Persistent Monitoring of Changing Environments using a Robot with Limited Range Sensing

Stephen L. Smith   Mac Schwager   Daniela Rus

***Abstract*— This paper presents controllers that enable a mobile robot to persistently monitor or sweep a changing environment. The changing environment is modeled as an accumulation function which grows in areas that are not within range of the robot, and decreases in areas that are within range of the robot. The robot must continually move through the environment to prevent the accumulation of any area from growing unbounded. We consider the case in which a pre-defined path is given for the robot, and we focus on controlling the robot's speed along the path. We characterize necessary and sufficient conditions on the speed controller of the robot for keeping the accumulation function bounded. We then search among the space of speed controllers that are parametrized by a finite set of basis functions. We develop a linear program to compute the optimal speed controller; that which minimizes the accumulation over the environment. Simulation results illustrate the performance of the controllers.**

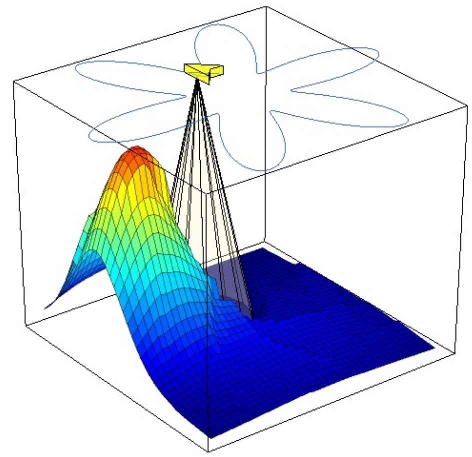


Fig. 1: A persistent monitoring task using a robot with limited range sensing. The robot follows the flower-shaped path and has a circular footprint. The surface shows the accumulation function, giving the uncertainty or quantity of material at each point. The robot controls its speed along the path so as to minimize the height of the surface.

I. Introduction

In this paper we treat the problem of controlling a robot to perpetually act in a changing environment, for example to clean an environment where material is constantly collecting, or to monitor an environment where uncertainty is continually growing. Each robot has only a small footprint over which to act (e.g. to sweep or to sense). The difficulty is in controlling the robot so that the amount of time it spends over a location is proportional to that location's rate of change. This scenario is distinct from most other sweeping and monitoring scenarios in the literature because the task cannot be completed: The robots must continually move to satisfy the objective. We consider the situation in which a robot is constrained to move on a fixed path, along which we must control its speed (see Figure 1 for an example).

We model the changing environment as a scalar valued function, which we call the *accumulation function*, The function captures the uncertainty at each point for a sensing task, or the quantity of material at each point for a cleaning task. The accumulation function grows at a constant rate at points not within range of a robot, and decreases at a constant rate at points within range of a robot. The rate of growth and decrease can be different at different points in the environment. This model is relevant to applications in which information over a large-scale, changing environment has to be kept up-to-date using robots with limited senor footprints. Examples include automatically patrolling a city or a building for suspicious activity, or continually vacuuming the floor of an office building.

Ideally, we would control the path and speed of the robot for persistent monitoring. However, obtaining performance guarantees for this problem is very difficult, as even the path planning component is NP-hard [1]. For this reason, we consider the robot path to be pre-planned, and we focus on controlling the speed of the robot along its path. Computing efficient paths has been addressed in our recent work [2]. Decoupling path planning from speed control is a well-established technique for dealing with complex trajectory planning problems [3]. Furthermore a particular path may be required in certain applications. An example is ocean sampling, where paths are commonly pre-specified by oceanographers [4].

