

# Trajectory Optimization for Self-Calibration and Navigation

James A. Preiss, Karol Hausman, Gaurav S. Sukhatme  
Department of Computer Science  
University of Southern California, Los Angeles, USA

Stephan Weiss  
Institute of Smart System Technologies  
Alpen-Adria-Universitat Klagenfurt, Klagenfurt, Austria

**Abstract**—Trajectory generation approaches for mobile robots generally aim to optimize with respect to a cost function such as energy, execution time, or other mission-relevant parameters within the constraints of vehicle dynamics and obstacles in the environment. We propose to add the cost of state observability to the trajectory optimization in order to ensure fast and accurate state estimation throughout the mission while still respecting the constraints of vehicle dynamics and the environment. Our approach finds a dynamically feasible estimation-optimized trajectory in a sequence of connected convex polytopes representing free space in the environment. In addition, we show a statistical procedure that enables observability-aware trajectory optimization for heterogeneous states in the system both in magnitude and units, which was not supported in previous formulations. We validate our approach with extensive simulations of a visual-inertial state estimator on an aerial platform as a specific realization of our general method. We show that the optimized trajectories lead to more accurate navigation while eliminating the need for a separate calibration procedure.

## I. INTRODUCTION

Successful mobile robot control requires accurate and robust state estimation. While state estimation algorithms have received significant attention from the robotics community, it is also possible to improve state estimation from the planning perspective. A carefully selected trajectory based on information gain can provide better information to the exteroceptive sensors of the system. This so-called *active perception* has been shown to noticeably reduce failure modes of estimators in the past [3, 4, 25].

On the other hand, the accuracy of state estimation often depends not only on the perceived exteroceptive information but also on the proprioceptive system input. For mobile robots these inputs are typically inertial readings of acceleration and angular velocity, and they are a direct result of the executed trajectory and the vehicle dynamics. The proprioceptive system inputs are particularly important for more complex sensor fusion frameworks which include self-calibration states and multiple sensors. To render high-dimensional state vectors observable, non-zero system inputs are generally required [16]. As a result, to improve the estimator’s accuracy, trajectory planning should take the required system input for observability into account.

Hausman et al. [8] demonstrated an approach to generate trajectories that render specific states better observable in a system performing online self-calibration. The method respects the system’s dynamical constraints and supports waypoint navigation problems, but assumes a free space without

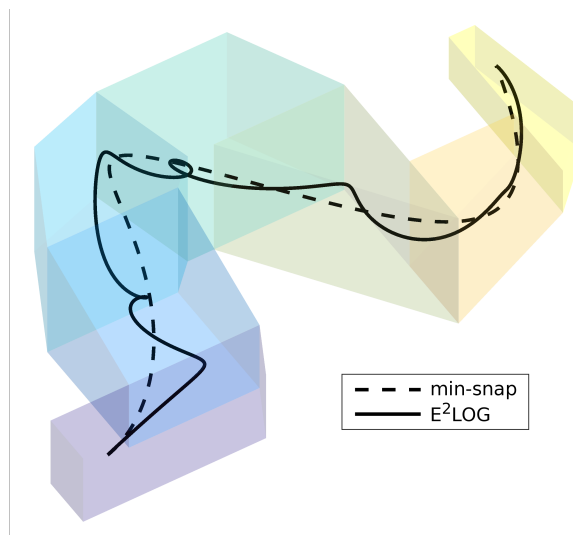


Fig. 1. Minimum-snap and  $E^2\text{LOG}$ -optimized trajectories for a UAV navigation task. The  $E^2\text{LOG}$  trajectory optimizes our proposed observability-based cost function for a visual-inertial navigation system while remaining inside a corridor of convex polytopes. The additional movement 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.

obstacles. In this paper, we remove the obstacle-free requirement and provide a method that optimizes a trajectory for state observability while moving through a free space corridor described by a sequence of convex polytopes. We also describe a 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 demonstrate our approach using a visual-inertial state estimator on an aerial platform. The proposed approach, however, is not specific to a particular system or estimator type but optimizes the trajectories by analyzing the nonlinear system dynamics and measurement model.

The key contributions of our approach are:

- An optimization framework including a multi-state balancing strategy to optimize trajectories for estimation of high-dimensional state vectors.
- A suitable formulation of the trajectory optimization in obstacle-free corridors using Bézier curves.
- Explicit application of the above general approach to a visual-inertial system on an aerial platform.

We evaluate the proposed method in extensive simulation experiments. We show that multi-state optimized trajectories effectively balance the self-calibration goals of the individual states, achieving competitive results to trajectories optimized for each state in isolation. We demonstrate that an optimized trajectory from our framework outperforms common heuristic self-calibration trajectories and a state-of-the-art optimization method. Our results on the navigation task yield  $2\times$  better position error than a common energy-efficient trajectory, showing the benefits of optimizing for navigation and self-calibration simultaneously using our approach.

## II. 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 [3, 13, 15]. More recent work considered dense photometric image information by seeking highly textured surfaces [4].

