Multi-Objective Path Planning in GPS Denied Environments under Localization Constraints

Shaunak D. Bopardikar Brendan Englot Alberto Speranzon

Abstract— **The main contribution of this paper is a novel planning algorithm that, starting from a probabilistic roadmap, efficiently constructs an expanded graph used to search for the optimal solution of a multi-objective problem. The primary cost is the shortest path from start to goal and the secondary cost is related to the state estimation error covariance. This needs to be optimized as we assume the navigation to be in a GPS denied environment. The proposed algorithm is efficient as it relies on a scalar metric, related to the largest eigenvalue of the error covariance, and adaptively quantizes the secondary cost, yielding a graph whose number of vertices and edges provides a good tradeoff between optimality and computational complexity. Numerical examples show the advantage of the proposed approach compared to methods where the expanded graph is built by quantizing the secondary cost uniformly.**

Index Terms— **Motion Planning; Multi-objective Optimization; GPS-denied Localization; Autonomous Systems.**

I. Introduction

Autonomous missions in complex and partially known environments require the generation of trajectories that not only enable safe navigation around obstacles, but also minimize multiple objectives. In this paper, we focus on navigation in GPS denied environments, and thus requiring the state estimation error to be bounded is crucial to avoiding paths that will lead to large errors, place the mission vehicle at risk of getting lost, and jeopardize the success of the mission. In the framework we propose, we assume that the environment is known (or at least partially known) and that sensor models are available so that their response to the environment model can be obtained. Of course, the quality of the chosen path will strongly depend on the quality of these "strong" priors. Given the large amount of Geographic Information Systems databases available, e.g., Google maps, prior information is generally very rich and goes well beyond just geometric information about obstacles.

A natural approach to solve a multi-objective optimization problem is to pose it as an optimization where a single primary cost is optimized and other costs are treated as constraints (secondary costs). In this paper, we will consider the path length (or a function of the the path length, such as fuel consumption or time-to-goal) as primary cost and the estimation error as secondary cost. Given the wide-spread use of Kalman filtering techniques, we will use a function of the estimation error covariance as the secondary objective.

Navigation in GPS-denied environments has been a topic of active research for the past two decades. Early works were focused solely on improving navigation estimation accuracy, e.g., [1], [2]. More recently, various researchers have been working on path planning under state estimation uncertainty. One of the seminal works is the belief roadmap (BRM) [3] which builds a probabilistic roadmap (PRM) [4] in a robot's state space, propagates beliefs over the roadmap using an extended Kalman filter (EKF) [5], and plans a path of minimum goal-state uncertainty. This approach has been extended [6] to bias the PRM samples using a Sensory Uncertainty Field (SUF) [7], which expresses the spatial variation in sensor performance over the workspace. Rapidly-exploring random belief trees (RRBTs) [8] use the EKF to propagate belief states over a rapidly-exploring random graph (RRG) [9], to find asymptotically optimal paths that minimize goal-state uncertainty subject to chance constraints. In [10], the authors propose an approach for planning in belief space, based on differential dynamic programming which relies on an initial guess computed using a rapidly-exploring random tree (RRT). Compared to this body of work, our paper introduces a novel method that trades computational complexity (measured in terms of both, time and storage space) with optimality, while leveraging the flexibility of sample-based approaches.

Optimal planning under multiple objectives is a classic problem, and several approaches have been proposed to explore the *Pareto-front*, i.e., the set of solutions that yield the same optimal primary value. However, the main impediment has been the high computational requirement. For problems in which both objectives, the primary and the secondary, are monotonic non-decreasing functions of the vehicle state, a computationally efficient algorithm has been recently proposed [11]. The approach is to represent the underlying state space using a graph, compute the primary and secondary objectives over edges of the graph, and search efficiently over a larger-sized *product graph*. However, with the estimation error covariance as the secondary objective, the non-negativity assumption does not hold any longer, since the existence of useful visual features will decrease the error covariance over an edge. Additionally, the change in the error covariance over an edge is a function of the covariance at the source vertex of that edge, which introduces a *history-dependence* to the problem. This calls for a new approach which will need to trade-off accuracy in terms of the primary cost for computational efficiency in computing a *good* path that still meets the secondary objective constraint.

In this paper, we propose a methodology to obtain approximate solutions to the multi-objective path planning problem that is history-independent during the search over the product graph. Our contributions are three-fold. First, using the max-

The authors would like to acknowledge United Technologies Research Center for supporting this research. The authors are all with United Technologies Research Center. Emails: {BopardSD, EnglotBJ, SperanA}@utrc.utc.com

imum eigenvalue of the estimation error covariance matrix, we derive a novel bound on its evolution given the underlying graph (possibly the output of a sample based planning algorithm such as PRM/PRM* or RRG [9]) and the sensing information over the edges of the graph. We show how this eigenvalue bound can be used to determine the secondary-objective quantization of the product graph. Second, we provide two algorithms; one based on uniform quantization scheme, i.e., every vertex of the original graph has the same number of copies in the product graph; a second based on an adaptive quantization scheme in which every vertex is replicated a different number of times depending upon the increase/reduction in uncertainty when arriving at that vertex in different ways. We formally establish that the secondary cost over the path obtained with either algorithm meets a hard constraint on the secondary objective. We also establish the computational complexity of the algorithms. Third and finally, we demonstrate the tradeoff between accuracy and computational efficiency of the two algorithms on a simple small-sized,on which we can compare against the globally optimal solution, as well as a more complex example of path planning over a given environment.

This paper is organized as follows. The problem formulation is presented in Section II. The description of our proposed approach along with the main technical results are presented in Section III. The results of the application of our algorithms on numerical examples is presented in Section IV. Finally, concluding remarks along with directions for future research are summarized in Section V.

II. Problem Formulation

In this section, we describe how we model the vehicle motion, the environment and present the problem formulation.