Our main contributions are to formalize the idea of persistent tasks, and to develop speed controllers for a robot along a prescribed path. The path may be arbitrarily complex and have self intersections. Our approach to designing a speed controller is to formulate the controller as the solution to a linear program (LP) [5]. We consider controllers that can be parametrized by a finite number of basis functions. This enables the LP formulation, and also allows us to incorporate speed and safety constraints on the robots. The use of basis functions is a common method for function approximation [6], and is frequently used in areas such as

This material is based upon work supported in part by ONR-MURI Award N00014-09-1-1051

The authors are with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge MA 02139 (slsmith@mit.edu, schwager@mit.edu, rus@csail.mit.edu). M. Schwager is also affiliated with the GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104.

adaptive control [7], and machine learning [8]. We derive an LP which produces an optimal controller, in the sense that it maintains the accumulation function as low as possible. In [9] we extend these results to multiple robots and we provide all mathematical details that are omitted in this paper.

Our problem is related to sweep coverage [16], [17], lawn mowing and milling problems [1], and patrolling problems [18], [19], [20], [21] where robots with finite sensor footprints must sweep their sensor over every point in the environment. However, these works differ from persistent tasks in that they either consider only the finite time problem, or consider only the case where each point must be visited with equal frequency. In [22], [23], the authors define a different notion of persistent surveillance in which task completion takes much longer than the life of a robot, and then tackle the power management issues that arise.

Our work is also related to environmental monitoring. In this literature, authors often model the environment probabilistically, and estimate the state of that model using a Kalman-like filter. Then, robots are controlled so as to maximize a metric on the quality of the state estimate [10], [11], [12], [13], [14]. Unfortunately, planning optimal trajectories under these models typically requires the solution of an intractable dynamic program, even for a static environment. One must resort to myopic methods, such as gradient descent (as in [10], [15], [12], [11]), or solve the dynamic program approximately (as in [13], [14]). As a result, performance guarantees are difficult to obtain. The approach we take in this paper circumvents the question of estimation by formulating a new model of growing uncertainty in the environment. Under this model, we can solve the speed planning problem over *infinite time*, while maintaining *guaranteed* levels of uncertainty in a *time-changing* environment. Thus we have used a less sophisticated environment model in order to obtain stronger results on the control strategy.

In Section II we formalize the problem, and define persistent tasks. In Section III two LPs are formulated, the solutions of which give a stabilizing controller and an optimal controller, respectively. Finally, numerical simulations are presented in Section IV.

II. Speed Control for Persistent Tasks

Consider a compact environment $\mathcal{E} \subset \mathbb{R}^2$, and a finite set of points of interest $Q \subseteq \mathcal{E}$. The environment contains a closed curve $\gamma : [0,1] \to \mathbb{R}^2$, where $\gamma(0) = \gamma(1)$. The curve is parametrized by $\theta \in [0,1]$, and we assume without loss of generality that θ is the arc-length parametrization. The environment also contains a single robot whose motion is constrained along the path γ. The robot's position at a time t can be described by $\theta(t)$, its position along the curve γ. The robot is equipped with a finite sensor footprint $\mathcal{B}(\theta) \subset \mathcal{E}$ (for example, the footprint could be a disk of radius r centered at the robot's position). Our objective is to control the speed v of the robot along the curve. We assume that for each point θ on the curve, the maximum possible robot speed is $v_{\max}(\theta)$ and the minimum robot speed is $v_{\min}(\theta) > 0$. This allows us to express constraints on the robot speed at different points on the curve. For example, for safety considerations, the robot may be required to move more slowly in certain areas of the environment, or on high curved sections of the path. To summarize, the robot is described by the triple $\mathcal{R} := (\mathcal{B}, v_{\min}, v_{\max})$.

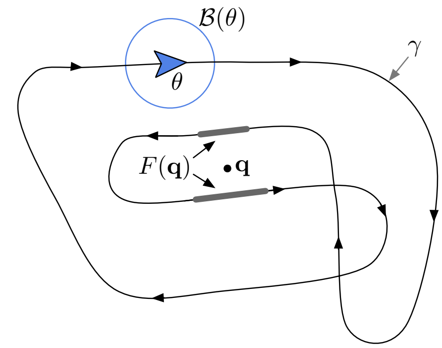


Fig. 2: An illustration of a curve γ followed by the robot. The robot is located at θ and has footprint $\mathcal{B}(\theta)$. The set $F(\mathbf{q})$ of robot positions θ for which the footprint covers q are shown as thick grey segments of the curve.

