III details our constrained optimization problem for a differentially flat system, where the optimization variables are the control points of the B-Splines that parametrize the trajectories of the flat outputs. In Sect. IV an online gradient-based solution is presented, while in Sect. VI, a number of simulation results are reported for the case studies introduced in Sect. V for showing the effectiveness of our method. The paper ends with some conclusions and future works.

## II. PRELIMINARIES

Let us consider a generic nonlinear system with noisy nonlinear outputs and negligible actuation/process noise

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

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

where  $\mathbf{q}(t) \in \mathbb{R}^n$  represents the state of the system,  $\mathbf{u}(t) \in \mathbb{R}^m$  is the control input,  $\mathbf{z}(t) \in \mathbb{R}^p$  is the sensor output (the measurements available through sensors),  $\mathbf{f}(\cdot)$  and  $\mathbf{h}(\cdot)$  are smooth functions (i.e.,  $C^\infty$ ), and  $\boldsymbol{\nu} \sim \mathcal{N}(0, \mathbf{R}(t))$  is a normally-distributed Gaussian output noise with zero mean and covariance matrix  $\mathbf{R}(t)$ .

The chosen formulation is (purposely) kept quite general for covering a broad class of practical cases. For instance, the state  $\mathbf{q}$  can include the pose of a mobile robot, its linear/angular velocity (in case the vehicle dynamics is taken into account), disturbances, as well as the environment (e.g. locations of landmarks) and/or calibration parameters (e.g. the focal length of a camera, sensor biases or physical/geometrical parameters). Likewise the inputs  $\mathbf{u}$  can represent velocity or force/torque commands, and the measurements  $\mathbf{z}$  can include typical sensor readings such as distances, bearing angles, forces, and so on.

The goal of this work is to minimize the state estimation uncertainty of the employed observer at the final time  $t_f$  in order to recover at best the (unmeasurable) state  $\mathbf{q}(t)$  by processing the collected (noisy) sensor readings  $\mathbf{z}(t)$  and applied inputs  $\mathbf{u}(t)$  over an interval  $[t_0, t_f]$ . We therefore need a suitable metric for capturing the information content of candidate trajectories  $\mathbf{q}(t)$  over  $[t_0, t_f]$ . Towards this end, we now briefly summarize some known concepts of nonlinear observability

<sup>1</sup>In the simple case studies considered in [14] the transition matrix was equal to the identity matrix and, hence, as it will be clear in next sections, the OG was equal to the CG. Of course, in more general situations this identity does not hold.

for arriving at the metric used in this work (which, as explained in the previous section, is the Constructibility Gramian (CG)).

The ability of determining the initial state  $\mathbf{q}_0 = \mathbf{q}(t_0)$  from knowledge of present and future system output  $\mathbf{z}(t)$  and input  $\mathbf{u}(t)$  over a time interval  $[t_0, t_f]$  revolves around the notion of *Observability* [15]. The initial state  $\mathbf{q}_0$  can be retrieved if one can distinguish, from the output measurements  $\mathbf{z}(t)$ , various initial states in a small neighborhood of  $\mathbf{q}_0$  without going too far from  $\mathbf{q}_0$ , or equivalently, if one cannot *locally* admit *indistinguishable* states. When this holds, system (1)–(2) is called *locally weakly observable*. A well-known observability criterion to check this property for a nonlinear system in the form (1)–(2) is the Observability Rank Condition (ORC) [5]. However, the ORC can only provide a “binary answer” about the local weak observability of the system, i.e. whether there exists (or not) at least one input, and hence one state trajectory, for (1)–(2) that allows recovering the initial state  $\mathbf{q}_0$ . An alternative more quantitative criterium, and hence more amenable to be used as performance index for quantifying the amount of information collected along a trajectory, is instead the so-called *Observability Gramian* (OG) (see [15], [16])  $\mathcal{G}_o(t_0, t_f) \in \mathbb{R}^{n \times n}$  defined as

$$\mathcal{G}_o(t_0, t_f) \triangleq \int_{t_0}^{t_f} \Phi(\tau, t_0)^T C(\tau)^T W(\tau) C(\tau) \Phi(\tau, t_0) d\tau \quad (3)$$

where  $C(\tau) = \frac{\partial \mathbf{h}(\mathbf{q}(\tau))}{\partial \mathbf{q}(\tau)}$ ,  $W(\tau) \in \mathbb{R}^{p \times p}$  is a symmetric positive definite weight matrix (a design parameter), and matrix  $\Phi(t, t_0) \in \mathbb{R}^{n \times n}$  is the state transition matrix (see [17] and [15] for its definition and properties) of the linear time-varying system obtained after linearizing the nonlinear system (1)–(2) around a trajectory. This matrix, also known as *sensitivity matrix* [16], is formally defined as  $\Phi(t, t_0) = \frac{\partial \mathbf{q}(t)}{\partial \mathbf{q}_0}$  and obeys the following differential equation

$$\dot{\Phi}(t, t_0) = \mathbf{A}(t)\Phi(t, t_0), \quad \Phi(t_0, t_0) = I, \quad (4)$$

where  $\mathbf{A}(t) \triangleq \frac{\partial \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t))}{\partial \mathbf{q}(t)}$ . If the (symmetric and semi-positive definite) OG  $\mathcal{G}_o$  is *full rank* over the time interval  $[t_0, t_f]$ , then system (1)–(2) is *locally weakly observable* [6]. Unlike linear systems, in the nonlinear case the OG is a function of the specific state trajectory  $\mathbf{q}(t)$  followed during the time interval  $[t_0, t_f]$ : therefore, one can attempt optimization of (some norm of) the OG w.r.t. the state trajectory  $\mathbf{q}(t)$  for determining the input  $\mathbf{u}(t)$  that can maximize the information about the initial state  $\mathbf{q}_0$  contained in the collected output  $\mathbf{z}(t)$  (see, e.g., [16], [14]).

Most robotics applications are, however, more concerned with the performance in reconstructing the *current state*  $\mathbf{q}(t)$  rather than observing the initial state  $\mathbf{q}_0$ , since knowledge of  $\mathbf{q}(t)$  is needed at runtime for implementing the required control action. The ability of determining the *current state*  $\mathbf{q}(t)$  (rather than the initial one) from knowledge of the present and past system output  $\mathbf{z}(t)$  and input  $\mathbf{u}(t)$  over  $[t_0, t]$  is instead captured by the so-called *Constructibility Gramian* [15], [16], which appears to be a less popular choice than the OG in the existing robotics literature on active sensing/perception. 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), the CG is defined as

$$\mathcal{G}_c(t_0, t_f) \triangleq \int_{t_0}^{t_f} \Phi(\tau, t_f)^T C(\tau)^T W(\tau) C(\tau) \Phi(\tau, t_f) d\tau. \quad (5)$$

From the semigroup property  $\Phi(t_0, t_f) = \Phi(t_0, \tau)\Phi(\tau, t_f) = \Phi^{-1}(\tau, t_0)\Phi(\tau, t_f)$ , one has  $\Phi(\tau, t_f) = \Phi(\tau, t_0)\Phi(t_0, t_f)$ . This can be used to show that the CG is related to the OG by

$$\mathcal{G}_c(t_0, t_f) = \Phi^T(t_0, t_f)\mathcal{G}_o(t_0, t_f)\Phi(t_0, t_f). \quad (6)$$

Since  $\Phi(t_f, t_0)$  is always nonsingular for continuous-time systems, it follows that  $\text{rank}(\mathcal{G}_c(t_0, t_f)) = \text{rank}(\mathcal{G}_o(t_0, t_f))$ : if a state trajectory  $\mathbf{q}(t)$  allows for recovering  $\mathbf{q}_0$  (full-rankness of  $\mathcal{G}_o$ ), it also allows for recovering  $\mathbf{q}_f$  (full-rankness of  $\mathcal{G}_c$ ) and vice-versa. However, optimization of the CG w.r.t.  $\mathbf{q}(t)$  will result in a state trajectory that maximizes the performance in reconstructing the *current state*  $\mathbf{q}_f$  rather than observing the initial state  $\mathbf{q}_0$ . It is also worth noting the role of matrix  $\Phi(t_0, t_f)$  in (6): its pre/post multiplication shifts at time  $t_f$  the information content of  $\mathcal{G}_o$  at time  $t_0$  about the initial state  $\mathbf{q}_0$ . This temporal shifting action will be often exploited in the following developments. Notice that, as  $\Phi(t_0, t_f)$  depends in general on the trajectory, the information content of  $\mathcal{G}_o$  may be shifted at time  $t_f$  in different manners.

**Remark 1** *Despite their similar definitions, the OG and the CG may represent two very different optimization objectives, and hence generate different optimal trajectories. Consider, for instance, a unicycle vehicle measuring its planar position in a global reference frame and needing to estimate its heading: in [12], the authors show that maximization of the smallest eigenvalue of the OG results in an optimal path barycentric w.r.t. the initial position of the vehicle (by considering the path as a continuous uniform distribution of unitary mass). Since the only difference between the OG and the CG is the use of  $\Phi^T(t, t_0)$  in place of  $\Phi^T(t, t_f)$  in their definitions, the CG would depend on the final position of the vehicle rather than on the initial one. By using the procedure in [12], it is hence straightforward to show that, when optimizing the CG, the optimal path results barycentric w.r.t. the final position. Therefore, the optimal paths in terms of the OG and of the CG are completely different since, as explained, they optimize two different (indeed opposite) objectives.*

We conclude by showing an important link between the CG and the optimal error covariance matrix  $\mathbf{P}$  for the linearization of system (1)–(2). Consider the linear time-varying system

$$\begin{aligned} \dot{\mathbf{q}}(t) &= \mathbf{A}(t)\mathbf{q}(t) + \mathbf{B}(t)\mathbf{u}(t), & \mathbf{q}(t_0) &= \mathbf{q}_0 \\ \mathbf{z}(t) &= \mathbf{C}(t)\mathbf{q}(t) + \boldsymbol{\nu} \end{aligned} \quad (7)$$

where  $\mathbf{A}(t) = \frac{\partial \mathbf{f}(\mathbf{q}, \mathbf{u})}{\partial \mathbf{q}}$ ,  $\mathbf{B}(t) = \frac{\partial \mathbf{f}(\mathbf{q}, \mathbf{u})}{\partial \mathbf{u}}$  and  $\mathbf{C}(t) = \frac{\partial \mathbf{h}(\mathbf{q})}{\partial \mathbf{q}}$ , that is, the linearization of (1)–(2) around a nominal trajectory  $\mathbf{q}(t)$ . In the absence of process noise, the optimal covariance matrix  $\mathbf{P}(t)$  for the estimation error is governed by the Continuous Riccati Equation (CRE) [19]

$$\dot{\mathbf{P}}(t) = \mathbf{A}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{A}^T(t) - \mathbf{P}(t)\mathbf{C}(t)^T \mathbf{R}^{-1}\mathbf{C}(t)\mathbf{P}(t),$$

which, exploiting the matrix identity  $\dot{P}^{-1} = -P^{-1}\dot{P}P^{-1}$ , can be rewritten as

$$\dot{P}^{-1}(t) = -P^{-1}(t)A(t) - A^T(t)P^{-1}(t) + C^T(t)R^{-1}C(t). \quad (8)$$

Considering the initial condition  $P(t_0) = P_0$ , the solution of (8) is (see [17], [20])

$$P^{-1}(t) = \Phi^T(t_0, t)P_0^{-1}\Phi(t_0, t) + \int_{t_0}^t \Phi^T(\tau, t)C^T(\tau)R^{-1}(\tau)C(\tau)\Phi(\tau, t)d\tau. \quad (9)$$

Since the second term of (9) is exactly the Constructibility Gramian  $\mathcal{G}_c(t_0, t)$  when  $W(t) = R^{-1}(t)$ , one has

$$P^{-1}(t) = \Phi^T(t_0, t)P_0^{-1}\Phi(t_0, t) + \mathcal{G}_c(t_0, t). \quad (10)$$

This expression can be interpreted as follows: the first term represents the contribution of the *a priori* information  $P_0$  available at time  $t_0$  but *shifted* at time  $t$  by the operator  $\Phi(t_0, t)$ , while the second term is the contribution of the information actually collected during the interval  $[t_0, t]$ .