A. Motion and Sensing Model

We consider a general model of an agent whose state evolves as per a non-linear discrete-time dynamical system

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t), \mathbf{n}(t)), \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$ is the state describing the system at time t, $\mathbf{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_n} \to \mathbb{R}^{n_x}$ describes the state transition map of the system and $\mathbf{n} \in \mathbb{R}^{n_n}$ is the process noise. The agent is equipped with m sensors in order to estimate the state $\mathbf{x}$. Sensors' output is modeled as

$$\mathbf{y}_j(t) = \mathbf{h}_j(\mathbf{x}(t), \mathbf{v}_j(t)), \qquad \forall j \in \{1, \ldots, m\}, \tag{2}$$

where $\mathbf{v}_j \in \mathbb{R}^{n_j}$ is the process noise of the j-th sensor and $\mathbf{h} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_j} \to \mathbb{R}^{n_{y_j}}$ describes the relation between state and measurement. We assume that the noise vectors $\mathbf{n}$ and $\mathbf{v}_j$ are independently generated zero mean Gaussian random vectors.

In this work, we will make the following assumptions.

Assumption 2.1 (Consistency and Synchrony): 1) The system is linearizable at each time instant t. In other words, an Extended Kalman Filter (EKF) provides a consistent estimate $\hat{\mathbf{x}}_t$ and the estimation error covariance $\mathbf{P}_t$ of the filter is a measure of the precision of the EKF.

2) The state estimate $\hat{\mathbf{x}}_t$ follows the nominal trajectory for the vehicle.

3) The measurements from all sensors/sources are time synchronized. □

The first two items of this assumption imply that we have a reasonable nominal model for the motion of the vehicle, and that there exists a control action that keeps the state estimate $\hat{\mathbf{x}}_t$ close to the nominal trajectory (c.f. [3]). Our work is concerned with the level of confidence measured through $\mathbf{P}_t$ that we can obtain in our state estimate $\hat{\mathbf{x}}_t$. The third item from Assumption 2.1 allows for a compact representation of our results, and may be relaxed to incorporate asynchrony between measurements.

Under these assumptions, analogous to the Kalman Filter case (see equations 185 and 186 in [12]), an EKF based estimator of the state $\mathbf{x}$ can be written as:

$$\mathbf{P}_{t+1}^{-1} = (\mathbf{F}_t\mathbf{P}_t\mathbf{F}_t' + \mathbf{Q})^{-1} + \sum_{j=1}^{m} \gamma_{j,t+1}\mathbf{H}_j'\mathbf{R}_j^{-1}\mathbf{H}_j, \tag{3}$$

$$\hat{\mathbf{x}}_{t+1} = \mathbf{P}_{t+1}\Big((\mathbf{F}_t\mathbf{P}_t\mathbf{F}_t' + \mathbf{Q})^{-1}\mathbf{f}(\hat{\mathbf{x}}_t, \mathbf{0}) + \\ + \sum_{j=1}^{m} \gamma_{j,t+1}\mathbf{H}_j'\mathbf{R}_j^{-1}(\mathbf{y}_j(t+1) - \mathbf{h}_j(\mathbf{f}(\hat{\mathbf{x}}_t), \mathbf{0}))\Big),$$

where $\hat{\mathbf{x}}_t$ is the state estimate, $\mathbf{P}_t$ is the expected error covariance with respect to the process and sensor noise terms, $\mathbf{F}_t$ is the linearization of $\mathbf{f}$ around $(\hat{\mathbf{x}}_t, \mathbf{0})$ and $\mathbf{H}_j$ is the linearization of $\mathbf{h}_j$ around $(\mathbf{f}(\hat{\mathbf{x}}_t), \mathbf{0})$. The matrix $\mathbf{Q}$ is the process noise covariance and the $\mathbf{R}_j$ is the measurement noise covariance associated to the j-th sensor. The variables $\gamma_{j,t+1}$ are binary, $0/1$ variables that model the absence/presence of the measurement provided by the j-th sensor.

In this work, we will consider the maximum eigenvalue of the error covariance matrix, $\bar{\lambda}(\mathbf{P}_t)$, as the scalar metric for capturing state estimate uncertainty. In the following, we will denote with $\underline{\lambda}(\mathbf{A})$ and with $\bar{\lambda}(\mathbf{A})$ the minimum and maximum eigenvalue of $\mathbf{A}$, respectively, where $\mathbf{A}$ is a positive definite/semi-definite matrix.

B. Environment Model

We model the environment $\mathcal{E} \subset \mathbb{R}^{n_x}$ as a directed graph $G(V, E)$, where V denotes a finite set of vertices and $E = V \times V \times \{0, 1\}$ denotes the set of edges. Here the extra $\{0, 1\}$ tag is required as we are working with directed graphs. Given a pair of vertices i, j, denote an edge e_{ij} from i to j if it exists. Two edges e_{ij}, e_{jk} are said to be *linked* if both e_{ij} and e_{jk} exist. Let $s, d \in V$ denote a start/destination pair of vertices. A *path* from s to d is a collection of linked edges $\pi_{sd} := \{e_{si_1}, e_{i_1i_2}, \ldots, e_{i_nd}\}$. The collection of paths from s to d is denoted as $\mathcal{P}_{sd}$.

Given a primary cost function, we assign a cost c_{ij} corresponding to each edge e_{ij}. The cost of a path π_{sd} is defined as

$$C(\pi_{sd}) := \sum_{e_{ij} \in \pi_{sd}} c_{ij}$$

Given a path π_{sk}, where $k \in V$ is an intermediate vertex, we associate the state estimate $\hat{\mathbf{x}}(\pi_{sk})$ and the filter error covariance $\mathbf{P}(\pi_{sk})$ corresponding to propagation of the EKF along the path π_{sk}. We say that the path π_{sk} is a *sub-path* of the path π_{sd} if $\pi_{sk} \subseteq \pi_{sd}$.

In this work, we are interested in computing minimum cost paths, subject to constraints on a secondary cost, namely the error covariance of the EKF propagated over the path. We wish to solve the following optimization problem:

$$\begin{aligned} \min_{\pi_{sd} \in \mathcal{P}_{sd}} \quad & C(\pi_{sd}) \\ \text{s.t.} \quad & \bar{\lambda}(\mathbf{P}(\pi_{sk})) \leq p_{\text{max}}\,, \forall \pi_{sk} \subseteq \pi_{sd} \in \mathcal{P}_{sd}\,, \end{aligned} \tag{4}$$

where $\bar{\lambda}$ denotes the maximum eigenvalue.

Note that this formulation may easily be generalized to include a separate *end* constraint such as

$$\bar{\lambda}(\mathbf{P}(\pi_{sd})) \leq p_{\text{end}}.$$

This will be clear after we present our approach.