Defined on the points of interest Q is a time-varying field (or accumulation function) $Z : Q \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. We assume that at each point $\mathbf{q} \in Q$, the field Z increases (or is produced) at a constant rate $p(\mathbf{q})$. When the robot footprint is covering $\mathbf{q}$, it consumes Z at a constant rate $c(\mathbf{q})$. Thus, when a point $\mathbf{q}$ is covered, the net rate of decrease is $p(\mathbf{q}) - c(\mathbf{q})$. Additionally, we assume that Z is non-negative. Thus, Z evolves according to the following differential equation (with initial conditions $Z(\mathbf{q}, 0)$ and $\theta(0)$):

$$\dot{Z}(\mathbf{q}, t) = \begin{cases} p(\mathbf{q}), & \text{if } \mathbf{q} \notin \mathcal{B}\big(\theta(t)\big), \\ p(\mathbf{q}) - c(\mathbf{q}), & \text{if } \mathbf{q} \in \mathcal{B}\big(\theta(t)\big) \text{ and } Z(\mathbf{q}, t) > 0, \\ 0, & \text{if } \mathbf{q} \in \mathcal{B}\big(\theta(t)\big) \text{ and } Z(\mathbf{q}, t) = 0, \end{cases} \tag{1}$$

where for each $\mathbf{q} \in Q$, we have $c(\mathbf{q}) > p(\mathbf{q}) > 0$. We assume that the robot knows the model parameters $p(\mathbf{q})$ and $c(\mathbf{q})$. It is reasonable to assume knowledge of $c(\mathbf{q})$ since it pertains to the performance of the robot. As for the production rate, $p(\mathbf{q})$, this must be estimated from the physics of the environment, from a human expert (e.g. an oil mining engineer in the case of an oil well leak), or it can be measured in a preliminary survey of the environment. However, the accuracy of the model is not crucial, as we show that our method has some robustness to errors in $p(\mathbf{q})$.

Next, let us define a function which maps each point $\mathbf{q} \in Q$, to the curve positions θ for which $\mathbf{q}$ is covered by the robot footprint. To this end, we define

$$F(\mathbf{q}) := \{\theta \in [0,1] \mid \mathbf{q} \in \mathcal{B}(\theta)\}.$$

An illustration of the curve, the robot footprint, and the set $F(\mathbf{q})$ is shown in Figure 2.

In general, the commanded speed at a point θ may depend on the current position θ, the field Z, the initial conditions $\theta(0)$ and $Z(\mathbf{q}, 0)$, and time. Thus, defining the set of initial conditions as $\mathrm{IC} := (\theta(0), Z(\mathbf{q}, 0))$, a general controller has the form $v(\theta, Z, \mathrm{IC}, t)$.

A. Stability and Feasibility

In this section we prove basic properties about the stability of a persistent task under a given controller and the feasibility of the problem for any controller. We begin by formalizing our notion of a *persistent task*.

Definition II.1 (Persistent Tasks)**.** *A persistent task is a tuple* $(\mathcal{R}, \gamma, Q, p, c)$*, where* $\mathcal{R}$ *is the robot model,* γ *is the curve followed by the robot,* Q *is the set of points of interest, and* p *and* c *are the production and consumption rates of the field, respectively.*

As a first consideration, a suitable controller should keep the field bounded everywhere, independent of the initial conditions. This motivates the following definition of stability.

Definition II.2 (Stabilizing Controller)**.** *A speed controller stabilizes a persistent task if the field is always eventually bounded, independent of initial conditions. That is, if there exists a* $Z_{\max} < +\infty$ *such that for every* $\mathbf{q} \in Q$ *and initial condition* $Z(\mathbf{q}, 0)$ *and* $\theta(0)$*, we have*

$$\limsup_{t \to +\infty} Z(\mathbf{q}, t) \leq Z_{\max}.$$

Note that in this definition of stability, for every initial condition, the field eventually enters the interval $[0, Z_{\max}]$. There are some persistent tasks for which no controller is stabilizing. This motivates the notion of *feasibility*.

