
Simultaneous Self-Calibration and Navigation using Trajectory Optimization

International Journal of Robotics
Research
Preprint(X):1-37
© The Author(s) 2018
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/



James A. Preiss¹, Karol Hausman¹, Gaurav S. Sukhatme¹, and
Stephan Weiss²

Abstract

We describe a trajectory optimization framework that maximizes observability of one or more user-chosen states in a nonlinear system. Our framework is based on a novel metric for quality of observability that is state-estimator agnostic and offers improved numerical stability over prior methods in some cases where the states of interest do not appear directly in the observation. We apply this metric to trajectory optimization problems for closed-loop self-calibration, maintaining observability while navigating through an environment, and rapidly modifying an already-planned trajectory for online recalibration. We include a statistical procedure to balance observability of several states with heterogeneous units and magnitudes. As an example, we apply our framework to online calibration of GPS-IMU and visual-inertial navigation systems on a quadrotor helicopter. Extensive simulations and a real-robot experiment demonstrate the effectiveness of our framework, showing better convergence of the states and the resulting higher precision in navigation.

Keywords

Robotics, State Estimation, Self-Calibration, Observability Analysis, Sensor-Fusion

¹University of Southern California, Los Angeles, USA

²Institute of Smart System Technologies, Alpen-Adria-Universitat Klagenfurt, Klagenfurt, Austria
The research leading to these results has received funding from the Austrian Ministry for Transport, Innovation and Technology (BMVIT) under grant agreement n. 855777 and from the ARL within the BAA W911NF-12-R-0011 under grant agreement W911NF-16-2-0112.

Corresponding author:

James A. Preiss, USC Dept. of Computer Science, 941 Bloom Walk, Los Angeles, CA 90016, USA
Email: jpreiss@usc.edu

1 Introduction

Robotic state estimation and control often require knowledge of the robot's calibration parameters. The set of calibration parameters for a complex robot may be large and diverse, including geometric properties such as sensor poses and link kinematics, optical camera intrinsics, electromechanical relationships in sensors and actuators, or dynamics properties such as moments of inertia and friction coefficients. Incorrect estimates of calibration parameter values can degrade state estimation accuracy or introduce a discrepancy between expected and actual results of control inputs. The stability guarantee of a feedback controller may even become invalid if a calibration parameter is estimated incorrectly.

Many robotic applications require the robot to operate in harsh environments or for a long periods of time. Under such conditions, it may be impractical or impossible to interrupt the robot's operation and perform an offline calibration routine. Still, calibration parameters usually change over time due to component wear, environmental conditions, or transient changes, e.g. after a collision. Online self-calibration addresses this issue by calibrating the relevant system states continuously during nominal operation, often within the same computational framework used to estimate configuration states such as position and orientation.

Including self-calibration states in the estimator comes at an important cost: the dimensionality of the state vector increases while the dimensionality of measurements remains unchanged. This cost may lead to the requirement of a nonzero, "exciting" system input to render all states observable (Kelly and Sukhatme 2011). As a result, motion planning and self-calibration are coupled: a pathological motion can make self-calibration impossible, but a motion optimized for observability can make self-calibration easier. Awareness of observability can be incorporated into a robot's planning and control framework to improve state estimation quality while simultaneously completing the robot's primary task. Note that motion planning based on state observability is fundamentally different from planning in the active perception context. Whereas the latter focuses on planning to obtain optimal measurements (i.e. system outputs), observability-aware planning focuses on generating optimal system excitations (i.e. system inputs).

In this paper, we describe such a framework for observability-aware motion generation. Our work is built upon the theory of nonlinear observability, as developed in the controls community. In particular, we optimize for a cost function closely related to the Local Observability Gramian, which measures the quality of observability for a nonlinear system following a particular trajectory. This theory applies to any nonlinear system that has smooth dynamics, a differentiable sensor model, and is observable in the user-chosen states*. Moreover, our method is not specific to a particular state estimation method, as it acts directly on the nonlinear time-continuous system definition. By choosing appropriate polynomial bases to represent trajectories, we implement our method for both closed-loop self-calibration trajectories and for

*Note that any partially observable system can be transformed into a fully observable system (given specific inputs) with a state vector dimension of the rank of the partially observable original system following Martinelli (2011)

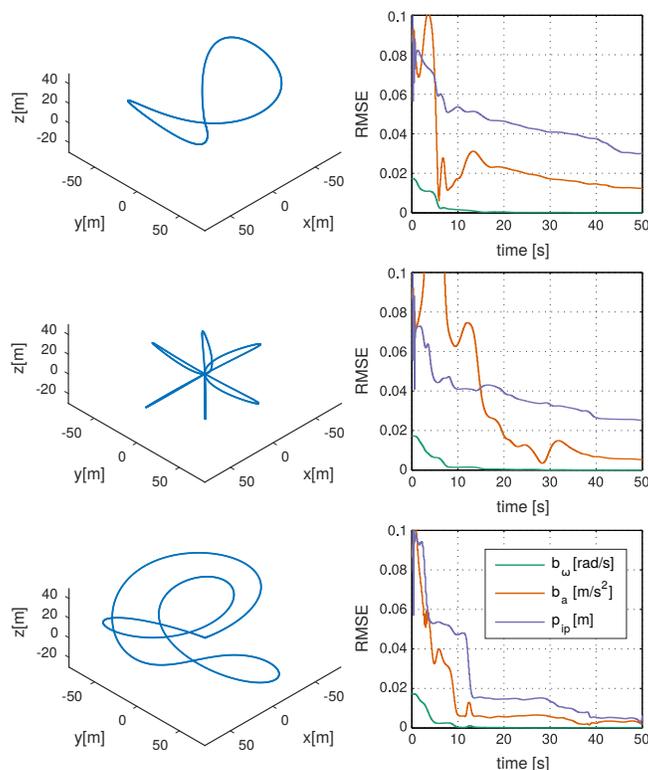


Figure 1. Root Mean Squared Error (RMSE) convergence of EKF self-calibration states for a quadrotor with GPS-IMU sensor suite: accelerometer bias \mathbf{b}_a , gyroscope bias \mathbf{b}_ω , and GPS-IMU offset \mathbf{p}_{ip}^p . From top to bottom: figure-eight heuristic trajectory, star heuristic trajectory, optimal trajectory from our method. Figure-eight and star trajectories include added yaw movement for improved observability.

collision-free trajectories through an environment with obstacles. We address the online replanning scenario with a fast sampling-based approach that improves observability over a short time horizon while maintaining continuity with an existing plan. We also describe a statistical scaling procedure that accounts for the varying units, parameter value distributions, dynamics, and measurement model in a system with multiple self-calibration states. This procedure allows us to generate trajectories that balance the goals of converging multiple self-calibration states.

We evaluate our methods with a series of experiments on a real and a simulated quadrotor using both GPS-IMU and visual-inertial sensor suites. The experiments show that our method compares favorably against manually designed heuristics and an EKF-specific covariance approach in the self-calibration task. We show that our observability-aware environment navigation trajectory leads to more accurate state estimation than a common energy-minimizing trajectory in both the full trajectory planning and online replanning scenarios. Some of the real robot experiments are shown in the video: <https://youtu.be/v8UkOtRJEsw>.

2 Related Work

Past work on planning for state estimation can roughly be divided into two groups: *exteroceptive*, or environment-based, and *proprioceptive*, or movement-based. Exteroceptive methods focus on analyzing the environment around a robot and biasing the motion planning towards the most informative areas, generally by maximizing an information theoretic metric (Bry and Roy 2011; Julian et al. 2012; Indelman et al. 2015). More recent work considered dense photometric image information by seeking highly textured surfaces (Costante et al. 2016).

Proprioceptive methods, which include the method shown in this paper, focus on *how* the robot should move to obtain the most accurate state estimates regardless of the environment. Much work in this area selects a specific realization of a state estimator and minimizes the final state uncertainty. With simple systems it may be possible to obtain an analytic solution (Martinelli and Siegwart 2006), but more commonly on complex systems a sampling-based approach with simulation is used (Achtelik et al. 2013; Bähmann et al. 2017). However, simulating the state estimator exposes the trajectory optimization method to any shortcomings of the state estimator, particularly with regard to linearization inconsistency as demonstrated by Hesch et al. (2014). These methods also inherit the potentially large computational cost of the state estimator.

Several works have proposed methods to select highly informative subsequences from a long trajectory. This step is necessary to make large-scale bundle adjustment calibration computationally feasible on a small mobile platform. In Maye et al. (2016), the authors evaluate the mutual information between the current parameter estimate and an incoming batch of data, taking advantage of intermediate results from the Gauss-Newton optimization used to estimate the parameters. The method is applied to a 2D SLAM problem with landmarks. A similar approach was applied in the visual-inertial odometry setting in Schneider et al. (2017) using an approximated, more computationally efficient information metric. Other works focus on detecting changes in the self-calibration parameters and directly modeling the parameter drift in the estimation stage, such as Nobre et al. (2017). Khosoussi et al. (2016) identified a connection between the SLAM pose graph structure and state estimate uncertainty, and derived methods to prune the pose graph to a desired sparseness nearly optimally. In contrast, our framework focuses on generating informative segments rather than evaluating the information content of given segments.

Krener and Ide (2009) propose a continuous measure of observability building upon the non-linear observability analysis suggested by Hermann and Krener (1977). This method analyzes the system dynamics and sensor model directly and is not specific to any particular state estimator. For states directly appearing in the sensor model, Hinson and Morgansen (2013) made use of this measure of observability to generate observability-aware trajectories. For states that *do not* appear in the sensor model, these methods require numerical integration of the system dynamics, which potentially introduces numerical stability issues. Our prior work (Hausman et al. 2017) addressed this issue by optimizing a slightly different, but closely related measure of observability. However, that method only optimizes trajectories for estimation of a single self-calibration state at a time, and does not handle environmental obstacles. In

our follow-up work (Preiss et al. 2017), we remove those two limitations by introducing a multi-state scaling technique and a trajectory representation based on Bézier curves that allows optimization with guaranteed obstacle avoidance. In this paper, we extend these methods to the online replanning scenario, where a robot modifies an existing trajectory plan on the fly in reaction to drift or sudden changes in self-calibration parameter values. Our replanning approach is similar in spirit to the planning under uncertainty method of Sun et al. (2015), but we sample directly in the spline basis, whereas they select the best plan from the outputs of a set of randomized motion planners.

We also present new analysis of the mathematical structure of our optimization objective that illuminates the difficulties of applying dynamic programming or RRT-based methods to planning for self-calibration. The belief-space RRT variant introduced by Bry and Roy (2011) for planning under uncertainty contains some techniques that could be applied to observability, but direct application of their method would require treating observability as a constraint rather than the main objective. We additionally show that a straightforward graph discretization of the problem is NP-complete, suggesting a fundamental hardness gap between observability and typical trajectory optimization objectives. Besides these extensions, this paper unifies our prior work into a comprehensive, self-contained article with extended derivations, improved notation, and additional details.

Other work in observability-aware motion planning has optimized similar measures of observability, but the problem of optimizing for states that do not appear in the measurement function is either avoided or resolved only for specific systems by manual analysis. Hinson et al. (2013) analyzed simple systems where the (generally intractable) state transition matrix is available, allowing direct derivation of the optimal control inputs. Bryson and Sukkarieh (2008) expressed 3D inertial SLAM in an error-state form, which allowed exact computation of the system’s unobservable modes and subsequent manual design of a set of maneuvers to improve observability. Tribou et al. (2015) identified unobservable modes of a multi-camera SLAM system in the unknown dynamics setting, where the task is reduced to analysis of the rank of the measurement Jacobian. Travers and Choset (2015) derived closed-form measures of observability for the specific system of series-elastic actuated manipulator and applied the measure in an online control setting. Hernandez et al. (2015) presented an alternative approach to observability analysis based on the volume of the set of indistinguishable trajectories for a given input sequence. Unlike most techniques in the literature, their approach can account for unknown noise inputs in the system.

These works all address similar problems to ours, but they require in-depth manual derivations for each system analyzed. In contrast, our method provides a “recipe” that produces an observability-aware trajectory optimization method given the user-chosen states of interest, system dynamics, and measurement equations. Our method is based on numerical optimization and can be applied to a new system by a non-expert.

3 Preliminaries

We consider a nonlinear dynamical system of the following form:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \delta), \quad \mathbf{z} = h(\mathbf{x}, \epsilon), \quad (1)$$

where \mathbf{x} is the state, \mathbf{u} are the control inputs, \mathbf{z} are the outputs (sensor readings), and δ, ϵ are noise values caused by modeling errors, imperfect sensors, and imperfect actuators. In this work, we use the term *self-calibration states* to describe those states whose dynamics are constant in expectation and independent of the control inputs and other state variables. We denote the self-calibration states as \mathbf{x}_{sc} . Some examples of self-calibration states are given in the introduction of this paper.

3.1 Nonlinear Observability Analysis

The observability of a system is defined as the possibility to compute the initial system state given a sequence of inputs $\mathbf{u}(t)$ and measurements $\mathbf{z}(t)$. A system is *globally observable* if there exist no two points $\mathbf{x}_0(0), \mathbf{x}_1(0)$ in the state space with the same input-output $\mathbf{u}(t)$ - $\mathbf{z}(t)$ maps for any control inputs. A system is *weakly locally observable* at the state $\mathbf{x}_0(0)$ if there is no point $\mathbf{x}_1(0)$ with the same input-output map in a neighborhood of $\mathbf{x}_0(0)$ for a specific control input (Krener and Ide 2009).