Additionally, we assume that even with the best possible sensing scenario and sensor suite, the metric on the estimation error covariance cannot go below a particular lower bound, given by p_{best}. This can happen in the intermittent GPS scenario, in which this lower bound is given by the precision of the GPS.

III. Proposed Approach

In this section, we present a computationally efficient but approximate approach to solve the problem (4).

A. Construction of the Product graph

The main idea behind this approach is to uniformly discretize the maximum allowed value p_{max} of the secondary cost into $N := \lceil (p_{\text{max}} - p_{\text{best}})/\delta \rceil$ *levels*, given the parameter δ. Then, we create a *product graph* defined as follows:

1) Create N copies of the vertex set V of the graph G, one copy at each level starting from p_{best}. We can now denote the i-th vertex in V at the ℓ-th level in the product graph by $V_{i,\ell}$.
2) For every pair of vertices $i, j \in V$ for which an edge $e_{ij} \in E$, and for every level ℓ from $1, \ldots, N$, compute the maximum possible value of the covariance metric $\bar{\lambda}_{e_{ij}}(\mathbf{P})$ after propagation of the EKF over the edge e_{ij}, with the initial covariance satisfying
$$\bar{\lambda}(\mathbf{P}_0) = p_{\text{best}} + \delta\ell.$$
If this maximum value is less than p_{max}, then we construct an edge $\bar{e}$ directed from $V_{i,\ell}$ to $V_{j,\bar{\ell}}$, where $\bar{\ell}$ is the least integer for which the final covariance $\bar{\lambda}(\mathbf{P}) \leq \delta\bar{\ell}$.
3) Quantize and round up $\bar{\lambda}_{e_{ij}}(\mathbf{P}_0)$ in the interval $[p_{\text{best}}, p_{\text{max}}]$ using the discretization δ. In other words, construct an edge $\bar{e}$ directed from $V_{i,\ell}$ to $V_{j,\bar{\ell}}$, where $\bar{\ell}$ is the least integer for which the final covariance $\bar{\lambda}(\mathbf{P}) \leq p_{\text{best}} + \delta\bar{\ell}$.

Figures 1 and 2 illustrate the construction of the product graph given a graph G for a simple example.

Finally, we apply any shortest path algorithm, e.g., Dijkstra's algorithm [13], from the initial vertex $V_{s,1}$ to the final vertex $V_{d,N}$ to obtain the optimal path.

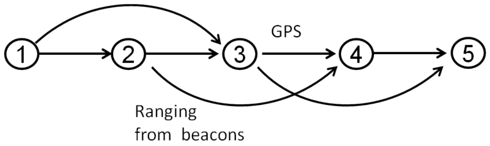


Fig. 1. The directed graph G that describes the primary cost and the secondary cost associated with each edge.

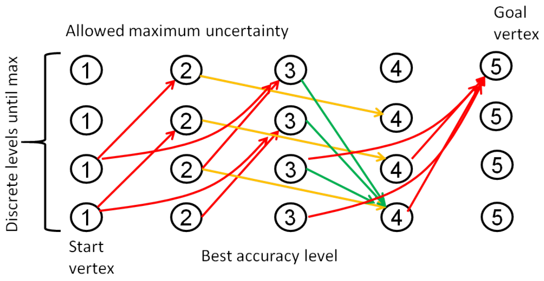


Fig. 2. Construction of the product graph. In this example, the red edges denote open-loop propagation of the filter, yellow edges denote the ones on which the vehicle can range to beacons, and the green ones denote the edges over which GPS signals are available.

In this proposed approach, the main issues to be addressed are: 1) the choice of the discretization level δ; and 2) the computation of the maximum possible value of the covariance metric $\bar{\lambda}_{e_{ij}}(\mathbf{P})$ over every edge e_{ij}. The smaller the choice of δ, the closer to optimality the final path cost will be, but at the cost of a larger resulting product graph. The computation of $\bar{\lambda}_{e_{ij}}(\mathbf{P})$ requires the propagation of the EKF over an edge for *every possible* initial value of the covariance metric, a computationally prohibitive step. However, if we can derive a uniform bound on the *increase* of the covariance metric over any edge, then we can mitigate the requirement of running the entire EKF over every edge and for every possible covariance metric value given by the N levels. This bound will also drive the choice of the δ parameter.

B. A Bound on the Maximum Eigenvalue $\bar{\lambda}$ metric

In this section, we derive a uniform upper bound for $\bar{\lambda}(\mathbf{P}_t)$ as a function of the initial value $\bar{\lambda}(\mathbf{P}_0)$. For this result, we introduce the following notation. Let $\mathcal{X} := \{\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_T\}$, denote a set of estimates at different times along an edge $e \in E$. Under Assumption 2.1, this set is identical to the nominal trajectory sampled at the corresponding times. Define,

$$\mathcal{X}_S := \Big\{ \hat{\mathbf{x}} \in \mathcal{X} \,:\, \underline{\lambda}\Big(\mathbf{H}(\mathbf{f}(\hat{\mathbf{x}}, \mathbf{0}))'\mathbf{R}^{-1}\mathbf{H}(\mathbf{f}(\hat{\mathbf{x}}, \mathbf{0}))\Big) = 0 \Big\},$$

denoting the subset of $\mathcal{X}$ at which the sensors provide no useful information. The filter runs open-loop at these locations. Then, the following result holds.

Theorem 3.1: Suppose that over the edge $e \in E$, the cardinality of the open-loop set $\mathcal{X}_S$ is κ. Then, under

Assumption 2.1, at the final estimation time instant T,

$$\bar{\lambda}(\mathbf{P}_T) \leq b \sum_{j=1}^{\kappa} a^{j-1} - a^{\kappa}\zeta + a^{\kappa} \Bigg/ \left(\left(\frac{d-\zeta c}{\zeta c + a}\right)^{T-\kappa} \frac{1}{\zeta + \bar{\lambda}(\mathbf{P}_0)} + \frac{c}{\zeta c + a} \left(\frac{1 - \frac{(d-\zeta c)^{T-\kappa}}{(\zeta c + a)^{T-\kappa}}}{1 - \frac{(d-\zeta c)}{(\zeta c + a)}} \right) \right) =: B_e(\bar{\lambda}(\mathbf{P}_0)),$$

where

$$a := \max_{\hat{\mathbf{x}} \in \mathcal{X}} \bar{\lambda}(\mathbf{F}(\hat{\mathbf{x}}))\,, \qquad b := \bar{\lambda}(\mathbf{Q})\,,$$

$$c := a \min_{\hat{\mathbf{x}} \in \mathcal{X} \setminus \mathcal{X}_S} \underline{\lambda}\left(\mathbf{H}(\mathbf{f}(\hat{\mathbf{x}}, \mathbf{0}))' \mathbf{R}^{-1} \mathbf{H}(\mathbf{f}(\hat{\mathbf{x}}, \mathbf{0}))\right)\,,$$

$$d := bc/a + 1\,,\ \zeta := (d - a + \sqrt{(d-a)^2 + 4bc})/(2c)\,. \square$$

The proof of this result is presented in the Appendix. To extend both of these results for m functioning sensors, we simply replace the term $\mathbf{H}'\mathbf{R}^{-1}\mathbf{H}$ by $\sum_{j=1}^{m} \mathbf{H}'_{j,t}\mathbf{R}_j^{-1}\mathbf{H}_{j,t}$.

