

# Generating Informative Paths for Persistent Sensing in Unknown Environments

Daniel E. Soltero    Mac Schwager    Daniela Rus

**Abstract**—We present an online algorithm for a robot to shape its path to a locally optimal configuration for collecting information in an unknown dynamic environment. As the robot travels along its path, it identifies both where the environment is changing, and how fast it is changing. The algorithm then morphs the robot’s path online to concentrate on the dynamic areas in the environment in proportion to their rate of change. A Lyapunov-like stability proof is used to show that, under our proposed path shaping algorithm, the path converges to a locally optimal configuration according to a Voronoi-based coverage criterion. The path shaping algorithm is then combined with a previously introduced speed controller to produce guaranteed persistent monitoring trajectories for a robot in an unknown dynamic environment. Simulation and experimental results with a quadrotor robot support the proposed approach.

## I. INTRODUCTION

Robots operating in dynamic and unknown environments are often faced with the problem of deciding where to go to get the most relevant information for their current task. Informative path planning addresses this problem. Given a dynamic unknown environment, and a robot with a sensor to measure the environment, we want to find a path for the robot that will maximize information gathering. To achieve this, the robot needs to do two things: 1) learn the structure of the environment by identifying the areas within the environment that are dynamic and the rate of change for these areas; 2) compute a path which allows it to sense the dynamic areas. Such a path is referred to as an *informative path*.

In this paper we present a new online path shaping algorithm for generating closed informative paths in unknown dynamic environments. The key insight is to use a parameter adaptation law inspired by [1] to learn a function representing the rate of change of each point in the environment. The robot moves along the path, marking the areas it observes as dynamic or static, and learning the rate of change of the dynamic areas. While learning the environment model, the algorithm simultaneously modifies its path based on this model by changing its waypoints to optimize a cost function related to the informativeness of the path. The cost function attempts to place the path waypoints at the centroids of their Voronoi cells, while also making sure the waypoints are not too far from one another. An example of the results of our path shaping algorithm can be seen in Figure 1.

This material is based upon work supported in part by ONR-MURI Award N00014-09-1-1051, the NSF Graduate Research Fellowship Award 0645960 & The Boeing Company.

D. E. Soltero and D. Rus are with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA (soltero@mit.edu, rus@csail.mit.edu). M. Schwager is with the Mechanical Engineering Department, Boston University, Boston, MA (schwager@bu.edu).

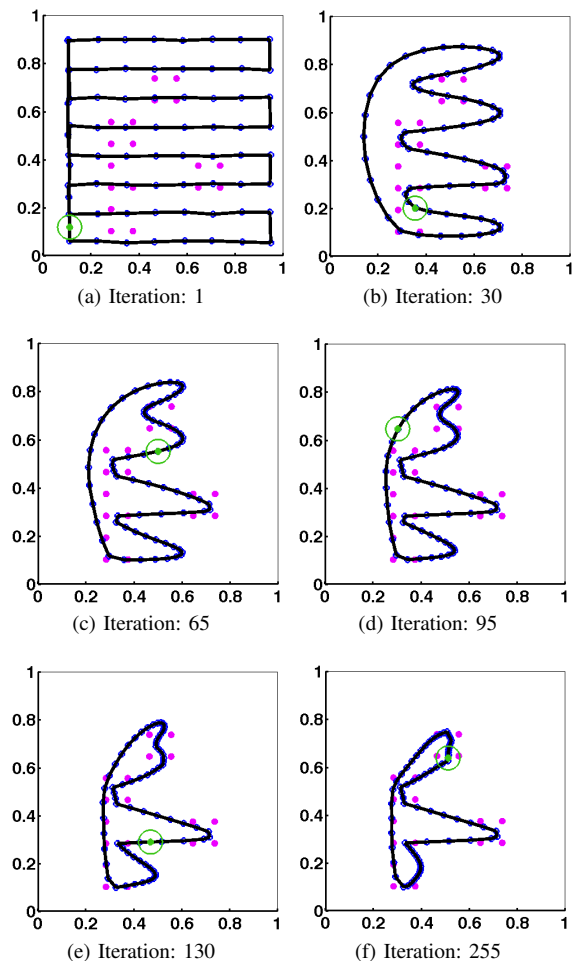


Fig. 1: Simulated system during the path shaping phase. The path, shown as the black line, connects all the waypoints, shown as blue circles. The points of interest are shown as purple dots. The simulated robot and its sensor footprint are shown as a green dot and green circle, respectively.

Many important applications call for robots to move along informative paths in unknown dynamic environments. For example, our algorithm could be used to provide surveillance of a city by allowing a robot to learn the regions where crime is frequently committed, and generate paths to visit the crime-ridden areas more frequently. Our algorithm could also be used for a robot vacuum cleaning task, where we want the robot to visit locations that accumulate dust and avoid locations that remain clean, or an estimation task, where we want the robot to keep an updated mental model of reality by visiting dynamic locations more frequently than static locations.

In this paper we apply our path shaping algorithm in combination with the speed control algorithm presented in [2] in order to design an informative *trajectory* (by which we mean the path and the speed along the path) to provide *persistent sensing* of an environment. In this application we wish for the robot, assumed to have a finite sensor footprint, to gather information so as to guarantee a bound on the difference between the robot's current model of the environment and the actual state of the environment for all time and over all locations. Since its sensor has a finite footprint, the robot cannot collect the data about all of the environment at once. As data about a dynamic region becomes outdated, the robot must return to that region repeatedly to collect new data. More generally, if different parts of the environment change at different rates, we wish to have the robot visit these areas in proportion to their rates of change to ensure a bounded uncertainty in the estimation. The algorithm from [2] calculates the speed of the robot at each point along a given path in order to perform a persistent sensing task, i.e. to prevent the robot's model of the environment from becoming too outdated. We use the path morphing algorithm in this paper as the input to the speed controller from [2] to produce informative trajectories online with provable performance guarantees.

More specifically, the persistent sensing problem is defined in [2] as an optimization whose goal is to keep a time changing environment, modeled as an *accumulation function*, as low as possible everywhere. The accumulation function grows where it is not covered by the robot's sensor, indicating a growing need to collect data at that location, and shrinks where it is covered by the robot's sensor, indicating a decreasing need for data collection. This accumulation function can be thought of as the amount of dust in a cleaning task, as the difference between the robot's mental model and reality in a mapping or estimation task, or as the chance of crime in a surveillance task. The goal of a persistent sensing task is to maintain this accumulation function bounded for the entire environment and for all time. In this paper we describe an algorithm for informative path planning that enables the robot to identify where and when to collect information.

The contributions of this paper are:

- a provably stable adaptive controller for learning the location of dynamic events in an environment, and computing an informative path for these events,
- a provably stable extension to the adaptive controller for computing informative paths to perform locally optimal persistent sensing tasks,
- simulation and hardware implementation.

