Optimal Active Sensing with Process and Measurement Noise

Marco Cognetti[†], Paolo Salaris[*] and Paolo Robuffo Giordano[†]

Abstract— The goal of this paper is to increase the estimation performance of an Extended Kalman Filter for a nonlinear differentially flat system by planning trajectories able to maximize the amount of information gathered by onboard sensors in presence of both process and measurement noises. In a previous work, we presented an online gradient descent method for planning optimal trajectories along which the smallest eigenvalue of the Observability Gramian (OG) is maximized. As the smallest eigenvalue of the OG is inversely proportional to the maximum estimation uncertainty, its maximization reduces the maximum estimation uncertainty of any estimation algorithm employed during motion. However, the OG does not consider the process noise that, instead, in several applications is far from being negligible. For this reason, this paper proposes a novel solution able to cope with non-negligible process noise: this is achieved by minimizing the largest eigenvalue of the a posteriori covariance matrix obtained by solving the Continuous Riccati Equation as a measure of the total available information. This minimization is expected to maximize the information gathered by the outputs while, at the same time, limiting as much as possible the negative effects of the process noise. We apply our method to a unicycle robot. The comparison between the novel method and the one of our previous work (which did not consider process noise) shows significant improvements in the obtained estimation accuracy.

I. INTRODUCTION

Evidence from neuroscience and biology shows that the humans' brain dedicates a large effort for reducing the negative effects of noise affecting both the motor and the sensing apparatus [1]. Indeed, humans take into account the quality of sensory feedback when planning their future actions. This is achieved by coupling feedforward strategies, aimed at reducing the effects of sensing noise, with feedback actions, mainly intended to accomplish given motor tasks and reduce the effects of control uncertainties [2].

Robots, analogously to humans, need to localize themselves w.r.t. the environment in order to move safely in unstructured environments while, at the same time, a map of the surrounding is built. This capability is highly influenced by the quality and amount of sensor information, especially in case of limited sensing capabilities and/or low cost sensors. Moreover, if one wishes to include self-calibration states in the estimation, the dimensionality of the state vector clearly increases, while the number of measurements remains unchanged [3]. As a consequence, it is important to determine inputs/trajectories that render all states and calibration parameters as observable as possible by maximizing the

information gathered along the trajectory. The problem of optimal information gathering has been indeed studied in the literature in a variety of contexts such as: (i) optimal sensor placements, c.f. [4] for optimal placing the sensors on a wearable sensing glove or [5] for studying the gyroscopic sensing distribution of an insect wings; (ii) localization and exploration for mobile robots, c.f. [6] that proposes a complete observability analysis of the planar bearing-only localisation and mapping problem.

For these reasons, understanding whether the problem of estimating the state of the robot and the environment from knowledge of the inputs and the outputs of the system admits a solution is of fundamental importance. This is referred to as observability problem as detailed in [7], [8]. The problem is even harder to solve if the system is non-linear since, in this case, the observability property depends not only on the state but also on the inputs of the system. In some cases, one may also find some *singular inputs*, i.e. inputs that do not allow the reconstruction of the whole state from measured output (c.f., [6]). In order to avoid this situation, some authors have proposed intelligent control strategies able to maximize the amount of information gathered by sensors or, in other words, to maximise the "distance" from the singular inputs/trajectories. This problem is known in the literature as *active perception*, *active sensing control* or *optimal information gathering*. One example in this context is given by [9], where the authors maximize the minimum eigenvalue of the Observability Gramian matrix in order to find optimal observability trajectories for first-order nonholonomic systems. Another example can be found in [10], where the condition number of the Observability Gramian matrix is used for finding the optimal observability trajectory for an aerial vehicle.

While several works take into account the measurement noise in their formulations, quite a few explicitly consider the actuation noise, which, on the other hand, is far from being negligible for several robotic applications (a prominent example being aerial vehicles). One exception is represented by the POMPD-based methods such as [11], [12] where the authors propose a motion planning algorithm that minimises the motion and the sensing uncertainties in environments with static obstacles. Another approach is the one in [13], where an optimal control algorithm selects trajectories that optimises a cost function composed by the state tracking goals, the control effort, and the sensitivity of planned motion to variations in model parameters, including actuation noise. However, these approaches are more devoted to an offline implementation (also because of their computational complexity), while the goal of this paper is to propose an

[†] are with CNRS, Univ Rennes, Inria, IRISA, Rennes, France, e-mail: {marco.cognetti,prg}@irisa.fr
[*] is with INRIA Sophia-Antipolis Méditerranée, 2004 Route des Lucioles, 06902 Sophia Antipolis, France, e-mail: paolo.salaris@inria.fr.

online solution to the active sensing problem.

The rest of the paper is organised as follows. In Section II, some preliminary concepts are introduced while in Section III the optimal control problem is formalized. In Section IV, a solution that combines an estimator (i.e., an Extended Kalman Filter) and a gradient-descend optimisation strategy is proposed while in Section V we test our method on a unicycle robot. Finally, some conclusions and future works are discussed in Section VI.

II. PRELIMINARIES

Let us consider a generic nonlinear dynamics

$$\dot{\boldsymbol{q}}(t) = \boldsymbol{f}(\boldsymbol{q}(t), \boldsymbol{u}(t) + \boldsymbol{w}), \quad \boldsymbol{q}(t_0) = \boldsymbol{q_0} \qquad (1)$$

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

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

The onboard sensors can typically provide only partial information about the state of the robot, which may include self-calibration parameters and features of the surrounding world. Therefore, an estimation algorithm is needed for recovering online those state variables not directly available from the raw sensor readings. Its performance, that is accuracy, precision and convergence rate, depends not only on the quality and amount of information about the unknown state variables gathered by the onboard sensors but also on the level of actuation noise. For nonlinear systems, both aspects depend on the inputs and hence the trajectories followed by the system. The goal of this paper is hence to improve the performance of the employed estimation algorithm by determining *online* the control inputs that maximize the information gathered by the onboard sensors and, at the same time, limit the negative effects of actuation noise (which basically degrades the above-mentioned information).

In our previous work on this subject [14], we considered the particular case where $\boldsymbol{w} = \boldsymbol{0}$, i.e. without actuation noise. As a consequence, the only objective was to determine the control inputs that maximize the amount of information gathered by the onboard sensors along the trajectory followed during motion. In order to obtain a measure of such information [15], we considered a well-known observability criterion for (1)–(2), related to the concept of local *indistinguishable* states [8], [16], [14], i.e. the *Observability Gramian* (OG) $\boldsymbol{\mathcal{G}}_o(t_0, t_f) \in \mathbb{R}^{n \times n}$,

$$\boldsymbol{\mathcal{G}}_o(t_0, t_f) \triangleq \int_{t_0}^{t_f} \boldsymbol{S}(\tau, t_0)^T \boldsymbol{H}(\tau)^T \boldsymbol{W}(\tau) \boldsymbol{H}(\tau) \boldsymbol{S}(\tau, t_0) \, \mathrm{d}\tau \qquad (3)$$

where $t_f > t_0$, $\boldsymbol{H}(\tau) = \frac{\partial \boldsymbol{h}(\boldsymbol{q}(\tau))}{\partial \boldsymbol{q}(\tau)}$, and $\boldsymbol{W}(\tau) \in \mathbb{R}^{p \times p}$ is a symmetric positive definite weight matrix (a design parameter) that usually (and also in this paper) is taken equal to $\boldsymbol{R}^{-1}$ in order to weight the outputs w.r.t. the level of noise.