Interestingly, expression (10) can also be reformulated in terms of the sole CG: let  $\mathcal{G}_c(-\infty, t)$  represent the CG computed over the (infinite) interval  $(-\infty, t]$ , and consider the partition

$$\mathcal{G}_c(-\infty, t) = \int_{-\infty}^{t_0} \Phi^T(\tau, t)C^T(\tau)R^{-1}(\tau)C(\tau)\Phi(\tau, t)d\tau + \int_{t_0}^t \Phi^T(\tau, t)C^T(\tau)R^{-1}(\tau)C(\tau)\Phi(\tau, t)d\tau. \quad (11)$$

Since  $\Phi(\tau, t) = \Phi(\tau, t_0)\Phi(t_0, t)$ , one has

$$\mathcal{G}_c(-\infty, t) = \Phi^T(t_0, t)\mathcal{G}_c(-\infty, t_0)\Phi(t_0, t) + \mathcal{G}_c(t_0, t). \quad (12)$$

By comparing (10) with (12), and by interpreting the *a priori* information  $P_0^{-1}$  as the information encoded by the CG over the interval  $[-\infty, t_0]$ , it follows that

$$P^{-1}(t) = \mathcal{G}_c(-\infty, t). \quad (13)$$

We can then conclude that maximization of (some norm of)  $\mathcal{G}_c(-\infty, t)$  is equivalent to minimization of (some norm of) the estimation error covariance  $P(t)$ . Maximization of some norm of CG is then expected to produce a state trajectory  $q(t)$  that results in an estimated state with *minimum uncertainty*. As a side effect, by reducing the estimation error covariance, both the precision and the convergence rate may also improve [21], even though these two objectives are not directly encoded in the CG (and, hence, not explicitly optimized by the machinery proposed in the following sections).

### III. PROBLEM FORMULATION

We now detail the optimal sensing control problem addressed in this paper. Let us consider the nonlinear dynamics (1)–(2), a time window  $[t_0, t_f]$ ,  $t_f > t_0$ , and a EKF built on system (1)–(2) for recovering an estimation  $\hat{q}(t)$  of the true (but unknown) state  $q(t)$  during motion. The goal is to develop an *online* optimization strategy for continuously solving, at each time  $t$ , the following optimal sensing control problem.

**Problem 1 (Optimal Sensing Control)** For all  $t \in [t_0, t_f]$ , find the optimal control strategy

$$u^*(t) = \arg \max_u \|\mathcal{G}_c(-\infty, t_f)\|, \quad (14)$$

s.t.

$$E(t_0, t_f) = \int_{t_0}^{t_f} \sqrt{u(\tau)^T M u(\tau)} d\tau = \bar{E} \quad (15)$$

where  $\|\cdot\|$  is a suitable norm for the CG (discussed in the next section),  $E(t_0, t_f)$  represents a measure of the “control effort” (or energy) needed for moving along the trajectory from  $t_0$  to  $t_f$ , and  $M > 0$  and  $\bar{E} > 0$  are design parameters. Note that, in our context, the final time  $t_f$  is not treated as a fixed parameter but, rather, as the time needed for spending the whole available energy  $\bar{E}$  during the robot motion.

**Remark 2** It is important to note that, in general,  $\|\mathcal{G}_c(-\infty, t_f)\|$  could be unbounded w.r.t. the terminal time  $t_f$  and/or the state  $q(t)$ . For nonlinear dynamic systems with or without drift, constraint (15) ensures well-posedness of Problem 1 preventing the state trajectories to grow unbounded in any non-idealized case. Indeed, any robot system is subject to dissipative effects (e.g., friction between wheels and ground, aerodynamic drag, joint friction, and so on) that in practice require a positive control effort for sustaining motion. Moreover, even if one decides to consider dissipative effects negligible (e.g., a UAV subject to gravity with negligible drag), any collected measurements w.r.t. the environment (e.g., relative distance, bearing, position) would eventually become uninformative as the robot travels too far away because of sensor noise, limited maximum range/resolution, or any other practical sensor limitation. Finally, other constraints needed by the optimization problem may also prevent an unbounded growth of the state trajectories in presence of drift. For instance, a ‘bounding box’ constraint for keeping the state trajectories confined (e.g., inside a room) would obviously avoid the issue, and analogously any other constraint requiring a positive norm of the control action over time (thus, imposing a continuous expense of control effort). Some of these possibilities have, indeed, been exploited in the quadrotor UAV simulations of Sect. VI-E where one feasibility requirement (flatness regularity) translates into demanding a positive thrust at all times, and a sensor noise covariance growing with the distance to the measured landmarks has been considered.

As explained, the need of an *online* solution is motivated by the fact that  $\mathcal{G}_c$  is a function of the state trajectory  $q(t)$  which is not assumed available. On the other hand, during the robot motion, it is possible to exploit a state estimation algorithm, such as a EKF, for improving *online* the current estimation  $\hat{q}(t)$  of the true state  $q(t)$ , with  $\hat{q}(t) \rightarrow q(t)$  in the limit. A converging state estimation  $\hat{q}(t)$  makes it possible to continuously refine (online) the previously optimized future path by exploiting the newly acquired information during motion.

We now proceed to better detail the structure of Problem 1 and of the proposed optimization strategy.

### A. Choice of the CG Performance Index

Among the many possible (matrix) norms, in this work we consider the smallest eigenvalue of the CG as performance index, i.e.  $\|\mathcal{G}_c(-\infty, t_f)\| = \lambda_{\min}(\mathcal{G}_c(-\infty, t_f))$ . Since the inverse of the smallest eigenvalue of the CG is a measure of the maximum estimation uncertainty (see (13)), maximizing  $\lambda_{\min}(\mathcal{G}_c(-\infty, t_f))$  is expected to *minimize* the maximum estimation uncertainty of the estimated state  $\hat{\mathbf{q}}(t)$  [11]. The use of the smallest eigenvalue as a cost function can, however, be ill-conditioned from a numerical point of view in case of repeated eigenvalues. For this reason, we replace  $\lambda_{\min}(\mathcal{G}_c(-\infty, t_f))$  with the so-called Schatten norm

$$\|\mathcal{G}_c(-\infty, t_f)\|_\mu = \sqrt[\mu]{\sum_{i=1}^n \lambda_i^\mu(\mathcal{G}_c(-\infty, t_f))} \quad (16)$$

where  $\mu \ll -1$  and  $\lambda_i(\mathbf{A})$  is the  $i$ -th smallest eigenvalue of a matrix  $\mathbf{A}$ : it is indeed possible to show that (16) represents a *differentiable* approximation of  $\lambda_{\min}(\cdot)$  [22]. The choice of taking the smallest eigenvalue as matrix norm for the CG is also known as **E-Optimality** criterium, which is related to the maximum estimation uncertainty.

**Remark 3** We note that the used cost function (Shatten norm or smallest eigenvalue of the CG) does not satisfy the Bellman principle, i.e., it is not additive since given two square matrices  $A$  and  $B$ ,  $\lambda_{\min}(A+B) \neq \lambda_{\min}(A) + \lambda_{\min}(B)$  in general. As a consequence truncations of optimal paths need not to be optimal: for instance, given the optimal path obtained for an energy  $\bar{E}_1$ , if one was interested in reducing the uncertainty by spending less energy  $\bar{E}_2 < \bar{E}_1$ , in general one would need to follow a completely different optimal path (and not the simple “truncation” at  $\bar{E}_2$  of the path optimized for  $\bar{E}_1$ ) because of the non-additivity of the cost function.

Clearly, other choices are also possible, see [23] for an overview. However, we believe that the **E-Optimality** criterium is the most appropriate choice when addressing observability optimization problems, since other existing criteria may lead to undesired behaviors. Consider, for instance, the (popular) *trace* operator taken as matrix norm of the CG (also known as **A-Optimality** criterium), which is related to the average estimation uncertainty. On one side, this measure is additive and hence satisfies the Bellman principle. On the other side, it can lead to contradictory results w.r.t. the notion of observability as illustrated in the following Remark 4.

**Remark 4** Consider a planar omnidirectional robot (modeled as a first order kinematic integrator) that measures its distance w.r.t. a beacon at the origin of a world reference frame, and that needs to estimate its world-frame position  $(x, y)$  from the collected measurements. In this case, the transition matrix is the identity. As a consequence, the OG and the CG computed in a time interval  $(t_0, t_f)$  coincide (see [18]) and their trace is equal to  $t_f - t_0 = T$  for any path with length  $T$ . This would hold also in case the chosen path results in a null smallest eigenvalue for  $\mathcal{G}$  (as long as all eigenvalues sum up to  $T$ ), thus clearly indicating a non-observable mode (i.e., the component of the state associated to the zero eigenvalue).

Similar considerations can also be drawn for the **D-Optimality** criterium, which is related to the volume of the estimation uncertainty ellipsoid by maximizing the *determinant* of CG (OG). The choice of adopting the smallest eigenvalue (**E-Optimality**) index guarantees, instead, optimization of the *worst-case* performance and, thus, prevents the occurrence of undesired results like the ones discussed above.

### B. CG decomposition

We now detail a decomposition of the CG instrumental for solving online Problem 1. Let  $t \in [t_0, t_f]$  be the current time during the robot motion along the planned trajectory: similarly to (12),  $\mathcal{G}_c(-\infty, t_f)$  can be expanded as

$$\begin{aligned} \mathcal{G}_c(-\infty, t_f) &= \Phi(t, t_f)^T \mathcal{G}_c(-\infty, t) \Phi(t, t_f) + \mathcal{G}_c(t, t_f) = \\ &= \Phi(t, t_f)^T \mathcal{G}_c(-\infty, t) \Phi(t, t_f) + \\ &+ \int_t^{t_f} \Phi(\tau, t_f)^T \mathbf{C}(\tau)^T \mathbf{W}(\tau) \mathbf{C}(\tau) \Phi(\tau, t_f) d\tau. \end{aligned} \quad (17)$$

By partitioning  $\Phi(\tau, t_f) = \Phi(\tau, t) \Phi(t, t_f)$  one has

$$\begin{aligned} \mathcal{G}_c(-\infty, t_f) &= \Phi(t, t_f)^T \left( \mathcal{G}_c(-\infty, t) + \right. \\ &+ \left. \int_t^{t_f} \Phi(\tau, t)^T \mathbf{C}(\tau)^T \mathbf{W}(\tau) \mathbf{C}(\tau) \Phi(\tau, t) d\tau \right) \Phi(t, t_f) \\ &= \Phi(t, t_f)^T \left( \mathcal{G}_c(-\infty, t) + \mathcal{G}_o(t, t_f) \right) \Phi(t, t_f). \end{aligned} \quad (18)$$

This decomposition of the CG is quite insightful for our goals: the first term  $\mathcal{G}_c(-\infty, t)$  represents a *memory* of the information about the current  $\mathbf{q}(t)$  collected while moving during  $[t_0, t]$  plus any additional “a priori” information that was possibly available at time  $t_0$  (representative of the interval  $[-\infty, t_0]$ ). The term  $\mathcal{G}_c(-\infty, t)$  is obviously “fixed” and cannot be optimized any longer at the current time  $t$ . On the other hand, the second term  $\mathcal{G}_o(t, t_f)$  stands for the information yet to be collected during the future interval  $[t, t_f]$ : this term can still be optimized at time  $t$ . Note that this contribution (correctly) takes the form of a OG since it encodes the information collected on the future period  $[t, t_f]$  about the ‘initial state’  $\mathbf{q}(t)$  (see Sect. II). Finally, the pre/post multiplication by  $\Phi(t, t_f)$  shifts all contributions at the final time  $t_f$  (this term is also not fixed and can be optimized at the current time  $t$ ).

From an implementation point of view, we also note that  $\Phi(t, t_f)$  and  $\mathcal{G}_o(t, t_f)$  in (18) are function of the state evolution  $\mathbf{q}(t)$  over  $[t, t_f]$  which, as explained, is assumed unknown: therefore, these two terms must be evaluated on a *predicted* state trajectory generated from the current estimated state  $\hat{\mathbf{q}}(t)$ . The term  $\mathcal{G}_c(-\infty, t)$  in (18) can instead be directly obtained via (13) as the inverse of the current (estimated) covariance matrix  $\hat{\mathbf{P}}(t)$  generated by the EKF<sup>2</sup>.

### C. Flatness and B-Spline parametrization

In order to reduce the complexity of the optimization procedure adopted to solve Problem 1 and, hence, to better

<sup>2</sup>If a different estimation algorithm is used then  $\mathcal{G}_c(-\infty, t)$  should be computed online via (11)-(12) on the estimated robot trajectory.

cope with the real-time constraint of an *online* implementation, we make two simplifying working assumptions.

First, we restrict our attention to the case of non-linear *differentially flat systems* [13]: for these systems, it is possible to find a set of outputs  $\zeta(\mathbf{q}) \in \mathbb{R}^\kappa$ , named *flat*, such that the state  $\mathbf{q}$  and inputs  $\mathbf{u}$  of the original system can be algebraically expressed in terms of the outputs  $\zeta$  and of a finite number of their time derivatives. In our context, the flatness assumption for system (1) allows avoiding any numerical integration of the nonlinear dynamics (1) for generating the state evolution  $\hat{\mathbf{q}}(\tau)$ ,  $\tau \in [t, t_f]$ , from the current estimated state  $\hat{\mathbf{q}}(t)$  by applying the planned inputs  $\mathbf{u}(t)$ .