Observability of linear as well as nonlinear systems can be determined by performing a *rank test* where the system is observable if the rank of the observability matrix (defined shortly) is equal to the number of states. In the case of a nonlinear system, the nonlinear observability matrix is constructed using the Lie derivatives of the sensor model $h(\mathbf{x})$. Lie derivatives are defined recursively with a zero-noise assumption. The 0-th Lie derivative is the sensor model itself, i.e.:

$$L_0^h = h(\mathbf{x}), \quad (2)$$

the next Lie derivative is constructed as:

$$L_{i+1}^h = \frac{\partial}{\partial t} L_i^h = \frac{\partial L_i^h}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial t} = \frac{\partial L_i^h}{\partial \mathbf{x}} f(\mathbf{x}, \mathbf{u}). \quad (3)$$

One can observe that Lie derivatives with respect to the sensor model are equivalent to the respective time derivatives of the sensory output \mathbf{z} :

$$\dot{\mathbf{z}} = \frac{\partial}{\partial t} \mathbf{z}(t) = \frac{\partial}{\partial t} h(\mathbf{x}(t)) = \frac{\partial h}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial t} = \frac{\partial h}{\partial \mathbf{x}} f(\mathbf{x}, \mathbf{u}) = L_1^h. \quad (4)$$

Consecutive Lie derivatives form the matrix:

$$O(\mathbf{x}, \mathbf{u}) = [\nabla L_0^h \quad \nabla L_1^h \quad \nabla L_2^h \quad \dots]^T, \quad (5)$$

where $\nabla L_0^h = \frac{\partial L_0^h}{\partial \mathbf{x}}$. The matrix $O(\mathbf{x}, \mathbf{u})$ formed from the sensor model and its Lie derivatives is known as the nonlinear observability matrix. This matrix has (theoretically) infinite number of rows and number of columns equal to the number of states. Following Hermann and Krener (1977), if the observability matrix evaluated at a state \mathbf{x}_0 has full column rank, then the nonlinear system is weakly locally observable

at \mathbf{x}_0 . Unlike linear systems, nonlinear observability is a local property that is input- and state-dependent.

It is worth noting that the observability of the system is a binary property and does not quantify how *well* observable the system is, which limits its utility for gradient-based methods. Observability analysis also does not take into account the noise properties of the system.

4 Expanded Empirical Local Observability Gramian (E²LOG): A Metric for Quality of Observability

Following Krener and Ide (2009) and according to the definition presented in Sec. 3.1, we introduce the notion of quality of observability. A state is *well* observable if the system output changes significantly when the state is marginally perturbed (Weiss 2012). A state with this property is robust to measurement noise and is highly distinguishable within some proximity where this property holds. Conversely, a state that leads to a small change in the output, even though the state value was extensively perturbed, is defined as *poorly* observable. In the limit, the measurement does not change even if we move the state value through its full range. In this case, the state is unobservable (Hermann and Krener 1977).

4.1 Taylor Expansion of the Sensor Model

In order to model the variation of the output in relation to a perturbation of the state, we approximate the sensor model using the n -th order Taylor expansion about a time point t_0 :

$$h_{t_0}(\mathbf{x}(t), \mathbf{u}(t)) = \sum_{i=0}^n \frac{(t-t_0)^i}{i!} h^i(\mathbf{x}(t_0), \mathbf{u}(t_0)), \quad (6)$$

where h_{t_0} represents the Taylor expansion of h about t_0 with the following Taylor coefficient: h^i :

$$h^i(\mathbf{x}(t_0), \mathbf{u}(t_0)) = \frac{\partial^i}{\partial t^i} (h(\mathbf{x}(t_0), \mathbf{u}(t_0))) = L_i^h(\mathbf{x}(t_0), \mathbf{u}(t_0)) \quad (7)$$

Using this result, one can also approximate the state derivative of the sensor model $\frac{\partial}{\partial \mathbf{x}} h(\mathbf{x}(t), \mathbf{u}(t))$. For brevity, we introduce the notation $\delta t = t - t_0$ and omit the arguments of the Lie derivatives:

$$\frac{\partial}{\partial \mathbf{x}} h_{t_0}(\mathbf{x}(t), \mathbf{u}(t)) = \sum_{i=0}^n \frac{\delta t^i}{i!} \nabla L_i^h. \quad (8)$$

This result in matrix form is:

$$\frac{\partial}{\partial \mathbf{x}} h_{t_0}(t) = \left[I \quad \delta t I \quad \frac{\delta t^2}{2} I \quad \dots \quad \frac{\delta t^n}{n!} I \right] O(\mathbf{x}(t_0), \mathbf{u}(t_0)), \quad (9)$$

where $O(\mathbf{x}(t), \mathbf{u}(t))$ is the nonlinear observability matrix, whose theoretically infinite row dimension corresponds with the theoretically infinite Taylor series.

Eq. 9 describes the Jacobian of the sensor model h with respect to the state \mathbf{x} around the time t_0 . Using this Jacobian, we are able to predict the change of the measurement with respect to a small perturbation of the state. This prediction not only incorporates the sensor model but it also models the dynamics of the system via high order Lie derivatives. Hence, in addition to showing the effect of the states that directly influence the measurement, Eq. 9 also reveals the effects of the varying control inputs and the states that are not included in the sensor model. This will prove useful in Sec. 4.3.

4.2 Observability Gramian

In this section, we develop observability metrics independently of any particular trajectory parameterization. We suppose an abstract parameterization θ such that the system state $\mathbf{x}_\theta(t)$ and control inputs $\mathbf{u}_\theta(t)$ at any moment in time t can be computed from the parameters θ . Additionally, θ is implied to include the total time duration T of the trajectory.

In addition to the change in the output with respect to the state perturbation, one must take into account the fact that different states can have different influence on the output. Thus, a large effect on the output caused by a small change in one state can swamp a similar effect on the output caused by a different state and therefore, *weaken* its observability. In order to model these interactions, following Krener and Ide (2009), we employ the local observability Gramian (LOG):

$$W_o(\theta) = \int_0^T \Phi_\theta^T(t) H_\theta^T(t) H_\theta(t) \Phi_\theta(t) dt. \quad (10)$$

In (10), $\Phi_\theta(t)$ is the state transition matrix induced by the trajectory θ , defined as the solution to the differential equation:

$$\frac{d}{dt} \Phi(t) = F_\theta(t) \Phi(t) \quad (11)$$

where $F_\theta(t)$ represents linear time-varying dynamics linearized about the trajectory θ (Krener and Ide 2009). $H_\theta(t)$ is the Jacobian of the sensor model $H_\theta(t) = \frac{\partial}{\partial \mathbf{x}} h(\mathbf{x})$ evaluated at the state $\mathbf{x}_\theta(t)$. Since a nonlinear system can be approximated by a linear time-varying system by linearizing its dynamics about a nominal trajectory, one can also use the local observability Gramian for nonlinear observability analysis. If the rank of the local observability Gramian is equal to the number of states, the original nonlinear system is locally weakly observable (Hermann and Krener 1977).

Krener and Ide (2009) introduce measures of observability based on the condition number or the smallest singular value of the local observability Gramian. Unfortunately, the local observability Gramian is difficult to compute for many nonlinear systems. In fact, it can only be computed in closed form for certain simple nonlinear systems. To deal with this, the local observability Gramian can be approximated numerically by simulating the sensor model for small state perturbations, resulting in the *empirical local observability Gramian* (ELOG) (Krener and Ide 2009):

$$W_o(\theta) \approx \frac{1}{4\epsilon^2} \int_0^T [\Delta \mathbf{z}_\theta^1(t) \dots \Delta \mathbf{z}_\theta^n(t)]^T [\Delta \mathbf{z}_\theta^1(t) \dots \Delta \mathbf{z}_\theta^n(t)] dt, \quad (12)$$

where $\Delta \mathbf{z}_\theta^i$ is the change in simulated measurement caused by perturbing the i^{th} component of the initial state \mathbf{x}_0 by a small amount and integrating the control sequence \mathbf{u}_θ induced by the trajectory θ :

$$\Delta \mathbf{z}_\theta^i(t) = h(\mathbf{x}(\mathbf{x}_0 + \varepsilon \mathbf{e}_i, \mathbf{u}_\theta, t)) - h(\mathbf{x}(\mathbf{x}_0 - \varepsilon \mathbf{e}_i, \mathbf{u}_\theta, t)) \quad (13)$$

where \mathbf{e}_i are the standard basis vectors. Krener and Ide (2009) state it can be shown that the empirical local observability Gramian in Eq. 12 converges to the local observability Gramian in Eq. 10 for $\varepsilon \rightarrow 0$.

The main disadvantage of this approximation is that, to obtain $\mathbf{x}(\mathbf{x}_0 + \varepsilon \mathbf{e}_i, \mathbf{u}_\theta, t)$, it depends on integrating the forced ordinary differential equation of the system dynamics defined by the perturbed initial states $\mathbf{x}_0 + \varepsilon \mathbf{e}_i$ and the control sequence \mathbf{u}_θ . Systems of interest in the robotics domain are often second-order, nonlinear, unstable, and have no closed-form solutions. Combined with the potentially long time duration of the trajectories under consideration, this poses a challenging task for numerical integration. If the integration scheme accumulates error, it will contribute to the measure of observability in a way that is indistinguishable from the contribution of the trajectory itself. (We show an example of such error in Sec. 8.3.) This issue can be avoided only if the states of interest appear exclusively in the measurement model and not in the system dynamics. In such cases, the ELOG can be computed by perturbing the state trajectory \mathbf{x}_θ directly without numerical integration. In the following section, we propose an alternative measure of observability that captures the contribution of states that appear in the system dynamics without requiring numerical integration.

4.3 Definition of E^2 LOG

In order to present the hereby proposed measure of observability concisely, we introduce the following notation:

$$K_{\theta, t_0}(t) = \frac{\partial}{\partial \mathbf{x}} h_{t_0}(\mathbf{x}_\theta(t), \mathbf{u}_\theta(t)). \quad (14)$$

Note that computing K requires a nominal value of all the self-calibration states \mathbf{x}_{sc} . In the spirit of the local observability Gramian, we define the integral of the Taylor expansion of the sensor model over a limited time horizon as follows:

$$\widetilde{W}_{t_0, H}(\theta) = \int_0^H K_{\theta, t_0}(t_0 + t)^T K_{\theta, t_0}(t_0 + t) dt. \quad (15)$$

Note that $\widetilde{W}_{t_0, H}$, encompasses the dependency of the input-output map on both the measurement function and the system dynamics via the higher-order Lie derivatives contained in K_{θ, t_0} . By capturing these dependencies analytically via a suitably high-order Taylor expansion, we replace $2N$ numerical integrals (for N -dimensional state) required by the LOG with a single closed-form expression. However, this expression is only useful for short time horizons H due to the approximation error of the Taylor series. We thus propose an alternate measure of observability defined by summing Eq. 15 at multiple points along the trajectory θ :

$$\widetilde{W}_o(\theta) = \sum_{k=0}^N \widetilde{W}_{k\Delta t, \Delta t}(\theta), \quad \text{where } \Delta t = \frac{T}{N}, \quad (16)$$

where the number of steps N is a fixed parameter, chosen empirically, that enables us to see the effects of the system dynamics while maintaining reasonable approximation error in the Taylor expansion. To measure the quality of observability we use the smallest singular value of $\widetilde{W}_o(\theta)$ [†]. We refer to the matrix \widetilde{W}_o as the *Expanded Empirical Local Observability Gramian* (E²LOG). While this matrix is not a direct approximation of the local observability Gramian, we retain the name due to the very similar structures and goals of the two quantities. We also note that the local observability Gramian itself is based on the approximation of a nonlinear system by a linear time-varying system. By using a higher-order Taylor expansion of the dynamics in Eq. 14, the E²LOG analytically captures higher-order properties of the input-output map which are not captured in the LOG due to the linearization step.

The E²LOG can be seen as a cross-correlated measure of the sensitivity of the measurements with respect to state variations. Maximizing the smallest singular value leads to maximizing the observability of the least observable subspace of \mathbf{x}_{sc} . In contrast, maximizing the condition number or trace of the E²LOG are less appropriate. The condition number captures only the ratio between the most observable and least observable subspaces and does not favor large values *per se*. Similarly, maximizing the trace, which can be expressed as the sum of the eigenvalues, may reward trajectories that render one state very well observable while other states are unobservable. Maximizing the determinant, as the product of the eigenvalues, could be considered as an alternative but would still mix influences of well and weak observable dimensions.

To measure the observability of a subset of the states, one can select the submatrix of the E²LOG corresponding to the states of interest, and analyze the singular values of this submatrix only. We use this technique to focus on different self-calibration states of the system.

4.4 Multi-State E²LOG

In general, entries in the K matrices (14) may have widely different magnitudes. These magnitudes depend on many factors, including the physical units, measurement model, system dynamics, and the expected values of the self-calibration states. As mentioned in Krener and Ide (2009), scaling of these states is needed to ensure that the E²LOG smallest-singular-value metric balances the influence of all states equally. We introduce a column scaling in the form of

$$K' = K \mathbf{diag}(s)^{-1} \quad (17)$$

as each column of K reflects the sensitivity of the measurement function with respect to one state. In general, it is not possible to obtain a closed-form solution for the expected values of the K matrix for a given system due to the presence of nonlinear dynamics and nonlinear sensor models. The values of s are therefore determined with a Monte Carlo approximation to a uniform sampling from the distribution of K matrices for the given system using the following procedure:

[†]The matrix \widetilde{W}_o is positive semidefinite, so maximizing the smallest eigenvalue instead of the smallest singular value would be equivalent, stable, and more computationally efficient by constant factors (Trefethen and Bau 1997). We retain the singular value formulation to be consistent with Krener and Ide (2009).