#### A. Relation to Previous Work

Most of the previous work in path planning focuses on reaching a goal while avoiding collision with obstacles, e.g. [3], or on computing an optimal path according to some metric, e.g. [4]. Other works have focused on probabilistic approaches to path planning, e.g. [5]. More relevant to our work is the prior work in adaptive path planning, e.g. [6], [7], which considers adapting the path to changes in the system's

state. Even more relevant to our work is that of informative sensing, where the goal is to calculate paths that provide the most information for robots [8], provide the most information while maintaining periodic connectivity in order for robots to share information and synchronize [9], provide the most information taking into account complex vehicle dynamics and sensor limitations [10], and other variations. All of the mentioned previous work in informative sensing, as with most in path planning, treats the path planning problem as a search and/or recursive problem, with running times that can be very high for large systems, and some of them only providing approximations of optimal solutions. In contrast, in this paper we took another approach to path planning; we treat the problem as a continuous-time dynamical systems control problem and use adaptive control tools to create a novel algorithm for computing informative paths.

The adaptive controller we use to solve the informative path planning problem is based on Voronoi partitions. Related work in Voronoi coverage includes [11], where the objective was to design a sampling trajectory that minimized the uncertainty of an estimated random field at the end of a time frame. A form of generalized Voronoi partitions was used to solve this optimization problem. In our work, we also use Voronoi partitions as the basis of our algorithms, but the field, although unknown, is not random. Also, we are not concerned with optimizing trajectories that minimize the predictive variance, but rather we generate paths by optimizing the location of waypoints defining the path, according to a Voronoi-based coverage criterion in an unknown environment. In that sense, this work builds upon [1], where a group of agents were coordinated to place themselves in static, locally optimal locations to sense an unknown environment. Voronoi partitions were used to position the robots, while a parameter adaptation law allowed the agents to learn the environment model. We build upon this work by letting the waypoints in a single robot's path define a Voronoi decomposition, and using a parameter adaptation law to learn both *where* and *how fast* the environment is changing.

The persistent sensing concept motivating this work was introduced in [2], where a linear program was designed to calculate the robot's speed at each point along a path in order for it to perform a persistent sensing task, i.e. maintain the accumulation function bounded. The robot was assumed to have full knowledge of the environment, and was given a pre-designed path. In this paper we alleviate these two assumptions by having the robot estimate the static/dynamic structure of the environment, and use these estimates to shape its path into a useful path. These two alleviations provide a significant step towards persistent sensing in dynamic environments.

In Section II we set up the problem, present locational optimization tools, and present a basis function approximation of the environment. Section III introduces the adaptive controller and proves stability of the system under this controller. In Section IV, we introduce the controller extension, designed to execute persistent sensing tasks. Finally, in Section V we provide simulated and hardware results.

## II. PROBLEM FORMULATION

We assume we are given a robot whose task is to sense an unknown dynamic environment while traveling along a closed path consisting of a finite number of waypoints. The goal is for the robot to identify the areas within the environment that are dynamic and compute a path which allows it to sense these dynamic areas. A formal mathematical description of the problem follows.

Let there be a robot, with position  $p_r$ , traveling along a finite number  $n$  of waypoints in a convex, bounded area  $Q \subset \mathbb{R}^2$ . An arbitrary point in  $Q$  is denoted  $\mathbf{q}$  and the position of the  $i^{\text{th}}$  waypoint is denoted  $p_i$ . Let  $\{p_1, \dots, p_n\}$  be the configuration of the path and let  $\{V_1, \dots, V_n\}$  be the Voronoi partitions of  $Q$ , with waypoint positions as the generator points, defined as

$$V_i = \{\mathbf{q} \in Q : \|\mathbf{q} - p_i\| \leq \|\mathbf{q} - p_j\|, \forall j \neq i\},$$

where  $\|\cdot\|$  denotes the  $l^2$ -norm. We assume that the robot is able to compute the Voronoi partitions based on the waypoint locations, as is common in the literature [12], [13].

Since the path is closed, each waypoint  $i$  has a previous waypoint  $i - 1$  and next waypoint  $i + 1$  related to it, which are referred to as the *neighbor waypoints* of  $i$ . Note that  $i + 1 = 1$  for  $i = n$ , and  $i - 1 = n$  for  $i = 1$ . Once the robot reaches a waypoint, it continues to move to the next waypoint, in a straight line interpolation.

A sensory function, defined as a map  $\phi : Q \mapsto \mathbb{R}_{\geq 0}$  (where  $\mathbb{R}_{\geq 0}$  refers to non-negative scalars) determines the constant rate of change of the environment at point  $\mathbf{q} \in Q$ . The function  $\phi(\mathbf{q})$  is not known by the robot, but the robot is equipped with a sensor to make a point measurement of  $\phi(p_r)$  at its position  $p_r$ .

*Remark II.1.* The interpretation of the sensory function  $\phi(\mathbf{q})$  may be adjusted for a broad range of applications. It can be any kind of weighting of importance for points  $\mathbf{q} \in Q$ . In this paper we treat it as a rate of change in a dynamic environment.

### A. Locational Optimization

Let  $p_{i,j} = p_i - p_j$ . Notice that  $p_{i,j} = -p_{j,i}$ . Building upon [1], we can formulate the cost incurred by the system over the region  $Q$  as

$$\mathcal{H} = \sum_{i=1}^n \int_{V_i} \frac{W_s}{2} \|\mathbf{q} - p_i\|^2 \phi(\mathbf{q}) d\mathbf{q} + \sum_{i=1}^n \frac{W_n}{2} \|p_{i,i+1}\|^2, \quad (1)$$

where  $\|\mathbf{q} - p_i\|$  can be interpreted as the unreliability of measuring the sensory function  $\phi(\mathbf{q})$  when the robot is at  $p_i$ , and  $\|p_{i,i+1}\|$  can be interpreted as the cost of a waypoint being too far away from a neighboring waypoint.  $W_s$  and  $W_n$  are constant positive scalar weights assigned to the sensing task and neighbor distance, respectively. Note that unreliable sensing and distance between neighboring waypoints are expensive. A formal definition of informative path follows.

**Definition II.2** (Informative Path). An informative path corresponds to a set of waypoint locations that locally minimize (1).

Next we define three properties analogous to mass-moments of rigid bodies. The mass, first mass-moment, and centroid of  $V_i$  are defined as  $M_{V_i} = \int_{V_i} W_s \phi(\mathbf{q}) d\mathbf{q}$ ,  $L_{V_i} = \int_{V_i} W_s \mathbf{q} \phi(\mathbf{q}) d\mathbf{q}$  and  $C_{V_i} = \frac{L_{V_i}}{M_{V_i}}$ , respectively. Also, let  $e_i = C_{V_i} - p_i$ .