Second, we choose to parameterize the flat outputs  $\zeta(\mathbf{q})$  (and, as a consequence, the state and inputs as well) by a family of curves function of a *finite* number of parameters. This choice further reduces the complexity of our optimization problem from an infinite-dimensional to a finite-dimensional one. Among the many possible parametric curves, we consider the class of B-Splines [26]. B-Spline 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 given as

$$\begin{aligned} \gamma(\mathbf{x}_c, \cdot) : S &\rightarrow \mathbb{R}^\kappa \\ s &\mapsto \sum_{j=1}^N \mathbf{x}_{c,j} B_j^\alpha(\mathbf{s}, s) = \mathbf{B}_s(s) \mathbf{x}_c \end{aligned} \quad (19)$$

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,  $\mathbf{B}_s(s)$  is the set of basis functions and  $B_j^\alpha$  is the  $j$ -th basis function evaluated in  $s$ , obtained by using the Cox-de Boor recursion formula [26].

**Remark 5** *Providing an analytical procedure for determining the minimum number of control points and degree  $\alpha$  that guarantees a solution to our optimization problem is very complex. However, besides ensuring continuity of all the state variables (i.e., of the flat outputs and a finite number of their derivatives), the parameter  $N$  should be chosen as a trade-off between the computational cost and the possibility of obtaining a more fine-tuned trajectory (thus increasing the value of the Shatten norm of the CG).*

By parameterizing the flat outputs  $\zeta(\mathbf{q})$  with a B-Spline curve  $\gamma(\mathbf{x}_c, s)$ , and by exploiting the differential flatness assumption, it follows that all quantities involved in Problem 1 (states  $\mathbf{q}$ , inputs  $\mathbf{u}$ , and, thus, any quantity needed for the CG computation) can be expressed as a function of the parameter  $s$  (the position along the spline) and of the control points  $\mathbf{x}_c$ . The latter will be then the (sole) optimization variables for Problem 1. In the following we will then let  $\mathbf{q}_\gamma(\mathbf{x}_c, s)$  and  $\mathbf{u}_\gamma(\mathbf{x}_c, s)$  represent the state  $\mathbf{q}$  and inputs  $\mathbf{u}$  determined (via the flatness) by the planned B-Spline path  $\gamma(\mathbf{x}_c, s)$ .

#### D. Additional requirements

In addition to the ‘bounded energy’ constraint (15) (necessary for ensuring well-posedness of Problem 1), in this work we also consider two additional requirements of interest for the optimal solution: *state coherency* and *flatness regularity*.

1) *State coherency*: when solving Problem 1 *online*, it is important to guarantee that, at the current time  $t$ ,  $\mathbf{q}_\gamma(\mathbf{x}_c(t), s(t)) = \hat{\mathbf{q}}(t)$  (i.e., it is indeed necessary to synchronize the B-Spline with the current state estimate of the robot), where  $\hat{\mathbf{q}}(t)$  is the current estimation of the true state  $\mathbf{q}(t)$  provided by the employed observer<sup>3</sup>. This requirement, already introduced in our previous work [14], then translates into some continuity constraints on the planned flat output path  $\gamma(\mathbf{x}_c(t), s(t))$  (and on some of its derivatives) at the current time  $t$  which, in turn, imposes some constraints on the motion of the B-Spline control points  $\mathbf{x}_c$ .

2) *Flatness regularity*: in order to always express  $\mathbf{q}$  and  $\mathbf{u}$  in terms of  $\zeta$  and of a finite number of their time derivatives, *intrinsic* and *apparent* singularities in flat differential systems (see [24], [25] for more details) must be avoided. While apparent singularities can be avoided by adopting a different set of flat outputs and different state space representations, intrinsic singularities must be handled by guaranteeing some constraints along the planned trajectories. Generally speaking, any intrinsic singularity can be expressed as a set of equalities  $\mathbf{f}(\mathbf{q}, \mathbf{u}) = \mathbf{0}$  and hence, in the contest of this work, as  $\mathbf{f}(\mathbf{x}_c, s) = \mathbf{0}$ . The flatness regularity requirement is then equivalent to move the control points in order to prevent function  $\mathbf{f}(\mathbf{x}_c, s)$  to vanish along the future planned path<sup>4</sup>.

#### E. Online Optimal Sensing Control

Exploiting (18) and letting  $s_0 = s(t_0)$ ,  $s_f = s(t_f)$  and, in general,  $s(t) = s_t$ , we can then reformulate Problem 1 as

**Problem 2 (Online Optimal Sensing Control)** *For all  $t \in [t_0, t_f]$ , find the optimal location of the control points*

$$\begin{aligned} \mathbf{x}_c^*(t) = \arg \max_{\mathbf{x}_c} & \|\Phi(\mathbf{x}_c(t), s_t, s_f)^T (\mathcal{G}_c(-\infty, s_t) + \\ & + \mathcal{G}_o(\mathbf{x}_c, s_t, s_f)) \Phi(\mathbf{x}_c(t), s_t, s_f)\|_\mu, \\ \text{s.t.} & \end{aligned}$$

- 1)  $\hat{\mathbf{q}}(t) - \mathbf{q}_\gamma(\mathbf{x}_c(t), s_t) \equiv \mathbf{0}$ ,
- 2)  $\mathbf{f}(\mathbf{x}_c(\tau), s_\tau) \neq \mathbf{0}$ ,  $\forall \tau \in [t, t_f]$
- 3)  $E(\mathbf{x}_c(t), s_t, s_f) = \bar{E} - E(s_0, s_t)$ ,

where

$$E(s_0, s_t) = \int_{s_0}^{s_t} \sqrt{\mathbf{u}(\sigma)^T \mathbf{M} \mathbf{u}(\sigma)} d\sigma$$

represents the control effort/energy already spent on the previous interval  $[t_0, t]$  (and, analogously,  $E(\mathbf{x}_c(t), s_t, s_f)$  the control effort/energy yet to be spent on the future interval  $[s_t, s_f]$ ).

The next section will be dedicated to detail the chosen optimization strategy for solving Problem 2.

<sup>3</sup>We note that this constraint is formally needed while the estimated state  $\hat{\mathbf{q}}(t)$  has not yet converged to the true one  $\mathbf{q}(t)$  since, after convergence, the requirement  $\mathbf{q}_\gamma(\mathbf{x}_c(t), s(t)) = \hat{\mathbf{q}}(t)$  would be trivially met.

<sup>4</sup>In our previous work [14], flatness regularity was not tackled because of the particularly simple case study which had neither apparent nor intrinsic singularities. This, however, does not translate to more realistic case studies such as the ones presented in this work.

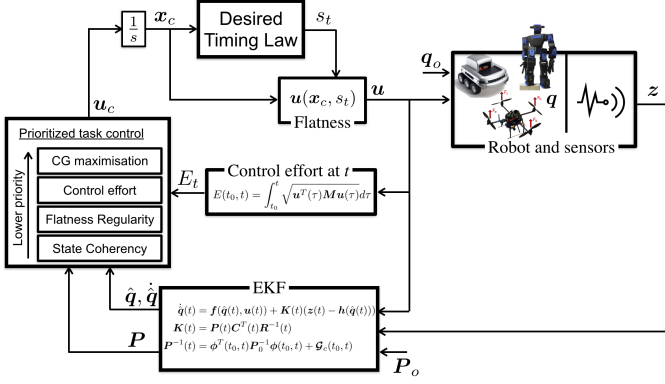


Fig. 1. Block diagram of the proposed method. The optimization action  $\mathbf{u}_c$  affects the positions of the control points that in turn are used to determine the desired timing law along the path. The control input  $\mathbf{u}$  for the robotic system is then computed by exploiting the flatness relationships from the position of the control points  $\mathbf{x}_c$  and  $s_t$ . While the robot moves along the trajectory, the control effort until the current instant  $t$  is computed online in order to correctly determine the control effort task. The EKF receives the sensor readings, the input  $\mathbf{u}$  and the initial covariance matrix  $\mathbf{P}_o$  and provides the current estimate of the robot's state  $\hat{\mathbf{q}}$  and the *a posteriori* covariance matrix  $\mathbf{P}$  that represents a memory of the past acquired information.  $\hat{\mathbf{q}}$  and  $\mathbf{P}$  are then used in the *Prioritized task control* to determine the next action  $\mathbf{u}_c$  for optimizing the location of the control points over the future path.

#### IV. AN ONLINE GRADIENT-BASED SOLUTION TO ACTIVE SENSING CONTROL

In this paper, we propose to solve Problem 2 by an online constrained gradient descent action affecting the location of the control points  $\mathbf{x}_c$ , and thus the overall shape of the trajectory followed by the robot (see Fig. 1). A feature of the chosen optimization strategy is its ability to handle different priorities for the various constraints/requirements and to cope with the real-time constraint of an online implementation. Towards this end, we also discuss how to obtain the gradient of both the cost function and the constraints w.r.t. the control points despite the assumed non-availability of a closed-form expression for the state transition matrix.

We then let

$$\dot{\mathbf{x}}_c(t) = \mathbf{u}_c(t), \quad \mathbf{x}_c(t_0) = \mathbf{x}_{c,0}, \quad (20)$$

where  $\mathbf{u}_c(t) \in \mathbb{R}^\kappa \times N$  is the optimization action to be designed, and  $\mathbf{x}_{c,0}$  the control points of a starting path (initial guess for the optimization problem).

Since Problem 2 involves optimization of the CG  $\mathcal{G}_c(-\infty, t_f)$  subject to multiple constraints, we design  $\mathbf{u}_c$  by resorting to the well-known general framework for managing multiple objectives (or tasks) at different priorities [27]. In short, let  ${}^i\mathbf{o}(\mathbf{x}_c)$  be a generic objective (or task/constraint) characterized by the differential kinematic equation  ${}^i\dot{\mathbf{o}} = \mathbf{J}_i(\mathbf{x}_c) {}^i\dot{\mathbf{x}}_c$ , where  $\mathbf{J}_i(\mathbf{x}_c)$  is the associated Jacobian matrix. Let also  $(\mathbf{J}_1, \dots, \mathbf{J}_r)$  be the stack of the Jacobians associated to  $r$  objectives ordered with decreasing priorities. Algorithm [27] allows computing the contributions of each task in the stack in a recursive way where  ${}^A\mathbf{N}_{i-1}$ , the projector into the null space of the augmented Jacobian  ${}^A\mathbf{J}_i = (\mathbf{J}_1, \dots, \mathbf{J}_i)$ , has the (iterative) expression  ${}^A\mathbf{N}_i = {}^A\mathbf{N}_{i-1} - (\mathbf{J}_i {}^A\mathbf{N}_{i-1})^\dagger (\mathbf{J}_i {}^A\mathbf{N}_{i-1})$  and  ${}^A\mathbf{N}_0 = \mathbf{I}$ .

Considering Problem 2, we then choose the following priority list (see ‘Prioritized task control’ in Fig. 1): the *state coherency* requirement should be the highest priority task, followed by the *regularity* constraint and then by the *bounded energy* constraint. Optimization of the CG is finally taken as the lowest priority task (thus projected in the null-space of all the previous constraints). This choice is motivated by the fact that the planned path  $\gamma$  should always be synchronized with the current estimated state  $\hat{\mathbf{q}}$  (*state coherency*) in order to generate the optimal path from the best available estimation of the true state. The generated optimal path should then avoid intrinsic flatness singularities (*flatness regularity*). Once these two basic requirements are satisfied, the bounded energy requirement must be also satisfied and maintained while the information metric is maximized. Different prioritizations are also clearly possible. Moreover, additional requirements can be also included in order to, e.g., avoid obstacles or reach a particular state value at  $t_f$ . We then now detail the various steps of this prioritized optimization.

##### A. State coherency

Let  ${}^1\mathbf{o}(t) = \mathbf{q}_\gamma(\mathbf{x}_c(t), s(t)) - \hat{\mathbf{q}}(t)$  represent the first task/requirement (state coherency), so that

$${}^1\dot{\mathbf{o}}(t) = \mathbf{J}_1 {}^1\mathbf{u}_c(t) + \mathbf{J}_s \dot{s} - \dot{\hat{\mathbf{q}}}(t) \quad (21)$$

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  $k \in \mathbb{N}$ . The order of derivative  $k$  is strictly related to the flatness expressions for the considered system: indeed,  $k$  is the maximum number of derivatives of the flat outputs needed for recovering the whole state and system inputs. The term  $\dot{\hat{\mathbf{q}}}(t)$  is, instead, the dynamics of the particular state estimation algorithm used to recover the state estimate  $\hat{\mathbf{q}}(t)$ . By choosing in (21)

$${}^1\mathbf{u}_c = -\mathbf{J}_1^\dagger (k_1 {}^1\mathbf{o}(t) - \dot{\hat{\mathbf{q}}}(t) + \mathbf{J}_s \dot{s}), \quad (22)$$

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  ${}^A\mathbf{N}_0 = \mathbf{I}_{\kappa N \times \kappa N}$ .