Proprioceptive methods, which include the method shown in this paper, focus on the *how* the robot should move to obtain the most accurate state estimates regardless of 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 [21], but more commonly on complex systems a sampling-based approach with simulation is used [1, 2]. 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 in [11]. These methods also inherit the potentially large computational cost of the state estimator.

In [18] the authors propose a continuous measure of observability from the non-linear observability analysis suggested in [9]. 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, [12] made use of this measure of observability to generate observability aware trajectories. The approach in [8] extends these methods to the quality of observability of hidden states, i.e. states that do not appear in the measurement model. However, the method in [8] only optimizes trajectories for estimation of a single self-calibration state at a time, and does not handle environmental obstacles. In this paper, 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 [10], the authors present an alternative approach to observability analysis based on the volume of the set of indistinguishable trajectories for a given input sequence. While their approach has some compelling advantages, such as accounting for unknown noise inputs, it requires in-depth manual derivations for each system analyzed, and is not amenable to analyzing subsets of self-calibration states. In contrast, our

method provides a “recipe” that produces a cost function in any user-chosen states from the system dynamics and measurement equations.

## III. PROBLEM FORMULATION

We assume a nonlinear system of the following form:

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

where  $\mathbf{x}$  is the state,  $\mathbf{u}$  are the control inputs,  $\mathbf{z}$  are the outputs (sensor readings) and  $\delta, \epsilon$  are noise values caused by modeling errors and non-perfect actuators. We summarize the states that have constant dynamics and are independent of the system inputs and other state variables as *self-calibration* states  $\mathbf{x}_{sc}$ .

### A. 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* 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 [18].

One way to determine whether or not a nonlinear system is weakly locally observable is based on the rank of the nonlinear observability matrix [9], constructed using the Lie derivatives of the sensor model  $h(\mathbf{x})$ :

$$O(\mathbf{x}, \mathbf{u}) = [\nabla L_0^h \quad \nabla L_1^h \quad \nabla L_2^h \quad \dots]^T, \quad (1)$$

where  $L_i^h$  is the  $i$ -th Lie derivative of the sensor model  $h(\mathbf{x})$ , defined recursively as:

$$L_0^h = h(\mathbf{x}), \quad L_{i+1}^h = \frac{\partial}{\partial t} L_i^h = \frac{\partial L_i^h}{\partial \mathbf{x}} f(\mathbf{x}, \mathbf{u}), \quad (2)$$

and  $\nabla L_i^h = \frac{\partial L_i^h}{\partial \mathbf{x}}$ . In nonlinear systems, the nonlinear observability is only a local property that depends on the state and the input. In this definition, the observability of a system is considered as a binary property, which makes it unsuitable for gradient-based optimization methods.

## IV. EXPANDED EMPIRICAL LOCAL OBSERVABILITY GRAMIAN (E<sup>2</sup>LOG)

In order to optimize trajectories for state estimation, we use the notion of quality of observability [8, 18]. A state is considered *poorly* observable if it leads to a small change in the output, even when extensively perturbed [18]. On the other hand, if a state causes a significant change in the output, even when marginally perturbed, this state is considered *well* observable. Well-observable states are easy to estimate even in the presence of high measurement noise [29].

Following the derivation presented in [8] and using the Taylor expansion of the sensor model about a point  $t_0$ , one can show that the Jacobian of the sensor model  $h(\mathbf{x})$  with respect to the state  $\mathbf{x}$  around the time  $t_0$  is:

$$\frac{\partial}{\partial \mathbf{x}} h_{t_0}(t) = \sum_{i=0}^n \frac{(t-t_0)^i}{i!} \nabla L_i^h(t). \quad (3)$$

In addition to showing the effect of the states that directly influence the measurement, Eq. 3 also reveals the effects of the varying control inputs and the states that are not included in the sensor model.

In order to model the interactions between the influences that different states can have on the output, we use the local observability Gramian [18]. We introduce the following notation for brevity:

$$K_{t_0}(t) = \frac{\partial}{\partial \mathbf{x}} h_{t_0}(t) = \frac{\partial}{\partial \mathbf{x}} h_{t_0}(\mathbf{x}(t), \mathbf{u}(t)). \quad (4)$$

Following [8], we use the Taylor expansion of the sensor model to approximate the local observability Gramian:

$$W_o(T, \Delta t) \approx \int_0^T K_t(t + \Delta t)^T K_t(t + \Delta t) dt, \quad (5)$$

where  $\Delta t$  is a fixed horizon, chosen empirically, that enables us to see the effects of the system dynamics.

We refer to this approximation of the local observability Gramian as *Expanded Empirical Local Observability Gramian* (E<sup>2</sup>LOG). To measure the quality of observability we use the smallest singular value of the E<sup>2</sup>LOG  $W_o(T, \Delta t)$ . Intuitively, the observability Gramian 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 dimension of  $\mathbf{x}_{sc}$ . In contrast, maximizing the condition number or trace of the E<sup>2</sup>LOG are less appropriate. The condition number captures only the ratio between the most observable and least observable subspaces. Similarly, maximizing the trace may reward trajectories that render one state very well observable while other states are unobservable.