- Sample a set of several physically plausible random trajectories ranging from stationary to near the physical limits of the dynamic model.
- For each trajectory, randomly sample multiple sets of realistic self-calibration parameter values \mathbf{x}_{sc} .
- For each trajectory-parameter pair, evaluate K at many points in time along the trajectory.
- Let s_i be the standard deviation of all entries in the i^{th} column of all K matrices generated with this procedure.

In Sec. 8.2, we demonstrate that for our example system, using this scaling process in a joint optimization for *all* self-calibration states can produce trajectories that perform nearly as well as trajectories optimized for the individual states in isolation.

This procedure aims to eliminate issues caused by different scales of the elements of the K matrices. In particular, states that minimally contribute to the magnitude of change of the measurement may be swamped by other states that have much larger absolute values (e.g. position of the vehicle in the world frame vs. the accelerometer bias). In our experiments, this problem caused our nonlinear optimization tools to fail at optimizing the trajectory jointly for multiple states according to the E²LOG objective, returning results close to the initial guess. By applying a scaling factor to the columns of the K matrix, we retain important properties of K such as the ratio between different partial measurement derivatives w.r.t. different states, while improving the behavior of E²LOG as an optimization objective. It is sufficient to perform this procedure once for a given system definition and use the stored s vector for different problem instances.

5 Trajectory Optimization Methods for E²LOG

The trajectory optimization problems considered in this paper can be stated as follows:

$$\begin{aligned} & \text{maximize} && \sigma_{\min}(\widetilde{W}_o(\theta)) \\ & \text{subject to} && \theta \text{ suitable for task} \\ & && \theta \text{ dynamically feasible} \end{aligned} \quad (18)$$

In this section, we introduce trajectory optimization methods for optimizing E²LOG for differentially flat systems. The class of differentially flat systems includes many vehicles which might require online self-calibration, such as cars, tractor-trailers, fixed-wing aircraft, and quadrotor helicopters (Martin et al. 2003; Mellinger and Kumar 2011). However, we emphasize that the E²LOG cost function itself is not restricted to differentially flat systems.

For a differentially flat system, there exists a set of *flat outputs* \mathbf{y} such that, given a trajectory $\mathbf{y}_\theta(t)$, the system states $\mathbf{x}_\theta(t)$ and control inputs $\mathbf{u}_\theta(t)$ can be computed as functions of the flat outputs \mathbf{y}_θ and a finite number of their derivatives:

$$\mathbf{x}_\theta = \zeta(\mathbf{y}_\theta, \dot{\mathbf{y}}_\theta, \ddot{\mathbf{y}}_\theta, \dots, \overset{(n)}{\mathbf{y}}_\theta), \quad \mathbf{u}_\theta = \psi(\mathbf{y}_\theta, \dot{\mathbf{y}}_\theta, \ddot{\mathbf{y}}_\theta, \dots, \overset{(m)}{\mathbf{y}}_\theta). \quad (19)$$

This assumption allows us to plan trajectories in the space of flat outputs that guarantee kinematic feasibility as long as the trajectory is sufficiently smooth.

We now describe two different realizations of the trajectory parameterization θ and the corresponding optimization problems. Both result in piecewise polynomial trajectories, but favor different tasks. For generating closed-loop self-calibration trajectories, we use a null-space representation that reduces the number of optimization variables. For planning trajectories through a map with obstacles, we use a Bézier curve formulation, where we can constrain the trajectory to lie inside a corridor of pairwise intersecting convex polytopes using only linear constraints on the decision variables.

5.1 Piecewise Polynomial Null-Space Basis

A d -degree, q -piece piecewise polynomial takes the form:

$$\mathbf{y}(t) = \begin{cases} p_1^T \mathbf{t}(t) & \text{if } t_0 \leq t < t_1 \\ \vdots & \\ p_q^T \mathbf{t}(t) & \text{if } t_{q-1} \leq t \leq t_q, \end{cases} \quad (20)$$

where $p_i \in \mathbb{R}^{d+1}$ is the vector of polynomial coefficients for the i^{th} polynomial piece, and \mathbf{t} is the time vector, i.e.:

$$\mathbf{t}(t) = [t^0 \quad t^1 \quad \dots \quad t^d]^T. \quad (21)$$

As detailed in Müller and Sukhatme (2014); Hausman et al. (2017), waypoint and continuity constraints on the trajectory can be represented as a linear system:

$$A [p_1^T \quad \dots \quad p_q^T]^T \triangleq Ap = b. \quad (22)$$

With a high enough degree d , this system is underdetermined. We can therefore represent any solution by the form:

$$p = p^* + \text{Null}(A)\rho, \quad (23)$$

where p^* is any particular solution of $Ap = b$, such as the minimum-norm solution provided by the Moore-Penrose pseudoinverse. This converts an optimization problem over the space of waypoint- and continuity-satisfying piecewise polynomials from a constrained, $q(d+1)$ -dimensional problem into a smaller, unconstrained problem over the null space weights ρ . The one-dimensional formulation given here extends naturally to higher-dimensional outputs. Since the gradient $\frac{\partial}{\partial p} \sigma_{\min}(\overline{W}_o)$ of the E²LOG objective with respect to the polynomial coefficients is not easily computed, we approximate the gradient by forward differences; therefore reducing the number of variables speeds up optimization significantly.

5.2 Bézier Basis

In practical robot deployments, it may be useful to have the ability to self-calibrate while performing some other task, rather than pausing to execute a closed-loop calibration trajectory. For a mobile robot, this means the robot should optimize its trajectory for self-calibration while moving from a start position to a goal position

and avoiding environmental obstacles. However, using polynomial coefficients as optimization variables is not well suited to problems with complex configuration space obstacles, and this property extends to the null-space basis. In previous work on polynomial trajectories Richter et al. (2013), the authors check collisions at a finite set of sampled points, and resolve them by adding waypoints from a known safe piecewise linear trajectory. However, this method may fail to detect collisions in between the sample points, and each resolved collision requires re-solving the optimization problem with more variables. Instead, we use a Bézier curve basis similar to Tang and Kumar (2016); Flores (2008) that provides collision avoidance guarantees.

We assume that a map of configuration space obstacles is available and that a high-level planner has identified a corridor of pairwise-overlapping convex polytopes containing some kinematically feasible path from start to goal position in the map. Such a corridor can be found using, e.g., the method of Deits and Tedrake (2015). We seek a trajectory from start to goal that minimizes our E²LOG cost function while remaining inside this corridor. From the high-level planner, we are given the start and goal positions $\mathbf{y}_{start}, \mathbf{y}_{goal} \in \mathbb{R}^k$ (where k is the dimensionality of the system's flat outputs), and a sequence of n convex polytopes C :

$$C = \mathcal{P}_1, \dots, \mathcal{P}_n, \quad \mathcal{P}_i = \{x \in \mathbb{R}^k : A_i x \leq b_i\}, \quad (24)$$

where $(A_i \in \mathbb{R}^{m \times k}, b_i \in \mathbb{R}^m)$ is the half-space representation of the polytope \mathcal{P}_i . Furthermore, we require that a path from \mathbf{y}_{start} to \mathbf{y}_{goal} exist in C :

$$\mathcal{P}_i \cap \mathcal{P}_{i+1} \neq \emptyset, \quad \mathbf{y}_{start} \in \mathcal{P}_1, \quad \mathbf{y}_{goal} \in \mathcal{P}_n. \quad (25)$$

Note that the requirement of overlap between adjacent \mathcal{P}_i is sufficient to ensure that a kinematically feasible path exists because we are working with a differentially flat system. Also note that there is no limit on the amount of overlap between any pair $\mathcal{P}_i, \mathcal{P}_j$ and that \mathcal{P}_i need not be bounded in general.

We seek a q -piece polynomial trajectory such that the i^{th} polynomial piece is contained in \mathcal{P}_i . Bézier curves provide a natural basis for expressing such trajectories. A degree- d Bézier curve is defined by a sequence of $d + 1$ control points $y_i \in \mathbb{R}^k$ and a fixed set of Bernstein polynomials, such that

$$\mathbf{y}(t) = b_{0,d}(t)y_0 + b_{1,d}(t)y_1 + \dots + b_{d,d}(t)y_d \quad (26)$$

for $t \in [0, 1]$, where each $b_{i,d}$ is a degree- d polynomial with coefficients given by Joy (2000). This form may be interpreted as a smooth interpolation between y_0 and y_d . The curve begins at y_0 and ends at y_d . In between, it does not pass through the intervening control points, but rather is guaranteed to lie in their convex hull. This follows directly from the fact that, on the interval $[0, 1]$, the Bernstein polynomials are nonnegative and form a partition of unity (Joy 2000), making any point in the form of Eq. 26 a convex combination of the control points y_i . Thus, when using control points as decision variables, constraining the control points to lie inside the polytope \mathcal{P}_i guarantees that the resulting curve will lie inside \mathcal{P}_i also. Polytope constraints on the control points are given by the linear inequalities in Eq. 24.

Enforcing arbitrary levels of continuity in piecewise Bézier curves is also easy. The derivative of $\mathbf{y}(t)$ as denoted in Eq. 26 is another Bézier curve of degree $d - 1$, with

control points that are scaled forward differences of the control points of $\mathbf{y}(t)$:

$$\mathbf{y}'(t) = db_{0,d-1}(t)(y_1 - y_0) + \dots + db_{d-1,d-1}(t)(y_d - y_{d-1}) \quad (27)$$

This is a linear transformation of the control points. We may apply this relationship recursively to generate equality constraints on the control points of adjacent pieces up to the desired level of smoothness.

In comparison to the null-space formulation, the Bézier basis is desirable because it guarantees a collision-free path through the corridor using only linear constraints. However, the number of optimization variables is larger than in the null-space formulation, and additional nonlinear constraints are still needed to enforce dynamic limits. Thus, optimization in the Bézier basis is somewhat slower than in the null-space basis. It is also true that the Bézier basis is conservative: for a polytope \mathcal{P} , there exist control points y_0, \dots, y_d such that some $y_i \notin \mathcal{P}$ but the Bézier curve through y_0, \dots, y_d lies inside \mathcal{P} . However, for the problem instances considered in this paper, the conservatism of the Bézier basis does not prevent our method from finding trajectories that perform well in experiments.

Both the null-space and Bézier bases require that the user specify the time interval for each polynomial piece. For the closed-loop self-calibration problem this is of little concern, but in the Bézier basis it introduces an undesirable coupling between the size of the polytopes \mathcal{P}_i and the speed of the robot moving through those polytopes. This issue can be addressed by several means:

1. allocating different durations to polytopes according to a size-based heuristic,
2. subdividing large polytopes until all polytopes are roughly the same size,
3. “growing” the polytopes so their overlap is maximized, allowing the optimizer more freedom to control the relative sizes of polynomial segments, or
4. including time allocations as additional optimization variables.

Of these, 3 is preferable because it relaxes the polytope-polynomial coupling without increasing the size of the optimization problem.

5.3 Numerical Optimization Method

In general, the E²LOG objective function is nonconvex in both the null-space and Bézier bases. We are therefore limited to local optimization methods. We use the MATLAB implementation of Sequential Quadratic Programming (SQP), which can enforce nonlinear constraints such as maximum motor thrust via barrier functions. Empirical tests showed that SQP performs faster than interior-point methods on our example problems. Multi-start optimization can be used to obviate the concern of picking an unusually bad initial guess. We compute the smallest singular values of the E²LOG using a standard Singular Value Decomposition (SVD). The SVD contributes negligible computation time, since evaluating the E²LOG cost function requires only one SVD, compared to the numerous nonlinear function evaluations required to approximate the integral (16). With an appropriate implementation, computing the SVD is numerically stable for any matrix (Trefethen and Bau 1997).

In closed-loop self-calibration problems using the null-space basis, we generate initial guesses of ρ in (23) by randomly sampling from a normal distribution and

discarding samples that violate the nonlinear physical constraints (e.g. thrust limits). However, for corridor problems in the Bézier basis, generating a feasible initial guess is nontrivial. One could solve a feasibility linear program to satisfy the continuity and polytope constraints, but this solution is not guaranteed to satisfy the nonlinear physical constraints, and often does not in our experience. Instead, we minimize an integrated-squared-derivative cost function using quadratic programming as in Tang and Kumar (2016) and use the solution as an initial guess. The order and relative weights of the derivatives in this cost function should be chosen based on an analysis of the system dynamics as they relate to the differentially flat variables.

5.4 Online Replanning

It may not always be desirable to modify the entire mission plan for the sake of self-calibration. We may prefer to ignore the observability objective during normal operation and modify the trajectory for observability only when the robot detects that its state estimation is performing badly. The modification should be bounded to a short time horizon, after which the robot returns to its previously defined plan. We explore this idea in this section.