##### B. Flatness regularity

The second constraint for Problem 2 consists in preserving flatness regularity along the path  $\gamma(\mathbf{x}_c(t), s(t))$  by avoiding intrinsic singularities, i.e., by avoiding that the control points  $\mathbf{x}_c$  make the flatness singularity functions  $\mathbf{fl}(\mathbf{x}_c, s)$  to vanish. We tackle this requirement by designing a repulsive potential acting on the control points when  $\delta_i(\mathbf{x}_c, s) = \|\mathbf{fl}_i(\mathbf{x}_c, s)\|_2$  is close to zero over some intervals  $S_i^*$ . Let us define a potential function  $\mathcal{U}_i(\delta_i)$  growing unboundedly for  $\delta_i \rightarrow \delta_{min}$  and vanishing (with vanishing slope) for  $\delta_i \rightarrow \delta_{MAX}$ , where  $\delta_{min}$  and  $\delta_{MAX} > \delta_{min}$  represent minimum and maximum thresholds for the potential. A possible potential function, also adopted in this paper, is given in Fig. 2. The total repulsive



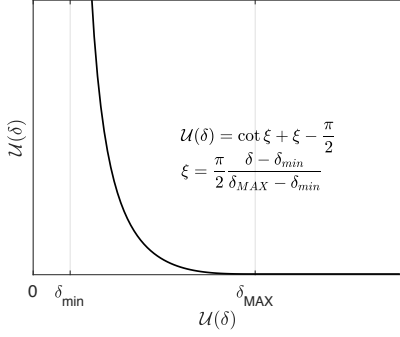


Fig. 2. A representative shape of the flatness regularity potential function.

potential associated to the  $i$ -th interval  $S_i^*$  is

$$U_i(\mathbf{x}_c, s(t)) = \int_{S_i^*} U_i(\delta_i(\mathbf{x}_c, \sigma)) d\sigma. \quad (23)$$

where  $S_i^* = S_i \cap [s_t, s_f]$  (indeed, the integral (23) is only evaluated on the future path) and, as a consequence,

$$U(\mathbf{x}_c, s(t)) = \sum_i \int_{S_i^*} U_i(\delta_i(\mathbf{x}_c, \sigma)) d\sigma \quad (24)$$

represents the repulsive potential for all  $N$  control points  $\mathbf{x}_{c,i}$ . The task is to minimize the potential (24), i.e.,  ${}^2o(t) = U(\mathbf{x}_c, s(t))$ . The time derivative of this task is

$${}^2\dot{o}(t) = \mathbf{J}_2 {}^1\mathbf{u}_c(t) \quad (25)$$

with  $\mathbf{J}_2 = \partial U / \partial \mathbf{x}_c$ . By choosing

$${}^2\mathbf{u}_c = {}^1\mathbf{u}_c - (\mathbf{J}_2^A \mathbf{N}_1)^\dagger (k_2 {}^2o(t) + \mathbf{J}_2 {}^1\mathbf{u}_c), \quad (26)$$

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

### C. Control effort

The third task in the priority (the control effort) can be implemented in a similar fashion. Let

$${}^3o(\mathbf{x}_c(t), s_t, s_f) = E(\mathbf{x}_c(t), s_t, s_f) - (\bar{E} - E(s_0, s_t))$$

where we consider that the control points  $\mathbf{x}_c(t)$  cannot obviously affect the past control effort  $E(s_0, s_t)$ . One then has (using Leibniz integral rule and simplifying terms)

$${}^3\dot{o}(t) = \mathbf{J}_3 {}^3\mathbf{u}_c(t)$$

where

$$\mathbf{J}_3 = \int_{s_t}^{s_f} \frac{\partial}{\partial \mathbf{x}_c} \sqrt{\mathbf{u}(\mathbf{x}_c, \sigma)^T \mathbf{M} \mathbf{u}(\mathbf{x}_c, \sigma)} d\sigma.$$

As a consequence, (26) is complemented as

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

and, again, the projector into the null space of all previous tasks is  ${}^A\mathbf{N}_3 = {}^A\mathbf{N}_2 - (\mathbf{J}_3^A \mathbf{N}_2)^\dagger (\mathbf{J}_3^A \mathbf{N}_2)$ .

### D. CG maximization

Finally, we consider the lowest priority task, that is, maximization of the Schatten norm of CG in the null-space of the previous tasks: the total control law for the control points to be plugged in (20) then becomes

$$\mathbf{u}_c = {}^3\mathbf{u}_c + {}^A\mathbf{N}_3 \nabla_{\mathbf{x}_c} \|\mathcal{G}_c(-\infty, s_f)\|_\mu.$$

The gradient of the Schatten norm of  $\mathcal{G}_c$  can be expanded as

$$\nabla_{\mathbf{x}_c} \|\mathcal{G}_c\|_\mu = \frac{1-\mu}{\mu} \sqrt{\sum_{i=1}^n \lambda_i^\mu(\mathcal{G}_c)} \left( \sum_{i=1}^n \lambda_i^{\mu-1}(\mathcal{G}_c) \frac{\partial \lambda_i(\mathcal{G}_c)}{\partial \mathbf{x}_c} \right),$$

where

$$\frac{\partial \lambda_i(\mathcal{G}_c)}{\partial \mathbf{x}_c} = \mathbf{v}_i^T \frac{\partial \mathcal{G}_c}{\partial \mathbf{x}_c} \mathbf{v}_i$$

and  $\mathbf{v}_i$  is the eigenvector associated to the  $i$ -th eigenvalue  $\lambda_i$  of  $\mathcal{G}_c$ . The expression of  $\frac{\partial \mathcal{G}_c}{\partial \mathbf{x}_c}$  can be obtained by using (18), and considering that  $\mathcal{G}_c(-\infty, s_t)$  is constant w.r.t.  $\mathbf{x}_c$ :

$$\begin{aligned} \frac{\partial \mathcal{G}_c}{\partial \mathbf{x}_c} &= \frac{\partial \Phi(\mathbf{x}_c, s_t, s_f)^T}{\partial \mathbf{x}_c} (\mathcal{G}_c(-\infty, s_t) + \mathcal{G}_o(\mathbf{x}_c, s_t, s_f)) \Phi(\mathbf{x}_c, s_t, s_f) + \\ &+ \Phi(\mathbf{x}_c, s_t, s_f)^T (\mathcal{G}_c(-\infty, s_t) + \mathcal{G}_o(\mathbf{x}_c, s_t, s_f)) \frac{\partial \Phi(\mathbf{x}_c, s_t, s_f)}{\partial \mathbf{x}_c} + \\ &+ \Phi(\mathbf{x}_c, s_t, s_f)^T \frac{\partial \mathcal{G}_o(\mathbf{x}_c, s_t, s_f)}{\partial \mathbf{x}_c} \Phi(\mathbf{x}_c, s_t, s_f) \end{aligned} \quad (27)$$

Evaluation of (27) requires availability of the following quantities: (i)  $\mathcal{G}_c(-\infty, s_t)$  (which, if a EKF is used, is directly estimated by the filter covariance matrix  $\mathbf{P}(t)$  as shown in (13), otherwise it can be computed by using (11)-(12)), (ii)  $\mathcal{G}_o(\mathbf{x}_c, s_t, s_f)$  which is obtained by forward integrating (3) over the future state trajectory  $\mathbf{q}_\gamma(\sigma)$ ,  $\sigma \in [s_t, s_f]$ ; (iii)  $\Phi(\mathbf{x}_c, s_t, s_f)$  which (being  $\Phi(t, t_f) = \Phi(\mathbf{x}_c, s_t, s_f) = \Phi(\mathbf{x}_c, s_f, s_t)^{-1} = \Phi(t_f, t)^{-1}$ ) is, again, obtained by forward integrating (4) over the future state trajectory  $\mathbf{q}_\gamma(\sigma)$ ,  $\sigma \in [s_t, s_f]$ ; and finally (iv) the gradients  $\frac{\partial \Phi(\mathbf{x}_c, s_t, s_f)}{\partial \mathbf{x}_c}$  and  $\frac{\partial \mathcal{G}_o(\mathbf{x}_c, s_t, s_f)}{\partial \mathbf{x}_c}$ . These can be obtained as follows: let us first consider  $\frac{\partial \Phi(\mathbf{x}_c, s_t, s_f)}{\partial \mathbf{x}_c}$  which we will denote, for convenience, as  $\Phi_{\mathbf{x}_c}(\mathbf{x}_c, s_t, s_f)$ . Since  $\Phi(\mathbf{x}_c, s_t, s_f) = \Phi(\mathbf{x}_c, s_f, s_t)^{-1}$ , one has

$$\Phi_{\mathbf{x}_c}(\mathbf{x}_c, s_t, s_f) = -\Phi(\mathbf{x}_c, s_f, s_t)^{-1} \Phi_{\mathbf{x}_c}(\mathbf{x}_c, s_f, s_t) \Phi(\mathbf{x}_c, s_f, s_t)^{-1}. \quad (28)$$

By leveraging relationship (4), the quantity  $\Phi_{\mathbf{x}_c}(\mathbf{x}_c, s_f, s_t)$  (needed to evaluate (28) and, thus, obtain the sought  $\Phi_{\mathbf{x}_c}(\mathbf{x}_c, s_t, s_f)$ ) can be obtained as the solution of the following differential equation over the future state trajectory  $\mathbf{q}_\gamma(\sigma)$ ,  $\sigma \in [s_t, s_f]$

$$\begin{aligned} \frac{d}{d\sigma} \frac{\partial \Phi(\mathbf{x}_c, \sigma, s_t)}{\partial \mathbf{x}_c} &= \frac{\partial}{\partial \mathbf{x}_c} \frac{d\Phi(\mathbf{x}_c, \sigma, s_t)}{d\sigma} = \\ &= \mathbf{A}_{\mathbf{x}_c}(\mathbf{x}_c, \sigma, s_t) \Phi(\mathbf{x}_c, \sigma, s_t) + \mathbf{A}(\mathbf{x}_c, \sigma, s_t) \Phi_{\mathbf{x}_c}(\mathbf{x}_c, \sigma, s_t), \\ \Phi_{\mathbf{x}_c}(\mathbf{x}_c, s_t, s_t) &= \mathbf{0} \end{aligned} \quad (29)$$

where  $\mathbf{A}_{\mathbf{x}_c}(\mathbf{x}_c, \sigma, s_t) = \frac{\partial \mathbf{A}(\mathbf{x}_c, \sigma, s_t)}{\partial \mathbf{x}_c}$  can be analytically computed (we note that the initial condition  $\Phi_{\mathbf{x}_c}(\mathbf{x}_c, s_t, s_t) = \mathbf{0}$  stems from the fact that  $\Phi(\mathbf{x}_c, s_t, s_t) = \mathbf{I}$  independently of  $\mathbf{x}_c$ ). Finally, by looking at (3), one can verify that  $\frac{\partial \mathcal{G}_o}{\partial \mathbf{x}_c}$  can be evaluated by exploiting all the quantities discussed so far (in particular the state transition matrix  $\Phi$  and its gradient  $\Phi_{\mathbf{x}_c}$ ).

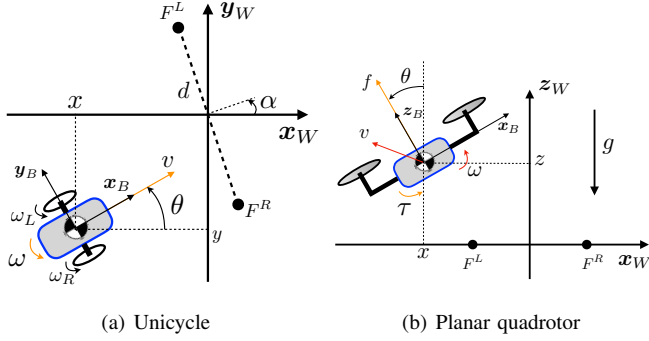


Fig. 3. Mobile robots and relevant quantities. The robot task is to localize itself with the smallest maximum estimation uncertainty by maximizing the information collected along the path through the outputs (i.e. distances w.r.t. landmarks  $F^R$  and  $F^L$ ).