In contrast to the empirical local observability Gramian (ELOG) proposed in [18], the E<sup>2</sup>LOG formulation is able to capture input-output dependencies that are not visible in the sensor model. This property is achieved by incorporating higher order Lie derivatives and evaluating the Taylor expansion over a short time horizon. These states can only be captured in the ELOG if the trajectory is generated by integrating the sequence of controls  $\mathbf{u}$ , but this results in numerical integration errors, especially for high-order systems. We refer the reader to [8] for more details on E<sup>2</sup>LOG.

## V. MULTI-STATE E<sup>2</sup>LOG

In general, entries in the  $K_t$  matrices Eq. 4 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 [18], scaling of these states is needed to ensure that the E<sup>2</sup>LOG smallest-singular-value metric balances the influence of all states equally. We introduce a column scaling in the form of

$$K'_t = K_t \mathbf{diag}(s)^{-1} \quad (6)$$

as each column of  $K_t$  reflects the sensitivity of the measurement function with respect to one state. The values of  $s$  are determined empirically by the following procedure:

- Generate a set of  $n$  physically plausible random trajectories ranging from stationary to near the physical limits of the dynamic model.
- For each trajectory, randomly sample  $m$  sets of realistic self-calibration parameter values.
- For each trajectory-parameter pair, evaluate  $K$  at  $p$  points.
- Let  $s_i$  be the standard deviation of all entries in the  $i^{th}$  column of all  $n \cdot m \cdot p$  generated  $K$  matrices.

This procedure approximates a uniform sampling from the distribution of  $K$  matrices for the given system. In Sec. VIII, we demonstrate that for our example system, 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 on the  $K_t$  matrices. In particular, states that minimally contribute to the magnitude of change of the measurement may be swamped by other states than 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 E<sup>2</sup>LOG objective, returning results close to the initial guess. By applying a scaling factor to the columns of the  $K_t$  matrix, we retain important properties of  $K_t$  such as the ratio between different partial measurement derivatives w.r.t. different states, while improving the behavior of E<sup>2</sup>LOG as an optimization objective. It is sufficient to perform this procedure once for a given system setup and use the stored  $s$  vector for different problem instances.

## VI. POLYNOMIAL TRAJECTORY BASES FOR OPTIMIZATION

In this section, we introduce trajectory optimization methods for optimizing E<sup>2</sup>LOG 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 [20, 22]. However, we emphasize that the E<sup>2</sup>LOG 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  $\mathbf{y}(t)$ , the system states  $\mathbf{x}(t)$  and control inputs  $\mathbf{u}(t)$  can be computed as functions of the flat outputs  $\mathbf{y}$  and a finite number of their derivatives:

$$\mathbf{x} = \zeta(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}, \dots, \overset{(n)}{\mathbf{y}}), \quad \mathbf{u} = \psi(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}, \dots, \overset{(m)}{\mathbf{y}}). \quad (7)$$

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.

The two trajectory representations described here 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.

### A. Piecewise Polynomial Null-Space Basis

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

$$y(t) = \begin{cases} p_1^T \mathbf{t}(t) & \text{if } t_0 \leq t < t_1 \\ \vdots & \\ p_q^T \mathbf{t}(t) & \text{if } t_{q-1} \leq t \leq t_q, \end{cases} \quad (8)$$

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

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

As detailed in [8, 23], waypoint and continuity constraints on the trajectory can be represented as a linear system:

$$A [p_1^T \quad \dots \quad p_q^T]^T \triangleq Ap = b. \quad (9)$$

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

$$p = p^* + \text{Null}(A)\rho, \quad (10)$$

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  $\rho$ . The one-dimensional formulation given here extends naturally to higher-dimensional outputs. Since the E<sup>2</sup>LOG objective function does not have an easily computed gradient with respect to the polynomial coefficients  $p$ , we approximate the gradient by forward differences; therefore reducing the number of variables speeds up optimization significantly.

### B. 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, planning with 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 [24], 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 [6, 26] 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 [5]. We seek a trajectory from start to goal that minimizes our E<sup>2</sup>LOG cost function while remaining inside this corridor. From a high-level planner, we obtain the start and goal positions  $p_{start}, p_{goal} \in \mathbb{R}^k$ , and a sequence of  $n$  convex polytopes  $C$ :

$$C = \mathcal{P}_1, \dots, \mathcal{P}_n, \quad \mathcal{P}_i = \{x \in \mathbb{R}^k : A_i x \leq b_i\}, \quad (11)$$

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  $p_{start}$  to  $p_{goal}$  exist in  $C$ :

$$\mathcal{P}_i \cap \mathcal{P}_{i+1} \neq \emptyset, \quad p_{start} \in \mathcal{P}_1, \quad p_{goal} \in \mathcal{P}_n. \quad (12)$$