From locational optimization [14], and from differentiation under the integral sign for Voronoi partitions [15] we have

$$\frac{\partial \mathcal{H}}{\partial p_i} = -M_{V_i} e_i - W_n p_{i+1,i} - W_n p_{i-1,i}. \quad (2)$$

An equilibrium is reached when  $\frac{\partial \mathcal{H}}{\partial p_i} = 0$ . Assigning to each waypoint dynamics of the form

$$\dot{p}_i = u_i, \quad (3)$$

where  $u_i$  is the control input, we propose the following control law for the waypoints to converge to an equilibrium configuration:

$$u_i = \frac{K_i (M_{V_i} e_i + \alpha_i)}{\beta_i}, \quad (4)$$

where  $\alpha_i = W_n p_{i+1,i} + W_n p_{i-1,i}$ ,  $\beta_i = M_{V_i} + 2W_n$  and  $K_i$  is a uniformly positive definite matrix.

*Remark II.3.*  $\beta_i > 0$  has the nice effect of normalizing the weight distribution between sensing and staying close to neighboring waypoints. Also,  $K_i$  could potentially be time-varying to improve performance.

### B. Sensory Function Approximation

We assume that the sensory function  $\phi(\mathbf{q})$  can be parameterized as an unknown linear combination of a set of known basis functions. That is,

*Assumption II.4* (Matching condition).  $\exists a \in \mathbb{R}_{\geq 0}^m$  and  $\mathcal{K} : Q \mapsto \mathbb{R}_{\geq 0}^m$ , where  $\mathbb{R}_{\geq 0}^m$  is a vector of non-negative entries, such that

$$\phi(\mathbf{q}) = \mathcal{K}(\mathbf{q})^T a, \quad (5)$$

where the vector of basis functions  $\mathcal{K}(\mathbf{q})$  is known by the robot, but the parameter vector  $a$  is unknown.

Let  $\hat{a}(t)$  be the robot's approximation of the parameter vector  $a$ . Then,  $\hat{\phi}(\mathbf{q}) = \mathcal{K}(\mathbf{q})^T \hat{a}$  is the robot's approximation of  $\phi(\mathbf{q})$ . Building on this, we define the mass-moment approximations as

$$\begin{aligned} \hat{M}_{V_i} &= \int_{V_i} W_s \hat{\phi}(\mathbf{q}) d\mathbf{q}, & \hat{L}_{V_i} &= \int_{V_i} W_s \mathbf{q} \hat{\phi}(\mathbf{q}) d\mathbf{q}, \\ \hat{C}_{V_i} &= \frac{\hat{L}_{V_i}}{\hat{M}_{V_i}}. \end{aligned}$$

Additionally, we can define  $\tilde{a} = \hat{a} - a$ , and the sensory function error, and mass-moment errors as

$$\tilde{\phi}(\mathbf{q}) = \hat{\phi}(\mathbf{q}) - \phi(\mathbf{q}) = \mathcal{K}(\mathbf{q})^T \tilde{a}, \quad (6)$$

$$\tilde{M}_{V_i} = \hat{M}_{V_i} - M_{V_i} = \int_{V_i} W_s \mathcal{K}(\mathbf{q})^T d\mathbf{q} \tilde{a}, \quad (7)$$

$$\tilde{L}_{V_i} = \hat{L}_{V_i} - L_{V_i} = \int_{V_i} W_s \mathbf{q} \mathcal{K}(\mathbf{q})^T d\mathbf{q} \tilde{a}, \quad (8)$$

$$\tilde{C}_{V_i} = \frac{\tilde{L}_{V_i}}{\tilde{M}_{V_i}}. \quad (9)$$

Finally, in order to compress the notation, let  $\mathcal{K}_r(t)$  and  $\phi_r(t)$  be the value of the basis function vector and the value of  $\phi$  at the robot's position  $p_r(t)$ , respectively.

### III. ADAPTIVE CONTROLLER

We design an adaptive control law and prove that it causes the path to converge to a locally optimal configuration according to (1), while causing the robot's estimate of the environment change rates to converge to the real description by integrating its sensory measurements along its trajectory.

Since the robot does not know  $\phi(\mathbf{q})$ , but has an estimate  $\hat{\phi}(\mathbf{q})$ , the control law from (4) becomes

$$u_i = \frac{K_i(\hat{M}_{V_i}\hat{e}_i + \alpha_i)}{\hat{\beta}_i}, \quad (10)$$

where

$$\alpha_i = W_n p_{i+1,i} + W_n p_{i-1,i}, \quad \hat{\beta}_i = \hat{M}_{V_i} + 2W_n,$$

$$\hat{e}_i = \hat{C}_{V_i} - p_i.$$

The parameter  $\hat{a}$  is adjusted according to an adaptation law which is described next. Let

$$\Lambda = \int_0^t w(\tau) \mathcal{K}_r(\tau) \mathcal{K}_r(\tau)^T d\tau, \quad (11)$$

$$\lambda = \int_0^t w(\tau) \mathcal{K}_r(\tau) \phi_r(\tau) d\tau, \quad (12)$$

where  $w(t)$  is a positive constant scalar if  $t < \tau_w$ , and zero otherwise, and  $\tau_w$  is some positive time at which part of the adaptation shuts down to maintain  $\Lambda$  and  $\lambda$  bounded. Let

$$b = \sum_{i=1}^n \int_{V_i} W_s \mathcal{K}(\mathbf{q})(\mathbf{q} - p_i)^T d\mathbf{q} \dot{p}_i, \quad (13)$$

$$\dot{\hat{a}}_{\text{pre}} = -b - \gamma(\Lambda \hat{a} - \lambda), \quad (14)$$

where  $\gamma > 0$  is the adaptation gain scalar. Since  $a(j) \geq 0, \forall j$  (where  $a(j)$  denotes the  $j^{\text{th}}$  element of  $a$ ), we enforce  $\hat{a}(j) \geq 0, \forall j$ . We do this by using a projection law [1],

$$\dot{\hat{a}} = \Gamma(\dot{\hat{a}}_{\text{pre}} - I_{\text{proj}} \dot{\hat{a}}_{\text{pre}}), \quad (15)$$

where  $\Gamma \in \mathbb{R}^{m \times m}$  is a diagonal positive definite adaptation gain matrix, and the diagonal matrix  $I_{\text{proj}}$  is defined element-wise as

$$I_{\text{proj}}(j) = \begin{cases} 0, & \text{if } \hat{a}(j) > 0, \\ 0, & \text{if } \hat{a}(j) = 0 \text{ and } \dot{\hat{a}}_{\text{pre}}(j) \geq 0, \\ 1, & \text{otherwise,} \end{cases} \quad (16)$$

where  $(j)$  denotes the  $j^{\text{th}}$  element for a vector and the  $j^{\text{th}}$  diagonal element for a matrix.