Definition II.3 (Feasible Persistent Task)**.** *A persistent task is feasible if there exists a stabilizing speed controller.*

As stated above, a speed controller in its most general form can be written as $v(\theta, Z, \mathrm{IC}, t)$. However, in the remainder of the paper we will focus on specific form of controller which we call *periodic position-feedback controllers*. In these controllers, the speed only depends on the robot's current position $\theta \in [0, 1]$:

$$v : [0, 1] \to \mathbb{R}_{>0}.$$

where each θ maps to a speed $v(\theta)$ satisfying the bounds $v_{\min}(\theta) \leq v(\theta) \leq v_{\max}(\theta)$. These controllers have the advantage that they do not require information on the current state of the field Z, only its model parameters $p(\mathbf{q})$ and $c(\mathbf{q})$. While it may seem restrictive to limit our controllers to this special form, the following result shows that it is not.

Proposition II.4 (Periodic Position-Feedback Controllers)**.** *If a persistent task can be stabilized by a general controller* $v(\theta, Z, \mathrm{IC}, t)$*, then it can be stabilized by a periodic position-feedback controller* $v(\theta)$*.*

The proof of Theorem II.4 is given in [9] and relies on the statement and proof of the upcoming result in Lemma II.5.

We will now investigate conditions for a controller to be stabilizing and for a persistent task to be feasible. Given a controller $\theta \mapsto v(\theta)$, we define two quantities: 1) the cycle time, or period, T, and 2) the coverage time per cycle $\tau(\mathbf{q})$. Since $v(\theta) > 0$ for all θ, the robot completes one full cycle of the closed curve in time

$$T := \int_0^1 \frac{1}{v(\theta)} d\theta. \tag{2}$$

During each cycle, the robot's footprint is covering the point $\mathbf{q}$ only when $\theta(t) \in F(\mathbf{q})$. Thus the point $\mathbf{q}$ is covered for

$$\tau(\mathbf{q}) := \int_{F(\mathbf{q})} \frac{1}{v(\theta)} d\theta, \tag{3}$$

time units during each complete cycle.

With these definitions we can give a necessary and sufficient condition for a controller to stabilize a persistent task.

Lemma II.5 (Stability condition)**.** *Given a persistent task, a controller* $v(\theta)$ *is stabilizing if and only if*

$$c(\mathbf{q}) \int_{F(\mathbf{q})} \frac{1}{v(\theta)} d\theta > p(\mathbf{q}) \int_0^1 \frac{1}{v(\theta)} d\theta \tag{4}$$

for every $\mathbf{q} \in Q$*. Applying the definitions in* (2) *and* (3)*, the condition can be expressed as* $\tau(\mathbf{q}) > p(\mathbf{q})/c(\mathbf{q})T$*.*

The lemma has a simple intuition. For stability, the consumption per cycle must exceed the production per cycle for each point $\mathbf{q} \in Q$. We now prove the result.

Proof. For a point $\mathbf{q} \in Q$, let us consider the change in the field from $Z(\mathbf{q}, t)$ to $Z(\mathbf{q}, t+T)$, where $t \geq 0$. Define an indicator function $\mathbf{I} : [0, 1] \times Q \to \{0, 1\}$ as $\mathbf{I}(\theta, \mathbf{q}) = 1$ for $\theta \in F(\mathbf{q})$ and 0 otherwise. Then, from (1) we have that

$$\dot{Z}(\mathbf{q}, t) \geq p(\mathbf{q}) - c(\mathbf{q})\mathbf{I}(\theta(t), \mathbf{q}),$$

for all values of $Z(\mathbf{q}, t)$, with equality if $Z(\mathbf{q}, t) > 0$. Integrating the above expression over $[t, t+T]$ we see that