Note that the requirement of overlap between adjacent  $\mathcal{P}_i$  is sufficient to ensure that a 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 an  $n$ -piece polynomial trajectory such that the  $i^{\text{th}}$  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  $x_i \in \mathbb{R}^k$  and a fixed set of Bernstein polynomials, such that

$$f(t) = b_{0,d}(t)x_0 + b_{1,d}(t)x_1 + \dots + b_{d,d}(t)x_d \quad (13)$$

for  $t \in [0, 1]$ , where each  $b_{i,d}$  is a degree- $d$  polynomial with coefficients given in [14]. This form may be interpreted as a smooth interpolation between  $x_0$  and  $x_d$ . The curve begins at  $x_0$  and ends at  $x_d$ . 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 [14]; thus any point in the form of Eq. 13 is a convex combination of the control points  $x_i$ . Thus, when using control points as decision variables instead of monomial coefficients, 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 take the form:

$$A_i x_j \leq b_i, \quad j \in [1 \dots d+1] \quad (14)$$

for each control point  $x_j$  in the polynomial piece corresponding to the  $i^{\text{th}}$  polytope.

Enforcing arbitrary levels of continuity in piecewise Bézier curves is also easy. The derivative of  $f(t)$  as denoted in Eq. 13 is another Bézier curve of degree  $d-1$ , with control points that are scaled forward differences of the control points of  $f(t)$ :

$$f'(t) = db_{0,d-1}(t)(x_1 - x_0) + \dots + db_{d-1,d-1}(t)(x_d - x_{d-1}) \quad (15)$$

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. (Note that  $b_{h,d}(0) = 0$  for  $h \neq 0$  and  $b_{h,d}(1) = 0$  for  $h \neq d$ .)

In comparison to the null-space formulation, the Bézier basis is desirable because it enforces 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  $x_0, \dots, x_d$  such that some  $x_i \notin \mathcal{P}$  but the Bézier curve through  $x_0, \dots, x_d$  lies inside  $\mathcal{P}$ . However, for the system 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 vehicle moving through those polytopes. This issue can be addressed by several means: 1) allocating different durations to each polytope 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, which allows the optimizer more freedom to control the relative sizes of the 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.

### C. Numerical Optimization Method

The E<sup>2</sup>LOG 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.

In closed-loop self-calibration problems using the null-space basis, we generate initial guesses of  $\rho$  by randomly sampling from a normal distribution and discarding samples that violate the nonlinear physical constraints. However, for corridor problems in the Bézier basis, the 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 practice. Instead, we minimize an integrated-squared-derivative cost function using quadratic programming as in [26] 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.

## VII. VISUAL-INERTIAL STATE ESTIMATOR AND QUADROTOR

As an example application of our method, we consider a quadrotor helicopter equipped with an Inertial Measurement Unit (IMU) and a camera. The IMU contains a gyroscope, yielding rotational velocity about the three body axes, and an accelerometer, yielding the sum of acceleration and gravity in the same axes. The camera is input to a visual odometry algorithm (e.g. [17]) which yields the system’s 3D position in undefined scale, and scale-free attitude estimations with respect to its own visual frame. The visual-inertial sensor suite is popular and performs well in practice. It also requires estimating a high-dimensional set of calibration parameters in different physical units and scales, creating a challenging task for trajectory optimization.

The full system state consists of the following:

$$\mathbf{x} = (\mathbf{p}_w^i, \mathbf{v}_w^i, \mathbf{q}_w^i, \mathbf{b}_\omega, \mathbf{b}_a, \lambda, \mathbf{p}_i^c, \mathbf{q}_i^c, \mathbf{p}_v^w, \mathbf{q}_v^w), \quad (16)$$

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,  $\mathbf{b}_\omega$  and  $\mathbf{b}_a$  are the gyroscope and accelerometer biases,  $\lambda$  is the visual scale, and  $\mathbf{p}_i^c, \mathbf{q}_i^c$  are the relative position and orientation between the camera and the IMU in the IMU frame.  $\mathbf{q}_v^w$  can be seen as the direction of the gravity vector in the visual map [16] 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 [28].

The state is governed by differential equations described in [16, 27]. To account for their temporal variations, the IMU biases are modeled as random processes. In this setup, the self-calibration states  $\mathbf{x}_{sc}$  are the gyroscope and accelerometer biases  $\mathbf{b}_\omega, \mathbf{b}_a$ , the visual scale  $\lambda$ , pose of the camera sensor in the IMU frame  $\mathbf{p}_i^c, \mathbf{q}_i^c$  and the drift states  $\mathbf{p}_v^w, \mathbf{q}_v^w$ .

Using the visual-inertial state vector in Eq. 16, the system dynamics, and assuming the connection between the IMU and the camera is rigid, we define the visual sensor model following [28]:

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

where  $\mathbf{n}_{z_{p_v}}$  and  $\mathbf{n}_{z_{q_v}}$  are white Gaussian measurement noise variables and  $\mathbf{C}_{(\mathbf{q})}$  is the rotation matrix obtained from the quaternion  $\mathbf{q}$ . The nonlinear observability analysis in [16] and [29] 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 4<sup>th</sup> Lie derivative, hence, this is the order of the Taylor expansion we use for generating the E<sup>2</sup>LOG in our experiments.