**Theorem III.1** (Convergence Theorem). *Under Assumption II.4, with waypoint dynamics specified by (3), control law specified by (10) and adaptive law specified by (15), we have*

$$(i) \lim_{t \rightarrow \infty} \|\hat{M}_{V_i}(t)\hat{e}_i(t) + \alpha_i(t)\| = 0 \quad \forall i \in \{1, \dots, n\},$$

$$(ii) \lim_{t \rightarrow \infty} \|\tilde{\phi}_r(\tau)\| = 0 \quad \forall \tau \mid w(\tau) > 0.$$

*Proof.* We define a Lyapunov-like function based on the robot's path and environment estimate, and use Barbalat's lemma to prove asymptotic stability of the system to a locally optimal equilibrium.

Let

$$\mathcal{V} = \mathcal{H} + \frac{1}{2} \tilde{a}^T \Gamma^{-1} \tilde{a}. \quad (17)$$

Taking the time derivative of  $\mathcal{V}$ , we obtain

$$\begin{aligned} \dot{\mathcal{V}} &= \sum_{i=1}^n \frac{\partial \mathcal{H}^T}{\partial p_i} \dot{p}_i + \tilde{a}^T \Gamma^{-1} \dot{\tilde{a}} \\ &= \sum_{i=1}^n -(M_{V_i} e_i + \alpha_r)^T \dot{p}_i + \tilde{a}^T \Gamma^{-1} \dot{\tilde{a}}. \end{aligned} \quad (18)$$

From (7), (8), (9), it is easy to check that

$$L_{V_i} = M_{V_i} C_{V_i} = M_{V_i} \hat{C}_{V_i} + \tilde{M}_{V_i} (\hat{C}_{V_i} - \tilde{C}_{V_i}).$$

Plugging this into (18),

$$\begin{aligned} \dot{\mathcal{V}} &= \sum_{i=1}^n -(M_{V_i} \hat{e}_i + \alpha_i)^T \dot{p}_i + (\tilde{L}_{V_i} - \tilde{M}_{V_i} \hat{C}_{V_i})^T \dot{p}_i \\ &\quad + \tilde{a}^T \Gamma^{-1} \dot{\tilde{a}}. \end{aligned}$$

Using (7), we have

$$\begin{aligned} \dot{\mathcal{V}} &= \sum_{i=1}^n -(\hat{M}_{V_i} \hat{e}_i + \alpha_i)^T \dot{p}_i + (\tilde{L}_{V_i} - \tilde{M}_{V_i} p_i)^T \dot{p}_i \\ &\quad + \tilde{a}^T \Gamma^{-1} \dot{\tilde{a}}. \end{aligned}$$

Substituting the dynamics specified by (3) and control law specified by (10), we obtain

$$\begin{aligned} \dot{\mathcal{V}} &= \sum_{i=1}^n -\frac{1}{\hat{\beta}_i} (\hat{M}_{V_i} \hat{e}_i + \alpha_i)^T K_i (\hat{M}_{V_i} \hat{e}_i + \alpha_i) \\ &\quad + \sum_{i=1}^n (\tilde{L}_{V_i} - \tilde{M}_{V_i} p_i)^T \dot{p}_i + \tilde{a}^T \Gamma^{-1} \dot{\tilde{a}}. \end{aligned}$$

Using (7) and (8),

$$\begin{aligned} \dot{\mathcal{V}} &= \sum_{i=1}^n -\frac{1}{\hat{\beta}_i} (\hat{M}_{V_i} \hat{e}_i + \alpha_i)^T K_i (\hat{M}_{V_i} \hat{e}_i + \alpha_i) \\ &\quad + \tilde{a}^T \sum_{i=1}^n \int_{V_i} W_s \mathcal{K}(\mathbf{q})(\mathbf{q} - p_i)^T d\mathbf{q} \dot{p}_i \\ &\quad + \tilde{a}^T \Gamma^{-1} \dot{\tilde{a}}. \end{aligned}$$

Plugging in the adaptation law from (15), (11) and (12),

$$\begin{aligned} \dot{\mathcal{V}} &= \sum_{i=1}^n -\frac{1}{\hat{\beta}_i} (\hat{M}_{V_i} \hat{e}_i + \alpha_i)^T K_i (\hat{M}_{V_i} \hat{e}_i + \alpha_i) \\ &\quad - \gamma \int_0^t w(\tau) (\tilde{\phi}_r(\tau))^2 d\tau - \tilde{a}^T I_{\text{proj}} \dot{\hat{a}}_{\text{pre}}. \end{aligned} \quad (19)$$

Denote the three terms in (19) as  $\xi_1(t)$ ,  $\xi_2(t)$  and  $\xi_3(t)$ , so that  $\dot{\mathcal{V}}(t) = \xi_1(t) + \xi_2(t) + \xi_3(t)$ . Notice that  $\xi_1(t) \leq 0$  since  $K_i$  is uniformly positive definite and  $\hat{\beta}_i > 0$ ,  $\xi_2(t) \leq 0$  since it is the negative integral of a squared quantity, and it was proven in [1] that  $\xi_3(t) \leq 0$ . Now consider the time

integral of each of these three terms,  $\int_0^t \xi_k(\tau) d\tau$ ,  $k = 1, 2, 3$ . Since each of the terms is non-positive,  $\int_0^t \xi_k(\tau) d\tau \leq 0$ ,  $\forall k$ , and since  $\mathcal{V} > 0$ , each integral is lower bounded by  $\int_0^t \xi_k(\tau) d\tau \geq -\mathcal{V}_0$ , where  $\mathcal{V}_0$  is the initial value of  $\mathcal{V}$ . Therefore, these integrals are lower bounded and non-increasing, and hence  $\lim_{t \rightarrow \infty} \int_0^t \xi_k(\tau) d\tau$  exists and is finite for all  $k$ . Furthermore, it was shown in [1] (Lemma 1) that  $\dot{\xi}_1(t)$  and  $\dot{\xi}_2(t)$  are uniformly bounded, therefore  $\xi_1(t)$  and  $\xi_2(t)$  are uniformly continuous. Hence, by Barbalat's Lemma [16],  $\xi_1(t) \rightarrow 0$  and  $\xi_2(t) \rightarrow 0$ . This implies (i) and (ii).  $\square$

*Remark III.2.* Property (i) from Theorem III.1 implies that the path will reach a locally optimal configuration for sensing, i.e. an informative path, where the waypoints reach a stable balance between providing good sensing locations and being close to their neighbor waypoints. Additionally, this balance can be tuned to the desired behavior (short paths vs. good coverage) by proper selection of  $W_n$  and  $W_s$ .