For such a replanning method to be useful, its computation must be fast. Our SQP-based optimization method is not suitable, as it requires a few seconds to plan even a short trajectory. However, when considering the specific mission of replanning while navigating through a corridor, we observe that the feasible solution set is relatively small compared to optimizing a full trajectory. It is bound by the short time horizon, dimensionality of the polynomial basis, corridor constraints, physical limits of the robot, and the requirement of high-order continuity between the modified segment and the original trajectory. With this fact in mind, a sampling-based approach becomes a valid alternative to full SQP optimization. A uniform sample from the set of feasible trajectory modifications can adequately explore the solution space without requiring many thousands of samples.

Suppose a robot is navigating through a polytope corridor, using energy-minimizing trajectory optimization, when the ground truth value of one of the self-calibration parameters shifts. There exist multiple ways to detect when recalibration is necessary. In this work, we assume a consistent probabilistic state estimator that keeps track of the state uncertainty, which can be used as an indicator of whether recalibration is needed. Similarly, recalibration could be triggered if several measurements in a row do not pass the probabilistic inlier-test, e.g. Brumback and Srinath (1987). In this case, recalibration could ensure that the measurements were not outliers and the system was indeed miscalibrated. It is out of the scope of this paper to discuss the methods to detect when recalibration is needed in detail, so we suppose that such a system is available and detects the problem[‡]. We wish to modify the trajectory for the next $k_m \in \mathbb{N}$ polynomial pieces to prioritize observability while remaining inside the corridor. In the Bézier

[‡]One such method is given by Hausman et al. (2016). This system may also need to interact with the state estimator and correct any overconfidence in the current estimate uncertainty.

basis, the set of feasible modifications can be expressed as a convex polytope:

$$T_m = \{y : A_m y \leq b_m, C_m y = d_m\} \quad (28)$$

where y denotes the concatenated control points of the modified polynomial pieces, $A_m y \leq b_m$ represents the corridor constraints, and $C_m y = d_m$ represents continuity with the preceding and subsequent pieces. The inequality and equality constraints are identical to those formulated in the original corridor trajectory optimization, except the position and derivatives at the boundaries are defined by the existing trajectory plan instead of user input. We denote by y^* the current plan for these pieces. Due to the physical limits of the robot, only a small subset $V_m \subset T_m$ can be expected to satisfy the nonlinear constraints. Since y^* is an energy-minimizing trajectory, we approximate the valid trajectory set by a unimodal distribution centered on y^* . We sample trajectories near y^* as follows:

$$y = y^* + \text{Null}(C_m)\rho, \text{ where } \rho \sim \mathcal{N}(0, \sigma^2), \text{ and } A_m y \leq b_m. \quad (29)$$

We note that, even when k_m is small, the dimensionality of ρ may be on the order of 10, and if the inscribed diameter of the polytopes are small relative to σ , a naive rejection sampling approach can become too slow. This problem is exacerbated by the fact that, in a nonconvex corridor, the energy-minimizing trajectory is often tight against some of the corridor inequalities. Instead, we employ *generalized hit-and-run* sampling (Bélisle et al. 1993), a Markov Chain Monte Carlo sampling approach, to sample according to (29) efficiently. We use the method of Botev (2017) as a subroutine to sample from a truncated one-dimensional Gaussian distribution.

After sampling according to (29), we discard candidate trajectories that violate the nonlinear physical limits to form a sample from V_m , and select the modification $y_m \in V_m$ that performs best according to the E²LOG metric. The variance parameter σ^2 should be selected such that a majority of trajectories are rejected, thus encouraging that the limits of V_m are reached. In the simulation experiments (Sec. 8.6), we show that this approach can execute in under 1/2 second and result in significant improvement of state estimation compared to following an energy-minimizing trajectory.

6 Remark on Structure of the Optimization Problem

Examining the structure of the $\sigma_{\min}(\widetilde{W}_o)$ objective suggests that its maximization is fundamentally more difficult than typical trajectory optimization objectives considered in robotics. Most trajectory optimization algorithms used in robotics assume some favorable structure to the objective. Some of these structural properties include:

Additivity:

$$R(\theta) = \int_0^T [r_x(\mathbf{x}_\theta(t)) + r_u(\mathbf{u}_\theta(t))] dt, \quad (30)$$

Monotonicity:

$$R(\theta_1|\theta_2) \geq R(\theta_1) \quad (31)$$

Submodularity:

$$R(\theta_1|\theta_2) \leq R(\theta_1) + R(\theta_2) \quad (32)$$

where r_x and r_u are arbitrary scalar functions, and the $|$ operator denotes trajectory concatenation in time. We state these properties in terms of a reward $R(\theta)$ to be maximized. Note that we use the term “submodular” in the sense of Hollinger and Sukhatme (2013), where it is defined in terms of concatenation of sequences. This is a non-standard definition meant to draw an analogy with the “diminishing returns” property of standard submodular set functions, which are defined in terms of set unions and intersections.

Additive objectives exhibit *optimal substructure*: if $A \rightarrow B \rightarrow C$ is an optimal trajectory from A to C , then $A \rightarrow B$ must also be an optimal trajectory from A to B . The optimal substructure can be exploited algorithmically via dynamic programming in “shooting” methods like iLQG (Todorov and Li 2005), and lends a favorable structure to the objective gradient in “collocation” methods such as CHOMP (Ratliff et al. 2009). Optimal sampling-based planners such as RRT* depend on monotonicity (Karaman and Frazzoli 2011). Submodular objectives are generally more difficult to optimize than additive or monotonic objectives, but they exhibit the *diminishing returns* property that allows pruning of the search tree using a heuristic, e.g. in the information-gathering planners proposed by Hollinger and Sukhatme (2013).

Unfortunately, the E²LOG objective is neither additive, monotonic, nor submodular. When the integration of (16) is discretized into a summation, the E²LOG objective takes the general form:

$$R(\theta) = \sigma_{\min}(K_1^T K_1 + \dots + K_N^T K_N) \quad (33)$$

where the matrices $K_i^T K_i$ are positive semidefinite, so the smallest singular values and eigenvalues are equal. Weyl’s inequality states that, for A and B symmetric,

$$\sigma_{\min}(A) + \sigma_{\min}(B) \leq \sigma_{\min}(A + B). \quad (34)$$

For example,

$$\begin{aligned} A &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, & B &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, & A + B &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ \sigma_{\min}(A) &= 0, & \sigma_{\min}(B) &= 0, & \sigma_{\min}(A + B) &= 1 \end{aligned} \quad (35)$$

Therefore, for the E²LOG objective,

$$R(\theta_1 | \theta_2) \geq R(\theta_1) + R(\theta_2) \quad (36)$$

This corresponds with the intuition behind the σ_{\min} metric: even if some of the states are well observable, the trajectory is considered poor unless *all* states are well observable. However, the simple example (35) also provides a counterexample to additivity, monotonicity, and submodularity for the E²LOG objective. In the terminology of Hollinger and Sukhatme (2013), (36) would classify the E²LOG objective as “supermodular”. However, standard greedy and bounded-suboptimal methods for supermodular set function maximization (described e.g. by Vondrák (2007)) do not directly apply to the concatenative definition of supermodularity.

The result (36) explains the difficulty of applying some widely-used trajectory optimization methods for continuous dynamical systems to the E²LOG. We may thus be motivated to consider graph-based discrete approaches instead. This does not lead to a tractable problem. In Appendix A, we show that a straightforward conversion of the continuous problem to a graph approximation results in an NP-complete problem. These results suggest that a major improvement in computational efficiency or global optimality would require a modification of the problem itself rather than the introduction of new computational methods.

7 Example Systems

In this section, we introduce the systems used to demonstrate our approach in simulation and real robot experiments. We analyze a quadrotor helicopter equipped with two different sensor suites: a GPS-IMU pair and a loosely coupled visual-inertial odometry (VIO) system.

We emphasize that the method presented in this work is generic. No part is specific to quadrotors or state estimation tasks related to $SE(3)$. We choose to analyze the quadrotor due to its relative simplicity and our familiarity from prior work.

7.1 Quadrotor Helicopter

Dynamical properties of the quadrotor helicopter have been extensively analyzed in the robotics and control literature. It is common to formulate the dynamics such that the IMU measurements, instead of motor thrusts, act as system inputs. Following this approach, we represent the system state as:

$$\mathbf{x}_{core} = \left[\mathbf{p}_w^i{}^T, \mathbf{v}_w^i{}^T, \mathbf{q}_w^i{}^T, \mathbf{b}_\omega^T, \mathbf{b}_a^T \right]^T \quad (37)$$

where \mathbf{p}_w^i , \mathbf{v}_w^i and \mathbf{q}_w^i are the position, velocity and orientation (represented as a quaternion) of the IMU in the world frame. \mathbf{b}_ω and \mathbf{b}_a are the gyroscope and accelerometer biases, respectively. We classify these as the *core states* because their dynamics are independent of the choice of measurement sensor suite. The core states are governed by the following differential equations:

$$\begin{aligned} \dot{\mathbf{p}}_w^i &= \mathbf{v}_w^i \\ \dot{\mathbf{v}}_w^i &= \mathbf{C}_{(\mathbf{q}_w^i)}^T (\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a) - \mathbf{g} \\ \dot{\mathbf{q}}_w^i &= \frac{1}{2} \Omega(\omega_m - \mathbf{b}_\omega - \mathbf{n}_\omega) \mathbf{q}_w^i \\ \dot{\mathbf{b}}_w &= \mathbf{n}_{\mathbf{b}_\omega}, \quad \dot{\mathbf{b}}_a = \mathbf{n}_{\mathbf{b}_a}, \end{aligned} \quad (38)$$

where $\mathbf{C}_{(\mathbf{q})}$ is the rotation matrix obtained from the quaternion \mathbf{q} , \mathbf{g} is the gravity vector in the world frame, $\Omega(\omega)$ is the quaternion multiplication matrix of ω , \mathbf{a}_m is the accelerometer signal, and ω_m is the gyroscope signal. The accelerometer and gyroscope measurements are corrupted by additive white Gaussian noise \mathbf{n}_a and \mathbf{n}_ω respectively. Since the IMU biases can change over time, they are modeled as random processes where $\mathbf{n}_{\mathbf{b}_\omega}$ and $\mathbf{n}_{\mathbf{b}_a}$ are zero-mean Gaussian random variables.

As shown by Mellinger and Kumar (2011), the quadrotor dynamics are differentially flat. This means that a quadrotor can execute any smooth trajectory in the space of flat outputs as long as the trajectory respects the physical limitations of the system. The flat outputs are x, y, z position and yaw angle (heading) θ . The remaining extrinsic states, i.e. roll and pitch angles, are functions of the flat outputs and their derivatives. The motor thrust commands map linearly to angular accelerations in the body frame, which correspond to the fourth derivative of position. We therefore require trajectories to be continuous up to the fourth derivative to ensure physical plausibility. We also place inequality constraints on thrust-to-weight ratio (≤ 1.5), angular velocity ($\leq \pi \text{ rad sec}^{-1}$), and deviation from vertical orientation ($\leq \pi/4 \text{ rad}$). These values allow fairly quick movements but disallow racing-level aggressiveness.

7.2 GPS-IMU Sensor

The combination of a Global Positioning System (GPS) receiver and a 6-DOF Inertial Measurement Unit (IMU) is a popular choice for quadrotors operating in open outdoor spaces for applications such as cinematography. The GPS receiver supplies a relatively noisy, low-bandwidth measurement of the vehicle's position in world coordinates. Noise and low bandwidth limit the quality of a velocity estimate obtained from the GPS alone. Thus, the accelerometer plays an important role in velocity estimation, so its calibration is important.

The GPS introduces the additional self-calibration state \mathbf{p}_i^p representing relative position between the GPS module and the IMU in the IMU frame. Assuming the connection between the IMU and the GPS sensor is rigid, we define the GPS sensor model:

$$\mathbf{z}_{gps} = h_{gps}(\mathbf{x}_{core}, \mathbf{p}_i^p, \mathbf{n}_{z_{gps}}) = \mathbf{p}_w^i + \mathbf{C}_{(\mathbf{q}_w^i)}^T \mathbf{p}_i^p + \mathbf{n}_{z_{gps}}, \quad (39)$$

where $\mathbf{n}_{z_{gps}}$ is white Gaussian measurement noise. The nonlinear observability analysis in Kelly and Sukhatme (2011) and Weiss (2012) shows that the system is fully observable with appropriate inputs. The nonlinear observability matrix of this system becomes full-rank after including the 5th Lie derivative, hence, this is the order of the Taylor expansion we used for experiments.

7.3 Visual-Inertial Sensor

Vision-based navigation is also popular for quadrotors. Visual input allows obstacle avoidance and operating in GPS-denied environments, while avoiding the weight and ambient light issues associated with active sensors such as LIDAR. The camera image is input to a visual odometry algorithm (e.g. Klein and Murray (2007)) which estimates the system's 3D position in undefined scale, and scale-free attitude estimations with respect to its own visual frame. The visual-inertial system requires estimating a high-dimensional set of calibration parameters in different physical units and scales, creating a challenging task for trajectory optimization. The additional system states consist of the following:

$$\mathbf{x}_{vis} = [\lambda, \mathbf{p}_i^c, \mathbf{q}_i^c, \mathbf{p}_v^w, \mathbf{q}_v^w] \quad (40)$$

where λ is the visual scale, \mathbf{p}_i^c and \mathbf{q}_i^c are the relative position and orientation between the camera and the IMU in the IMU frame, and \mathbf{q}_v^w can be seen as the direction of the gravity vector in the visual map Kelly and Sukhatme (2011) which drifts over time due to accumulated errors in the visual framework. \mathbf{p}_v^w is the analogous visual drift in position. This setup is also known as a loosely coupled visual-inertial odometry approach and is described in more detail in Weiss et al. (2013).