The next result essentially bounds the change in the metric $\bar{\lambda}(\mathbf{P})$ when the EKF is propagated over any edge of the PRM.

Corollary 3.1 (Change in $\bar{\lambda}$ over an edge): Suppose that the set $\mathcal{X}$ represents points along any fixed edge e of the PRM. Then, the change in $\bar{\lambda}$ over that edge, i.e., the quantity $\Delta_e(\bar{\lambda}(\mathbf{P}_0)) := B_e(\bar{\lambda}(\mathbf{P}_0)) - \bar{\lambda}(\mathbf{P}_0)$, is a concave function of $\bar{\lambda}(\mathbf{P}_0)$, with the maximum occurring for the value of

$$\bar{\lambda}^*(\mathbf{P}_0) = \max\left\{ \frac{\sqrt{\alpha} - \alpha}{\gamma} - \zeta, p_{\text{best}} \right\}$$

where

$$\alpha := \left(\frac{d-\zeta c}{\zeta c + a}\right)^{T-\kappa}, \gamma := \frac{c}{\zeta c + a} \left(\frac{1 - \frac{(d-\zeta c)^{T-\kappa}}{(\zeta c + a)^{T-\kappa}}}{1 - \frac{(d-\zeta c)}{(\zeta c + a)}} \right) \square$$

This result follows by applying standard calculus using the bound from Theorem 3.1 and considering the derivative of the maximum change in $\bar{\lambda}$ with respect to $\bar{\lambda}(\mathbf{P}_0)$.

C. Discretization for Product graph construction

Corollary 3.1 can now be used to determine the discretization level required to construct the product graph G_μ in the following manner:

1) For each edge $e \in E$, compute the maximum Δ_e, denoted as Δ_e^* (note that this maximum value may be negative for some edges, signifying that for any covariance value, the edge always reduces uncertainty in the position).
2) Find the minimum (away from zero) in the absolute value sense of all Δ_e^*, and set

$$\delta := \min_{e \in E} \{|\Delta_e^*| \ : \ |\Delta_e^*| > 0\},$$

This procedure is illustrated in Figure 3, when applied to the example in Figure 1. For ease of presentation, let us assume that the number of sampling instants over each edge are the same, i.e., T is the same for each edge. Then,

$$\delta = \begin{cases} |\Delta_e^*(e_{24})| & \text{if } |\Delta_e^*(e_{24})| \leq bT, \\ bT & \text{otherwise.} \end{cases}$$

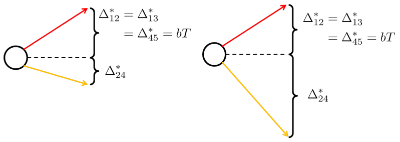


Fig. 3. Illustration of the choice of the parameter δ. In the left figure, the choice of δ would be Δ^*, while in the right figure, it would be equal to bT, where b is the process noise term and T is the number of sampling instants over the edge.

Algorithm 1 MOPA$(s, d, V, E, f, h, Q, R, p_{\text{best}}, p_{\max})$

```
$\delta := \min_{e \in E} |\Delta_e^*|$, using Corollary 3.1.
$L := \lceil (p_{\max} - p_{\text{best}})/\delta \rceil + 1$, $N := |V|L$.
Initialize matrices $A_{\text{exp}}, C_{\text{exp}}$ to zeros of size $N \times N$.
for $e_{ij} \in E$ do
    for $\ell \in \{1, \dots, L\}$ do
        if $B_{e_{ij}}(p_{\text{best}} + (\ell - 1)\delta) \leq p_{\max}$ then
            if $j = d$ then
                $A_{\text{exp}}((\ell-1)|V| + i, (L-1)|V| + j) = 1$
                $C_{\text{exp}}((\ell-1)|V| + i, (L-1)|V| + j) = c_{ij}$
            else
                $r_{ij} := \lceil \max\{0, B_{e_{ij}}(p_{\text{best}} + (\ell-1)\delta)/\delta\} \rceil$
                $A_{\text{exp}}((\ell-1)|V| + i, r_{ij}|V| + j) = 1$
                $C_{\text{exp}}((\ell-1)|V| + i, r_{ij}|V| + j) = c_{ij}$
            end if
        end if
    end for
end for
$\pi_{\text{approx}} :=$ Dijkstra$(A_{\text{exp}}, C_{\text{exp}}, s, (L-1)|V| + d)$.
return $\pi_{\text{MOPA}} =$ Project$(\pi_{\text{approx}})$.
```

We now give a Multi-Objective Planning Algorithm (MOPA) that summarizes these steps (cf. Algorithm 1). The construction of the product graph is performed in two nested for-loops. The outer for-loop runs over the set of edges E of the underlying original graph G. The inner for-loop runs over the set of discrete levels decided by the choice of the parameter δ using Corollary 3.1. The first if-loop checks whether the uncertainty bound when propagated over the edge e_{ij} starting from a certain level ℓ is less than the $p_{\max}$ threshold. If it is, then the particular edge is allowed in the product graph. The inner most if-loop wires those vertices that have an allowable edge in the product graph, depending on whether the end vertex is the destination d or not. Next, the algorithm computes an optimal path π_{approx} (if it exists) using a simple Dijkstra search over the product graph, commencing from the start vertex s and terminating at the destination vertex in the product graph. The final choice of the path is obtained by *projecting* the path from the product graph down to the original graph G.

The following result summarizes theoretical properties of Algorithm 1.