**Remark 6** We note that other requirements/constraints could also be imposed along the path, such as, e.g., reaching a desired configuration, avoiding obstacles or imposing control boundaries and sensor constraints. All these additional requirements can be easily included in the priority stack (at any desired level).

For example, for reaching a desired state value  $\bar{\mathbf{q}}_{t^*}$  at a time  $t^*$ , it is sufficient to apply the same procedure used for the state coherency requirement to the following new task:  $\mathbf{o}(t) = \mathbf{q}_\gamma(\mathbf{x}_c(t), s(t^*)) - \bar{\mathbf{q}}_{t^*}$ , where  $\mathbf{q}_\gamma(\mathbf{x}_c(t), s(t^*))$  is the state computed in  $s(t^*)$  by the flatness in terms of the control points position at the current time  $t$  ( $t^*$  could be  $t_f$ ) This is indeed exploited in the case study of Sect. VI-E.

For obstacle avoidance or similar constraints, similarly to [28], [29], it would be sufficient to define a repulsive potential function  $PF = PF(\mathbf{x}_c, s(t))$  that acts on the control points of the B-Spline and pushes the planned path away from the obstacles (or from any other constraint, such as limited actuation). The procedure for determining such potential function and the control law for the control points would be analogous to the one reported in Sec. IV-B for the flatness regularity requirement. It is sufficient to define  $\delta_i(\mathbf{x}_c, s) = \|p(\mathbf{x}_c, s) - p_{obs}\|_2$  where  $p(\mathbf{x}_c, s)$  and  $p_{obs}$  are the position of the robot and obstacle respectively, or  $\delta_i(\mathbf{x}_c, s) = \|u(\mathbf{x}_c, s) - \bar{u}\|_2$  where  $u(\mathbf{x}_c, s)$  is the control input of the robot and  $\bar{u}$  its boundary.

## V. CASE STUDIES

In order to prove the effectiveness and the flexibility of our machinery, in this section we apply the method to the cases of a unicycle vehicle and a 2D quadrotor UAV, both equipped with a sensor able to provide noisy distances from two fixed landmarks, denoted by  $F^L$  and  $F^R$  (see Fig. 3). In the unicycle case, we also consider different scenarios, including the cases where self-calibration parameters and landmark positions are a subset of the whole state space to be estimated. For both robots, we assume a right-handed reference frame  $\mathcal{F}_W$  defined with origin in  $\mathcal{O}_W$  and axes  $\mathbf{X}_W, \mathbf{Y}_W, \mathbf{Z}_W$ .

a) *Unicycle vehicle*: Let us consider a unicycle vehicle moving on the plane  $\mathbf{X}_W \times \mathbf{Y}_W$  (see Fig. 3(a)). The configuration of the vehicle is described by  $\mathbf{q}(t) = (x(t), y(t), \theta(t))$ ,

where  $(x(t), y(t))$  is the position on the plane  $\mathbf{X}_W \times \mathbf{Y}_W$  of a reference point of the vehicle, and  $\theta(t)$  is the vehicle heading with respect to the  $\mathbf{X}_W$  axis. Using this notation and denoting by  $v(t)$  and  $\omega(t)$  the forward and angular velocity, the unicycle kinematics is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \quad (30)$$

Moreover,  $v(t) = \frac{r(\omega_R(t) + \omega_L(t))}{2}$  and  $\omega(t) = \frac{r(\omega_R(t) - \omega_L(t))}{2b}$ , where  $\omega_R(t)$  and  $\omega_L(t)$  the right and left wheel angular velocities while  $r$  and  $b$  are the wheels radius and the axle length, respectively. In the following, we will assume that the nominal values of parameters  $r$  and  $b$  are 0.1 m and 0.25 m.

The flat outputs for the unicycle vehicle are  $\zeta = [\zeta_1, \zeta_2]^T = [x, y]^T$  and it is easy to show that  $\theta = \arctan(\dot{\zeta}_2/\dot{\zeta}_1)$ ,  $v = \sqrt{\dot{\zeta}_1^2 + \dot{\zeta}_2^2}$  and  $\omega = (\ddot{\zeta}_2\dot{\zeta}_1 - \dot{\zeta}_1\ddot{\zeta}_2)/(\dot{\zeta}_1^2 + \dot{\zeta}_2^2)$ .

b) *Planar quadrotor*: Let us consider now a planar quadrotor moving on the plane  $\mathbf{X}_W \times \mathbf{Z}_W$  (see Fig. 3(b)). Let us also consider a body frame  $\mathcal{F}_B$  attached to the quadrotor center of mass, with  $\mathbf{Z}_B$  aligned with the trust direction. Let  $\mathbf{q} = [x \ z \ \dot{x} \ \dot{z} \ \theta \ \omega]^T = [\mathbf{p} \ \mathbf{v} \ \theta \ \omega]^T$  be the state of the planar quadrotor and let  $\mathbf{u} = [f \ \tau]^T$  be the inputs, i.e., the total trust and torque, the planar quadrotor dynamic model considered in this paper is

$$\begin{cases} \dot{\mathbf{p}} = \mathbf{v} \\ \dot{\mathbf{v}} = \begin{bmatrix} 0 \\ -g \end{bmatrix} + \frac{f}{m} \begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix} \\ \dot{\theta} = \omega \\ \dot{\omega} = \frac{\tau}{I} \end{cases} \quad (31)$$

with  $m$  and  $I$  being the quadrotor mass and inertia, and  $g$  the gravity acceleration magnitude. The flat outputs are  $\zeta = [\zeta_1, \zeta_2]^T = [x, z]^T$  and it is easy to show that  $\dot{x} = \dot{\zeta}_1$ ,  $\dot{z} = \dot{\zeta}_2$ ,  $\theta = \arctan(-\dot{\zeta}_1/(\dot{\zeta}_2 + g))$ ,  $\omega = (\ddot{\zeta}_2\dot{\zeta}_1 - \dot{\zeta}_1\ddot{\zeta}_2)/(\dot{\zeta}_1^2 + (\dot{\zeta}_2 + g)^2)$ ,  $f = \sqrt{\dot{\zeta}_1^2 + (\dot{\zeta}_2 + g)^2}$ , and  $\tau = \dot{\omega}I$ .

c) *Sensory measurements*: For the unicycle case, we assume that the landmarks are located on the plane of motion. Let  $d$  and  $\alpha$  be the distance between the landmarks and the orientation of the segment in-between w.r.t.  $\mathbf{X}_W$ , respectively. The cartesian coordinates of these two points w.r.t.  $\mathcal{F}_W$  are  $F^R = (\frac{d}{2} \sin \alpha, -\frac{d}{2} \cos \alpha, 0)$  and  $F^L = (-\frac{d}{2} \sin \alpha, \frac{d}{2} \cos \alpha, 0)$ . Hence, the outputs, in this case, can be expressed as (see Fig. 3(a))

$$\begin{aligned} h_R &= \left(x - \frac{d}{2} \sin \alpha\right)^2 + \left(y + \frac{d}{2} \cos \alpha\right)^2, \\ h_L &= \left(x + \frac{d}{2} \sin \alpha\right)^2 + \left(y - \frac{d}{2} \cos \alpha\right)^2. \end{aligned} \quad (32)$$

In the following, we will assume that the actual position of the landmarks are such that  $d = 4$  m and  $\alpha = 0$  rad.

For the planar quadrotor, we will assume that the landmarks are located on a parallel plane w.r.t. the plane of motion. Let  $\bar{y}$  be the distance of such a plane from  $\mathbf{X}_W \times \mathbf{Z}_W$ . The cartesian coordinates of these two points w.r.t.  $\mathcal{F}_W$  are  $F^R = (d, \bar{y}, 0)$

and  $\mathbf{F}^L = (-d, \bar{y}, 0)$ . Hence, the outputs, in this case, can be expressed as (see Fig. 3(b))

$$\begin{aligned} h_R &= (x - d)^2 + \bar{y}^2 + z^2, \\ h_L &= (x + d)^2 + \bar{y}^2 + z^2. \end{aligned} \quad (33)$$

with  $d = 0.5$  m and  $\bar{y} = 1$  m in our simulations.

d) *Timing laws along B-Spline*: The parametrization imposed by the current B-Spline, i.e.  $s(t)$ , can be changed depending on the desired timing law chosen for traveling along the path (see ‘Desired Timing Law’ of Fig. 1). This new parametrization can be simply obtained by forward integrating

$$\dot{s} = v^*(s) \left\| \frac{\partial \gamma(\mathbf{x}_c, s)}{\partial s} \right\|_2^{-1}, \quad s(t_0) = 0, \quad (34)$$

where  $v^*(s)$  is the desired timing law along the trajectory. To avoid numerical problems with previous equation, it is important to ensure along the whole trajectory  $\partial \gamma(\mathbf{x}_c, s) / \partial s \neq 0$ . In the following, we will assume that the unicycle moves at constant velocity (i.e.  $v^*(s) \equiv 1$ ) along the path  $\gamma(\mathbf{x}_c, s)$  with, thus,  $s(t)$  representing the *arc length* parametrization. On the other hand, for the planar quadrotor, we will not change the parametrization imposed by the B-Spline. Finally, by exploiting the flatness, a feedback control law able to ensure the tracking of the planned B-Spline trajectory with desired timing law can be also simply designed (see [30] for details).

## VI. RESULTS

This section is dedicated to evaluate the improvement in performance in estimating the state (which also include self-calibration and environment parameters for the unicycle case) via a EKF when maximizing the smallest eigenvalue of CG.

### A. Estimation of the unicycle state

Starting from a given initial configuration  $\mathbf{q}_0$  of the vehicle and an initial estimation  $\hat{\mathbf{q}}_0$  with uncertainty  $\mathbf{P}_0$ , we generated 200 random paths with the same energy  $E(t_0, t_f) = \bar{E} = 15$ , which were then optimized by using our methodology. In this section, the parameters  $r$  and  $b$ , as well as the landmarks parameters  $d$  and  $\alpha$ , are assumed constant and equal to their nominal values, i.e.,  $r = 0.1$  m,  $b = 0.25$  m,  $d = 4$  m,  $\alpha = 0$  rad. We assume a normally-distributed Gaussian output noise with zero mean and identity covariance matrix  $\mathbf{R} = \mathbf{I}$ . Finally, the initial estimation error covariance matrix is  $\mathbf{P}_0 \approx 0.16\mathbf{I}$ . Fig. 4(a) shows a selection of the 200 random paths, and Fig. 4(b) the resulting optimal ones (after the optimization has converged). We note that, due to the local nature of our method, the optimization converges to two distinct locally optimal paths depending on the particular initial guess. The smallest eigenvalue of the CG attains its largest value along the path on the left w.r.t. the initial forward direction of the vehicle. Nonetheless, both paths are locally optimal and reduce the estimation uncertainty w.r.t. the corresponding random one which served as initial guess.

We also note that the path followed by the vehicle will be slightly different from those showed in Fig. 4(b), which are obtained *offline* by relying on the initial estimated configuration of the vehicle  $\hat{\mathbf{q}}_0$ . Indeed, as already explained, during

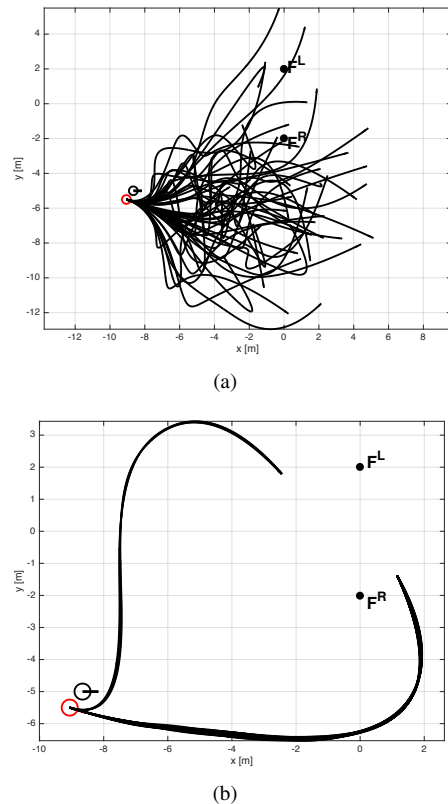


Fig. 4. Some of the 200 generated random path (a) and the optimal ones obtained after applying our optimization method (b).

motion the employed EKF improves the current estimation  $\hat{\mathbf{q}}(t)$ , making it possible to continuously refine (online) the previously optimized future path by exploiting the newly acquired information during motion.