*Remark III.3.* Property (ii) from Theorem III.1 implies that the sensory function estimate error,  $\hat{\phi}(\mathbf{q})$ , will converge to zero for all points on the robot's trajectory with positive weight  $w(t)$ , but not necessarily for all the environment. This means that the robot will learn the true sensory function for the environment if its trajectory is rich enough while the weight is positive. Since it is crucial for the sensory function estimate to approach the true sensory function for all the environment, the initial waypoint locations can be designed so the robot initially travels most of the environment in dynamic unknown environments (see Figure 2).

This adaptive controller is applicable to any sensing task done by a robot without full knowledge of the environment structure. In the next section we present an extension that is specific to persistent sensing tasks.

#### IV. ADAPTIVE CONTROLLER FOR PERSISTENT SENSING

The main motivation behind this work is to generate an informative path that can be used by a robot tasked with persistent sensing. We now extend the adaptive controller from Section III and prove that it enables a robot with a finite sensor footprint to learn the environment dynamics, in the form of growth rates of the environment's accumulation function, and to generate an informative path to follow and sense the growing accumulation function in order to maintain it bounded.

The robot is assumed to be equipped with a sensor with a finite footprint. Examples of sensors with finite footprint are a camera for a surveillance task and a vacuum cleaner for a cleaning task. Although any footprint shape can be used, we use a constant circular footprint for simplicity, defined as  $F(p_r) = \{\mathbf{q} \in Q : \|\mathbf{q} - p_r\| \leq \rho\}$ , when the robot is at location  $p_r$ , where  $\rho$  is a constant positive scalar.

We would like to use the stability criterion for a persistent sensing task given the speed of the robot at each point along the path, referred to as the *speed profile*, defined in [2], and plug it into the adaptive controller, such that the control action increases the *stability margin* of the persistent sensing task through time. However, since the robot does not

know the environment, but has an estimate of it, it uses the estimated version of this stability criterion, given by

$$\frac{\hat{\phi}(\mathbf{q}, t)}{c(\mathbf{q})} T(t) - \tau_c(\mathbf{q}, t) = \hat{s}(\mathbf{q}, t) < 0, \quad \forall \mathbf{q} \text{ s.t. } \hat{\phi}(\mathbf{q}, t) > 0, \quad (20)$$

where  $\hat{\phi}(\mathbf{q}, t)$  (the estimated sensory function) is the estimated rate at time  $t$  at which the environment's accumulation function grows at point  $\mathbf{q}$  (referred to as *growth rate*), the constant scalar  $c(\mathbf{q})$  is the rate at which the accumulation function shrinks (is consumed) when the robot's sensor is covering point  $\mathbf{q}$ , and  $c(\mathbf{q}) > \phi(\mathbf{q})$ ,  $\forall \mathbf{q}$  is necessary for a stable persistent task to be feasible. Also,  $T(t)$  is the time it takes the robot to complete the path at time  $t$ , and  $\tau_c(\mathbf{q}, t)$  is the time the robot's sensor covers point  $\mathbf{q}$  along the path at time  $t$ . These last two quantities are calculated using the speed profile for the path at time  $t$  and the robot's sensor footprint  $F(p_r)$ . The estimated stability margin of the system is  $\hat{S}(t) = -(\max_{\mathbf{q}} \hat{s}(\mathbf{q}, t))$ . A *stable persistent task* is one in which  $S$  (the true version of  $\hat{S}$ ) is positive, which means the robot is able to maintain the environment's accumulation function bounded at all points  $\mathbf{q}$ . Note that only points  $\mathbf{q}$  that satisfy  $\hat{\phi}(\mathbf{q}, t) > 0$  are considered in a persistent sensing task since it is not necessary to persistently sense a point that has no sensory interest. Points that satisfy this condition are referred to as *points of interest*.

In [2], a linear program was given which can calculate the speed profile for the path at time  $t$  that maximizes  $\hat{S}(t)$  (or  $S(t)$  for ground-truth). With the speed profiles obtained with this linear program, and using (20), we can formulate a new controller to drive the robot's path in a direction to perform stable persistent sensing tasks. Hence, from this point on, we assume that this maximizing speed profile is known and used to obtain  $\hat{s}(\mathbf{q}, t)$ ,  $\forall \mathbf{q}$ ,  $\forall t$ .

Let the waypoints have new dynamics of the form

$$\dot{p}_i = I_i u_i, \quad (21)$$

where  $u_i$  is defined in (10),

$$I_i = \begin{cases} 0, & \text{if } \frac{\partial \hat{S}}{\partial p_i} T < 0 \text{ and } t - t_u^i > \tau_{\text{dwell}}, \\ 1, & \text{otherwise,} \end{cases} \quad (22)$$

$\tau_{\text{dwell}}$  is a design parameter, and  $t_u^i$  is the most recent time at which  $I_i$  switched from zero to one (switched "up"). Equations (21) and (22) look at how each waypoint movement affects the stability margin through time, and ensure that this quantity does not decrease.

*Remark IV.1.* For positive  $\epsilon \rightarrow 0$ ,  $\frac{\partial \hat{S}}{\partial p_i}(t)$  is not always defined when  $\arg \max_{\mathbf{q}} \hat{s}(\mathbf{q}, t - \epsilon) \neq \arg \max_{\mathbf{q}} \hat{s}(\mathbf{q}, t + \epsilon)$ . In such cases  $\frac{\partial \hat{S}}{\partial p_i}(t)$  refers to  $\frac{\partial \hat{S}}{\partial p_i}(t + \epsilon)$ .

*Theorem IV.2* (Convergence Theorem for Persistent Sensing). *Under Assumption II.4, with waypoint dynamics specified by (21), control law specified by (10), and adaptive law specified by (15), we have*

$$(i) \lim_{t \rightarrow \infty} I_i(t) \|\hat{M}_{V_i}(t) \hat{e}_i(t) + \alpha_i(t)\| = 0, \quad \forall i \in \{1, \dots, n\},$$

(ii)  $\lim_{t \rightarrow \infty} \|\tilde{\phi}_r(\tau)\| = 0, \quad \forall \tau \mid w(\tau) > 0.$

*Proof.* We define a Lyapunov-like function based on the robot's path and environment-rates estimate, and prove asymptotic stability of the system to a locally optimal equilibrium. However, contrary to the previous section, a bit more work is needed to prove property (i) due to the piecewise differentiability of the new Lyapunov-like function caused by the new waypoint dynamics.