Theorem 3.2: Under Assumption 2.1, the proposed approach in Algorithm 1 has the following properties:

1) If the algorithm yields a solution $\pi_{\text{MOPA}} \neq \emptyset$, then the true uncertainty along the solution π_{MOPA} measured in

the $\bar{\lambda}$ metric is upper bounded by p_{max}, i.e.,

$$\bar{\lambda}(\mathbf{P}(\pi_{\text{MOPA}})) \leq p_{\text{max}}.$$

2) If the original graph $G = (V, E)$ is sparse, i.e., $|E| \ll |V|^2$, then the computational complexity of Algorithm 1 is

$$O\left(\frac{p_{\text{max}} - p_{\text{best}}}{\delta}\left(|E| + |V| \log\left(|V| \frac{p_{\text{max}} - p_{\text{best}}}{\delta}\right)\right)\right)$$

Proof: The first part follows from the fact that Algorithm 1 obtains a path based on an upper bound on $\bar{\lambda}$ propagated over every edge, optimized over *any* value of the covariance at the start of the edge. The second claim follows from the facts that (i) the product graph is sparse, (ii) the complexity of the construction of the product graph is $O(|E|(p_{\text{max}} - p_{\text{best}})/\delta)$ and (iii) the complexity of running Dijkstra's algorithm on a sparse graph is $O(|E| + |V| \log |V|)$. ■

D. Adaptive Quantization for Computational Efficiency

In this section, we present an approach that chooses a separate quantization level for each vertex in V. This approach allows for a reduction in the total number of vertices in the product graph, which in turn leads to a reduction in the worst-case completion time of the Dijkstra step.

The main idea is to run a for-loop over the set of all vertices in which for each vertex $i \in V$, we examine which are the incoming edges e (if they exist) into that vertex, and pick the minimum of $|\Delta_e^*|$ over those edges and set that as the quantization level δ_i. The number of levels L_i for the vertex i is computed using this specific δ_i. The rest of the procedure remains almost identical to that described for the uniform quantization approach in Algorithm 1. The quantization is adaptive in the sense that we make use of the fact that over some edges the change in the covariance may be more than over others. This procedure is described in Algorithm 2.

Note that as compared to the uniform quantization level approach, the adaptive approach will introduce more round-off error since the discretization can be coarse for some vertices. Additionally, the adaptive approach introduces an extra for-loop over the set of original vertices V. However, we will see numerical examples in the next section in which this approach is much faster because the overall number of levels and the total number of vertices in the product graph can be much smaller than for Algorithm 1.

IV. Numerical Results

In this section, we present a numerical implementation of our proposed approach on an illustrative example. We assume a planar environment $\mathcal{E}$ and a simple first order integrator motion model for the vehicle, with an onboard range sensor that is able to measure distances to fixed beacons located at $\mathbf{p} \in \mathcal{E}$. In this experiment, we assume the following models for the different parameters of the problem. The primary cost function c was chosen to be the Euclidean distance between two vertices if an edge exists.

$$\mathbf{f}(\mathbf{x}, \mathbf{n}) := \mathbf{x} + \mathbf{n},$$

$$\mathbf{h}_j(\mathbf{x}, \mathbf{v}_j) := \begin{cases} \|\mathbf{p}_j - \mathbf{x}\| + \mathbf{v}_j, & \text{if } \|\mathbf{p}_j - \mathbf{x}\| \leq r_{\text{sens}}, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Algorithm 2 Adaptive MOPA$(s, d, V, E, f, h, Q, R, p_{\text{best}}, p_{\text{max}})$

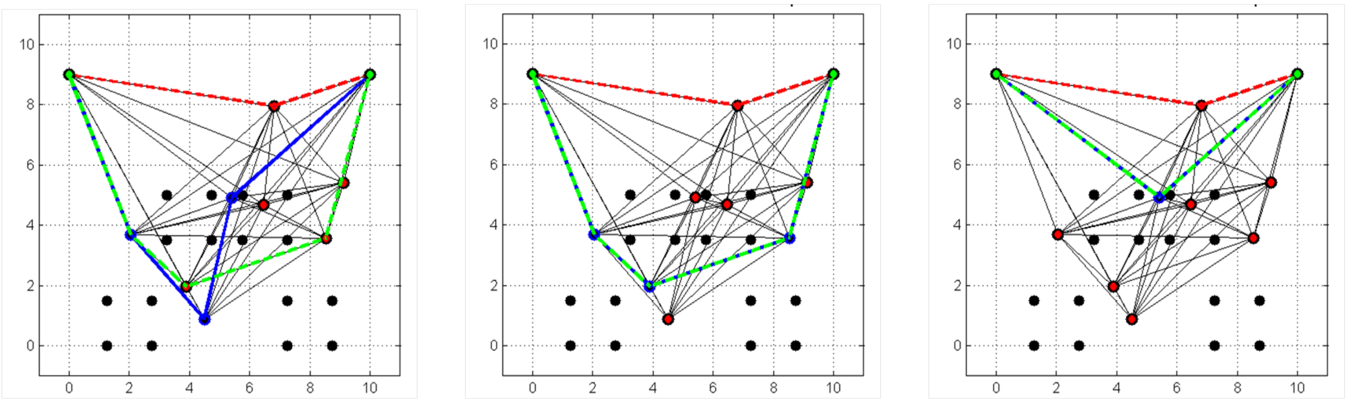
```
for  $i \in V$  do
    Let $e_i$ denote the incoming edges into $i$
    $\delta_i := \min_{e_i} |\Delta_{e_i}^*|$, using Corollary 3.1.
    $L_i := \lceil (p_{\text{max}} - p_{\text{best}})/\delta_i \rceil + 1$,
end for
$N := |V| \max\{L_i\}$.
Initialize matrices $A_{\text{exp}}, C_{\text{exp}}$ to zeros of size $N \times N$.
for $e_{ij} \in E$ do
    for $\ell \in \{1, \ldots, L_i\}$ do
        if $B_{e_{ij}}(p_{\text{best}} + (\ell - 1)\delta_i) \leq p_{\text{max}}$ then
            if $j = d$ then
                $A_{\text{exp}}((\ell - 1)|V| + i, (L_j - 1)|V| + j) = 1$
                $C_{\text{exp}}((\ell - 1)|V| + i, (L_j - 1)|V| + j) = c_{ij}$
            else
                $r_{ij} := \lceil \max\{0, B_{e_{ij}}(p_{\text{best}} + (\ell - 1)\delta_j)/\delta_j\} \rceil$
                $A_{\text{exp}}((\ell - 1)|V| + i, r_{ij}|V| + j) = 1$
                $C_{\text{exp}}((\ell - 1)|V| + i, r_{ij}|V| + j) = c_{ij}$
            end if
        end if
    end for
end for
$\pi_{\text{approx}} :=$ Dijkstra$(A_{\text{exp}}, C_{\text{exp}}, s, (L - 1)|V| + d)$.
return  $\pi_{\text{MOPA}} =$ Project$(\pi_{\text{approx}})$.
```