We then compared the estimation performances of the EKF during the robot motion along each random path and its corresponding optimized one in order to show the expected benefits in terms of estimation performance. For the sake of completeness, we performed a comparison not only in terms of the maximum estimation uncertainty, i.e.,  $\lambda_{MAX}(\mathbf{P}(t_f)) \equiv \lambda_{min}^{-1}(\mathcal{G}_c(t_0, t_f))$  (which is the metric actually optimized by our algorithm), but also in terms of the average, the volume and the shape of the estimation uncertainty, i.e.,  $\text{tr}(\mathbf{P}(t_f))$ ,  $\det(\mathbf{P}(t_f))$ , and  $\kappa(\mathbf{P}(t_f)) = \frac{\lambda_{MAX}(\mathbf{P}(t_f))}{\lambda_{min}(\mathbf{P}(t_f))}$ , respectively.

The performance was also compared in terms of the individual components of the final estimation errors, and of its Root Mean Squared (RMS), as well as in terms of the time of convergence of the estimation errors, defined here as the time needed to attain the same amount of estimation error at the end of the path. For instance, assuming the EKF performs better in the optimal case (as expected), the convergence time is defined as the time needed by the optimal strategy to reach the estimation error norm attained at the end of the corresponding random path (which served as initial guess). This definition of the convergence rate makes it possible to assess whether, with our method, the same final estimation error (in the example, the one at the end of the random path) can be obtained in a shorter time. Similarly, the corresponding energy consumption,

TABLE I

MEAN VALUES OF THE MAXIMUM AND AVERAGE ESTIMATION UNCERTAINTY AS WELL AS OF THE SHAPE AND VOLUME OF THE ESTIMATION UNCERTAINTY WITH THEIR STANDARD DEVIATIONS. THE PERCENTAGE AVERAGE IMPROVEMENT IS ALSO REPORTED IN THE LAST COLUMN

|                 | Random Path                   | Optimal Path                  | % decrease  |
|-----------------|-------------------------------|-------------------------------|-------------|
| $\lambda_{MAX}$ | $4.97e^{-3} \pm 2.47e^{-3}$   | $8.78e^{-4} \pm 6.03e^{-5}$   | $\sim 82\%$ |
| trace           | $5.72e^{-3} \pm 2.37e^{-3}$   | $1.57e^{-3} \pm 1.15e^{-4}$   | $\sim 72\%$ |
| $\kappa$        | $5.14e^2 \pm 2.37e^{-3}$      | $2.07e^1 \pm 1.15e^{-4}$      | $\sim 96\%$ |
| det             | $7.35e^{-11} \pm 6.81e^{-11}$ | $2.46e^{-11} \pm 4.29e^{-12}$ | $\sim 67\%$ |

hereafter called energy of convergence, was also computed.

Statistical differences were evaluated using classic tools, after having tested the normality and homogeneity of variances assumption on samples (through Lilliefors' composite goodness-of-fit test and Levene's test, respectively). In particular, a non-parametric test was adopted for the comparison (Wilcoxon rank sum test) as, in our case, the normality hypothesis always failed on our samples. A significance level of 5% was assumed and p-values less than  $10^{-4}$  were considered to be equal to zero.

In Table I, the average values of  $\lambda_{MAX}(\mathbf{P}(t_f))$ ,  $\text{tr}(\mathbf{P}(t_f))$ ,  $\kappa(\mathbf{P}(t_f))$ , and  $\det(\mathbf{P}(t_f))$  with their corresponding standard deviations are reported for both the random and the optimal paths. The resulting p-values are all zero, showing that in terms of uncertainty, the proposed optimization method is able to find more informative paths according to all the considered metrics besides  $\lambda_{MAX}(\mathbf{P}(t_f))$  (which, again, is the quantity directly optimized by the proposed algorithm). Furthermore, Fig. 5(a)–Fig. 5(b) show the average absolute estimation error and estimation uncertainties (with associated standard deviations) for each state component obtained by the EKF at the end of both the random and the optimal paths. In Fig. 5(b), the RMS of the whole state estimation error is also reported. Wilcoxon rank sum test confirms that there is statistical difference in the average absolute estimation errors and estimation uncertainty for all the state variables as well as for the RMS. Indeed, one can verify that the average absolute estimation error of  $x$  and  $y$  is much less along the optimal paths than along the random ones (p-value: 0). However, the same does not hold for  $\theta$  where the average absolute estimation error is slightly smaller along the random paths than along the optimal ones (p-value:  $6.4e^{-4}$ ). For variable  $\theta$ , there is no significant difference in terms of uncertainty at the end of the random and optimal paths (see Fig. 5(a)). Indeed, the largest uncertainty at the end of the random paths (used as initial guess for our optimization method) is on the states  $x$  and  $y$  and, as a consequence, our method acts more on these states since it aims at reducing the *maximum estimation uncertainty* (which is the goal encoded in the CG). However, the RMS of the whole state estimation error at the end of the optimal path is, on average, two times smaller than that at the end of the random path (reduction of about 54%), thus showing that, overall, the estimation performance was significantly better in the optimized case.

Fig. 5(c) shows, for each state variable and for the RMS of the whole state estimation error, the average time of convergence obtained by the EKF at the end of both the random and the optimal paths. Their corresponding standard

deviations are also reported. Also in this case, the Wilcoxon rank sum test confirms that there is statistical difference in the average time of convergence for all the cases (the p-values are indeed zero). In other words, while for  $x$  and  $y$  along the optimal paths we have a smaller time of convergence than along the random ones, the same conclusion cannot be drawn for  $\theta$ . The reason of this result is exactly the same that for the average absolute estimation errors shown in Fig. 5(b). However, the time of convergence computed for the RMS of the whole state estimation error allows to conclude that our method reduces the overall time of convergence of about 19%. Finally, Fig. 5(d) shows the energy consumption along the random and optimal path in order to achieve the largest final estimation error between the optimal and the random path. Their corresponding standard deviations are also reported. Also in this case the Wilcoxon rank sum test confirms that there is statistical difference in the average energy of convergence for all the cases except for the estimation error of  $\theta$ . The p-values are indeed zero,  $1.86e^{-4}$  and zero for the estimate of  $x$ ,  $y$  and the RMS, respectively, while it is 0.1 for  $\theta$ . In other words, while for  $x$  and  $y$  we have a smaller energy of convergence along the optimal paths than along the random ones, the same conclusion cannot be drawn for  $\theta$  with the 5% of confidence. However, looking at the energy of convergence for the RMS for the whole state estimation error (see Fig. 5(d)), with the optimization strategy proposed in this paper we can achieve the same results obtained with a random path with, in average, a 25% of energy saving.

We finally note that the proposed framework has been implemented in the Matlab<sup>®</sup>/Simulink<sup>®</sup> environment and executed on a laptop with an Intel Core i7-6600U at 2.60 GHz. Each iteration of the optimization routine has taken about 18 ms (in the non-optimized code used for our simulations), thus showing the possibility of the proposed approach to run in real-time during the robot motion.

### B. Comparison of optimizing $\mathcal{G}_c(t_0, t_f)$ vs. $\mathcal{G}_o(t_0, t_f)$ for the unicycle case

In this section, starting from a given initial configuration  $\mathbf{q}_0$  of the vehicle and an initial estimate  $\hat{\mathbf{q}}_0$  of this configuration with uncertainty  $\mathbf{P}_0$ , the path that maximizes the smallest eigenvalue of  $\mathcal{G}_c(t_0, t_f)$  (CG-based optimization) is compared, in terms of the estimation performances obtained with the EKF, with the one that maximizes the smallest eigenvalue of  $\mathcal{G}_o(t_0, t_f)$  (OG-based optimization), which, as explained, is a more popular choice in the existing literature. The objective is to confirm what stated in Remark 1. We assume the same output noise as in previous section and all the parameters (i.e.,  $r$ ,  $b$ ,  $d$  and  $\alpha$ ) are again set to their nominal values.

Fig. 6 shows the results of the simulation. First of all, one can note how the CG-based optimal path is completely different from the OG-based one, showing that the two metrics are indeed encoding two different objectives. Moreover, the smallest eigenvalue of the inverse of the covariance matrix provided by the EKF at the end of the CG-based optimal path becomes six times the one at the end of the OG-based optimal path (bottom-right in Fig. 6), with a percentage increment of

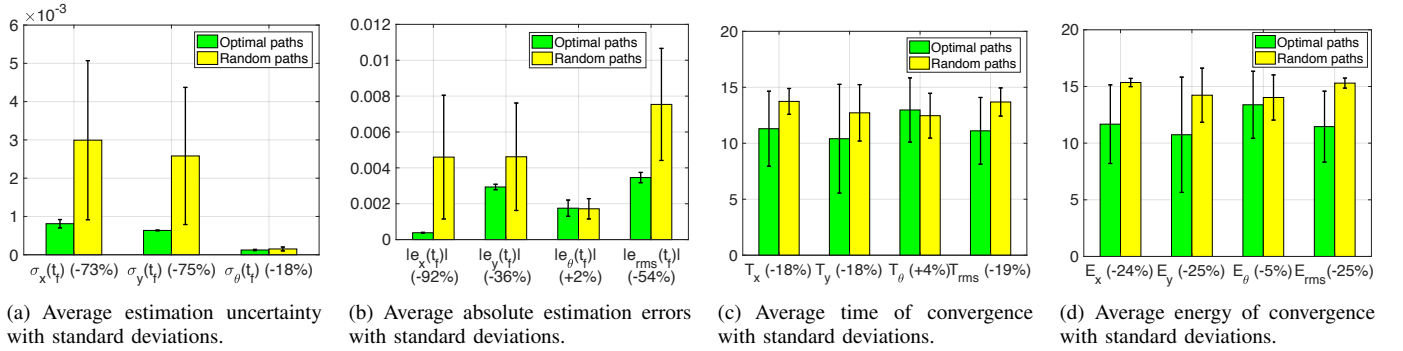


Fig. 5. Statistical differences in the average absolute estimation errors and convergence rate show that our method overall improves the estimation performances obtained with the EKF. The average estimation uncertainty is also reported to show the effectiveness of our method in increasing the overall collected information. The percentage average reductions/increments obtained with the optimal paths w.r.t. the random ones are also reported in each figure.

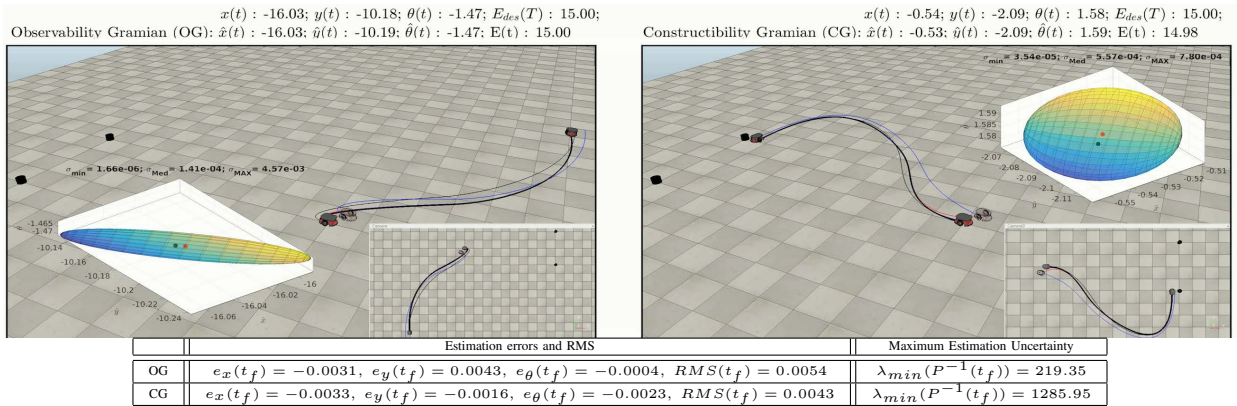


Fig. 6. Estimation performances obtained with the EKF along the CG-based (top-right) and the OG-based (top-left) optimal path (the shape of the final estimation uncertainty is also reported). The B-Splines is characterized by  $N = 6$  control points and degree  $\lambda = 4$ . The optimal paths from the estimated initial configuration are drawn in thin blue line, the optimal path from the real initial configuration are in thin black line, the real robot trajectory from the real robot configuration ( $\mathbf{q} = (-10, 0, 0)^T$  in black) are in thick black line, the estimated robot trajectory from the initial estimated robot configuration ( $\mathbf{q} = (-10.4, -0.5, -0.3)^T$  in red) are in thick red line.