Matrix $\boldsymbol{S}(t, t_0) \in \mathbb{R}^{n \times n}$, also known as *sensitivity matrix*, is given as $\boldsymbol{S}(t, t_0) = \frac{\partial \boldsymbol{q}(t)}{\partial \boldsymbol{q}_o}$ and it is easy to check that it verifies the following differential equation

$$\dot{\boldsymbol{S}}(t, t_0) = \frac{\partial \boldsymbol{f}(\boldsymbol{q}(t), \boldsymbol{u}(t))}{\partial \boldsymbol{q}(t)} \boldsymbol{S}(t, t_0), \qquad \boldsymbol{S}(t_0, t_0) = \boldsymbol{I}. \quad (4)$$

In [14] the smallest eigenvalue of the OG was used as performance index in order to determine the inputs and hence the trajectory over a future time horizon along which the information content is maximized. Moreover, as the inverse of the smallest eigenvalue of the OG is proportional to the maximum estimation uncertainty, its *maximisation* was also expected to *minimize* the maximum estimation uncertainty of any estimation algorithm that could be used during motion [9]. The choice of adopting the smallest eigenvalue (E-Optimality) index hence guarantees optimization of the worst-case performance. In particular, by using an Extended Kalman Filter (EKF) as estimation algorithm, in [14] we verified through simulations that the ellipsoid associated to the covariance matrix obtained by solving the Continuous Riccati Equation (CRE) is much less elongated along the eigenvector associated to the largest eigenvalue hence obtaining a smaller and a more uniform estimation uncertainty. We also obtained an increment of the convergence rate and the precision of the EKF. Of course, in case of actuation noise, which is far from negligible in several robotic applications (e.g. UAVs), the OG is not able to measure how the actuation noise degrades the amount of information gathered by the noisy sensory feedback.

The main idea of this paper is hence to use directly the CRE and determine the inputs that maximize the smallest eigenvalue of the inverse of the covariance matrix explicitly evaluated in presence of process noise. By doing this, the maximum estimation uncertainty will reduce not only because along the planned trajectory the gathered information from noisy sensory feedback is maximized, but also because, at the same time, the degrading effects of the process noise on such information are minimized.

III. PROBLEM FORMULATION

We now detail the optimal sensing problem addressed in this paper: consider the particular class of nonlinear dynamics (1–2) such that $\boldsymbol{f}(\boldsymbol{q}, \boldsymbol{0}) = \boldsymbol{0}$ (i.e. without drift), a time window $[t_0, t_f]$, $t_f > t_0$, and a EKF built on system (1–2) for recovering an estimation $\hat{\boldsymbol{q}}(t)$ of the true (but unknown) state $\boldsymbol{q}(t)$ during motion. The goal of the paper is to propose 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*

$$\boldsymbol{u}^*(t) = \arg\max_{\boldsymbol{u}} \ \lambda_{min} \left(\boldsymbol{P}^{-1}(t_0, t_f) \right), \qquad (5)$$

$$s.t.$$

$$E(t_0, t_f) = \int_{t_0}^{t_f} \sqrt{\boldsymbol{u}(\tau)^T \boldsymbol{M} \boldsymbol{u}(\tau)} \, \mathrm{d}\tau = \bar{E} \qquad (6)$$

where $\boldsymbol{M}$ is a constant weight matrix, $\bar{E}$ is a constant design parameter $\lambda_{min}(\cdot)$ is an operator that extracts the smallest eigenvalue of the matrix argument while $\boldsymbol{P}(t_0, t_f)$ is the covariance matrix of the EKF filter, such that $\boldsymbol{P}^{-1}(t_0, t_f)$ is solution of

$$\dot{\boldsymbol{P}}^{-1}(\tau) = -\boldsymbol{P}^{-1}(\tau)\boldsymbol{A}(\tau) - \boldsymbol{A}^T(\tau)\boldsymbol{P}^{-1}(\tau) +$$
$$+ \boldsymbol{H}^T(\tau)\boldsymbol{R}^{-1}\boldsymbol{H}(\tau) - \boldsymbol{P}^{-1}(\tau)\boldsymbol{B}(\tau)\boldsymbol{V}\boldsymbol{B}^T(\tau)\boldsymbol{P}^{-1}(\tau),$$
$$\boldsymbol{P}^{-1}(t_0) = \boldsymbol{P}_0^{-1} \quad (7)$$

$\tau \in [t_0, t_f]$ and initial condition $\boldsymbol{P}_0^{-1}$ representing the a priori information available at time t_0 about the initial $\boldsymbol{q}(t_0)$. Matrices $\boldsymbol{A}(\tau) = \frac{\partial \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{u})}{\partial \boldsymbol{q}}\big|_{\hat{\boldsymbol{q}}, \boldsymbol{u}}$, $\boldsymbol{B}(\tau) = \frac{\partial \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{u})}{\partial \boldsymbol{u}}\big|_{\hat{\boldsymbol{q}}, \boldsymbol{u}}$ and $\boldsymbol{C}(\tau) = \frac{\partial \boldsymbol{h}(\boldsymbol{q})}{\partial \boldsymbol{q}}\big|_{\hat{\boldsymbol{q}}, \boldsymbol{u}}$, are the dynamics, input and output matrices of the linear time-varying system obtained by linearization of (1)-(2) around the estimated trajectory $\hat{\boldsymbol{q}}(\tau)$:

$$\dot{\hat{\boldsymbol{q}}}(\tau) = \boldsymbol{A}(\tau)\hat{\boldsymbol{q}}(\tau) + \boldsymbol{B}(\tau)(\boldsymbol{u}(\tau) + \boldsymbol{w}(\tau))$$
$$\hat{\boldsymbol{z}}(\tau) = \boldsymbol{C}(\tau)\hat{\boldsymbol{q}}(\tau) + \boldsymbol{\nu} \quad (8)$$

The scalar quantity $E(t_0, t_f)$ is meant to represent the "control effort" (or energy) needed by the robot for moving along the trajectory from t_0 to t_f. Constraint (6) over the time horizon T is introduced for ensuring well-posedness of the optimization problem. Indeed, in general, as for $\lambda_{min}(\boldsymbol{\mathcal{G}}_o(t_0, t_f))$, also $\lambda_{min}(\boldsymbol{P}^{-1}(t_0, t_f))$ could be unbounded from above w.r.t. the control effort E, as the most likely optimal solution would consist in increasing the observation time and the state indefinitely. Notice that the final time t_f is hence not treated as a fixed parameter but, rather, as the time needed for "spending" the whole available energy $\bar{E}$ during the robot motion and depends also on the choice of the timing law along the trajectory.