We select a quadrotor helicopter as the example robot to carry the visual-inertial system. Visual-inertial systems are especially popular on quadrotors due to their light weight and low power requirements. As shown by Mellinger and Kumar [22], the quadrotor dynamics are differentially flat in

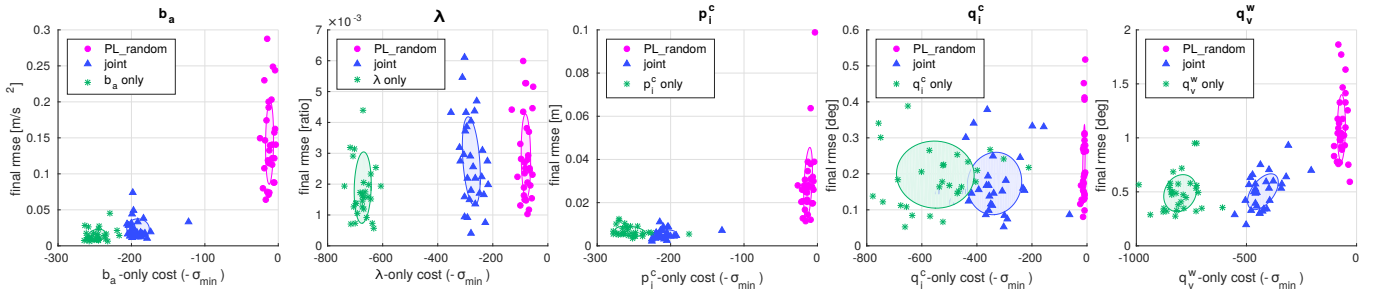


Fig. 2. Comparison between trajectories jointly optimized for all self-calibration states (blue triangles), trajectories separately optimized for individual self-calibration states (green stars), and randomly generated trajectories that excite the system near its physical limits (purple circles).  $x$ -axes represent  $E^2LOG$  cost for individual state only.  $y$ -axes represent EKF estimation error of individual state at trajectory termination. Shaded ellipses indicate one- $\sigma$  principal component boundaries of each trajectory class.

the flat outputs of  $x, y, z$  position and yaw  $\theta$ . The remaining extrinsic states, i.e. roll and pitch angles, are functions of these outputs and their derivatives. To ensure physically plausible trajectories we place inequality constraints on thrust-to-weight ratio ( $\leq 1.5$ ) and angular velocity ( $\leq \pi \frac{rad}{s}$ ). These values are well within the limits of a small quadrotor.

## VIII. EXPERIMENTAL SIMULATION RESULTS

We evaluate the proposed method on a simulated quadrotor helicopter equipped with the visual-inertial system described in Sec. VII. We demonstrate the system on two tasks: a closed-loop self-calibration trajectory using the null-space basis and a corridor navigation trajectory using the Bézier basis. In both cases, we represent trajectories as degree-7 polynomials with  $C^4$  continuity and require that all derivatives be zero at the beginning and end points. The integration step and the time horizon ( $\Delta t$ ) for  $E^2LOG$  are 0.1s.

To judge the effectiveness of a trajectory for self-calibration, we implement a state estimator for the visual-inertial system, simulate the system and estimator over the duration of the trajectory, and finally measure the accuracy of the estimator’s belief relative to known ground truth values. As a realization of the state estimator, we employ the popular Extended Kalman Filter (EKF). In particular, we use the indirect formulation of an EKF [19] where the state prediction is driven by IMU measurements. We choose this state estimator due its ability to work with various sensor suites and proven robustness in the quadrotor scenario.

In order to fairly evaluate trajectories across a distribution of plausible values for self-calibration states, our experiments simulate the EKF multiple times per trajectory with randomly sampled ground truth self-calibration state values and randomly sampled initial estimation errors in the EKF state. The distributions from which we sample are listed in Table I. Aggregating results over multiple random simulations reduces the effect of picking an unusually favorable or unfavorable set of ground-truth states. All input and measurement noise are zero-mean Gaussian with the following standard deviations:  $\sigma = 7.4 \times 10^{-3}$  deg/s in the gyroscope,  $\sigma = 8.3 \times 10^{-3}$  m/sec<sup>2</sup> in the accelerometer,  $\sigma = 0.01$  m in the visual position, and  $\approx \pm 0.2$  deg in the visual attitude.

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

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

### A. Self-Calibration

For the self-calibration task, we optimize a 6-piece, 15-second polynomial in the null-space basis, starting and ending at the origin. The trajectory is constrained to lie within a  $2.5 \times 2.5 \times 1.5$ -meter box and subject to the thrust and angular velocity constraints. The box constraint is small enough to easily fit in a typical indoor environment.