Let  $\mathcal{V}^f$  be the new Lyapunov-like function, and let  $\mathcal{V}^f = \mathcal{V}$  from (17). Then, following the procedure from Section III, but with  $\dot{p}_i$  defined by (21), we get

$$\dot{\mathcal{V}}^f = \sum_{i=1}^n -\frac{1}{\beta_i} (\hat{M}_{V_i} \hat{e}_i + \alpha_i)^T I_i K_i (\hat{M}_{V_i} \hat{e}_i + \alpha_i) - \gamma \int_0^t w(\tau) (\tilde{\phi}_r(\tau))^2 d\tau - \tilde{a}^T I_{\text{proj}} \dot{\hat{a}}_{\text{pre}}. \quad (23)$$

Using a similar analysis as in Section III, denote the three terms in (23) as  $-\xi_1(t)$ ,  $-\xi_2(t)$  and  $-\xi_3(t)$ , so that  $\dot{\mathcal{V}}^f(t) = -\xi_1(t) - \xi_2(t) - \xi_3(t)$ . In Section III we showed that  $\xi_1(t) \geq 0$ ,  $\xi_2(t) \geq 0$ , and  $\xi_3(t) \geq 0$ . Additionally, the time integral of each of these three terms,  $\lim_{t \rightarrow \infty} \int_0^t \xi_k(\tau) d\tau$ , exists and is finite for all  $k$ . It was shown in [1] (Lemma 1) that  $\dot{\xi}_2(t)$  is uniformly bounded, therefore  $\xi_2(t)$  is uniformly continuous. Hence, by Barbalat's Lemma,  $\xi_2(t) \rightarrow 0$ . This implies (ii).

Now, as proven before,  $\int_0^\infty \xi_1(\tau) d\tau$  exists and is finite, and we want to show that  $\lim_{t \rightarrow \infty} \xi_1(t) = 0$ . Let  $\xi_1^i$  be defined such that  $\xi_1 = \sum_{i=1}^n \xi_1^i$ . Let us assume that  $\lim_{t \rightarrow \infty} \xi_1^i(t) \neq 0$ , that is,  $\forall t \exists t_j \geq t, \epsilon > 0$ , such that  $\xi_1^i(t_j) \geq \epsilon$ . Let  $\{t_j\}_{j=1}^\infty$  be the infinite sequence of  $t_j$ 's separated by more than  $2\tau_{\text{dwell}}$  such that  $\xi_1^i(t_j) \geq \epsilon$ . That is,  $|t_j - t_{j'}| > 2\tau_{\text{dwell}}, \forall j \neq j'$  and  $t_j, t_{j'} \in \{t_j\}_{j=1}^\infty$ . Since, from [1] (Lemma 1),  $\dot{\xi}_1^i(t)$  is uniformly bounded by some value  $B$  when  $I_i = 1$  (i.e.  $|\dot{\xi}_1^i(t)| \leq B$ ), and whenever  $I_i = 1$ , it remains with this value for at least  $\tau_{\text{dwell}}$ , then we have that  $\forall t_j \in \{t_j\}_{j=1}^\infty$ ,

$$\int_{t_j - \tau_{\text{dwell}}}^{t_j + \tau_{\text{dwell}}} \xi_1^i(\tau) d\tau \geq \epsilon \delta > 0, \quad (24)$$

where

$$\delta = \min\left\{\frac{\epsilon}{2B}, \frac{\tau_{\text{dwell}}}{2}\right\} > 0. \quad (25)$$

Then

$$\begin{aligned} \int_0^\infty \xi_1^i(\tau) d\tau &\geq \sum_{t_j \in \{t_j\}_{j=1}^\infty} \int_{t_j - \tau_{\text{dwell}}}^{t_j + \tau_{\text{dwell}}} \xi_1^i(\tau) d\tau \\ &\geq \sum_{t_j \in \{t_j\}_{j=1}^\infty} \epsilon \delta, \end{aligned} \quad (26)$$

which is infinite, and contradicts the already proven fact that  $\int_0^\infty \xi_1(\tau) d\tau = \sum_{i=1}^n \int_0^\infty \xi_1^i(\tau) d\tau$  exists and is finite. Therefore, by contradiction,  $\lim_{t \rightarrow \infty} \xi_1^i(t) = 0$ , hence property (i) holds.  $\square$

*Remark IV.3.* The stability margin can theoretically worsen while  $I_i$ , for some  $i$ , cannot switch from one to zero because

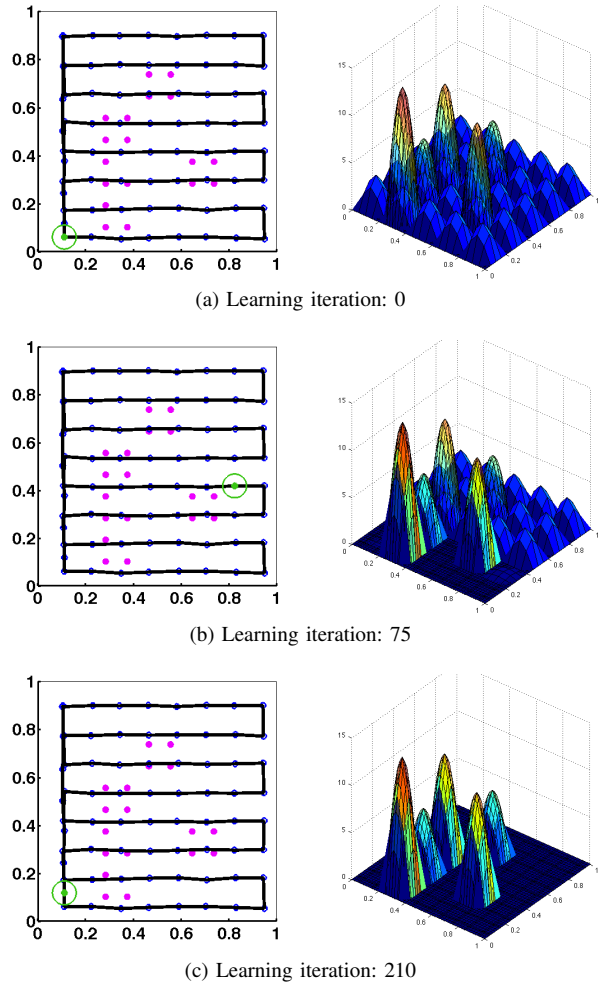


Fig. 2: Simulated system during the learning phase. Left: 1) the path, shown as the black line, connects all the waypoints, shown as blue circles; 2) the points of interest in the environment are shown as purple dots; 3) the simulated robot and its sensor footprint are shown as a green dot and green circle, respectively. Right: the environment growth rates, where the translucent data represents the true environment growth rates and the solid-colored data represents the estimated growth rates.