Assuming the connection between the IMU and the camera is rigid, we define the visual sensor model following Weiss et al. (2013):

$$\begin{aligned} \mathbf{z}_{p_v} &= h_p(\mathbf{x}, \mathbf{n}_{z_{p_v}}) = \mathbf{p}_v^w + \lambda \mathbf{C}_{(\mathbf{q}_v^w)}^T (\mathbf{p}_i^i + \mathbf{C}_{(\mathbf{q}_i^i)}^T \mathbf{p}_i^c) + \mathbf{n}_{z_{p_v}}, \\ \mathbf{z}_{q_v} &= h_q(\mathbf{x}, \mathbf{n}_{z_{q_v}}) = \mathbf{q}_i^c \otimes \mathbf{q}_i^i \otimes \mathbf{q}_v^w + \mathbf{n}_{z_{q_v}}, \end{aligned} \quad (41)$$

where $\mathbf{n}_{z_{p_v}}$ and $\mathbf{n}_{z_{q_v}}$ are white Gaussian measurement noise variables and \otimes denotes quaternion-quaternion multiplication. The nonlinear observability analysis in Kelly and Sukhatme (2011) and Weiss (2012) shows that the system is observable up to the global position and heading only with appropriate inputs. The nonlinear observability matrix of this system has maximal rank after including the 4th Lie derivative, hence, this is the order of the Taylor expansion we use for generating the E²LOG in our experiments.

8 Simulation Experimental Results

In this section, we present simulation experiments for both the GPS-IMU and visual-inertial sensor suites. A simulation environment provides the important benefit of known ground truth values for the self-calibration parameters. Many of the parameters considered in this paper are not easy to measure on real robots. In whole-trajectory planning experiments, we represent trajectories as degree-7 polynomials with C^4 continuity and require that all derivatives be zero at the beginning and end points. Online replanning requires a degree of 8 to maintain enough degrees of freedom. The integration step and the time horizon (Δt) for E²LOG are 0.1s.

As mentioned in earlier, our method is not specific to any particular type of state estimator. However, to judge the effectiveness of a trajectory for self-calibration, we wish to simulate a robot following the trajectory and measure the accuracy of the robot's state estimate relative to the ground truth. This requires a choice of state estimator to use for the experiments. We employ the popular Extended Kalman Filter (EKF) for all experiments. In particular, we use the indirect formulation of an EKF Lynen et al. (2013) to avoid the singular covariance matrix associated with the unit quaternion invariant Lefferts et al. (1982). We choose this state estimator due its ability to work with various sensor suites and proven robustness in the quadrotor scenario. Other possible choices include Unscented Kalman Filters, particle filters, or batch nonlinear least-squares estimation.

8.1 Correlation between E²LOG and estimation accuracy

We first aim to demonstrate that $\sigma_{\min}(\widetilde{W}_o(\theta))$ is correlated with the quality of the state estimate obtained from traversing the trajectory θ . We demonstrate this first on the GPS-IMU system in a closed-loop self-calibration task without obstacles. We generate random trajectories by sampling a zero-mean Gaussian distribution for each

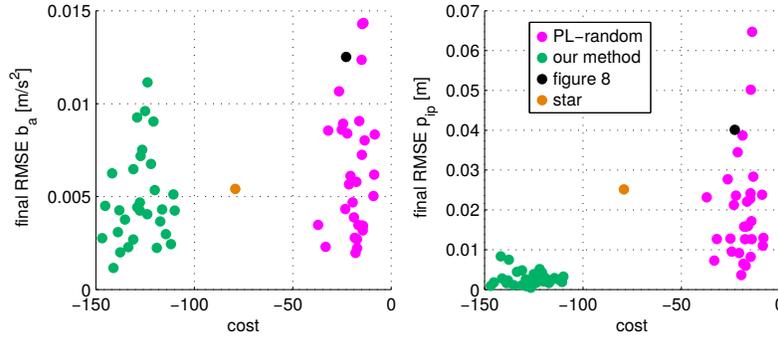


Figure 2. Self-calibration task: final RMS error values for the accelerometer bias \mathbf{b}_a and the GPS position in the IMU frame \mathbf{p}_i^p obtained using optimization (green) and 3 different heuristics: star and figure eight trajectories from Fig. 1 and randomly sampled trajectories that are close of the physical limits of the system.

optimization variable, i.e. each column of the null space of the piecewise polynomial constraint matrix described in Sec. 5.1. The standard deviation of the distribution is chosen to be large, such that some of the generated trajectories violate the system’s physical limits (listed in Sec. 7.1) and are discarded. This biases the remaining trajectories towards “exciting” system inputs that should lead to good observability. We then used each random trajectory as an initial guess for E²LOG optimization. In this experiment, we optimize for observability of only the GPS-IMU offset, \mathbf{p}_i^p . Results are shown in Fig. 2. In both plots, the x -axis corresponds to the optimization objective $-\sigma_{\min}(\widehat{W}_o(\theta))$ considering \mathbf{p}_i^p only. Note that the objective is negated for compatibility with the optimization package, which assumes a minimization problem. The y -axis corresponds to the final estimate error for the accelerometer bias \mathbf{b}_a (left) and for the GPS-IMU offset \mathbf{p}_i^p (right). *PL-random* are the randomized trajectories described above. *Figure 8* and *star* are the heuristic trajectories presented in Fig. 1, and *our method* are trajectories generated from our optimization framework using the *PL-random* trajectories as initial conditions. While the *star* trajectory and some of the *PL-random* trajectories perform well on \mathbf{b}_a , our approach outperforms all other methods on \mathbf{p}_i^p . These plots illustrate that the E²LOG objective is strongly correlated with the final estimation accuracy in the chosen states, but only weakly correlated with the accuracy in self-calibration states that were not included in the E²LOG.

8.2 Validation of multi-state E²LOG scaling

To validate our proposed multi-state E²LOG scaling procedure, we compare trajectories jointly optimized for all self-calibration states against trajectories individually optimized for a single self-calibration state. We generate closed-loop trajectories using the null space polynomial basis. We conduct this experiment on the visual-inertial system because its self-calibration states comprise many different physical units with different ranges, so scaling is essential. The experimental method is as follows:

- Generate a set \mathcal{T} of random trajectories near the physical limits of the system using the method described in Sec. 8.1.

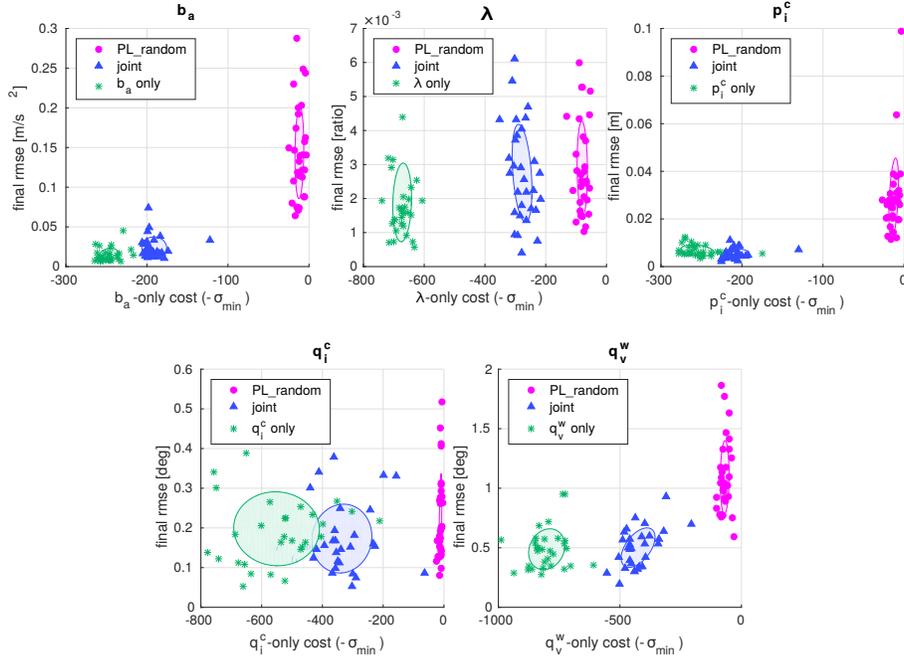


Figure 3. Comparison between trajectories jointly optimized for all self-calibration states (blue triangles), trajectories separately optimized for individual self-calibration states (green stars), and randomly generated trajectories that excite the system near its physical limits (purple circles). x -axes represent E²LOG cost for individual state only. y -axes represent EKF estimation error of individual state at trajectory termination. Shaded ellipses indicate one- σ principal component boundaries of each trajectory class.

- Generate a set \mathcal{T}_j of trajectories jointly optimized for all self-calibration states by using each trajectory in \mathcal{T} as an initial guess for gradient-based optimization, using the multi-state scaling described in Sec. 4.4.
- For each self-calibration state s , generate a set \mathcal{T}_s of trajectories optimized for s by using \mathcal{T} as initial guesses and removing the rows and columns from the E²LOG corresponding to other self-calibration states.
- For each self-calibration state s , measure how well the EKF estimates the ground truth value of s for each trajectory in \mathcal{T} , \mathcal{T}_j , and \mathcal{T}_s .

Results of this procedure are shown as scatter plots in Fig. 3. Note that we do not optimize for the state p_v^w as the global position of this system is unobservable Kelly and Sukhatme (2011). In each scatter plot, the x -axis corresponds to the negated σ_{\min} E²LOG cost function for the individual calibration state s , and the y -axis corresponds to the estimation error of s in the EKF at the end of the trajectory. Each marker represents one trajectory. Final error for each trajectory is averaged over five sets of randomly sampled ground truth self-calibration parameters. We see that, while the jointly optimized trajectories do not score as highly on the individual-state E²LOG cost functions, they perform equally well or nearly as well in the EKF error metric. This suggests that the multi-state E²LOG successfully balances the goals of optimizing each

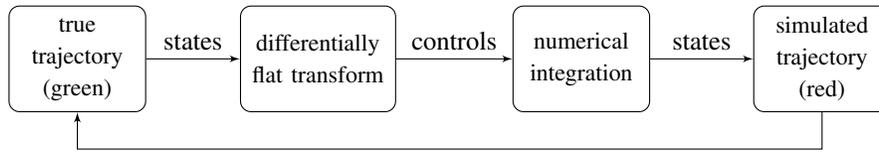
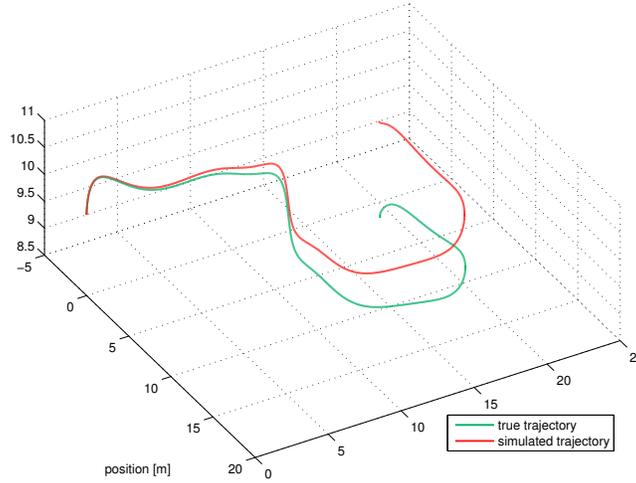


Figure 4. Illustration of numerical instability when applying ELOG to states that do not appear in sensor model. ELOG requires representing the trajectory as a control sequence and integrating the controls numerically to see the states in the ELOG. Inherent instability of high-order numerical integration leads to divergence from the true trajectory.

self-calibration state. Note that, in initial experiments without column scaling, SQP frequently terminated at a solution near the initial guess, indicating that the unscaled cost function is poorly conditioned with respect to the optimization variables.

Both the individual and joint optimized trajectories significantly exceed the performance of random trajectories on the \mathbf{b}_a , \mathbf{p}_i^c , and \mathbf{q}_v^w parameters. On \mathbf{q}_i^c , all three classes perform roughly equally, but we note that the typical estimation error of 0.2 degrees is very low and can be considered successfully converged in all cases. On λ , the joint optimized trajectories perform equally well as the random trajectories, but here the estimation error of 0.3% is also quite small.

8.3 Example of ELOG numerical stability issue

In Sec. 4.3, we mentioned that numerical instability can arise from using the Empirical Local Observability Gramian (ELOG) as proposed by Krener and Ide (2009). We now show an example of this issue. In the IMU-driven formulation of the quadrotor dynamics (38), the IMU bias terms appear only when the control signal is integrated into the state. To see these biases in the ELOG, we must thus represent the trajectory as a sequence of control inputs rather than as a sequence of states. Since the quadrotor

system is differentially flat in (x, y, z, θ) , we can design a trajectory in state space and derive the control inputs needed to achieve the trajectory using the flatness property. However, the quadrotor dynamics are second-order in the accelerometer input and third-order in the gyroscope input. The high-order dynamics make numerical integration of the control sequence unstable. In our experiments, even with accurate ODE integration scheme such as RK4 (Butcher 1996), the state-space trajectory obtained from integrating the controls drifts away from the nominal trajectory. An example is shown in Fig. 4. The integrated control sequence (red line) diverges significantly from the state-space trajectory (green).