The need for an *online* solution is motivated by the fact that, as for the OG in [14], also the solution $\boldsymbol{P}^{-1}$ of (7) depends on the state trajectory followed by the robot, which is not assumed available. However, during the robot motion, starting from an initial estimation $\hat{\boldsymbol{q}}(t_0)$, the EKF improves the current estimation $\hat{\boldsymbol{q}}(t)$ of the true state $\boldsymbol{q}(t)$, with $\hat{\boldsymbol{q}}(t) \to \boldsymbol{q}(t)$ in the limit. It is hence reasonable to exploit this improved estimate to continuously refine (online) the previously optimized path by leveraging the newly acquired information during motion.

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

A. Schatten norm as differentiable approximation of λ_{min}

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 $\lambda_{min}(\boldsymbol{P}^{-1}(t, t_f))$ with the so-called Schatten norm

$$\|\boldsymbol{P}^{-1}(t, t_f)\|_\mu = \sqrt[\mu]{\sum_{i=1}^n \lambda_i^\mu(\boldsymbol{P}^{-1}(t, t_f))} \quad (9)$$

where $\mu \ll -1$ and $\lambda_i(\boldsymbol{A})$ is the i-th smallest eigenvalue of a matrix $\boldsymbol{A}$: it is indeed possible to show that (9) represents a *differentiable* approximation of $\lambda_{min}(\cdot)$ [17].

B. Flatness and B-Spline parametrization

In order to reduce the complexity of the optimization procedure adopted to solve Problem 1 and, hence to better cope with the real-time constraint of an online implementation we now recall two simplifying working assumptions already adopted in [14]. First, we restrict our attention to the case of non-linear differentially flat[1] systems [19]: as well-known, for these systems one can find a set of outputs $\boldsymbol{\zeta} \in \mathbb{R}^\kappa$, termed *flat*, such that the state and inputs of the original system can be expressed *algebraically* in terms of these outputs and a finite number of their derivatives. The flat system assumption allows avoiding the numerical integration of the nonlinear dynamics (1) for generating the future state evolution $\hat{\boldsymbol{q}}(t)$, $t \in [\bar{t}, t_f]$, from the planned inputs $\boldsymbol{u}(t)$ and the current state estimate $\hat{\boldsymbol{q}}(\bar{t})$. Second, we represent the flat outputs (and, as a consequence, also the state and inputs of the considered system) with a family of curves function of a *finite* number of parameters. Among the many possibilities, and taking inspiration from [20], [21], as in [14], also in this work we leverage the family of B-Splines [22] as parametric curves. B-Spline curves are linear combinations, through a *finite* number of control points $\mathbf{x}_c = (\mathbf{x}_{c,1}^T, \mathbf{x}_{c,2}^T, \ldots, \mathbf{x}_{c,N}^T)^T \in \mathbb{R}^{\kappa \cdot N}$, of basis functions $B_j^\alpha : S \to \mathbb{R}$ for $j = 1, \ldots, N$. Each B-Spline is given as

$$\gamma(\mathbf{x}_c, \cdot) : S \to \mathbb{R}^\kappa$$
$$s \mapsto \sum_{j=1}^N \mathbf{x}_{c,j} B_j^\alpha(\mathbf{s}, s) = \mathbf{B_s}(s)\mathbf{x}_c \quad (10)$$

where S is a compact subset of $\mathbb{R}$, $\mathbf{B_s}(s) \in \mathbb{R}^{\kappa \times N}$. The degree $\alpha > 0$ and knots $\mathbf{s} = (s_1, s_2, \ldots, s_\ell)$ are constant parameters chosen such that $\gamma(\mathbf{x}_c, \cdot)$ is sufficiently smooth w.r.t. s (the degree of smoothness will be chosen taking into account the dynamics of the system). $\mathbf{B_s}(s)$ is the collection of basis functions and B_j^α is the j-th basis function evaluated in s by means of the Cox-de Boor recursion formula [22].

By parameterizing the flat outputs $\boldsymbol{\zeta}(\boldsymbol{q})$ with a B-spline curve $\gamma(\boldsymbol{x}_c, s)$, and by exploiting the differential flatness assumption, it follows that all quantities involved in Problem 1 can be expressed as a function of the parameter s (the position along the spline) and of the control points $\boldsymbol{x}_c$. In the following we will then let $\boldsymbol{q}_\gamma(\boldsymbol{x}_c, s)$ and $\boldsymbol{u}_\gamma(\boldsymbol{x}_c, s)$ represent the state $\boldsymbol{q}$ and inputs $\boldsymbol{u}$ determined (via the flatness relationships) by the planned B-spline path $\gamma(\boldsymbol{x}_c, s)$. We also note that the control points $\boldsymbol{x}_c$ become the (sole) optimization variables for Problem 1. The control points $\mathbf{x}_c$ will then become the new optimization variables for Problem 1 that can be reformulated in terms of these few parameters and s. This choice allows then reducing the complexity of Problem 1 from an infinite-dimensional optimization into a finite-dimensional one.

It is important to stress here that, in order to always express $\boldsymbol{q}$ and $\boldsymbol{u}$ in terms of $\boldsymbol{\zeta}$ and of a finite number of their

---

[1]The class of flat systems includes some of the most common robotic platforms such as, e.g., unicycles, cars with trailers and quadrotor UAVs, and in general any system which can be (dynamically) feedback linearized [18].

time derivatives, intrinsic and apparent singularities in flat differential systems (c.f. [23], [24]) must be avoided. However, while apparent singularities can be avoided by adopting a different set of flat outputs and different state space representations, intrinsic singularities must be treated by guaranteeing some constraints along the planned trajectories that depend on the system dynamics at hand. In our previous work [14], this problem has not been tackled because of the simplicity of the dynamics used in the simulation part that has neither apparent nor intrinsic singularities. In the following, we will describe our strategy to move the control points in order to generate trajectories that avoid intrinsic singularities.

C. Additional requirements

In addition to the energy constraint already introduced in Problem 1 for guaranteeing a finite maximum value for the cost function and hence well-posedness of our optimization problem, in this work we also consider two additional constraints: *state coherency* and *flatness regularity*.

1) State coherency: Already introduced in our previous work [14], this constraint guarantees that at the current time t, $\boldsymbol{q}_\gamma(\boldsymbol{x}_c(t), s(t)) = \hat{\boldsymbol{q}}(t)$, where $\hat{\boldsymbol{q}}(t)$ is the current estimation of the true state $\boldsymbol{q}(t)$ provided by the EKF. This constraint, while the estimated state $\hat{\boldsymbol{q}}(t)$ converges to the true one $\boldsymbol{q}(t)$, guarantees that the optimization over the time window $[t, t_f]$ will be done coherently with the current state estimation.