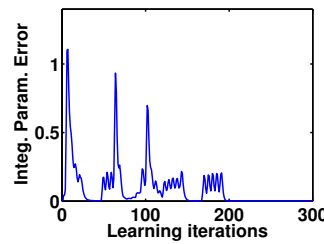


Fig. 3: Integral parameter error

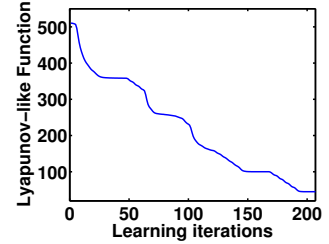


Fig. 4: Lyapunov-like function

it is waiting for  $t - t_u^i > \tau_{\text{dwell}}$ . However,  $\tau_{\text{dwell}}$  can be selected arbitrarily small and, in practice, any computer will enforce a  $\tau_{\text{dwell}}$  due to discrete time steps. Therefore, it is not a practical restriction. As a result, intuitively, (i) from Theorem IV.2 means that  $\lim_{t \rightarrow \infty} \|\hat{M}_{V_i}(t) \hat{e}_i(t) + \alpha_i(t)\| = 0$  only if this helps the persistent sensing task. Otherwise  $\lim_{t \rightarrow \infty} I_i(t) = 0$ , meaning that the persistent task will not benefit if waypoint  $i$  moves.



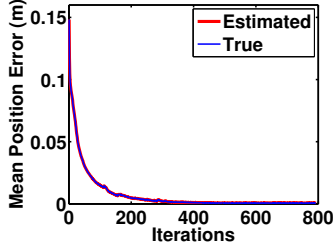


Fig. 5: Mean waypoint position error

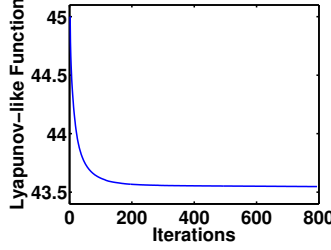


Fig. 6: Lyapunov-like function

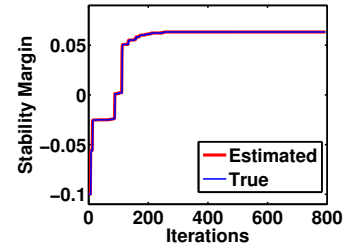


Fig. 7: Stability margin for persistent sensing task

## V. SIMULATION, IMPLEMENTATION AND RESULTS

### A. Numerical Simulation

The adaptive controller for persistent tasks was simulated in a MATLAB environment for many test cases. Here we present a case for  $n = 71$  waypoints. A fixed-time step numerical solver was used with a time step of 0.01 seconds, and  $\tau_{\text{dwell}} = 0.009$ . The region  $Q$  was taken to be the unit square. The sensory function  $\phi(\mathbf{q})$  was parametrized as a Gaussian network with 25 truncated Gaussians, i.e.  $\mathcal{K} = [\mathcal{K}(1) \cdots \mathcal{K}(25)]^T$ , where

$$\begin{aligned} G(j) &= \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(\mathbf{q} - \mu_j)^2}{2\sigma^2}\right\}, \\ G_{\text{trunc}} &= \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{\rho_{\text{trunc}}^2}{2\sigma^2}\right\}, \\ \mathcal{K}(j) &= \begin{cases} G(j) - G_{\text{trunc}}, & \text{if } \|\mathbf{q} - \mu_j\| < \rho_{\text{trunc}}, \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \quad (27)$$

$\sigma = 0.18$  and  $\rho_{\text{trunc}} = 0.1$ . The unit square was divided into an even  $5 \times 5$  grid and each  $\mu_j$  was chosen so that each of the 25 Gaussians was centered at its corresponding grid square. The parameters were chosen as  $a(6) = 20$ ,  $a(7) = 10$ ,  $a(8) = 16$ ,  $a(14) = 10$ ,  $a(17) = 16$ , and  $a(j) = 0$  otherwise. The environment growth rates created with these parameters can be seen in Figure 2c. The estimated parameter  $\hat{a}$  was initialized element-wise to 5, and  $\Lambda$  and  $\lambda$  were initialized to zero. The parameters for the controller were  $K_i = 30$ ,  $\forall i$ ,  $\Gamma = \text{identity}$ ,  $\gamma = 500$ ,  $W_n = 3$ ,  $W_s = 50$ ,  $w = 10$ , and  $\rho = 0.05$ . The spatial integrals were approximated by discretizing each Voronoi region into a  $7 \times 7$  grid and summing contributions of the integrand over the grid. Voronoi regions were computed using a decentralized algorithm similar to the one in [12]. The environment was discretized into a  $12 \times 12$  grid and only points in this grid that satisfied  $\hat{\phi}(\mathbf{q}) > 0$  were used as points of interest in (20). This discretization is only used in (20), and by using this discretized version of the environment, the running time for experiments is greatly reduced. For more sensitive systems, this grid can be refined.

The initial path was designed to “zig-zag” across the environment (see Figure 2) to provide a rich initial trajectory for the robot to sample the environment. We first allowed the robot to go through the initial path without reshaping it so that it could sample the space and learn the distribution of sensory information in the environment. Therefore, we present results in two separate phases: 1) learning phase, and

2) path shaping phase. The learning phase corresponds to the robot traveling through its full path once, without reshaping it, in order to learn the environment rates of change. The path shaping phase corresponds to when (21) is used to reshape the path into an informative path, and starts when the learning phase is done. In the path shaping phase,  $w = 0$ .

1) *Learning Phase*: The robot travels its entire path once, measuring the environment growth rates as it travels and using the adaptation law from (15) to estimate these rates. This process can be seen in Figure 2. As the robot travels its path, the adaptation law causes  $\dot{\hat{\phi}}(\mathbf{q}) \rightarrow 0$ ,  $\forall \mathbf{q} \in Q$ , as can be seen from Figures 2a, 2b, 2c, where the robot’s estimate of the environment growth rates ( $\hat{\phi}(\mathbf{q})$ , solid-colored data) converges to the real environment growth rates ( $\phi(\mathbf{q})$ , translucent data) for all of the space. This means that the robot’s trajectory was rich enough to generate accurate estimates of the environment structure. Figure 3 shows that  $\lim_{t \rightarrow \infty} \int_0^t w(\tau) (\hat{\phi}_r(\tau))^2 d\tau = 0$ , in accordance with (ii) from Theorem IV.2. Finally, for this learning phase, we see in Figure 4 that the Lyapunov-like function  $\mathcal{V}^f$  is monotonically non-increasing, which supports our theory.

2) *Path Shaping Phase*: Once the robot travels through its path once (and learns the environment growth rates), the controller from Section IV is activated. We can see how the path evolves under this controller in Figure 1. The robot learns the location of the points of interest (with positive growth rate) in the learning phase and, consequently, shapes its path to cover these points while improving the stability margin of the persistent sensing task. After 65 iterations (Figure 1c), the path already tends to go through all points of interest, while avoiding all other points. At 255 iterations (Figure 1f), the path clearly only goes through points of interest, enabling the robot to perform a locally optimal persistent sensing task. The path at 800 iterations is approximately the same to the one at 255 iterations.