A. Illustrative Example

In this example, we illustrate the uniform quantization approach. Figure 4 shows how the optimal path (solid blue) arising out of Algorithm 1 changes with different values of the maximum uncertainty p_{max}. There are 10 nodes in the original graph G. The start and destination vertices are the left and right green ones, respectively. The black dots located towards the bottom part of the environment represent the beacons that the vehicle can range to. The sensing radius r_{sens} was chosen to be 3 units. The unconstrained primary optimal solution is shown in dashed-red.

The optimal primary costs at different values of p_{max} are reported in Table I. We also report the true value of $\bar{\lambda}$ along the path π_{MOPA} obtained from Algorithm 1, by propagating the EKF equations (3), to verify the first claim from Theorem 3.2 and to compare the gap with the bound. For this small sized example, we can actually compare our result with the globally optimal solution (green curve in Figure 4) of the problem. We conclude that except at very small values of p_{max}, we obtain a solution which is consistent with the global optimum for this problem.

Remark 4.1 (Conservativeness of the bound): While the bound appears to be reasonably accurate in this example, there have been instances in which the bound can be lot more conservative, leading to a high gap with respect to the true value. Further, a *ranking* of two paths based on



(a) $p_{\max} = 0.61$ (b) $p_{\max} = 0.7$ (c) $p_{\max} = 0.8$

Fig. 4. Choice of different optimal paths at different levels of maximum uncertainty $p_{\max}$. The process uncertainty $\mathbf{Q} = 0.01\mathbf{I}$, and the measurement uncertainty $\mathbf{R}_j = 1, \forall j$. The best value of estimation precision $p_{\text{best}} = 0.001$.

TABLE I

OPTIMAL COSTS AT DIFFERENT $p_{\max}$ VALUES.

$p_{\max}$	True λ	Approx. Optimal	Global Optimal
< 0.61	N/A	N/A	N/A
0.61	0.60	19.67	18.751
0.65	0.60	18.754	18.751
0.7	0.60	18.751	18.751
0.8	0.7	12.91	12.91
≥ 1.1	1.07	10.24	10.24

TABLE II

OPTIMAL COSTS USING ADAPTIVE QUANTIZATION

$p_{\max}$	Optimal cost	True λ
< 8	-	-
8	1273.38	7.051
9	1214.60	2.512
13	1205.28	8.778
≥ 14	1174.29	8.168

TABLE III

OPTIMAL COSTS USING UNIFORM QUANTIZATION

$p_{\max}$	Optimal cost	True λ
< 7	-	-
7	1214.60	2.512
8	1214.60	2.512
9	1205.77	2.512
≥ 12	1174.29	8.168

TABLE IV

COMPARISON OF SPACE COMPLEXITY

$p_{\max}$	Adaptive		Uniform		Improvement
	Edges	Nodes	Edges	Nodes	
8	47413	2253	174542	9257	3.68
9	55354	2541	205268	10486	3.71
12	79104	3405	297642	14207	3.76
13	87023	3693	328368	15436	3.77
14	94930	3970	359192	16682	3.78

the computation of the bound may not be preserved in the true propagation of the entire filter over the paths, unless certain extra conditions hold. These conditions are a subject of current work. □

B. Adaptive Quantization approach

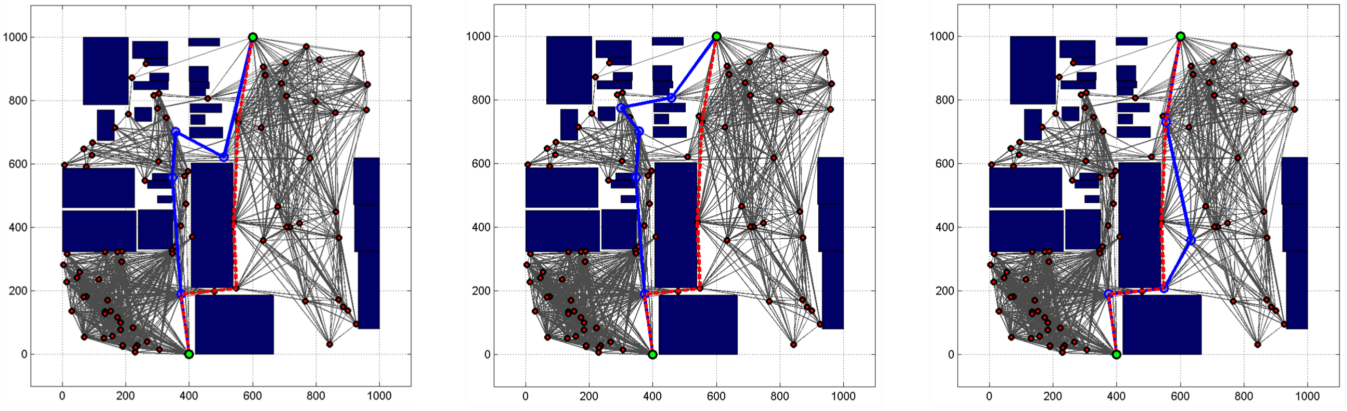
We experimented on a larger sized graph drawn out of a PRM. In this experiment, we investigated the effect of using the Adaptive Quantization level for the algorithm. Figure 5 shows how the optimal paths (solid blue) arising out of Algorithm 1 change with different values of the maximum uncertainty $p_{\max}$, using the Adaptive quantization levels. Figure 6 shows how the optimal paths (solid blue) arising out of Algorithm 1 change with different values of the maximum uncertainty $p_{\max}$, using the uniform quantization level. The optimal cost and the comparison with the true $\bar{\lambda}$ for each approach are reported in Tables II and III.