8.4 Comparison to EKF-trace-minimization and heuristics

To explore the characteristics of E²LOG-optimized trajectories in greater detail, we compare them to several competitive baselines. To the best of our knowledge, the only other widely-used cost function that reflects the convergence of the system states is based on the estimate covariance in a simulated state estimator. Minimizing the trace of the covariance results in minimizing the uncertainty about the state for all of its individual dimensions Beinhofer et al. (2013) and yields better results than optimizing its determinant (i.e. mutual information), as discussed in Hausman et al. (2015). Therefore, as one baseline of comparison for our approach, we implement the following cost function based on the covariance in a simulated EKF:

$$c_{trace} = \delta t \sum_{i=1}^n \text{tr}(P_{sc}) \quad (42)$$

where n is the length of the trajectory discretized into timesteps of length δt and P_{sc} is the submatrix of the EKF covariance estimate associated with the self-calibration states \mathbf{x}_{sc} . We optimize these trajectories using the same piecewise polynomial basis and SQP solver used for the E²LOG trajectories. Note that it is not feasible to generate many EKF-optimized trajectories in the manner of Fig. 3 because optimizing for (42) took over $50\times$ longer than optimizing for E²LOG.

We also compare against the common heuristic self-calibration trajectories *star* and *figure-8*. These trajectories are designed manually and spatially scaled so they are just within the same physical constraints used in the optimization procedure. They reflect the intuition that a good self-calibration trajectory should excite the system in multiple axes of rotation and translation.

8.4.1 GPS-IMU System For the GPS-IMU system, we collected statistics over 50 EKF simulations for a single representative trajectory from each strategy. Note that this experiment was performed before we developed the multi-state E²LOG scaling method, so our trajectory is optimized for the GPS-IMU offset \mathbf{p}_i^p only. Fig. 5 summarizes our results in terms of the RMSE integrated over the entire trajectory and the final RMSE for accelerometer bias \mathbf{b}_a and GPS position \mathbf{p}_i^p . Results show that our approach outperforms all baseline approaches in terms of the final and integrated RMSE of the GPS position \mathbf{p}_i^p . The only method that achieves a similar integrated RMSE value for GPS position is the covariance-trace-based optimization. However, computing that solution takes approximately 13 hours, versus approximately

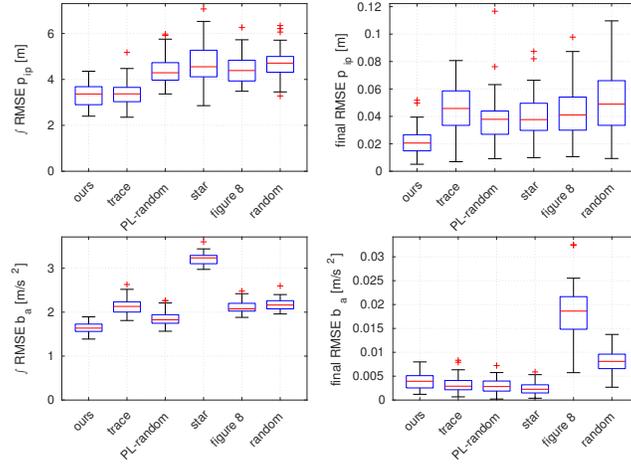


Figure 5. GPS-IMU Self-calibration task: statistics collected over 50 runs of the quadrotor EKF using 6 different trajectories: ours - E^2 LOG optimized for \mathbf{p}_i^p only; trace - optimized trajectory using the covariance-trace cost (42); PL-random - randomly sampled trajectory that is close to the physical limits of the system; star, figure 8 - heuristics-based trajectories presented in Fig. 1; random - randomly sampled trajectory that satisfies the constraints. Top left: GPS position integrated RMSE, top right: GPS position final RMSE, bottom left: accelerometer bias integrated RMSE, bottom right: accelerometer bias final RMSE.

10 minutes with our method. The main reason for this is the computational load of the EKF, including matrix inversion at every step, which is more expensive than the integration of the local observability Gramian used in our approach. The integrated RMSE of the accelerometer bias \mathbf{b}_a also suggests that our approach is able to make this state converge faster than in other methods. Nevertheless, a few other trajectories such as covariance-trace-based and PL-random were able to perform well in this test. This is also visible in the final RMSE of the accelerometer bias \mathbf{b}_a where the first four methods yield similar results. While our method is slightly worse than the covariance-trace-based and the two heuristic-based approaches, one needs to take into account that our method was optimizing for the \mathbf{p}_i^p objective.

The suboptimal performance of the covariance-based method can be explained by the linearization and Gaussian assumptions of the EKF. These assumptions potentially introduce inconsistencies to the estimator, in particular for highly nonlinear systems. Therefore, covariance-based optimization can produce a trajectory where the EKF under- or overestimates the true state covariance, thus producing a final estimate with worse RMSE than our method.

8.4.2 Visual-Inertial System Results for the same experiment on the visual-inertial system are shown in Fig. 7. In this case we use the multi-state E^2 LOG scaling to optimize our trajectory for all self-calibration states jointly. The optimized trajectory performs best in both integrated and final error in all self-calibration parameters, except for final error in b_a where the EKF-trace optimized trajectory is slightly better. Note that each boxplot represents one trajectory, so larger interquartile range indicates that the accuracy of state estimate when executing that trajectory varies

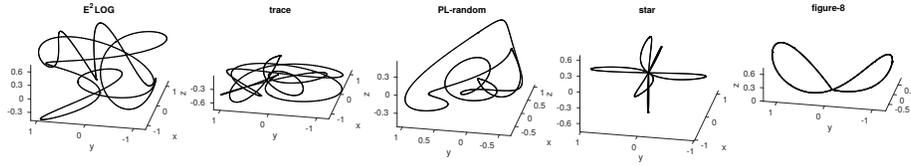


Figure 6. Trajectories used for the comparison in Fig. 7. Each trajectory lasts 15 seconds and is tight against at some physical or box constraint.

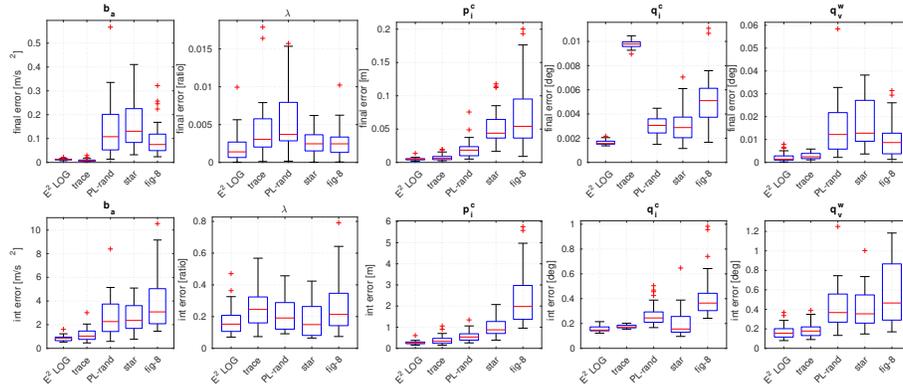


Figure 7. Quartile box plots summarizing final (top) and integrated (bottom) error of EKF state estimates, aggregated over 30 simulated runs. E^2LOG : trajectory jointly optimized for all self-calibration states using our framework. *trace*: trajectory minimizing final trace of EKF covariance in all self-calibration states. *PL-rand*: random trajectories near the system physical limits. *star*, *fig-8*: common manually designed heuristic self-calibration trajectories.

widely depending on the ground truth values and initialization. These results indicate that the E^2LOG -optimized closed-loop trajectory renders the self-calibration states more well observable than other trajectories.

An especially interesting result is the trajectories optimized for the visual scale parameter λ . An example is shown in Fig. 8. The optimizer exploits the entire optimization space to provide best information – linear acceleration input in the this case – while respecting the box and dynamic constraints.

8.5 Planning in a corridor

We demonstrate the corridor planning application on the visual-inertial system using a manually-designed corridor of moderate complexity. As a baseline, we use a minimum-snap trajectory. Minimum-snap trajectory planning with piecewise polynomials is the prevailing method of energy-minimizing trajectory planning for quadrotors Mellinger and Kumar (2011); Richter et al. (2013). However, the energy-minimizing characteristic that makes these trajectories desirable for graceful flight or aggressive maneuvers can also lead to trajectories that do not excite the system sufficiently to render the self-calibration states well observable. We compare this to a trajectory from our framework, optimized for the multi-state E^2LOG of all self-calibration states. The corridor and trajectories are visualized in Fig. 9. The min-snap

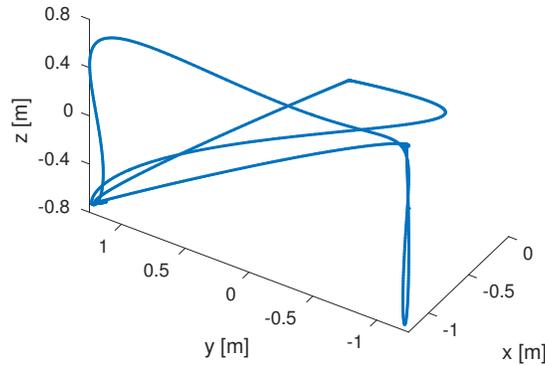


Figure 8. Self-calibration trajectory optimized for estimation of the visual scale parameter λ in the visual-inertial system. The trajectory fully exploits, but does not violate, the box constraints to provide best input (acceleration in this case) to the system.

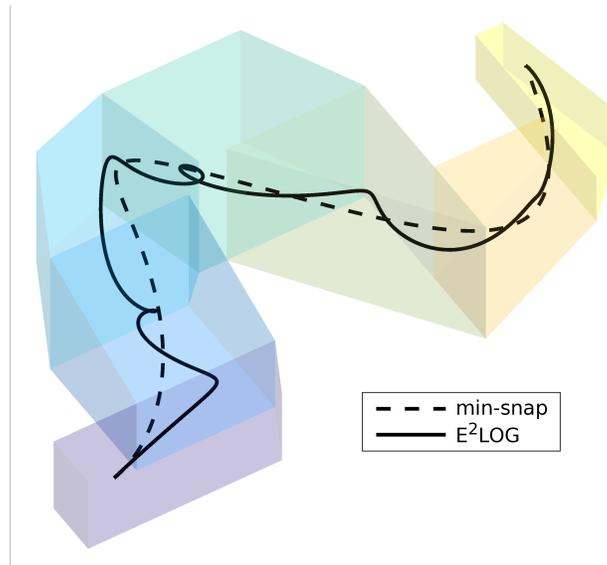


Figure 9. Minimum-snap and E^2 LOG-optimized trajectories for a UAV navigation task. The E^2 LOG trajectory optimizes our proposed observability-based cost function for a visual-inertial navigation system while remaining inside a corridor of convex polytopes. The additional movement produces better observability of proprioceptive self-calibration states, leading to $2\times$ lower position estimate error in a simulated EKF at the end of the trajectory.

trajectory is characteristically smooth. In contrast, the E^2 LOG-optimized trajectory displays significant additional movement to excite the system while remaining safely inside the corridor.

We compare the two trajectories' fitness for self-calibration by simulating the EKF over multiple trials ($N = 30$) for both trajectories. For each trial, we randomly sample ground truth self-calibration parameters \mathbf{x}_{sc} according to Table 1. The EKF is initialized with an incorrect estimate of the ground truth states, with the errors sampled

param.	distribution	error dist.	unit	description
b_ω	$\mathcal{N}(0, 0.286)$	$\mathcal{N}(0, 0.029)$	deg/s	gyroscope bias
b_a	$\mathcal{N}(0, 0.1)$	$\mathcal{N}(0, 0.02)$	m/s ²	accelerometer bias
λ	$\mathcal{N}(1, 0.1)$	$\mathcal{N}(0, 0.05)$	ratio	visual scale
p_i^c	$\mathcal{N}(0, 0.1)$	$\mathcal{N}(0, 0.02)$	meter	vision-IMU position
q_i^c	$\in (S)^3$	≈ 3	deg	vision-IMU attitude
q_v^w	$\in (S)^3$	≈ 1	deg	vision-world attitude

Table 1. Distributions of randomly sampled self-calibration parameters and initial self-calibration estimate errors for simulation experiments.

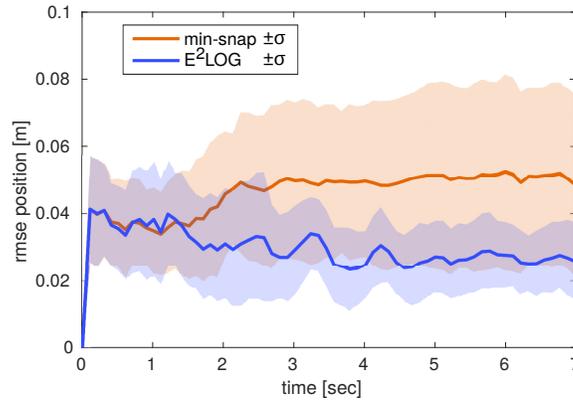


Figure 10. RMSE of position estimate in EKF averaged over 30 simulated trials for min-snap and E²LOG-optimized trajectories shown in Fig. 9. Ground truth self-calibration states and EKF initialization errors are randomly sampled for each trial. One- σ bands illustrate the variation in estimation accuracy over different trials.

randomly according to Table 1. To present the results concisely, we use the root-mean-square error of the EKF position estimate as a proxy for the overall accuracy of the self-calibration estimate. This corresponds with the end purpose of self-calibration, which is to improve the estimation quality of the robot’s fundamental configuration-space states.