As a baseline for comparison, we generate competitive random trajectories by randomly sampling sets of large null-space weights and discarding trajectories that violate the physical limits. The analysis in [10] suggests that trajectories with large angular velocities lead to well observable self-calibration states. Each random trajectory is then used as the initial guess for the SQP-based optimization procedure described in Sec. VI, yielding a set of optimized trajectories.

1) *Validation of multi-state  $E^2LOG$  scaling:* To validate our proposed multi-state  $E^2LOG$  scaling procedure, we compare a set of trajectories jointly optimized for all self-calibration states against trajectories individually optimized for the sub-matrix of a single self-calibration state. Results are shown in Fig. 2. Note that we do not optimize for the state  $p_v^w$  as the global position of this system is unobservable [16]. In each scatter plot, the  $x$ -axis corresponds to the  $E^2LOG$  cost function for the individual calibration state, and the  $y$ -axis corresponds to the error of the individual state estimate in the EKF at the end of the 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  $E^2LOG$  cost functions, they perform equally well or nearly as well in the EKF error metric. This suggests that the multi-state  $E^2LOG$  successfully balances the goals of optimizing each

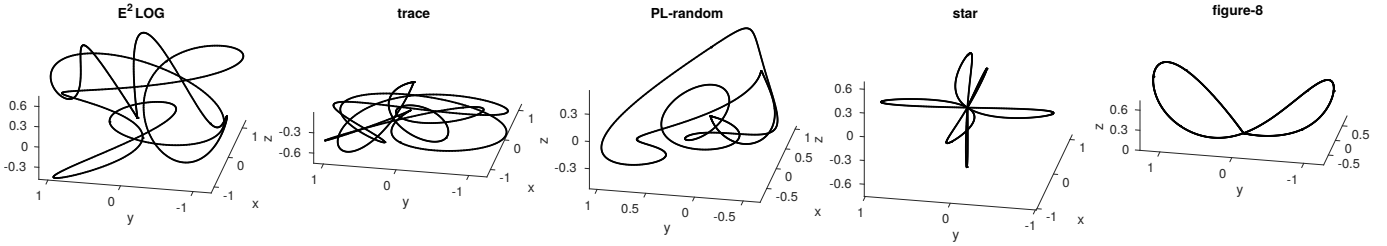


Fig. 3. Trajectories used for the comparison in Fig. 4. Each trajectory lasts 15 seconds and is tight against at some physical or box constraint.

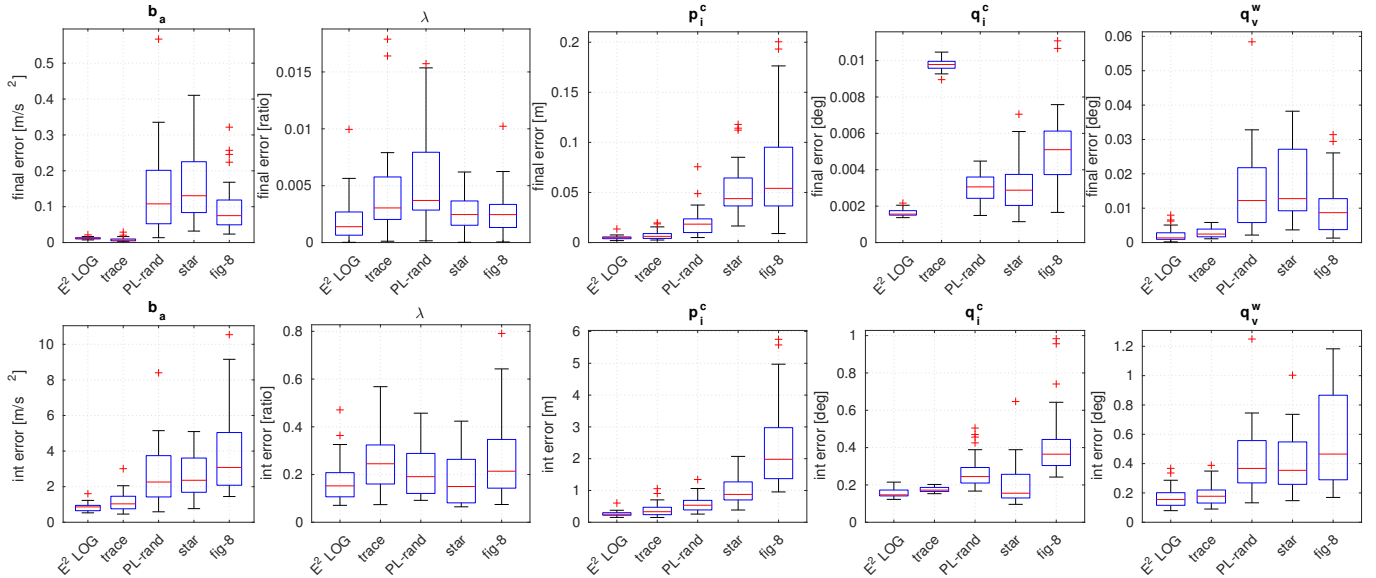


Fig. 4. Quartile box plots summarizing final (top) and integrated (bottom) error of EKF state estimates, aggregated over 30 simulated runs.  $E^2LOG$ : trajectory jointly optimized for all self-calibration states using our framework. *trace*: trajectory minimizing final trace of EKF covariance in all self-calibration states. *PL-rand*: random trajectories near the system physical limits. *star*, *fig-8*: common manually designed heuristic self-calibration trajectories.