$$\begin{aligned} Z(\mathbf{q}, t+T) - Z(\mathbf{q}, t) &\geq p(\mathbf{q})T - c(\mathbf{q}) \int_t^{t+T} \mathbf{I}(\theta(\tau), \mathbf{q}) d\tau \\ &= p(\mathbf{q})T - c(\mathbf{q})\tau(\mathbf{q}), \end{aligned} \tag{5}$$

where $\tau(\mathbf{q})$ is defined in (3). From (5) we see that a necessary condition for the field to be eventually bounded by some $Z_{\max}$ for all initial conditions $Z(\mathbf{q}, 0)$ is that $\tau(\mathbf{q}) > p(\mathbf{q})/c(\mathbf{q})T$ for all $\mathbf{q} \in Q$.

To see that the condition is also sufficient, suppose that $\tau(\mathbf{q}) > p(\mathbf{q})/c(\mathbf{q})T$. Then there exists $\epsilon > 0$ such that $p(\mathbf{q})T - c(\mathbf{q})\tau(\mathbf{q}) = -\epsilon$. If $Z(\mathbf{q}, t) > \big(c(\mathbf{q}) - p(\mathbf{q})\big)T$, then the field at the point $\mathbf{q} \in Q$ is strictly positive over the entire interval $[t, t+T]$, implying that $Z(\mathbf{q}, t+T) = Z(\mathbf{q}, t) - \epsilon$. Thus, from every initial condition, $Z(\mathbf{q}, t)$ moves below $\big(c(\mathbf{q}) - p(\mathbf{q})\big)T$. Additionally, note that for each $\bar{t}$ in the interval $[t, t+T]$, we trivially have that $Z(\mathbf{q}, \bar{t}) \leq Z(\mathbf{q}, t) + p(\mathbf{q})T$. Thus, we have that there exists a finite time $\bar{t}$ such that for all $t \geq \bar{t}$,

$$Z(\mathbf{q}, t) \leq \big(c(\mathbf{q}) - p(\mathbf{q})\big)T + p(\mathbf{q})T = c(\mathbf{q})T.$$

Since Q is compact, there exists a single $\epsilon > 0$ such that for every point $\mathbf{q} \in Q$ we have $\tau(\mathbf{q}) - p(\mathbf{q})/c(\mathbf{q})T > \epsilon$. Hence, letting $Z_{\max} = \max_{\mathbf{q} \in Q} c(\mathbf{q})T$, we see that Z is stable for all $\mathbf{q}$, completing the proof. □

In the following sections we will address two problems, determining a stabilizing controller, and determining a minimizing controller.

Problem II.6 (Persistent Task Metrics)**.** *Given a persistent task, determine a periodic position-feedback controller* v :

$[0,1] \to \mathbb{R}_{>0}$ *that satisfies the speed constraints (i.e.,* $v(\theta) \in [v_{\min}(\theta), v_{\max}(\theta)]$ *for all* $\theta \in [0,1]$*), and*

(i) *is stabilizing; or*

(ii) *minimizes the maximum steady-state field* $\mathcal{H}(v)$*:*

$$\mathcal{H}(v) := \max_{\mathbf{q}\in Q}\left(\limsup_{t\to+\infty} Z(\mathbf{q},t)\right).$$

In Section III we will show that by writing the speed controller in terms of a set of basis functions, problems (i) and (ii) can be solved using linear programs.

III. Speed Controllers: Stability and Optimality

We consider a finite number of points of interest $Q = \{\mathbf{q}_1, \dots, \mathbf{q}_m\}$. These m locations could be specific regions of interest, or they could be a discrete approximation of the continuous space obtained by, for example, laying a grid down on the environment. In the Section IV we show examples of both scenarios. Our two main results are given in Theorems III.1 and III.7, which show that a stabilizing controller, and a controller minimizing $\mathcal{H}(v)$, can each be found by solving linear programs.