Results are shown in Fig. 10. We plot the mean RMS error of the position estimate at each timestep. The shaded bands indicate the standard deviation of RMS errors across the trials at each timestep. These plots show that the E²LOG-optimized trajectory is generally able to improve its position estimate over time by correcting the initial self-calibration estimate errors in the EKF. In contrast, the min-snap trajectory displays poor convergence. The wider standard deviations indicate that the EKF is unable to correct the initialization errors and is thus highly sensitive to the correctness of the initial calibration.

8.6 Online Replanning

In this section, we demonstrate an application of the online replanning approach described in Sec. 5.4 to the visual-inertial system. We start with the same corridor and minimum-snap trajectory illustrated in Fig. 9. We introduce a small shift in the

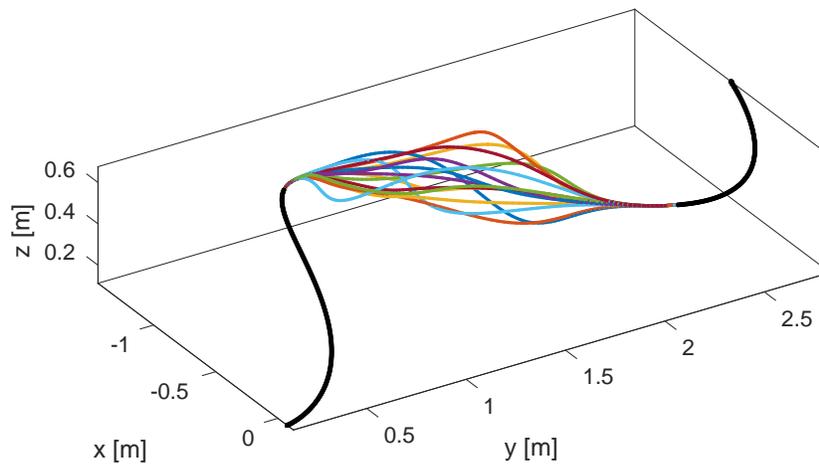


Figure 11. Set of 15 candidate modifications for two polynomial pieces in a seven-piece trajectory. 100 trajectories satisfying the linear constraints were sampled, of which 85 were rejected due to violating the nonlinear physical constraints.

ground truth vision-IMU orientation parameter \mathbf{q}_i^c on the order of 2 to 5 degrees, as might be caused by flight vibrations if the equipment mounting is not secure. The change is modeled as a linear shift, beginning 2 seconds into the flight and lasting for 1 second until the parameter stabilizes again. We suppose that a method is available that detects the increased uncertainty in state estimation and triggers observability-aware online replanning. We then use the sampling-based approach to replan the next two polynomial pieces. A sampled set of candidate trajectories V_m for this problem is illustrated in Fig. 11.

We illustrate results from this experiment in Fig. 12. In this experiment, we initialize the EKF with the correct values for the self-calibration parameters so we can focus on the effect when a sudden error is introduced. The shift in \mathbf{q}_i^c is introduced in the time interval highlighted by the light red box. The estimate errors in both \mathbf{q}_i^c and in the robot position \mathbf{p} increase quickly. The system plans the modified trajectory pieces in 0.4 seconds of computation time and flies those pieces in the time interval highlighted by the grey box. We see that the observability-aware maneuver causes a dramatic improvement in the \mathbf{q}_i^c estimate error and a significant reduction in the RMS error of the position estimate as well. In contrast, when flying the min-snap trajectory, the errors remain large because the trajectory does not excite the system enough to render the parameter well observable. The shaded one- σ bands represent the aggregated results over multiple trials in which we randomized the ground truth self-calibration values and the magnitude and direction of the \mathbf{q}_i^c shift. We also provide a breakdown of the computation time in Tab. 2.

9 Real-Robot Experimental Results

In order to show the applicability and effectiveness of our method on real robots, we deploy it on a Crazyflie 2.0, a small quadrotor that weighs 27 grams and

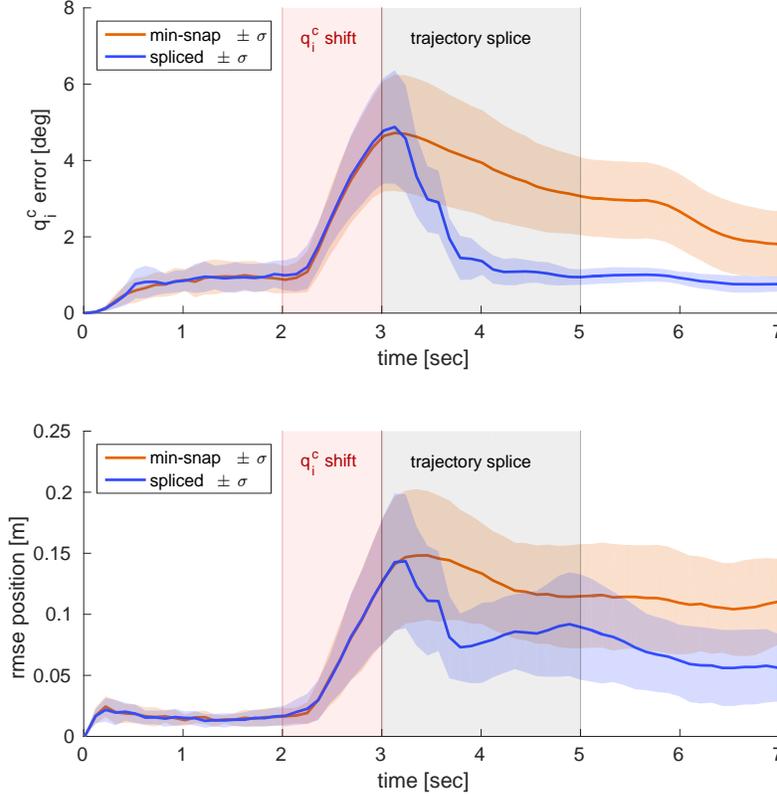


Figure 12. Simulation results for online replanning. During the 2-3sec interval, the ground truth camera-IMU orientation q_i^c shifts between 2-5 degrees in a random direction. We assume that the state estimation system is self-monitoring and detects the problem. The online replanning system replaces the next two polynomial pieces of the trajectory with an observability-optimized movement. Error in q_i^c estimate (top) and in overall position \mathbf{p} (bottom) are plotted over the duration of the trajectory. One- σ bands illustrate the variation in estimation accuracy with different random q_i^c shifts and noisy measurements, but same trajectories.

sampling (sec)	filtering (sec)	ranking (sec)	total time (sec)	sampled (count)	rejected (percent)	best σ_{\min} —
0.243	0.078	0.039	0.360	100	82.37	1.473
± 0.024	± 0.002	± 0.008	± 0.023	—	± 3.94	± 0.202

Table 2. Breakdown of computation time for online replanning over 30 trials with the same problem instance but different random seeds. Top row: mean. Bottom row: standard deviation. Maximum total computation time for any trial is 0.42 seconds.

measures 92 millimeters between diagonally opposed rotors. Our experiment compares performance of figure-eight trajectory, star trajectory, and an optimized trajectory from our framework on the task of estimating the position of the GPS sensor in the IMU frame \mathbf{p}_i^p . All trajectories have 5-second duration. To provide a fair comparison, we adjust the heuristic trajectories until they reach the same physical constraint limits

	figure 8	star	ours
\mathbf{p}_i^p	$0.102 \pm$	$0.105 \pm$	$0.055 \pm$
RMSE[m]	0.033	0.036	0.025
\mathbf{p}_i^p cov.	$0.0190 \pm$	$0.0177 \pm$	$0.0093 \pm$
trace[m ²]	0.0009	0.0007	0.0004

Table 3. Self-calibration task on a real quadrotor: statistics of the final RMSE values and final trace of the covariance matrix for GPS sensor position in the IMU frame \mathbf{p}_i^p . Data collected for 8 trials with random \mathbf{p}_i^p values sampled uniformly from the 0.3m sphere. All trajectories last 5 seconds.

used to generate the optimal trajectory. We focus on the \mathbf{p}_i^p self-calibration parameter because, unlike IMU biases, we can generate its ground-truth value.

The experiment consists of 8 trials performed in a motion capture setup. For each trial we generate a ground truth \mathbf{p}_i^p by taking a uniform random sample from the 0.3m sphere. We upload the target trajectory onto the vehicle, which performs estimation and control onboard using the previously described EKF and the trajectory tracking controller proposed in Mellinger and Kumar (2011). We simulate GPS measurements by corrupting motion capture position measurements with white Gaussian noise of 0.1m standard deviation, adding the groundtruth \mathbf{p}_i^p in the local coordinate frame, and throttling measurements to 10Hz. In each trial, we fly the three trajectories and log the filter’s \mathbf{p}_i^p estimate over time.

The results in Tab. 3 show that, with the optimized trajectory, the EKF estimates the GPS sensor position with 2x lower RMSE than other trajectories. In addition, our method yields smaller standard deviation of estimate errors and significantly smaller mean trace of the \mathbf{p}_i^p estimate covariance. This indicates that our trajectory produces more consistent results and is more certain about the converged values. Some of the real robot experiments are shown in the video: <https://youtu.be/v8UkOtRJEsw>.

10 Conclusion

In this work, we introduced a framework to optimize robot trajectories for estimation of self-calibration states. Our method is applicable to any locally observable nonlinear system with smooth dynamics and a differentiable sensor model. Our observability-aware E²LOG cost function employs a short-horizon Taylor expansion to capture the interactions between the self-calibration states, system dynamics, and measurements, allowing us to optimize for states that do not directly appear in the measurement model. We also described a statistical scaling technique to jointly optimize for multiple self-calibration states with varying units, ranges, and measurement functions. We applied the E²LOG objective to several trajectory optimization formulations: a piecewise polynomial null-space basis for optimized self-calibration loop trajectories, a Bézier spline basis for calibration-aware navigation through an obstacle-free corridor, and a sampling-based online replanning method that can modify an existing trajectory plan in less than 1/2 second to improve observability while respecting corridor and dynamic constraints.

We demonstrated our methods in extensive simulation experiments with both GPS-IMU and visual-inertial odometry sensor suites. Our experiments showed a strong correlation between the E²LOG cost function value and the estimation accuracy of self-calibration states in an EKF, and validated that our statistical scaling technique produces trajectories that balance convergence of all the self-calibration states simultaneously. We showed that a single joint-optimized closed loop trajectory from our framework outperforms a trajectory minimizing the EKF covariance trace and common heuristic self-calibration trajectories, while taking $\sim 50\times$ less time to optimize than the EKF-based cost function.

For the corridor navigation problem, we showed example results from our method and analyzed the overall performance of the trajectories by considering the error of the vehicle position estimate. Our optimized trajectory finished with $2\times$ better RMS position error than a minimum-snap trajectory, showing that a calibration-aware trajectory can help a mobile robot maintain an accurate overall state estimate while moving collision-free through the environment. We also demonstrated that our online replanning framework can modify a short segment of a corridor-constrained trajectory for improved observability. We simulated a scenario where a self-calibration parameter is perturbed and a hypothetical self-monitoring state estimation system detects the increased uncertainty. The replanned trajectory caused the estimator to converge on the new correct value whereas the energy-minimizing trajectory retained the incorrect estimate.

Finally, we evaluated our method on a real quadrotor, using a motion capture system to simulate noisy GPS data with known ground truth self-calibration values. The experiments on the real quadrotor confirmed the simulation results, yielding $2\times$ improvement over the heuristic trajectories in terms of the GPS-IMU position state estimation.

Acknowledgements

The authors would like to thank Nicholas Rotella for numerous discussions on the non-linear observability analysis and Christoph Böhm for the evaluation of various cost functions.