2) Flatness regularity: a second requirement of this work consists in avoiding intrinsic singularities, in order to always express $\boldsymbol{q}$ and $\boldsymbol{u}$ in terms of $\boldsymbol{\zeta}$ and of a finite number of their time derivatives. Any intrinsic singularity can be generically expressed as set of equalities $\boldsymbol{fl}(\boldsymbol{q}, \boldsymbol{u}) = \boldsymbol{0}$ and hence, in the contest of this work, as $\boldsymbol{fl}(\boldsymbol{x}_c, s) = \boldsymbol{0}$. The flatness regularity requirement is then equivalent to move the control points in order to prevent all the functions $\boldsymbol{fl}(\boldsymbol{x}_c, s)$ to be zero along the future planned trajectory.

D. Online Optimal Sensing Control

Letting $s_0 = s(t_0)$, $s_f = s(t_f)$ and, in general, $s(t) = s_t$, based on previous consideration, 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*

$$\boldsymbol{x}_c^*(t) = \arg\max_{\boldsymbol{x}_c} \|\boldsymbol{P}^{-1}(s_0, s_t) + \boldsymbol{P}^{-1}(\boldsymbol{x}_c(t), s_t, s_f)\|_\mu,$$

s.t.

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

where $\boldsymbol{P}^{-1}(s_0, s_t)$ represents a memory *of the information available at the current time t about the current $\boldsymbol{q}(t)$ collected while moving during $[t_0, t]$ plus* any *a priori information available at time t_0, while*

$$E(s_0, s_t) = \int_{s_0}^{s_t} \sqrt{\boldsymbol{u}(\boldsymbol{x}_{\boldsymbol{c}}, \sigma)^T \boldsymbol{M} \boldsymbol{u}(\boldsymbol{x}_{\boldsymbol{c}}, \sigma)} \, \mathrm{d}\sigma$$

is the control effort/energy spent on the already traveled interval $[t_0, t]$ (and, analogously, $E(\boldsymbol{x}_c(t), s_t, s_f)$ the control effort/energy to be spent on the future interval $[s_t, s_f]$). Finally, $\boldsymbol{P}^{-1}(\boldsymbol{x}_c, s_t, s_f)$, for $s \in [s_t, s_f]$, is solution of

$$\dot{\boldsymbol{P}}^{-1}(\boldsymbol{x}_c, s) = -\boldsymbol{P}^{-1}(\boldsymbol{x}_c, s)\boldsymbol{A}(\boldsymbol{x}_c, s) - \boldsymbol{A}^T(\boldsymbol{x}_c, s)\boldsymbol{P}^{-1}(\boldsymbol{x}_c, s) + $$
$$- \boldsymbol{P}^{-1}(\boldsymbol{x}_c, s)\boldsymbol{B}(\boldsymbol{x}_c, s)\boldsymbol{V}\boldsymbol{B}^T(\boldsymbol{x}_c, s)\boldsymbol{P}^{-1}(\boldsymbol{x}_c, s) + $$
$$+ \boldsymbol{H}^T(\boldsymbol{x}_c, s)\boldsymbol{R}^{-1}\boldsymbol{H}(\boldsymbol{x}_c, s)v(\boldsymbol{x}_c, s), \quad \boldsymbol{P}^{-1}(s_t) = \boldsymbol{0}. \quad (11)$$

We now detail the chosen optimization strategy for solving Problem 2, taking into account what has been introduced so far.

IV. ONLINE GRADIENT-BASED TRAJECTORY OPTIMIZATION

This section is devoted to adapt and extend to Problem 1 the *online* solution proposed in [14]. The method still combines an online constrained gradient-descent optimization strategy with an Extended Kalman Filter meant to recover an estimation $\hat{\boldsymbol{q}}(t)$ of the true (but unknown) state $\boldsymbol{q}(t)$ during motion. The constrained gradient descent action affects the location of the control points $\mathbf{x}_c$ and, thus, the overall shape of the planned trajectory followed by the system over the future time window $[t, t_f]$ in order to minimize the maximum estimation uncertainty. For this reason, we introduce a time dependency $\mathbf{x}_c(t)$ so that the B-Spline path becomes a time varying path. We assume hence that the control points move according to the following simple update law

$$\dot{\boldsymbol{x}}_c(t) = \boldsymbol{u}_c(t), \qquad \boldsymbol{x}_c(t_0) = \boldsymbol{x}_{c,0}, \quad (12)$$

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

Since Problem 1 involves optimization of $\boldsymbol{P}^{-1}$ subject to multiple constraints, we design $\boldsymbol{u}_c$ by resorting to the well-known general framework for managing multiple objectives (or tasks) at different priorities [25]. In short, let ${}^i\boldsymbol{o}(\boldsymbol{x}_c)$ be a generic objective (or task/constraint) characterized by the differential kinematic equation ${}^i\dot{\boldsymbol{o}} = \boldsymbol{J}_i(\boldsymbol{x}_c){}^i\dot{\boldsymbol{x}}_c$, where $\boldsymbol{J}_i(\boldsymbol{x}_c)$ is the associated Jacobian matrix. Let also $(\boldsymbol{J}_1, \ldots, \boldsymbol{J}_r)$ be the stack of the Jacobians associated to r objectives ordered with decreasing priorities. Algorithm [25] allows computing the contributions of each task in the stack in a recursive way where ${}^A\boldsymbol{N}_{i-1}$, the projector into the null space of the augmented Jacobian ${}^A\boldsymbol{J}_i = (\boldsymbol{J}_1, \ldots, \boldsymbol{J}_i)$, has the (iterative) expression ${}^A\boldsymbol{N}_i = {}^A\boldsymbol{N}_{i-1} - (\boldsymbol{J}_i{}^A\boldsymbol{N}_{i-1})^\dagger(\boldsymbol{J}_i{}^A\boldsymbol{N}_{i-1})$ and ${}^A\boldsymbol{N}_0 = \boldsymbol{I}$.

Considering Problem 1 and all the additional requirements in section III-C, we then choose the following priority list: the *state coherency* requirement should be the highest priority task, followed by the *regularity* constraint and then by the *bounded energy* constraint. Optimization of $\boldsymbol{P}^{-1}$ 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 we consider state coherency as a primary requirement (the planned path $\boldsymbol{\gamma}$ should always be synchronized with the current estimated state $\hat{\boldsymbol{q}}$), followed

by flatness regularity and then by the bounded energy requirement.

A. State Coherency

As mentioned before, the state coherency constraint ensures that the B-Spline is deformed so as to always pass through the current robot state estimation[2].