We also report a comparison of the space complexity and computational times[1] in using the uniform and the adaptive quantization approaches. In particular, we compare the ratio of the number of edges of the extended graphs, as well as the ratio of the computation time for both approaches.

The results summarized in Tables IV and V demonstrate the benefit of the adaptive quantization approach with respect to space complexity and computation time. However, this advantage comes at a cost of loss of accuracy as is seen from Figure 5(c), in which the adaptive quantization approach has selected a path that is different from the shortest path, although it falls within the maximum $\bar{\lambda}$ threshold.

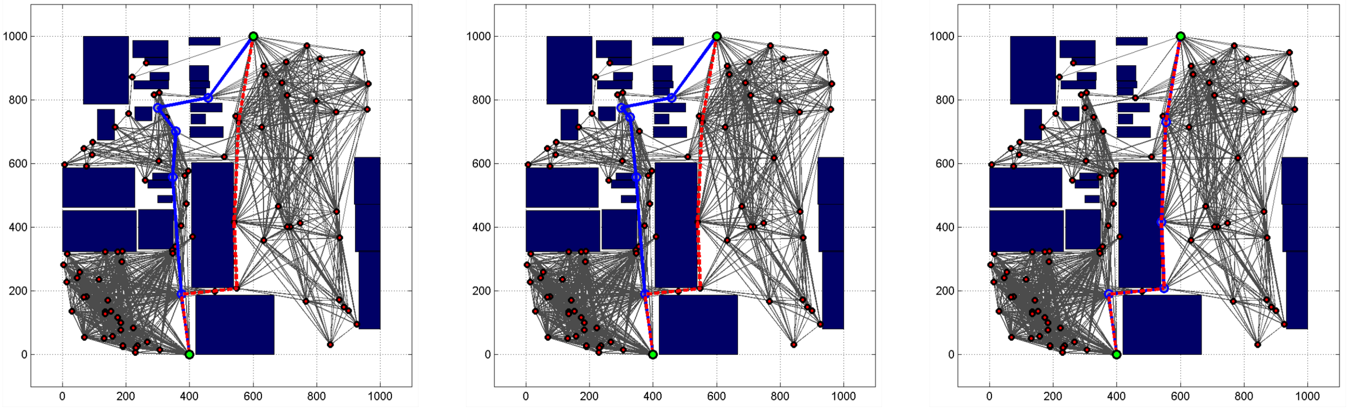
Remark 4.2 (Alternate metrics for uncertainty: Trace): While the use of $\bar{\lambda}(\mathbf{P})$ has a computational advantage due to the closed form expressions, that metric can provide conservative solutions because we are approximating the localization error by the worst case scenario. In this aspect, the trace should provide us with a less conservative solution, but at the expense of higher computation time. Our

[1] Times refer to a MATLAB implementation of the algorithms executed on an i7 Dual-core 2.8GHz, 8MB RAM. Matrices are represented as `sparse` data objects.



(a) $p_{\text{max}} = 8$ (b) $p_{\text{max}} = 9$ (c) $p_{\text{max}} = 13$

Fig. 5. Choice of different optimal paths using an Adaptive quantization level, for different values of maximum uncertainty p_{max}. The process uncertainty $\mathbf{Q} = 0.01\mathbf{I}$, and the measurement uncertainty $\mathbf{R}_j = 1, \forall j$. The best value of estimation precision $p_{\text{best}} = 0.001$.



(a) $p_{\text{max}} = 8$ (b) $p_{\text{max}} = 9$ (c) $p_{\text{max}} = 13$

Fig. 6. Choice of different optimal paths using the uniform quantization level for different values of maximum uncertainty p_{max}. The process uncertainty $\mathbf{Q} = 0.01\mathbf{I}$, and the measurement uncertainty $\mathbf{R}_j = 1, \forall j$. The best value of estimation precision $p_{\text{best}} = 0.001$.

TABLE V
COMPARISON OF TIME COMPLEXITY

p_{max}	Adaptive [s]	Uniform [s]	Improvement
8	41.94	73.45	1.75
9	45.46	97.30	2.14
12	48.82	175.65	3.60
13	51.04	215.62	4.22
14	52.95	262.78	4.96

approach for this metric leads to the following semi-definite optimization problem (SDP):

$$\begin{aligned} \min_{\Gamma, \mathbf{P} \in \mathbb{S}_{++}} \quad & \mathrm{Tr}\left(\Gamma \mathbf{H}_t (\mathbf{H}_t \Gamma \mathbf{H}_t' + \mathbf{R})^{-1} \mathbf{H}_t' \Gamma - \Gamma + \mathbf{P}\right) \\ \text{s.t.} \quad & \Gamma = \mathbf{F}_t \mathbf{P} \mathbf{F}_t' + \mathbf{Q} \\ & \mathrm{Tr}\left(\mathbf{P}\right) \geq p_{\text{best}}, \end{aligned}$$

which is a convex problem (see, e.g., [14] for a proof of concavity of the Riccati operator). Therefore, a global minimum exists, and is bounded away from zero. However, this SDP needs to be solved at each sampling instant, making this approach lot more computationally intensive than for $\bar{\lambda}$ as the metric, in which case Corollary 3.1 yields a closed-form expression. □

V. CONCLUSION AND FUTURE DIRECTIONS

This paper presented a novel approach to the problem of optimal path planning in GPS-denied environments, under a hard constraint on the size of the estimation error covariance. A novel bound on the maximum eigenvalue of the error covariance matrix was developed to produce an approximate solution in a computationally efficient manner. Two novel algorithms were proposed, one of which was more accurate while the other was more computationally efficient. Formal guarantees on the resulting solution in satisfying the secondary constraint as well as the complexity of the computation were provided. The application of the two algorithms on numerical examples showed promising results in terms of accuracy even though they are known to be sub-optimal.

A major open direction is to be able to provide guarantees on the factor of optimality or design new approaches with a factor of optimality guarantee with respect to the primary cost function. Extension to multiple secondary costs is another open direction. Additional directions include the presence of sensor misdetections in certain regions of the environments, which is the topic of our recent work [15].

References

[1] W. Ding, J. Wang, C. Rizos, and D. Kinlyside, "Improving adaptive Kalman estimation in GPS/INS integration," *The Journal of Navigation*, vol. 60, pp. 517–529, 2007.

[2] C. Hide, T. Moore, and M. Smith, "Adaptive Kalman filtering for low-cost INS/GPS," *The Journal of Navigation*, vol. 56, no. 1, 2003.