References

- Achtelik M, Weiss S, Chli M and Siegwart R (2013) Path planning for motion dependent state estimation on micro aerial vehicles. In: *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*. pp. 3926–3932.
- Bähnemann R, Burri M, Galceran E, Siegwart R and Nieto JJ (2017) Sampling-based motion planning for active multirotor system identification. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 3931–3938.
- Beinhofer M, Müller J, Krause A and Burgard W (2013) Robust landmark selection for mobile robot navigation. In: *In Int. Conf. on Intelligent Robots and Systems (IROS)*. pp. 3637–2643.
- Bélisle CJP, Romeijn HE and Smith RL (1993) Hit-and-run algorithms for generating multivariate distributions. *Mathematics of Operations Research* 18(2): 255–266.
- Botev ZI (2017) The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79(1): 125 – 148.
- Brumback B and Srinath MD (1987) A chi-square test for fault-detection in kalman filters. *IEEE Transactions on Automatic Control* 32(6): 552–554.
- Bry A and Roy N (2011) Rapidly-exploring random belief trees for motion planning under uncertainty. In: *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*. pp. 723–730.
- Bryson M and Sukkarieh S (2008) Observability analysis and active control for airborne SLAM. *IEEE Transactions on Aerospace and Electronic Systems* 44(1): 261–280. DOI: 10.1109/TAES.2008.4517003.
- Butcher JC (1996) A history of runge-kutta methods. *Applied numerical mathematics* 20(3): 247–260.
- Costante G, Forster C, Delmerico J, Valigi P and Scaramuzza D (2016) Perception-aware path planning. *arXiv preprint arXiv:1605.04151*.
- Deits R and Tedrake R (2015) Computing large convex regions of obstacle-free space through semidefinite programming. In: *Algorithmic Foundations of Robotics XI*. Springer, pp. 109–124.
- Flores ME (2008) *Real-time trajectory generation for constrained nonlinear dynamical systems using non-uniform rational b-spline basis functions*. Ph.D. Dissertation, California Institute of Technology.
- Hausman K, Müller J, Hariharan A, Ayanian N and Sukhatme G (2015) Cooperative multi-robot control for target tracking with onboard sensing. *The International Journal of Robotics Research* 34(13): 1660–1677.
- Hausman K, Preiss JA, Sukhatme GS and Weiss S (2017) Observability-aware trajectory optimization for self-calibration with application to uavs. *IEEE Robotics and Automation Letters* 2(3): 1770–1777.
- Hausman K, Weiss S, Brockers R, Matthies L and Sukhatme G (2016) Self-calibrating multi-sensor fusion with probabilistic measurement validation for seamless sensor switching on a uav. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Stockholm, Sweden, pp. 4289–4296.
- Hermann R and Krener A (1977) Nonlinear controllability and observability. *IEEE Transactions on automatic control* 22(5): 728–740.
- Hernandez J, Tsotsos K and Soatto S (2015) Observability, identifiability and sensitivity of vision-aided inertial navigation. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 2319–2325.
- Hesch JA, Kottas DG, Bowman SL and Roumeliotis SI (2014) Camera-imu-based localization: Observability analysis and consistency improvement. *International Journal of Robotics Research* 33(1): 182–201.

- Hinson B and Morgansen K (2013) Observability optimization for the nonholonomic integrator. In: *American Control Conference (ACC), 2013*. pp. 4257–4262.
- Hinson BT, Binder MK and Morgansen KA (2013) Path planning to optimize observability in a planar uniform flow field. In: *2013 American Control Conference*. pp. 1392–1399.
- Hollinger GA and Sukhatme GS (2013) Sampling-based motion planning for robotic information gathering. In: *Robotics: Science and Systems IX*.
- Indelman V, Carlone L and Dellaert F (2015) Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments. *The International Journal of Robotics Research* 34(7): 849–882.
- Joy KI (2000) Bernstein polynomials. *On-Line Geometric Modeling Notes* 13.
- Julian BJ, Angermann M, Schwager M and Rus D (2012) Distributed robotic sensor networks: An information-theoretic approach. *I. J. Robotics Res.* 31(10): 1134–1154.
- Karaman S and Frazzoli E (2011) Sampling-based algorithms for optimal motion planning. *Int. Journal of Robotics Research* 30(7): 846–894.
- Kelly J and Sukhatme GS (2011) Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *I. J. Robotics Res.* 30(1): 56–79.
- Khosoussi K, Sukhatme GS, Huang S and Dissanayake G (2016) Designing sparse reliable pose-graph SLAM: A graph-theoretic approach.
- Klein G and Murray D (2007) Parallel tracking and mapping for small ar workspaces. In: *IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE, pp. 225–234.
- Krener A and Ide K (2009) Measures of unobservability. In: *Proceedings of the 48th IEEE Conference on Decision and Control*. pp. 6401–6406.
- Lefferts EJ, Markley FL and Shuster MD (1982) Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics* 5(5): 417–429.
- Lynen S, Achtelik M, Weiss S, Chli M and Siegwart R (2013) A robust and modular multi-sensor fusion approach applied to mav navigation. In: *In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. pp. 3923–3929.
- Martin P, Murray RM and Rouchon P (2003) Flat systems, equivalence and trajectory generation. Technical report, California Institute of Technology.
- Martinelli A (2011) State estimation based on the concept of continuous symmetry and observability analysis: The case of calibration. *Robotics, IEEE Transactions on* 27(2): 239–255.
- Martinelli A and Siegwart R (2006) Observability properties and optimal trajectories for on-line odometry self-calibration. In: *Decision and Control, 2006 45th IEEE Conference on*. pp. 3065–3070.
- Maye J, Sommer H, Agamennoni G, Siegwart R and Furgale PT (2016) Online self-calibration for robotic systems. *The International Journal of Robotics Research* 35(4): 357–380.
- Mellinger D and Kumar V (2011) Minimum snap trajectory generation and control for quadrotors. In: *In Int. Conf. on Robotics and Automation (ICRA)*. IEEE, pp. 2520–2525.
- Müller J and Sukhatme G (2014) Risk-aware trajectory generation with application to safe quadrotor landing. In: *In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. pp. 3642–3648.
- Nobre F, Kasper M and Heckman C (2017) Drift-correcting self-calibration for visual-inertial SLAM. In: *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*. pp. 6525–6532.
- Preiss J, Hausman K, Sukhatme G and Weiss S (2017) Trajectory optimization for self-calibration and navigation. In: *Proceedings of Robotics: Science and Systems*. Cambridge, Massachusetts.
- Ratliff ND, Zucker M, Bagnell JA and Srinivasa SS (2009) CHOMP: gradient optimization techniques for efficient motion planning. In: *IEEE International Conference on Robotics*

- and Automation, (ICRA). pp. 489–494.
- Richter C, Bry A and Roy N (2013) Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In: *In Proc. of the International Symposium on Robotics Research (ISRR)*. pp. 649–666.
- Schneider T, Li M, Burri M, Nieto JJ, Siegwart R and Gilitschenski I (2017) Visual-inertial self-calibration on informative motion segments. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 6487–6494.
- Sun W, Patil S and Alterovitz R (2015) High-frequency replanning under uncertainty using parallel sampling-based motion planning. *IEEE Transactions on Robotics* 31(1): 104–116. DOI:10.1109/TRO.2014.2380273.
- Tang S and Kumar V (2016) Safe and complete trajectory generation for robot teams with higher-order dynamics. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 1894–1901.
- Todorov E and Li W (2005) A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In: *Proceedings of the American Control Conference*, volume 1. pp. 300–306.
- Travers M and Choset H (2015) Use of the nonlinear observability rank condition for improved parametric estimation. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 1029–1035. DOI:10.1109/ICRA.2015.7139303.
- Trefethen L and Bau D (1997) *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics. ISBN 9780898713619.
- Tribou MJ, Wang DWL and Waslander SL (2015) Degenerate motions in multicamera cluster SLAM with non-overlapping fields of view. *CoRR* abs/1506.07597.
- Vondrák J (2007) Submodularity in combinatorial optimization .
- Weiss S (2012) *Vision Based Navigation for Micro Helicopters*. Phd thesis, ETH Zurich.
- Weiss S, Achtelik M, Lynen S, Achtelik M, Kneip L, Chli M and Siegwart R (2013) Monocular vision for long-term micro aerial vehicle state estimation: A compendium. *Journal of Field Robotics* 30(5): 803–831.

A Hardness of Graph Approximation

In this appendix, we construct a graph discretization of the observability-aware trajectory optimization problem and show that it is NP-complete. We call this problem OBSERVABILITY-PATH. While this result does not directly apply to the continuous form of the problem, it illustrates the fundamental difference between observability-maximizing trajectory optimization and time- or energy-minimizing trajectory optimization.

Let $G = (V, E)$ be a connected, directed graph where each vertex $v \in V$ is associated with a valid location in the system's state space. For each $v_1, v_2 \in V$, the directed edge $e = (v_1, v_2)$ exists in E if there is a collision-free trajectory segment from v_1 to v_2 that respects the system's dynamic limits (e.g. maximum actuator force). Note that the state space may include velocities, etc., as needed to ensure that a graph path corresponds to a physically plausible trajectory.

Each edge $e \in E$ is associated with a symmetric positive semidefinite $n \times n$ matrix K_e corresponding to the E²LOG integrated over the trajectory segment. A path $p = e_1, e_2, \dots, e_k$ is mapped to the objective:

$$\sigma_{\min}(W_o(p)), \text{ where } W_o(p) = (K_{e_1} + K_{e_2} + \dots + K_{e_k}), \quad (43)$$

where σ_{\min} denotes the smallest singular value. We wish to find the path between two vertices $s, t \in V$ that maximizes this objective. If the covering of the free space by V is sufficiently dense, then this problem approximates the continuous problem reasonably. In the decision version of OBSERVABILITY-PATH, we query if an $s - t$ path p with $\sigma_{\min}(W_o(p)) \geq \alpha$, $\alpha \in \mathbb{R}_{>0}$ exists.

Theorem 1. OBSERVABILITY-PATH is NP-complete.

Proof. Given a path p of ℓ edges, we can sum the matrices K_{e_i} in $O(\ell n^2)$ time, compute the singular values of $W_o(p)$ in $O(n^3)$ time (Trefethen and Bau 1997), and find the minimum in $O(n)$ time. Thus, OBSERVABILITY-PATH is in NP.

We give a reduction from the well-known NP-complete problem SET-COVER to OBSERVABILITY-PATH. In SET-COVER, we are given a set U indexed by $i \in 1 \dots n$, a collection of subsets $S \subseteq 2^U$ indexed by $j \in 1 \dots m$, and a positive integer $L \in \mathbb{N}^+$. We wish to decide if there exists a cover $C \subseteq S$ such that $|C| \leq L$ and $\bigcup C = U$.

To each set $s_j \in S$ we associate the diagonal $n \times n$ matrix K_j , with diagonal entries defined by:

$$K_j[i, i] = \begin{cases} 1 & : u_i \in s_j \\ 0 & : \text{otherwise} \end{cases} \quad (44)$$

for all $u_i \in U$. Note that the smallest singular value $\sigma_{\min}(K_j) = 0$ unless $s_j = U$. We construct a directed graph following the illustration in Fig. 13. We construct L "layers" where layer ℓ contains a directed edge $e_{j,\ell}$ for each $s_j \in S$. The edge $e_{j,\ell}$ connects the vertices $v_{j,\ell}, v_{j,\ell'}$ and is associated with matrix K_j . We connect layer ℓ to layer $\ell + 1$ with a complete bipartite subgraph such that $v_{j,\ell'}$ is connected to $v_{\gamma,\ell+1}$ for all $j, \gamma \in 1 \dots m$. All of the edges in this bipartite subgraph are associated with a zero matrix. Finally, we construct the source s connected to all $v_{j,1}$ with zero-matrix edges, and the sink t connected to all $v_{j,L'}$ with zero-matrix edges.

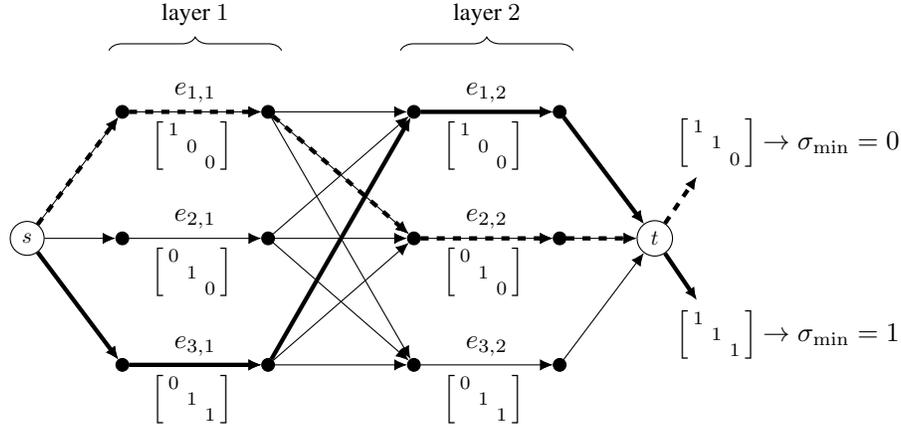


Figure 13. Example of graph used to reduce SET-COVER to OBSERVABILITY-PATH. In this small problem, $U = \{1, 2, 3\}$, $L = 2$, and $S = \{\{1\}, \{2\}, \{2, 3\}\}$. Edges without a matrix drawn correspond to zero matrices. The dotted path p_d corresponds to the choice of sets $\{1\}$ and $\{2\}$. It is not a covering of U , and $W_o(p_d) = 0$. The solid path p_s corresponds to the choice of sets $\{1\}$ and $\{2, 3\}$. It is a covering of U , and $W_o(p_s) = 1$.

This graph has $O(Lm)$ vertices and $O(Lm^2)$ edges, and $O(Lmn^2)$ time needed to construct the K_j edge matrices.

Any $s - t$ path p must contain exactly L of the edges $e_{j,\ell}$ and corresponds to a choice C of between 1 and L elements of S , with some elements possibly chosen more than once. By the construction of the K_j , the matrix sum $W_o(p)$ will contain nonzero entries only for those diagonal elements $W_o[i, i]$ where $u_i \in s$ for some $s \in C$. Since the singular values of a diagonal matrix are exactly the diagonal entries, $\sigma_{\min}(W_o(p)) \geq 0$ if and only if C is a cover of U . Furthermore, by construction of the K_j matrices, all nonzero singular values of $W_o(p)$ must be ≥ 1 . Since OBSERVABILITY-PATH seeks to find an $s - t$ path p with $\sigma_{\min}(W_o(p)) \geq \alpha$, OBSERVABILITY-PATH with $\alpha = 1$ will find a path corresponding to an L -cover of U , if it exists. \square