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

$$^1\dot{\boldsymbol{o}}(t) = \boldsymbol{J}_1\,^1\boldsymbol{u}_c(t) + \boldsymbol{J}_s\dot{s} - \dot{\hat{\boldsymbol{q}}}(t) \qquad (13)$$

where $\boldsymbol{J}_s = \dfrac{\partial \boldsymbol{q}_\gamma}{\partial s}$, the Jacobian $\boldsymbol{J}_1 = \dfrac{\partial \boldsymbol{q}_\gamma}{\partial \boldsymbol{x_c}} = \dfrac{\partial \boldsymbol{q}_\gamma}{\partial \boldsymbol{\Gamma}} \dfrac{\partial \boldsymbol{\Gamma}}{\partial \boldsymbol{x_c}}$, and matrix $\boldsymbol{\Gamma} = \left[\boldsymbol{\gamma}(\boldsymbol{x_c}(t), s_t), \dfrac{\partial \boldsymbol{\gamma}(\boldsymbol{x_c}(t), s_t)}{\partial s}, \dots, \dfrac{\partial^{(k)} \boldsymbol{\gamma}(\boldsymbol{x_c}(t), s_t)}{\partial s^{(k)}} \right]$ for a suitable $k \in \mathbb{N}$. Here, 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 $\dot{\hat{\boldsymbol{q}}}(t)$ is, instead, the dynamics of state updating rule EKF used to recover the state estimate $\hat{\boldsymbol{q}}(t)$. By choosing in (13)

$$^1\boldsymbol{u}_c = -\boldsymbol{J}_1^\dagger (K_1\,^1\boldsymbol{o}(t) - \dot{\hat{\boldsymbol{q}}}(t) + \boldsymbol{J}_s\dot{s}), \qquad (14)$$

one obtains *exact* exponential regulation of the highest priority task $^1\boldsymbol{o}(t)$ with rate K_1. The projector into the null space of this (first) task is just $^A\boldsymbol{N}_1 = {}^A\boldsymbol{N}_0 - (\boldsymbol{J}_1\,^A\boldsymbol{N}_0)^\dagger(\boldsymbol{J}_1\,^A\boldsymbol{N}_0)$ with $^A\boldsymbol{N}_0 = \boldsymbol{I}_{\kappa N \times \kappa N}$. Notice that, if other requirements along the path should be imposed, as e.g. the desired configuration of the robot at the end of the path or obstacles avoidance, they could be easily included at this level of priority.

B. Flatness regularity

The second constraint for Problem 1 consists in preserving flatness regularity by avoiding that the control points $\boldsymbol{x_c}$ zeroing the flatness singularity functions $\boldsymbol{fl}(\boldsymbol{x_c}, s)$. We tackle this requirement by designing a repulsive potential acting on the control points when $\delta_i(\boldsymbol{x_c}, s) = \|fl_i(\boldsymbol{x_c}, s)\|_2$ is close to zero over some intervals S_i^*. Let us define a potential function $\mathcal{U}_i(\delta_i)$ growing unbounded for $\delta_i \to \delta_{min}$ and vanishing (with vanishing slope) for $\delta_i \to \delta_{MAX}$, where $\delta_{MAX} > \delta_{min}$ represent minimum/maximum thresholds for the potential. An example of such repulsive function, which will be used in our simulation (see Section V), is

$$\mathcal{U}_i(\delta_i(\boldsymbol{x_c}, \sigma)) = \cot d + d - \frac{\pi}{2}, \quad d = \frac{\pi}{2}\frac{\delta_i - \delta min}{\delta_{MAX} - \delta_{min}}.$$

The total repulsive potential associated to the i-th interval S_i^* is

$$U_i(\boldsymbol{x_c}, s(t)) = \int_{S_i^*} \mathcal{U}_i(\delta_i(\boldsymbol{x_c}, \sigma))\,\mathrm{d}\sigma. \qquad (15)$$

where $S_i^* = S_i \cap [s_t, s_f]$ (indeed, the integral (15) is only evaluated on the future path) and, as a consequence,

$$U(\boldsymbol{x_c}, s(t)) = \sum_i \int_{S_i^*} \mathcal{U}_i(\delta_i(\boldsymbol{x_c}, \sigma))\,\mathrm{d}\sigma \qquad (16)$$

represents the repulsive potential for all N control points $\boldsymbol{x}_{c,i}$. The task here is to zero the potential (16), i.e. $^2\boldsymbol{o}(t) = U(\boldsymbol{x_c}, s(t))$. The time derivative of this task is

$$^2\dot{\boldsymbol{o}}(t) = \boldsymbol{J}_2\,^1\boldsymbol{u}_c(t) \qquad (17)$$

Analogously to previous tasks, by choosing

$$^2\boldsymbol{u}_c = {}^1\boldsymbol{u}_c - (\boldsymbol{J}_2\,^A\boldsymbol{N}_1)^\dagger(K_2\,^2\boldsymbol{o}(t) + \boldsymbol{J}_2\,^1\boldsymbol{u}_c), \qquad (18)$$

with $\boldsymbol{J}_2 = \partial U / \partial \boldsymbol{x_c}$, one obtains *exact* exponential regulation of the highest priority task $^2\boldsymbol{o}(t)$ with rate K_2. The projector into the null space of both previous objectives can be computed (recursively) as $^A\boldsymbol{N}_2 = {}^A\boldsymbol{N}_1 - (\boldsymbol{J}_2\,^A\boldsymbol{N}_1)^\dagger(\boldsymbol{J}_2\,^A\boldsymbol{N}_1)$.

C. Control effort

The third task in the priority (control effort) can be implemented in a similar fashion. Let $^3o(\boldsymbol{x_c}(t), s_t, s_f) = E(\boldsymbol{x_c}(t), s_t, s_f) - (\bar{E} - E(s_0, s_t))$ where we consider (as usual) that the control points $\boldsymbol{x_c}(t)$ cannot affect the past control effort $E(s_0, s_t)$ already spent on the path. One then has (using Leibniz integral rule and simplifying terms)

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

where

$$\boldsymbol{J}_3 = \int_{s_t}^{s_f} \frac{\partial}{\partial \boldsymbol{x_c}} \sqrt{\boldsymbol{u}(\boldsymbol{x_c}, \sigma)^T \boldsymbol{M} \boldsymbol{u}(\boldsymbol{x_c}, \sigma)}\,\mathrm{d}\sigma.$$

As a consequence, (18) is complemented as

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

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

D. Maximization of $\boldsymbol{P}^{-1}$

Finally, we consider the lowest priority task, that is, maximization of the Schatten norm of $\boldsymbol{P}^{-1}$ in the null-space of the previous tasks: the total control law for the control points to be plugged in (12) then becomes

$$\boldsymbol{u}_c = {}^3\boldsymbol{u}_c + {}^A\boldsymbol{N}_3 \nabla_{\boldsymbol{x_c}} \|\boldsymbol{P}^{-1}(s_t, s_f)\|_\mu.$$