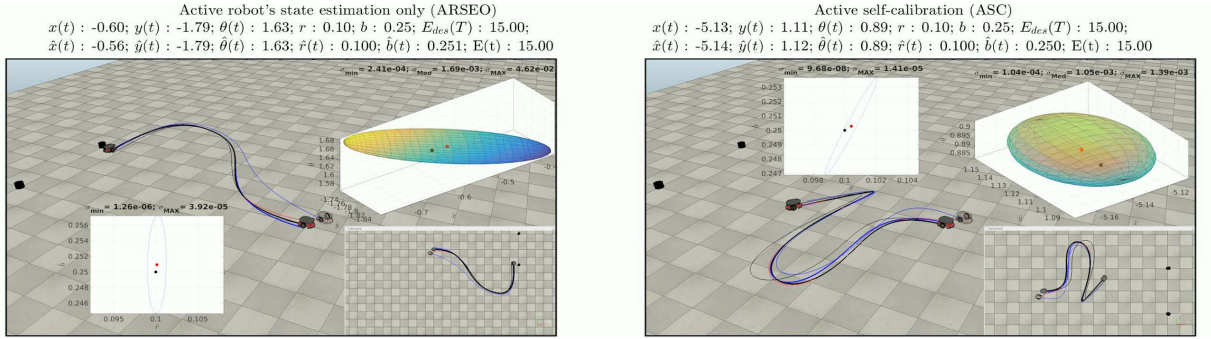
about 487%, thus confirming Remark 1. Of course, this has then an effect on the RMS of the estimation error that at the end of the CG-based optimal path is smaller than at the end of the OG-based optimal path as expected.

### C. Active sensing control for self-calibration for unicycle

In this section, we consider an instance of the Active Self-Calibration (ASC) problem for improving not only the estimation of the configuration of the vehicle  $\mathbf{q}(t) \in \mathbb{R}^3$ , but also the estimation of parameters  $r$  and  $b$ , of which only an initial (wrong) estimation is available, by maximizing the information collected along the path about the extended state  ${}^{sc}\mathbf{q}_e(t) = [\mathbf{q}(t)^T, r(t), b(t)]^T \in \mathbb{R}^5$  (see Fig. 3(a)). The objective is to maximize the smallest eigenvalue of the CG associated to the extended state  ${}^{sc}\mathbf{q}_e(t)$  ( $\mathcal{G}_c(t_0, t_f) \in \mathbb{R}^{5 \times 5}$ ). A parallel simulation (hereafter named Active Robot's State Estimation Only (ARSEO)) has also been performed where the objective is to maximize the amount of information concerning only the state of the vehicle  $\mathbf{q}(t)$ , i.e., without including the wheels' radius  $r$  and the axle length  $b$ . Of course, also during this simulation, the EKF is estimating the extended state  ${}^{sc}\mathbf{q}_e(t)$ , although not along a path optimal w.r.t. the estimation of the extended state. For both simulations, the

output noise was the same as in previous sections. A video of this simulation can be found in the multimedia attachment.

Fig. 7 shows the final results of the simulations, also available in the attached multimedia video. The optimal path for ASC is significantly different from the one for ARSEO. The collected information along the two paths until about 9 s of simulation is almost the same. In particular, in terms of the collected information (bottom-right in Fig. 7), at the beginning and until 2 s of simulation, the ASC outperforms the ARSEO, then from 2 s to 6 s the ARSEO outperforms the ASC and finally from 6 s to 9 s of simulation is again the ASC that outperforms the ARSEO. This behavior is due to the uncertainty about the calibration parameters that acts on the EKF as a sort of actuation/process noise which degrades the information collected through the outputs. After 9 s of simulation, the ASC definitely outperforms the ARSEO and indeed the maximum estimation uncertainty at the end of the ASC optimal path is almost 35 times less than the one at the end of the ARSEO optimal path, with a percentage decrement of about 2784%. Indeed, the RMS of the whole state estimation error is smaller along the ASR optimal path (bottom-left plot in Fig. 7). Finally, the same RMS of the state estimation error obtained at the end of the ARSEO optimal



|       | Estimation errors and RMS  | Maximum Estimation Uncertainty                 |
|-------|--|--|
| ARSEO | $e_x(t_f) = -0.0339$ , $e_y(t_f) = 0.0014$ , $e_\theta(t_f) = 0.0018$ , $e_r(t_f) = -0.0001$ , $e_b(t_f) = -0.0009$ , $RMS(t_f) = 0.0340$  | $\lambda_{min}(\mathbf{P}^{-1}(t_f)) = 79.99$  |
| ASC   | $e_x(t_f) = 0.0050$ , $e_y(t_f) = -0.0079$ , $e_\theta(t_f) = -0.0068$ , $e_r(t_f) = -0.0004$ , $e_b(t_f) = -0.0003$ , $RMS(t_f) = 0.0116$ | $\lambda_{min}(\mathbf{P}^{-1}(t_f)) = 178.79$ |

Fig. 7. Estimation performances obtained with an EKF along the optimal paths for ASC (top-right) and the ARSEO (top-left). The shape of the final estimation uncertainty for state  $x$ ,  $y$  and  $\theta$  and parameters  $r$  and  $b$ , separately, are also reported. The B-Splines is characterized by  $N = 6$  control points and degree  $\lambda = 4$ . The optimal paths from the estimated initial configuration are drawn in thin blue line, the optimal path from the real initial configuration are in thin black line, the real robot trajectory from the real robot configuration ( $\mathbf{q} = (-10, 0, 0, 0.1, 0.25)^T$  in black) are in thick black line, the estimated robot trajectory from the initial estimated robot configuration ( $\hat{\mathbf{q}} = (-10.4, -0.5, -0.3, 0.11, 0.3)^T$  in red) are in thick red line.

path can be achieved also along the ASC optimal one with about a 78% of energy saving.

#### D. Active sensing control for scene reconstruction for unicycle

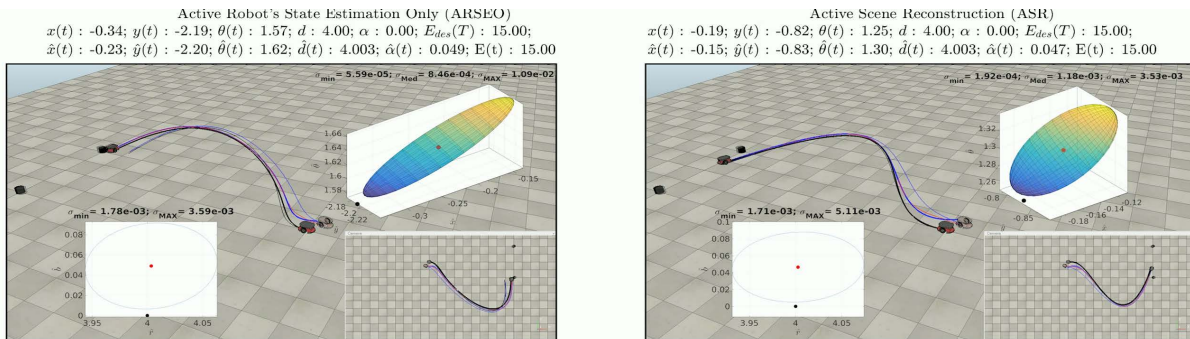
In this last case, the objective is to apply our optimization method to the problem of simultaneous scene reconstruction and unicycle state estimation (hereafter named Active Scene Reconstruction (ASR)). We hence assume that the landmarks represent the doorposts of a door through which the robot has to pass, perpendicularly to the segment between the landmarks. We hence consider the problem of estimating the position of the middle of the door and the orientation of the door, i.e., the extended state  ${}^{er}\mathbf{q}_e(t) = [\mathbf{q}(t)^T, d(t), \alpha(t)]^T$  (see Fig. 3(a)), as accurately as possible. The model of the vehicle (parameters  $r$  and  $b$ ) are assumed perfectly known and equal to their nominal values. A video of this simulation can be found in the multimedia attachment of this paper. Also in this case, the ARSEO solution has also been obtained for comparison. For both simulations, the output noise was the same as in previous sections. Fig. 8 shows the results of the simulations. The optimal path for ASR is not very different from the optimal path for ARSEO especially until 9 s of simulations. The main differences can be remarked in the final part of the optimal paths where the estimation uncertainty is more than two time less along the ASR optimal path, with a percentage decrement w.r.t. the ARSEO optimal path of about 107%. Indeed, the RMS of the whole state estimation error is in average smaller along the ASR optimal path (bottom-left plot in Fig. 8). Moreover, the same RMS of the state estimation error obtained at the end of the ARSEO optimal path can be achieved also along the ASR optimal one with about a 8.4% of energy saving.

#### E. Estimation performances of the 2D UAV state

In this section, we consider the case of a planar UAV that needs to estimate its state by exploiting two distance measurements from two landmarks whose positions are  $\mathbf{F}^R = [d, \bar{y}, 0]^T$  and  $\mathbf{F}^L = [-d, \bar{y}, 0]^T$ , (refer to (33) for the

meaning of  $\bar{y}$ ). Contrarily to the unicycle case, in order to show how further requirements can be easily included in our machinery, we imposed a final desired configuration for the UAV: the robot needs to reach the middle point between the markers on the plane of motion and remain there in hovering. Furthermore, we also considered a measurement noise that increases with the distance from the landmarks for representing a more realistic setting in which the quality of a measurement degrades with its range from the measured point. This has been obtained by weighting the measurement covariance matrix  $\mathbf{R}^{-1}$  by a weight matrix  $\mathbf{W}$  so that  $\mathbf{R}_W^{-1} = \mathbf{W}^T \mathbf{R}^{-1} \mathbf{W}$  where  $\mathbf{W} = \text{diag}(w_L, w_R)$ . A possible shape for weight  $w_{\{L,R\}}$  is shown in Fig. 9. As a consequence, if the robot moves further than the distance  $D_2$  from the marker, then the measurement covariance matrix becomes infinity and the measure from that marker is no longer available. In our simulations,  $D_1 = 6$  m and  $D_2 = 7$  m.

The results of our optimization strategy are reported in Fig. 10, also available in the attached multimedia video. A comparison with another trajectory, along which a small amount of information is collected, is also reported in order to show the estimation improvement obtained. The non-optimal path is planned offline and then executed without any adjustment during motion. The optimal path is also planned offline but then, during motion, it is refined *online* as usual by taking into account the current state estimation. The maximum estimation uncertainty at the end of the optimal path is about 5 times less than at the end of the non optimal path. Our method also provides a significant improvement of the overall estimation performance, in particular in terms of convergence rate (see Fig. 10 bottom-right). The RMS at the end of the optimal path is 98% less than at the end of the non optimal path. Moreover, the same RMS of the state estimation error obtained at the end of the non optimal path can be achieved also along the optimal one with about a 77% of energy saving and with about 76% less time. Finally, one step of the proposed algorithm for the planar UAV, implemented with non optimized Matlab<sup>®</sup> code, has taken about 29 ms on an



|       | Estimation errors and RMS   | Maximum Estimation Uncertainty                 |
|-------|---|--|
| ARSEO | $e_x(t_f) = -0.1059$ , $e_y(t_f) = 0.0116$ , $e_\theta(t_f) = -0.0487$ , $e_d(t_f) = -0.0035$ , $e_\alpha(t_f) = -0.0491$ , $RMS(t_f) = 0.1270$ | $\lambda_{min}(\mathbf{P}^{-1}(t_f)) = 79.99$  |
| ASR   | $e_x(t_f) = -0.0391$ , $e_y(t_f) = 0.0072$ , $e_\theta(t_f) = -0.0462$ , $e_d(t_f) = -0.0026$ , $e_\alpha(t_f) = -0.0465$ , $RMS(t_f) = 0.0767$ | $\lambda_{min}(\mathbf{P}^{-1}(t_f)) = 178.79$ |

Fig. 8. Estimation performances obtained with an EKF along the optimal paths for ASR (top-right) and the ARSEO (top-left). The B-Splines is characterized by  $N = 6$  control points and degree  $\lambda = 4$ . The shape of the final estimation uncertainty for state  $x$ ,  $y$  and  $\theta$  and parameters  $d$  and  $\alpha$ , separately, are also reported. The optimal paths from the estimated initial configuration are drawn in thin blue line, the optimal path from the real initial configuration in thin black line, the real robot trajectory from the real robot configuration ( $\mathbf{q} = (-10, 0, 0, 4, 0)^T$  in black) are in thick black line, the estimated robot trajectory from the initial estimated robot configuration ( $\hat{\mathbf{q}} = (-10.4, -0.5, -0.3, 0.5, \pi/12)^T$  in red) are in thick red line.

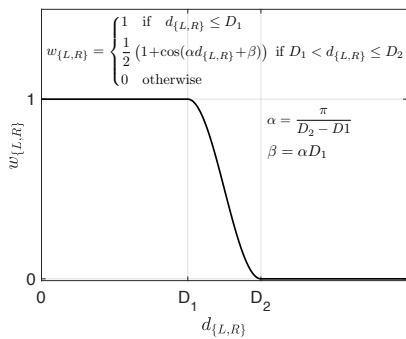


Fig. 9. A representative shape of the weight adopted to increase the measurement noise.