[3] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *International Journal of Robotics Research*, vol. 28, no. 11–12, pp. 1448–1465, 2009.

[4] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics*, vol. 12, no. 4, pp. 556–580, 1996.

[5] A. Gelb, Ed., *Applied Optimal Estimation.* MIT Press, 1974.

[6] R. He, S. Prentice, and N. Roy, "Planning in information space for a quadrotor helicopter in a GPS-denied environment," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008, pp. 814–820.

[7] H. Takeda, C. Facchinetti, and J.-C. Latombe, "Planning the motions of a mobile robot in a sensory uncertainty field," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 10, pp. 1002–1017, 1994.

[8] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *IEEE International Conference on Robotics and Automation*, 2011.

[9] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[10] J. van den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, 2011.

[11] R. Takei, W. Chen, Z. Clawson, S. Kirov, and A. Vladimirsky, "Optimal control with budget constraints and resets," *SIAM Journal on Control and Optimization*, 2011, http://arxiv.org/abs/1110.6221.

[12] H. Durrant-Whyte, "Multi sensor data fusion," http://www.acfr.usyd.edu.au/pdfs/training/multiSensorDataFusion/dataFusionNotes.pdf, 2001.

[13] T. H. Cormen, C. E. Lieserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms.* Prentice Hall, 2001.

[14] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[15] S. D. Bopardikar, B. Englot, and A. Speranzon, "Robust belief roadmap: Planning under uncertain and intermittent sensing," in *International Conference on Robotics and Automation*, 2014, to appear.

[16] S. D. Bopardikar, A. Speranzon, S. Zhang, and B. Sinopoli, "Performance analysis of linear estimators with unknown changes in sensors characteristics," in *American Control Conference*, 2013.

Appendix

To prove Theorem 3.1, we require the following intermediate result, which establishes a bound for the value of a scalar variable l which may evolve as per one out of two equations at any given time.

Lemma 1.1: Suppose that in the time interval $[0, 1, \dots, T]$, a scalar variable ℓ evolves as per

$$\ell_{t+1} = \begin{cases} \dfrac{a\ell_t + b}{c\ell_t + d}, & \text{for some } T-\kappa \text{ instants,} \\ al_t + b, & \text{for the remaining } \kappa \text{ instants,} \end{cases}$$

where a, b, c, d are some finite positive scalars, then

$$\ell_T \leq b\sum_{j=1}^{\kappa} a^{j-1} - \zeta a^{\kappa} + a^{\kappa} \Big/ \left(\left(\frac{d-\zeta c}{\zeta c + a}\right)^{T-\kappa} \frac{1}{\zeta + \ell_0} + \frac{c}{\zeta c + a} \left(\frac{1 - \frac{(d-\zeta c)^{T-\kappa}}{(\zeta c+a)^{T-\kappa}}}{1 - \frac{(d-\zeta c)}{(\zeta c + a)}} \right) \right)$$

where ζ is defined in Theorem 3.1.

Proof: Observing that for the above evolution of ℓ,

$$\frac{a\ell + b}{c\ell + d} + b \geq \frac{a(a\ell + b) + b}{c\ell + d},$$

which means that for *any* sequence of κ occurrences of the second equation, one can always upper bound the resulting ℓ trajectory by considering all the occurrences of evolution by the first equation, followed by the second.

The evolution given by the first equation in the time interval $[0, T-\kappa]$ can be simplified as follows. Set

$$\mu_t := 1 \Big/ (\zeta + \ell_t)$$

Since $bc > 0$, we get

$$\mu_t \geq \frac{d - \zeta c}{\zeta c + a}\mu_{t-1} + \frac{c}{\zeta c + a}$$

$$\Rightarrow \mu_t \geq \left(\frac{d - \zeta c}{\zeta c + a}\right)^k \mu_0 + \frac{c}{\zeta c + a}\left(\frac{1 - (d - \zeta c)^k/(\zeta c + a)^k}{1 - (d - \zeta c)/(\zeta c + a)}\right),$$

Therefore,

$$\ell_{T-\kappa} \leq 1 \Big/ \left(\left(\frac{d-\zeta c}{\zeta c + a}\right)^{T-\kappa} \frac{1}{\zeta + \ell_0} + \frac{c}{\zeta c + a} \left(\frac{1 - \frac{(d-\zeta c)^{T-\kappa}}{(\zeta c+a)^{T-\kappa}}}{1 - \frac{(d-\zeta c)}{(\zeta c + a)}} \right) \right) - \zeta.$$

The claim now follows since ℓ_T can be at most the above right hand side subject to the second, linear evolution for κ time steps. ■

Next, we recall the following recursion, originally proven in [16]. In this result, the notation $\underline{\lambda}$ refers to the minimum eigenvalue.

Theorem 1.1: At every time instant t,

$$\bar{\lambda}(\mathbf{P}_t) \leq \frac{\bar{\lambda}^2(\mathbf{F}_t)\bar{\lambda}(\mathbf{P}_{t-1}) + \bar{\lambda}(\mathbf{Q}_t)}{\underline{\lambda}\left(\mathbf{H}_t'\mathbf{R}_t^{-1}\mathbf{H}_t\right)\left(\bar{\lambda}^2(\mathbf{F}_t)\bar{\lambda}(\mathbf{P}_{t-1}) + \bar{\lambda}(\mathbf{Q}_t)\right) + 1}, \tag{5}$$

where the Jacobians $\mathbf{F}_t, \mathbf{H}_t$ are evaluated at $\mathbf{f}(\hat{\mathbf{x}}_{t-1}, \mathbf{0})$.

We can now prove Theorem 3.1.

Proof: [Proof of Theorem 3.1] Consider the recursion from Theorem 1.1. Substituting $z_t := \bar{\lambda}(\mathbf{P}_t)$, we obtain the following linear rational recurrence,

$$z_t \leq \frac{az_{t-1} + b}{cz_{t-1} + d} \tag{6}$$

where we used the definition of a, b, c and d.

Now, whenever $\hat{x} \in \mathcal{X} \setminus \mathcal{X}_S$, z will evolve as per (6). Otherwise, z evolves as per

$$z_t \leq az_{t-1} + b,$$

which happens at most κ times as per the assumption. Therefore, applying Lemma 1.1, the claim is established. ■