The gradient of the Schatten norm of $\boldsymbol{P}^{-1}$ can be expanded as

$$\nabla_{\boldsymbol{x_c}} \|\boldsymbol{P}^{-1}\|_\mu = {}^{\frac{1-\mu}{\mu}}\sqrt{\sum_{i=1}^n \lambda_i^\mu(\boldsymbol{P}^{-1})} \left(\sum_{i=1}^n \lambda_i^{\mu-1}(\boldsymbol{P}^{-1})\frac{\partial \lambda_i(\boldsymbol{P}^{-1})}{\partial \boldsymbol{x_c}} \right),$$

where

$$\frac{\partial \lambda_i(\boldsymbol{P}^{-1})}{\partial \boldsymbol{x_c}} = \mathbf{v}_i^T \frac{\partial \boldsymbol{P}^{-1}}{\partial \boldsymbol{x_c}} \mathbf{v}_i$$

and $\mathbf{v}_i$ is the eigenvector associated to the i-th eigenvalue λ_i of $\boldsymbol{P}^{-1}$ [26]. By leveraging relationship (7), the quantity $\frac{\partial \boldsymbol{P}^{-1}}{\partial \boldsymbol{x_c}}$ can be obtained as the solution of the following

differential equation over the future state trajectory $\boldsymbol{q}_\gamma(\boldsymbol{x}_c, s)$ $s \in [s_t, s_f]$

$$\frac{\mathrm{d}}{\mathrm{d}s} \frac{\partial \boldsymbol{P}^{-1}(\boldsymbol{x}_c, s)}{\partial \boldsymbol{x}_c} = \frac{\partial}{\partial \boldsymbol{x}_c} \frac{\mathrm{d}\boldsymbol{P}^{-1}(\boldsymbol{x}_c, s)}{\mathrm{d}s} =$$

$$= \frac{\partial}{\partial \boldsymbol{x}_c} \Big(-\boldsymbol{P}^{-1}(\boldsymbol{x}_c, s)\boldsymbol{A}(\boldsymbol{x}_c, s) - \boldsymbol{A}^T(\boldsymbol{x}_c, s)\boldsymbol{P}^{-1}(\boldsymbol{x}_c, s) -$$

$$-\boldsymbol{P}^{-1}(\boldsymbol{x}_c, s)\boldsymbol{B}(\boldsymbol{x}_c, s)\boldsymbol{V}\boldsymbol{B}^T(\boldsymbol{x}_c, s)\boldsymbol{P}^{-1}(\boldsymbol{x}_c, s) +$$

$$+\boldsymbol{H}^T(\boldsymbol{x}_c, s)\boldsymbol{R}^{-1}\boldsymbol{H}(\boldsymbol{x}_c, s)v(\boldsymbol{x}_c, s) \Big), \; \boldsymbol{P}_{\boldsymbol{x}_c}^{-1}(s_t) = \boldsymbol{0}$$

$$\tag{19}$$

where $v(\boldsymbol{x}_c, s) = \frac{\partial \gamma(\boldsymbol{x}_c, s)}{\partial s}$. Notice that, all derivative $\boldsymbol{A}_{\boldsymbol{x}_c}(\boldsymbol{x}_c, s) = \frac{\partial \boldsymbol{A}(\boldsymbol{x}_c, s)}{\partial \boldsymbol{x}_c}$, $\boldsymbol{B}_{\boldsymbol{x}_c}(\boldsymbol{x}_c, s) = \frac{\partial \boldsymbol{B}(\boldsymbol{x}_c, s)}{\partial \boldsymbol{x}_c}$ and $\boldsymbol{H}_{\boldsymbol{x}_c}(\boldsymbol{x}_c, s) = \frac{\partial \boldsymbol{H}(\boldsymbol{x}_c, s)}{\partial \boldsymbol{x}_c}$ can be analytically computed (the initial condition $\boldsymbol{P}_{\boldsymbol{x}_c}^{-1}(s_t) = \boldsymbol{0}$ steams from the fact that $\boldsymbol{P}^{-1}(s_t)$ is independent from $\boldsymbol{x}_c$). Moreover, forward integration of (19) needs also the concurrent forward integration of (11).

V. SIMULATION RESULTS

In order to prove the effectiveness of our optimal active sensing control strategy, in this section we apply the proposed method to a unicycle robot that needs to estimate its configuration by using, as its only outputs, the squared distances from two fixed markers. Let us hence consider a unicycle robot that moves in a plane where a right-handed reference frame $\mathcal{W}$ is defined with origin in $\mathcal{O}_\mathcal{W}$ and axes $X_\mathcal{W}$, $Y_\mathcal{W}$. The robot pose $\boldsymbol{q}(t) = [x(t), \; y(t), \; \theta(t)]^T$ in the world frame defines the state of our estimation algorithm, where $(x(t), y(t))$ is the position of a representative point of the robot in $\mathcal{W}$ and $\theta(t)$ is the robot heading with respect to $X_\mathcal{W}$. Then, the robot kinematic model is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r/2 & r/2 \\ r/b & -r/b \end{bmatrix} (\boldsymbol{u} + \boldsymbol{w}) = \boldsymbol{B}\boldsymbol{T}(\boldsymbol{u} + \boldsymbol{w}),$$

where $\boldsymbol{u} = [\omega_R, \; \omega_L]^T$ are the angular velocities of the right and left wheels, respectively. Moreover, r is the radius of the robot wheels and b is the distance between their centers. We assume that the robot is equipped with a sensor able to measure the squared distances between the robot and two fixed markers that, without loss of generality, are located at $(0, d)$ and $(0, -d)$, respectively. Formally, the outputs can be expressed as

$$\boldsymbol{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} x^2 + (y-d)^2 \\ x^2 + (y+d)^2 \end{bmatrix} + \boldsymbol{\nu}. \tag{20}$$

To show the effectiveness of considering also the process noise during the optimization phase, we will compare the results obtained by applying the optimal active sensing control strategy proposed in this paper, hereafter named *CRE-based* method, with the one proposed in [14], hereafter named *OG-based* method, where the actuation noise was not taken into account. Of course, for both cases, the actuation noise acts on the inputs ω_L and ω_R. While the OG-based method is executed, the EKF does not take into account actuation noise, i.e. $\boldsymbol{V} = \boldsymbol{0}$ in the covariance update step.

On the other hand, while the CRE-based method is executed, the noise is re-introduced.

Fig. 1 compares the estimation performance of a EKF filter when the system moves either along the optimal path obtained by applying the *CRE-based* method developed in this paper or the *OG-based* method developed in [14] for the considered system[3]. The robot spent the same level of energy along both paths. The two plots on the left depict the trajectories traveled by the robot when using the *CRE-based* method (top) and the *OG-based* method (bottom). It is important to note that, starting from the initial state estimation $\hat{\boldsymbol{q}}(t_0)$, the optimal path from this estimated configuration is obtained. Then, the robot starts moving at constant velocity $v = 1$ and the EKF reduces the estimation error while the gradient descent algorithm keeps updating *online* the shape of the optimal path. For this reason, the final path will differ, in general, with respect to the initial one (compare the thick blue lines which represents the final B-Spline, with the thin blue lines, which represents the starting B-Spline, in the plots on the left in Fig. 1).

In the upper right corner of Fig. 1, the EKF performances for the two methods are reported while the minimum eigenvalues of $\boldsymbol{P}^{-1}$ are shown in the bottom right corner of the same figure. The last plot is particularly important since it shows that the proposed method outperforms the one in [14] since the minimum eigenvalue of $\boldsymbol{P}^{-1}$ reaches a higher values in the former case. Notice that the two methods generate paths having different length (the robot move at constant velocity $v = 1$) but equal energy spent.