Figure 5 shows the true and estimated mean waypoint position errors, where the estimated error refers to the quantity  $I_i(t) \|\hat{M}_{V_i}(t) \hat{e}_i(t) + \alpha_i(t)\|$ . As shown,  $\lim_{t \rightarrow \infty} I_i(t) \|\hat{M}_{V_i}(t) \hat{e}_i(t) + \alpha_i(t)\| = 0$ , in accordance to (i) from Theorem IV.2. Figure 6 shows the Lyapunov-like function  $\mathcal{V}^f$  monotonically non-increasing during this path morphing phase and settling to a local minimum, meaning that the path is driven to a locally optimal configuration that is helpful for the persistent sensing task. The initial value of this function in the path morphing phase is the final value of the function in the learning phase.

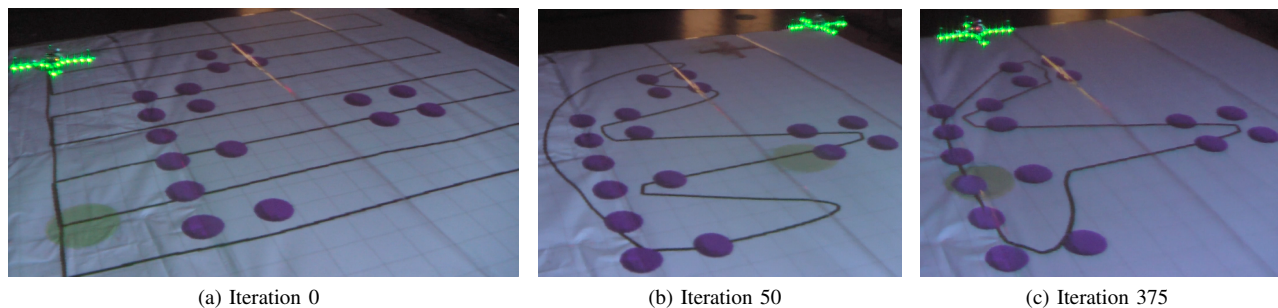


Fig. 8: Hardware implementation of the system using a quadrotor robot. Three snapshots of the path shaping phase at different iteration values are shown. The path, shown as the black line, connects all the waypoints. The points of interest in the environment are shown as purple dots. The robot is the green-lit quadrotor, and its sensor footprint is represented by the green circle under the robot's position.

Finally, Figure 7 shows the persistent sensing task's stability margin evolving through time, using the speed profiles from [2]. The estimated stability margins is shown, as well as the true value for ground-truth. The stability margin starts off with a negative value, indicating that the persistent sensing task is initially unstable, and increases through time, indicating that the path is morphing to make the persistent sensing task "more stable". By the end of the simulation, the persistent sensing task is stable, i.e. the stability margin is positive, and the robot achieves an informative trajectory. The discrete jumps in the stability margin are due to the discrete time steps, and can be minimized by shortening these steps.

### B. Implementation

The adaptive controller for persistent tasks was implemented with a quadrotor robot for the same simulated environment as described earlier and for a path consisting of 71 waypoints. The accompanying video submission presents the results of this implementation. The path shaping phase was run for 375 iterations. Results for the implementation were practically identical to simulated results and are omitted to avoid redundancy. Figure 8 shows snapshots of the implementation at different iteration values. Figure 8c shows the final informative path, which is approximately the same to that in Figure 1f from the simulations.

## VI. CONCLUSION

This paper uses a Voronoi-based coverage approach, building upon previous work in [1], to generate an adaptive controller for a robot to learn the rates at which the environment changes, through parameter estimation, and shape its path into an informative path, i.e. a locally optimal path for sensing dynamic regions in the environment. A Lyapunov-like proof showed that the controller will shape the path to a locally optimal configuration, and drive the estimated parameter error to zero, assuming the robot's trajectory is rich enough. An extension of the adaptive controller was designed and proven to drive the path to a locally optimal configuration that is beneficial to a persistent sensing task. The adaptive control law was simulated and implemented for a robot moving through 71 waypoints, generating results that support the theory, and producing stable persistent tasks based only on the robot's estimate of the environment.

## REFERENCES

- [1] M. Schwager, D. Rus, and J.-J. Slotine, "Decentralized, adaptive coverage control for networked robots," *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, 2009. [Online]. Available: <http://ijr.sagepub.com/content/28/3/357.abstract>
- [2] S. L. Smith, M. Schwager, and D. Rus, "Persistent robotic tasks: Monitoring and sweeping in changing environments," *Robotics, IEEE Transactions on*, vol. PP, no. 99, pp. 1–17, 2011.
- [3] A. Piazzì, C. Guarino Lo Bianco, and M. Romano, " $\eta^3$ -splines for the smooth path generation of wheeled mobile robots," *Robotics, IEEE Transactions on*, vol. 23, no. 5, pp. 1089–1095, oct. 2007.
- [4] K. Kyriakopoulos and G. Saridis, "Minimum jerk path generation," in *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, apr 1988, pp. 364–369 vol.1.
- [5] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, pp. 566–580, aug 1996.
- [6] C. Cunningham and R. Roberts, "An adaptive path planning algorithm for cooperating unmanned air vehicles," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 4, 2001, pp. 3981–3986 vol.4.
- [7] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, may 1994, pp. 3310–3317 vol.4.
- [8] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *J. Artif. Int. Res.*, vol. 34, no. 1, pp. 707–755, Apr. 2009. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1622716.1622735>
- [9] G. Hollinger and S. Singh, "Multi-robot coordination with periodic connectivity," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, may 2010, pp. 4457–4462.
- [10] D. Levine, B. Luders, and J. P. How, "Information-rich path planning with general constraints using rapidly-exploring random trees," in *AIAA Infotech@Aerospace Conference*, Atlanta, GA, April 2010, (AIAA-2010-3360).
- [11] R. Graham and J. Cortes, "Adaptive information collection by robotic sensor networks for spatial estimation," *Automatic Control, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2011.
- [12] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 2, pp. 243–255, april 2004.
- [13] S. Salapaka, A. Khalak, and M. Dahleh, "Constraints on locational optimization problems," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 2, dec. 2003, pp. 1741–1746 Vol.2.
- [14] Z. Drezner, "Facility location: A survey of applications and methods," in *Springer Series in Operations Research*. New York: Springer-Verlag, 1995.
- [15] L. C. A. Pimenta, M. Schwager, Q. Lindsey, V. Kumar, D. Rus, R. C. Mesquita, and G. A. S. Pereira, "Simultaneous coverage and tracking (scat) of moving targets with robot networks," in *Proceedings of the Eighth International Workshop on the Algorithmic Foundations of Robotics (WAFR 08)*, December 2008.
- [16] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Englewood Cliffs, NJ: Prentice-Hall, 1996.