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  $b_a$ ,  $p_i^c$ , and  $q_v^w$  parameters. On  $q_i^c$ , 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  $\lambda$ , the joint optimized trajectories perform equally well as the random trajectories, but here the estimation error of 0.3% is also quite small.

## 2) Comparison to EKF-trace-minimization and heuristics:

To explore the characteristics of these trajectories in greater detail, we randomly pick one representative joint-optimized trajectory and one random trajectory, and compare them to several competitive baselines, with results over many simulated EKF runs with randomized self-calibration parameters ( $N = 30$ ) displayed as box plots.

As a baseline for comparison, we generate one trajectory that minimizes the integrated trace of the covariance matrix in a simulated EKF [7]. This trajectory is optimized using

the null-space formulation with the same parameters and constraints used for the  $E^2LOG$ -optimized trajectories. (Note that it is not feasible to generate many EKF-optimized trajectories in the manner of Fig. 2 because optimizing for the EKF trace took over  $50\times$  longer than optimizing for  $E^2LOG$ .) 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. All trajectories are visualized in Fig. 3.

Results are shown in Fig. 4. The multi-state  $E^2LOG$ -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 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

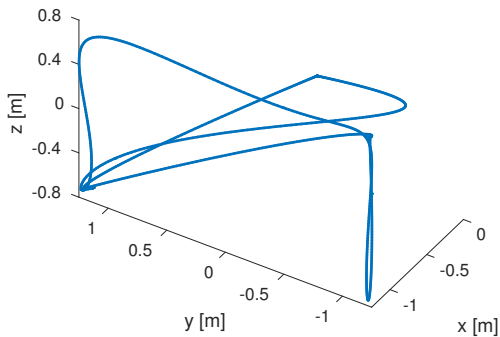


Fig. 5. Self-calibration trajectory optimized for estimation of the visual scale parameter  $\lambda$ . The trajectory fully exploits, but does not violate, the box constraints to provide best input (acceleration in this case) to the system.

for the visual scale parameter  $\lambda$ . An example is shown in Fig. 5. 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.

### B. Planning in a corridor

We demonstrate the corridor planning application in a manually-designed corridor of moderate complexity. As a baseline, we use a minimum-snap trajectory, computed using quadratic programming as in [26]. Minimum-snap trajectory planning is widely used for quadrotors [22, 24]. 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  $E^2\text{LOG}$  of all self-calibration states. The corridor and trajectories are visualized in Fig. 1 at the beginning of this paper. The min-snap trajectory is characteristically smooth. In contrast, the  $E^2\text{LOG}$ -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 multiple times ( $N = 30$ ) for both trajectories with randomly sampled ground truth  $\mathbf{x}_{sc}$  and initialization errors according to Table I. 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. 6. 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  $E^2\text{LOG}$ -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.

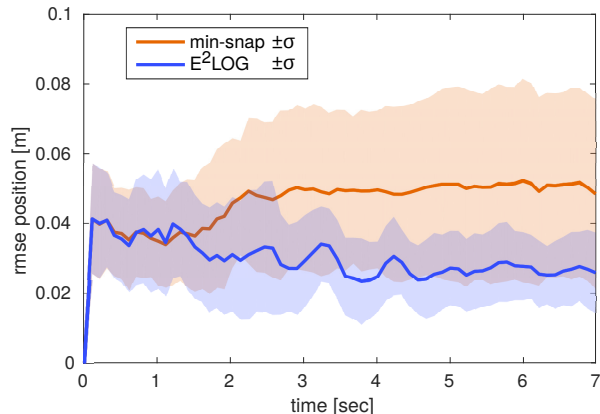


Fig. 6. RMSE of position estimate in EKF averaged over 30 simulated trials for min-snap and  $E^2\text{LOG}$ -optimized trajectories shown in Fig. 1. Ground truth self-calibration states and EKF initialization errors are randomly sampled for each trial. One- $\sigma$  bands illustrate the variation in estimation accuracy over different trials.

## IX. CONCLUSION

In this work, we introduced a framework to optimize robot trajectories for estimation of self-calibration states. Our  $E^2\text{LOG}$  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, allowing us to balance the goals of estimating each of the five self-calibration states in a visual-inertial odometry system.

We described two trajectory representations: a piecewise polynomial null-space basis for optimized self-calibration loop trajectories, and a Bézier spline basis for calibration-aware navigation through a corridor of intersecting convex polytopes. We demonstrated our method in extensive simulation experiments with an EKF-based visual-inertial odometry system. Our experiments showed a strong correlation between the  $E^2\text{LOG}$  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 conversion of all the self-calibration states simultaneously. We showed that a single joint-optimized trajectory from our framework outperforms a trajectory minimizing the EKF covariance trace and common heuristic self-calibration trajectories, while taking significantly less time to optimize than the EKF-based cost function.