Moreover, between 1 s and 1.6 s, in the bottom right corner of Fig. 1, the minimum eigenvalue of $\boldsymbol{P}^{-1}$ has a higher value when using the *OG-based* method. This is due to the fact that we are maximizing the smallest eigenvalue of $\boldsymbol{P}^{-1}$ at the final time t_f. For this reason, due to the presence of the actuation noise, it may happen that locally the *OG-based* method outperforms the *CRE-based* one. However, at the end of the path the smallest eigenvalue of $\boldsymbol{P}^{-1}$ assumes an higher value by using the *CRE-based* method as expected.

Table I reports the numerical data of Fig. 1. In particular, notice again the minimum eigenvalue of $\boldsymbol{P}^{-1}$, whose value at t_f obtained with the *CRE-based* method reaches a value almost three times larger than the value raised by the *OG-based* method. This confirms the validity of the proposed approach.

We have also performed a comparison considering three different kind of paths: (i) paths generated by the *CRE-based* method; (ii) paths generated by the *OG-based* method; and (iii) random paths. Along each path the energy spent is ≈ 15.46. Both optimal methods use as initial guess the random paths. Fig. 2 reports the maximum eigenvalue of $\boldsymbol{P}(t_0, t_f)$ (i.e., the maximum estimation uncertainty) and its trace (i.e., a measure of the average estimation uncertainty) for the three approaches. As expected, the *CRE-based* method significantly outperforms the other two methods. On the other hand, surprisingly, the *OG-based* paths are worse

---

[3]A video of the simulation is also attached to the paper.

TABLE I

NUMERICAL SIMULATION RESULTS OF FIG. 1. FOR BOTH SIMULATIONS, THE VEHICLE STARTS FROM $\boldsymbol{q}(t_0) = [-10.0 \text{ M}, 0.0 \text{ M}, 0.0 \text{ RAD}]^T$. THE INITIAL STATE ESTIMATION IS $\hat{\boldsymbol{q}}(t_0) = [-10.4 \text{ M}, -0.5 \text{ M}, -0.3 \text{ RAD}]^T$ WITH $\boldsymbol{P}_o = 0.16\,\boldsymbol{I}$. THE INITIAL ESTIMATION ERROR IS $\boldsymbol{e}(t_0) = [-0.4 \text{ M}, -0.5 \text{ M}, -0.3 \text{ RAD}]^T$. THE OUTPUT NOISE COVARIANCE MATRIX $\boldsymbol{R} = 9 \cdot 10^{-1}\boldsymbol{I}$ AND THE ACTUATION NOISE COVARIANCE MATRIX IS $\boldsymbol{V} = 9 \cdot 10^{-1}\boldsymbol{I}$. THE NUMBER OF CONTROL POINTS IS $N = 6$ AND THE DEGREE OF THE B-SPLINE IS $\alpha = 3$.

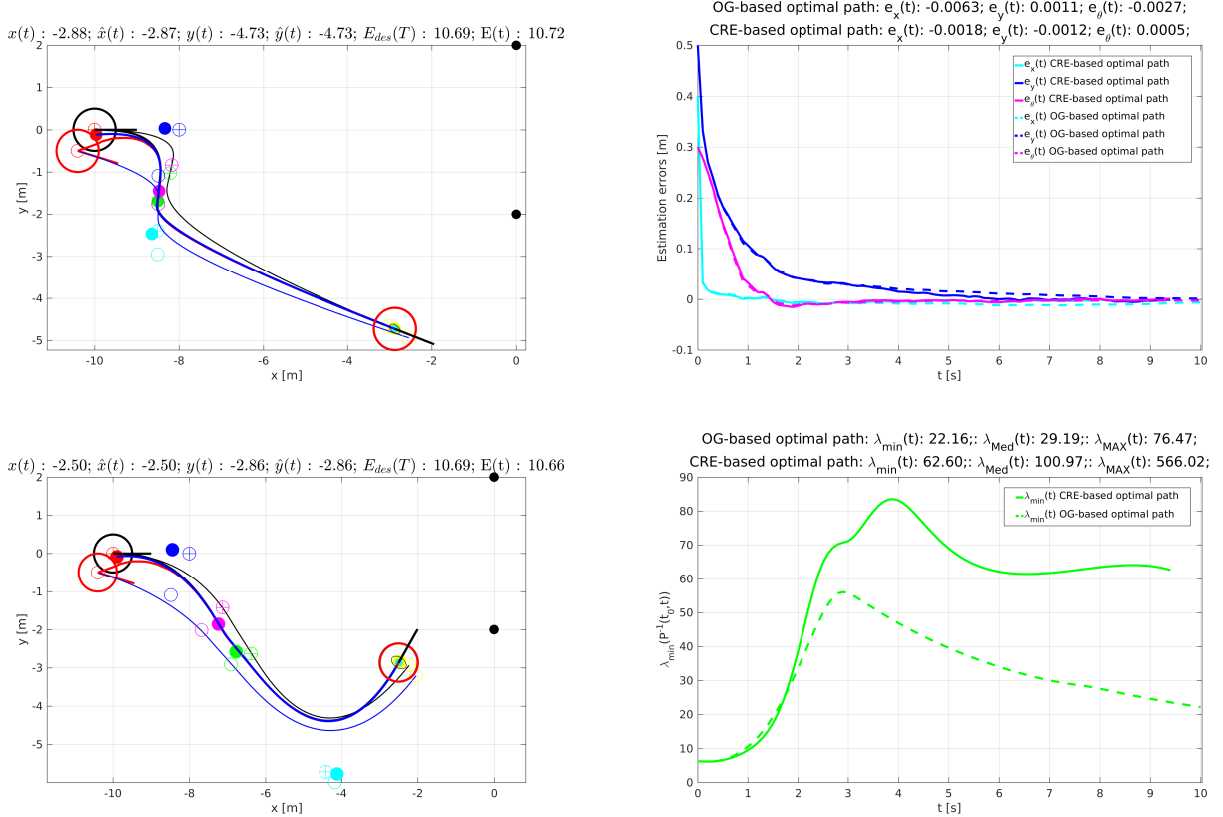|	$\boldsymbol{q}(t_f)$	$\hat{\boldsymbol{q}}(t_f)$	$\boldsymbol{e}(t_f)$ $[\times 10^{-3}]$	$\lambda_{min}((\boldsymbol{P}^{-1}(t_f))$	$\lambda_{MAX}(\boldsymbol{P}^{-1}(t_f))$
OG-based optimal path	$x(t_f) = -2.498$ m	$\hat{x}(t_f) = -2.504$ m	$e_x(t_f) = -6.3$ m	22.16	76.47
	$y(t_f) = -2.86$ m	$\hat{y}(t_f) = -2.874$ m	$e_y(t_f) = 1.1$ m		
	$\theta(t_f) = -1.058$ rad	$\hat{\theta}(t_f) = -1.059$ rad	$e_\theta(t_f) = -2.7$ rad		
CRE-based optimal path	$x(t_f) = -2.875$ m	$\hat{x}(t_f) = -2.874$ m	$e_x(t_f) = -1.8$ m	62.60	566.02
	$y(t_f) = -4.727$ m	$\hat{y}(t_f) = -4.726$ m	$e_y(t_f) = -1.2$ m		
	$\theta(t_f) = -0.372$ rad	$\hat{\theta}(t_f) = -0.371$ rad	$e_\theta(t_f) = 0.5$ rad		