Intel Core i7-6600U running at 2.60 GHz.

## VII. CONCLUSIONS AND FUTURE WORKS

In this paper, the problem of active sensing control for non-linear differentially flat systems has been tackled by considering the smallest eigenvalue of the CG as metric for quantifying the acquired information during motion. The computational complexity of the optimization problem has been reduced by parametrizing the flat outputs with a family of B-Splines. By applying our strategy to a unicycle vehicle and a planar quadrotor, we showed that an improved estimation of the state can be consistently achieved in a large number of tested conditions, thus demonstrating the effectiveness of the proposed framework. Future works will consist in applying our methodology to more complex robotic platforms such as a complete 3D quadrotor UAV and multi-robot systems for localization purposes. Moreover, we are also interested in including the actuation/process noise in our optimization problem, which can be quite relevant when dealing with uncertain robotics systems such as UAVs (for which the aerodynamics can be hardly modeled accurately). In [31], we have already proposed a possible solution consisting in minimizing the largest eigenvalue of the covariance matrix directly, as the solution of the Riccati differential equation.

However, this solution is only valid if an EKF is used as an observer and hence limits its application domain. A better possibility is probably to leverage tools coming from the nonlinear reachability analysis instead in order to quantify how much the actuation/process noise degrades the collected information during motion. Another possibility when in presence of parametric uncertainty in the robot model is to combine the CG with a *parameter sensitivity metric*, such as the one proposed in [30], for taking into account at the same time state observability and robustness against model uncertainty (which could be seen as a form of process noise).

## REFERENCES

- [1] K. P. Kording and D. M. Wolpert, "Bayesian decision theory in sensorimotor control," *Trends in Cognitive Sciences*, vol. 10, no. 7, pp. 319 – 326, 2006, special issue: Probabilistic models of cognition.
- [2] S.-H. Yeo, D. W. Franklin, and D. M. Wolpert, "When Optimal Feedback Control Is Not Enough: Feedforward Strategies Are Required for Optimal Control with Active Sensing," *PLoS computational biology*, vol. 12, no. 12, p. e1005190, 2016.
- [3] K. Hausman, J. Preiss, G. S. Sukhatme, and S. Weiss, "Observability-aware trajectory optimization for self-calibration with application to uavs," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1770–1777, July 2017.
- [4] T. Hamel and C. Samson, "Position estimation from direction or range measurements," *Automatica*, vol. 82, no. Sup. C, pp. 137 – 144, 2017.
- [5] R. Hermann and A. J. Krener, "Nonlinear controllability and observability," *IEEE Transactions on automatic control*, vol. 22, no. 5, pp. 728–740, 1977.
- [6] G. Besançon, *Nonlinear observers and applications*. Springer, 2007, vol. 363.
- [7] F. A. Belo, P. Salaris, D. Fontanelli, and A. Bicchi, "A complete observability analysis of the planar bearing localization and mapping for visual servoing with known camera velocities," *International Journal of Advanced Robotic Systems*, vol. 10, 2013.
- [8] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, no. 2, pp. 177–196, Feb 2018.
- [9] A. J. Krener and K. Ide, "Measures of unobservability," in *Proceedings of the 48th IEEE Conference on Decision and Control, held jointly with the 28th Chinese Control Conference. CDC/CCC*, Dec 2009, pp. 6401–6406.
- [10] R. W. Brockett, *Control Theory and Singular Riemannian Geometry*. New York, NY: Springer New York, 1982, pp. 11–27.
- [11] B. T. Hinson and K. A. Morgansen, "Observability optimization for the nonholonomic integrator," in *American Control Conference (ACC)*, June 2013, pp. 4257–4262.

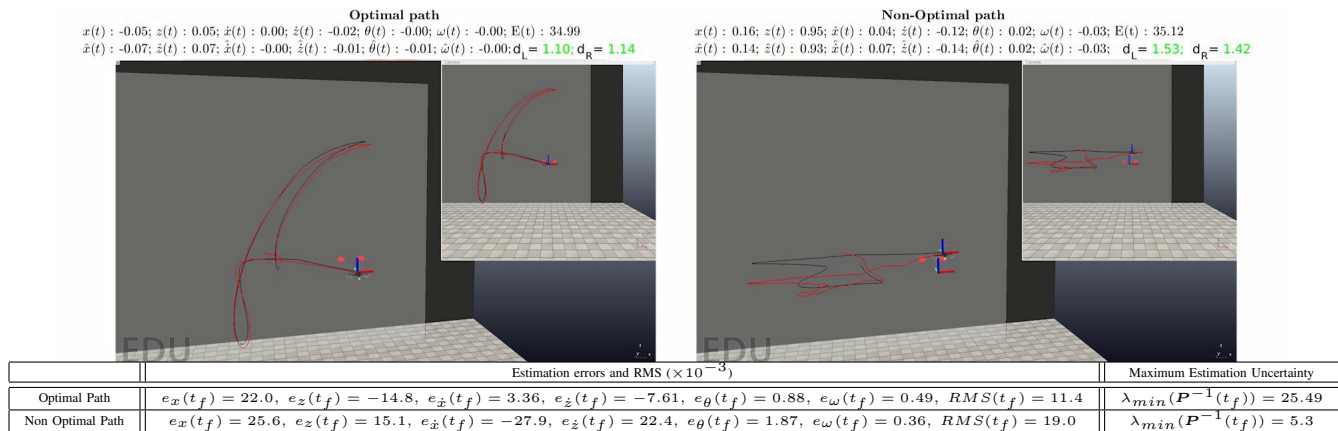


Fig. 10. Estimation performances obtained with an EKF along the optimal path (top-right) and a random path (top-left). The B-Splines is characterized by  $N = 16$  control points and degree  $\lambda = 8$ . For both cases, the energy spent is  $\bar{E} = 35$ . The planar quadrotor has a mass  $m = 0.4 \text{ kg}$ ,  $I = 0.1 \text{ kg m}^2$ , and  $0.6 \text{ m}$  of diameter. The optimal paths from the estimated initial configuration is drawn in thin black line, the real robot trajectory from the real robot configuration ( $\mathbf{q} = (-3.5, 0.5, 0, 0, 0)^T$ ) are in black line, the estimated robot trajectory from the initial estimated robot configuration ( $\hat{\mathbf{q}} = (-4.2, 1.25, 0.098, -0.25, 0.12, -0.15)^T$ ) are in red line. The initial state covariance matrix is  $\mathbf{P}_0 \approx \text{diag}(1, 1, 0.25, 0.25, 0.15)$ . Notice that, the maximum estimation uncertainty at the end of the optimal path is about 5 times less than at the end of the non optimal path.

- [12] B. T. Hinson, M. K. Binder, and K. A. Morgansen, "Path planning to optimize observability in a planar uniform flow field," in *American Control Conference (ACC)*, June 2013, pp. 1392–1399.
- [13] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of nonlinear systems: Introductory theory and examples," *International Journal of Control*, vol. 61, no. 6, pp. 1327–1361, 1995.
- [14] P. Salaris, R. Spica, P. Robuffo Giordano, and P. Rives, "Online Optimal Active Sensing Control," in *International Conference on Robotics and Automation (ICRA)*, Singapore, Singapore, May 2017.
- [15] P. J. Antsaklis and A. N. Michel, *Linear systems*. Springer Science & Business Media, 2006.
- [16] F. Lorussi, A. Marigo, and A. Bicchi, "Optimal exploratory paths for a mobile rover," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2001, pp. 2078–2083.
- [17] A. Gelb, *Applied Optimal Estimation*, A. Gelb, J. F. Kasper, R. A. Nash, C. F. Price, and A. A. Sutherland, Eds. Cambridge, MA: MIT Press, 1974.
- [18] M. Rafieisakhaei, S. Chakravorty, and P. R. Kumar, "On the use of the observability gramian for partially observed robotic path planning problems," in *IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec 2017, pp. 1523–1528.
- [19] J. H. Taylor, "The cramer-rao estimation error lower bound computation for deterministic nonlinear systems," *IEEE Transactions on automatic control*, vol. 24, no. 2, pp. 343–344, 1979.
- [20] A. Bryson and Y. Ho, *Applied optimal control*. Wiley New York, 1975.
- [21] T. Hamel and C. Samson, "Riccati observers for position and velocity bias estimation from direction measurements," in *IEEE 55th Conference on Decision and Control (CDC)*, Dec 2016, pp. 2047–2053.
- [22] R. Spica and P. Robuffo Giordano, "Active decentralized scale estimation for bearing-based localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2016.
- [23] F. Pukelsheim, *Optimal design of experiments*. siam, 1993, vol. 50.
- [24] D. E. Chang and Y. Eun, "Construction of an atlas for global flatness-based parameterization and dynamic feedback linearization of quadcopter dynamics," in *IEEE 53rd Annual Conference on Decision and Control (CDC)*. IEEE, 2014, pp. 686–691.
- [25] Y. J. Kaminski, J. Levine, and F. Ollivier, "Intrinsic and Apparent Singularities in Flat Differential Systems," *arXiv.org*, jan 2017.
- [26] L. Biagiotti and C. Melchiorri, *Trajectory planning for automatic machines and robots*. Springer Science & Business Media, 2008.
- [27] B. Siciliano and J. J. E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Fifth International Conference on Advanced Robotics (ICAR): 'Robots in Unstructured Environments'*, June 1991, pp. 1211–1216 vol.2.
- [28] Y. Yang and O. Brock, "Elastic roadmaps—motion generation for autonomous mobile manipulation," *Autonomous Robots*, vol. 28, no. 1, p. 113, Sep 2009.
- [29] C. Masone, P. Robuffo Giordano, H. H. Bühlhoff, and A. Franchi, "Semi-autonomous trajectory generation for mobile robots with integral haptic

shared control," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 6468–6475.

- [30] P. Robuffo Giordano, Q. Delamare, and A. Franchi, "Trajectory generation for minimum closed-loop state sensitivity," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 286–293.

- [31] M. Cognetti, P. Salaris, and P. Robuffo Giordano, "Optimal active sensing with process and measurement noise," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 2118–2125.



**Paolo Salaris** received the M.Sc. degree in Electrical Engineering in 2007 and his PhD in Automation and Bioengineering in 2011, both from the university of Pisa. He has been Visiting Scholar at Beckman Institute for Advanced Science and Technology, University of Illinois, Urbana-Champaign in 2009. He has been a PostDoc at the Research Center "E.Piaggio" from March 2011 to January 2014 and at LAAS-CNRS in Toulouse from February 2014 to July 2015. Currently is a permanent research (CRCN) in the Chorale team at Inria Sophia Antipolis Méditerranée.

His main research interests are optimal motion planning and control, active sensing, optimal sensors placements and optimal estimation, multi-robot and nonholonomic systems.



**Marco Cognetti** (S'14, M'17) received the Ph.D. degree in Control Engineering from University of Rome "La Sapienza", Italy, in 2016. In 2011, he was Visiting Scholar at the Human-Robot Interaction group at the Max Planck Institute for Biological Cybernetics (MPI-KYB), Tübingen, Germany. In 2015, he was Visiting Scholar at the Personal Robotics Lab of The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. He is currently a postdoctoral researcher in the Rainbow group at Inria and Inria in Rennes, France. His research interests

include active perception, state estimation, multi-robot control/localization and whole-body motion planning and control for humanoid robots.





**Riccardo Spica** (M'12) received his M.Sc. degree in Electronic Engineering from the University of Rome "La Sapienza" in 2012. He worked first as a Master's Student and later as a Graduate Research Assistant at the Max Planck Institute for Biological Cybernetics in Tübingen, Germany for one year between 2012 and 2013. He received his Ph.D. in Signal Processing from the University of Rennes, France in 2015. He also spent one year as a postdoc in the Lagadic team with CNRS at IRISA Rennes before moving to the University of Stanford where he joined the Multi-

Robot Systems Lab as a postdoctoral scholar. His research deals with vision-based estimation and control strategies for manipulators and mobile robots.



**Paolo Robuffo Giordano** (M'08-SM'16) received his M.Sc. degree in Computer Science Engineering in 2001, and his Ph.D. degree in Systems Engineering in 2008, both from the University of Rome "La Sapienza". In 2007 and 2008 he spent one year as a PostDoc at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR), and from 2008 to 2012 he was Senior Research Scientist at the Max Planck Institute for Biological Cybernetics in Tübingen, Germany. He is currently a senior CNRS researcher head of the Rainbow group

at Irisa and Inria in Rennes, France.