For the corridor navigation problem, we showed example results from our method using the Bézier curve trajectory representation 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.



## REFERENCES

- [1] M.W. Achtelik, S. Weiss, M. Chli, and R. Siegwart. Path planning for motion dependent state estimation on micro aerial vehicles. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2013.
- [2] Rik Bähnemann, Michael Burri, Enric Galceran, Roland Siegwart, and Juan Nieto. Sampling-based motion planning for active multirotor system identification. *arXiv preprint arXiv:1612.05143*, 2016.
- [3] A. Bry and N. Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.
- [4] Gabriele Costante, Christian Forster, Jeffrey Delmerico, Paolo Valigi, and Davide Scaramuzza. Perception-aware path planning. *arXiv preprint arXiv:1605.04151*, 2016.
- [5] Robin Deits and Russ Tedrake. Computing large convex regions of obstacle-free space through semidefinite programming. In *Algorithmic Foundations of Robotics XI*, pages 109–124. Springer, 2015.
- [6] Melvin E Flores. *Real-time trajectory generation for constrained nonlinear dynamical systems using non-uniform rational b-spline basis functions*. Ph.D. Dissertation, California Institute of Technology, 2008.
- [7] K. Hausman, J. Müller, A. Hariharan, N. Ayanian, and G. Sukhatme. Cooperative multi-robot control for target tracking with onboard sensing. *The International Journal of Robotics Research*, 34(13):1660–1677, 2015.
- [8] Karol Hausman, James Preiss, Gaurav Sukhatme, and Stephan Weiss. Observability-aware trajectory optimization for self-calibration with application to uavs. *arXiv preprint arXiv:1604.07905*, 2016.
- [9] R. Hermann and A. Krener. Nonlinear controllability and observability. *IEEE Transactions on automatic control*, 22(5): 728–740, 1977.
- [10] J. Hernandez, K. Tsotsos, and S. Soatto. Observability, identifiability and sensitivity of vision-aided inertial navigation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2319–2325, May 2015.
- [11] J. Hesch, D. Kottas, S. Bowman, and S. Roumeliotis. Cameraimu-based localization: Observability analysis and consistency improvement. *The International Journal of Robotics Research*, 2013.
- [12] B.T. Hinson and K.A. Morgansen. Observability optimization for the nonholonomic integrator. In *American Control Conference (ACC), 2013*, pages 4257–4262, June 2013.
- [13] Vadim Indelman, Luca Carlone, and Frank Dellaert. 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, 2015.
- [14] Kenneth I Joy. Bernstein polynomials. *On-Line Geometric Modeling Notes*, 13, 2000.
- [15] B.J. Julian, M. Angermann, M. Schwager, and D. Rus. Distributed robotic sensor networks: An information-theoretic approach. *Int. Journal of Robotics Research*, 31(10), 2012.
- [16] J. Kelly and G. Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *The International Journal of Robotics Research*, 30(1), 2011.
- [17] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.
- [18] A. Krener and K. Ide. Measures of unobservability. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 6401–6406, 2009.
- [19] S. Lynen, M. Achtelik, S. Weiss, M. Chli, and R. Siegwart. A robust and modular multi-sensor fusion approach applied to mav navigation. In *In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3923–3929, 2013.
- [20] Phillipe Martin, Richard M Murray, and Pierre Rouchon. Flat systems, equivalence and trajectory generation. 2003.
- [21] A. Martinelli and R. Siegwart. Observability properties and optimal trajectories for on-line odometry self-calibration. In *Decision and Control, 2006 45th IEEE Conference on*, pages 3065–3070, Dec 2006.
- [22] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *In Int. Conf. on Robotics and Automation (ICRA)*, pages 2520–2525. IEEE, 2011.
- [23] J. Müller and G.S. Sukhatme. 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)*, Chicago, IL, USA, September 2014.
- [24] C. Richter, A. Bry, and N. Roy. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In *In Proc. of the International Symposium on Robotics Research (ISRR)*, 2013.
- [25] D. N. Ta, M. Kobilarov, and F. Dellaert. A factor graph approach to estimation and model predictive control on unmanned aerial vehicles. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 181–188, May 2014.
- [26] Sarah Tang and Vijay Kumar. Safe and complete trajectory generation for robot teams with higher-order dynamics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.
- [27] S. Weiss, M. Achtelik, S. Lynen, M. Chli, and R. Siegwart. Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.
- [28] S. Weiss, M. Achtelik, S. Lynen, M. Achtelik, L. Kneip, M. Chli, and R. Siegwart. Monocular vision for long-term micro aerial vehicle state estimation: A compendium. *Journal of Field Robotics*, 30(5):803–831, 2013.
- [29] Stephan Weiss. *Vision Based Navigation for Micro Helicopters*. Phd thesis, ETH Zurich, March 2012.