Fig. 1. The estimation performance of the *CRE-based* method is compared with the *OG-based* method proposed in [14]. At the end of the *CRE-based* optimal path, the smallest eigenvalue reaches a value that is almost three times the one reached at the end of the *OG-based* optimal path. On the left, the optimal path from the estimated initial configuration (thin blue line), the optimal path from the real initial configuration (thin black line), the final B-pline (thick blue line), the real robot trajectory (thick black line) and the estimated robot trajectory (red line) obtained with *CRE-based* optimal control method (top-left) and with the *OG-based* optimal control method (bottom-left); (top-right): the estimation errors of the EKF for the *CRE-based* (solid lines) and *OG-based* (dashed lines) optimal control algorithms; (bottom-right): the minimum eigenvalue of $\boldsymbol{P}^{-1}$ obtained with the *CRE-based* method (solid lines) and with the *OG-based* method (dashed lines).

than the random ones, confirming the destructive effects of the actuation noise when not properly considered in the optimization phase. This result is statistically significant, confirmed by the *Wilcoxon rank sum test* (after having rejected the normality of variance assumption on samples) that results in less than 0.05 significance level. Obviously, the results obtained on this comparison depend on different factors, such as the actuation noise amplitude. In fact, the larger the noise, the worse the *OG-based* should perform.

Finally, the computation time at each iteration of our algorithm for on-line optimal sensing control, implemented in MATLAB®/Simulink® with not fully optimized code, is around 23 ms on an Intel Core i7-6600U running at 2.60 GHz. This confirm the possibility of a real-time implementation.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper, the problem of active sensing control has been considered for differentially flat systems in presence of actuation noise. The objective was to maximize the smallest
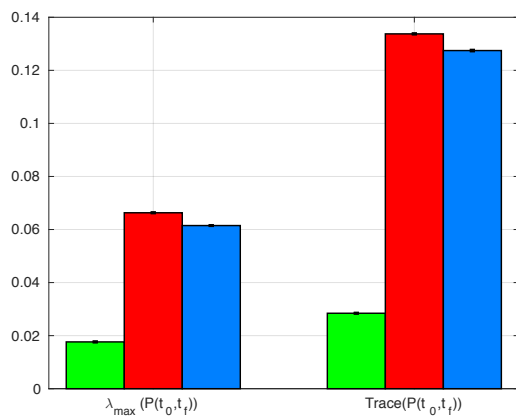
Fig. 2. Statistical analysis over 40 different runs of the *CRE-based* method (green), the *OG-based* method (red) and random paths (blue). The robot starts from the same initial conditions. On the left, the maximum eigenvalue of $\boldsymbol{P}(t_0, t_f)$, that is the maximum estimation uncertainty; on the right, the trace of $\boldsymbol{P}(t_0, t_f)$, that is a measure of the average estimation uncertainty.

eigenvalue of the inverse of the covariance matrix, that is a measure of the maximum estimation uncertainty taking into account both process noise and noisy sensory feedback. We have tested the proposed algorithm for the unicycle robot via several simulations and we have compared it to the approach in [14] which did not consider the actuation noise. We have shown that, by using our approach, the maximum estimation uncertainty is significantly reduced along the optimal path, thus giving rise to an improved estimation of the state at the end of the time horizon. We have also conducted a series of simulations, comparing the two methods described above with random paths. As one might have expected, if not properly taken into account the actuation noise can have disruptive effects in terms of maximum estimation uncertainty, yielding results similar to what one could have obtained with random (non-optimized) paths. This is, instead, not the case when correctly considering the actuation noise in the optimizaton problem, as proposed in this work. This, then, further validates the proposed approach.

In the next future, we will apply the proposed method to more complex systems such as quadrotor UAVs. It would also be interesting to include the environment within the variables to be estimated, i.e. to include some targets instead of markers as well as calibration parameters.

REFERENCES

[1] G. Orbán and D. M. Wolpert, "Representations of uncertainty in sensorimotor control," *Current Opinion in Neurobiology*, vol. 21, no. 4, pp. 629 – 635, 2011, sensory and motor systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959438811000948

[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, 2017.

[4] M. Bianchi, P. Salaris, and A. Bicchi, "Synergy-based hand pose sensing: Optimal glove design," *International Journal of Robotics Research (IJRR)*, vol. 32, no. 4, pp. 407–424, April 2013.

[5] B. T. Hinson and K. A. Morgansen, "Gyroscopic sensing in the wings of the hawkmoth manduca sexta : the role of sensor location and directional sensitivity," *Bioinspiration & Biomimetics*, vol. 10, no. 5, p. 056013, 2015.

[6] 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.

[7] R. Hermann and A. J. Krener, "Nonlinear controllability and observability," *IEEE Transactions on automatic control*, vol. 22, no. 5, pp. 728–740, 1977.

[8] G. Besançon, *Nonlinear observers and applications*. Springer, 2007, vol. 363.

[9] B. T. Hinson and K. A. Morgansen, "Observability optimization for the nonholonomic integrator," in *American Control Conference (ACC)*, June 2013, pp. 4257–4262.

[10] 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.

[11] S. Candido and S. Hutchinson, "Minimum uncertainty robot path planning using a pomdp approach," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 1408–1413.

[12] ——, "Minimum uncertainty robot navigation using information-guided pomdp planning," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 6102–6108.

[13] A. Ansari and T. Murphey, "Minimum sensitivity control for planning with parametric and hybrid uncertainty," *The International Journal of Robotics Research*, vol. 35, no. 7, pp. 823–839, 2016.

[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, pp. 672–678.

[15] 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.

[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] 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, pp. 5084–5091.

[18] A. D. Luca and G. Oriolo, "Trajectory planning and control for planar robots with passive last joint," *International Journal of Robotics Research (IJRR)*, vol. 21, no. 5–6, pp. 575–590, 2002.

[19] 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.

[20] C. Masone, A. Franchi, H. H. Blthoff, and P. R. Giordano, "Interactive planning of persistent trajectories for human-assisted navigation of mobile robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2012, pp. 2641–2648.

[21] C. Masone, P. R. Giordano, H. H. Blthoff, 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.

[22] L. Biagiotti and C. Melchiorri, *Trajectory planning for automatic machines and robots*. Springer Science & Business Media, 2008.

[23] D. E. Chang and Y. Eun, "Construction of an atlas for global flatness-based parameterization and dynamic feedback linearization of quadcopter dynamics," in *2014 IEEE 53rd Annual Conference on Decision and Control (CDC)*. IEEE, 2014, pp. 686–691.

[24] Y. J. Kaminski, J. Levine, and F. Ollivier, "Intrinsic and Apparent Singularities in Flat Differential Systems," *arXiv.org*, Jan 2017.

[25] 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.

[26] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*, 2012, http://matrixcookbook.com.