To begin, it will be more convenient to talk about the reciprocal speed controller $v^{-1}(\theta) := 1/v(\theta)$, with its corresponding constraints $1/v_{\max}(\theta) \le v^{-1}(\theta) \le 1/v_{\min}(\theta)$. Now, our approach is to consider a finite set of basis functions $\{\beta_1(\theta), \dots, \beta_n(\theta)\}$. Example basis functions include (a finite subset of) the Fourier basis or Gaussian basis [6]. In what follows we will use rectangular functions as the basis:

$$\beta_j(\theta) = \begin{cases} 1, & \text{if } \theta \in [(j-1)/n, j/n) \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

for each $j \in \{1, \dots, n\}$.

Then consider reciprocal speed controllers of the form

$$v^{-1}(\theta) = \sum_{j=1}^{n} \alpha_j \beta_j(\theta), \tag{7}$$

where $\alpha_1, \dots, \alpha_n$ are free parameters that we will use to optimize the speed controller. A rich class of functions can be represented as a finite linear combination of basis functions, though not all functions can be represented this way. Limiting our speed controller to a linear parametrization allows us to find an optimal controller within that class, while preserving enough generality to give complex solutions that would be difficult to find in an ad hoc manner. In the following subsection we will consider the problem of synthesizing a stabilizing controller.

A. Synthesis of a Stabilizing Controller

In this section we will show that a stabilizing speed controller of the form (7) can be found through the solution of a linear program. This result is summarized in Theorem III.1. To begin, let us consider reciprocal speed controllers in the form of (7). Then for $\mathbf{q}_i \in Q$, the stability condition in Lemma II.5 becomes

$$\sum_{j=1}^{n} \alpha_j \int_{F(\mathbf{q}_i)} \beta_j(\theta) d\theta > \frac{p(\mathbf{q}_i)}{c(\mathbf{q}_i)} \sum_{j=1}^{n} \alpha_j \int_0^1 \beta_j(\theta) d\theta$$

Rearranging, we get $\sum_{j=1}^{n} \alpha_j K(\mathbf{q}_i, \beta_j) > 0$, where we have defined

$$K(\mathbf{q}_i, \beta_j) := \int_{F(\mathbf{q}_i)} \beta_j(\theta) d\theta - \frac{p(\mathbf{q}_i)}{c(\mathbf{q}_i)} \int_0^1 \beta_j(\theta) d\theta. \tag{8}$$

Finally, to satisfy the speed constraints we have that

$$\frac{1}{v_{\max}(\theta)} \le \sum_{j=1}^{n} \alpha_j \beta_j(\theta) \le \frac{1}{v_{\min}(\theta)} \tag{9}$$

For the rectangular basis in (6), the speed constraints become $1/v_{\max}(j) \le \alpha_j \le 1/v_{\min}(j)$, where

$$v_{\max}(j) = \inf_{\theta \in [(j-1)/n, j/n)} v_{\max}(\theta), \text{ and}$$
$$v_{\min}(j) = \sup_{\theta \in [(j-1)/n, j/n)} v_{\min}(\theta).$$

Thus, we obtain the following result.

Theorem III.1 (Existence of a Stabilizing Controller)**.** *A persistent task is stabilizable by a speed controller of the form* (7) *if and only if the following linear program is feasible:*

$$\begin{aligned} &\textit{minimize} && 0 \\ &\textit{subject to} && \sum_{j=1}^{n} \alpha_j K(\mathbf{q}_i, \beta_j) > 0 && \forall\, i \in \{1, \dots, m\} \\ &&& \frac{1}{v_{\max}(j)} \le \alpha_j \le \frac{1}{v_{\min}(j)}, && \forall\, j \in \{1, \dots, n\}, \end{aligned}$$

where $K(\mathbf{q}_i, \beta_j)$ *is defined in* (8)*, and* $\alpha_1, \dots, \alpha_n$ *are the optimization variables.*

Hence, we can solve for a stabilizing controller using a simple linear program. The program has n variables (one for each basis function coefficient), and $2n + m$ constraints (two for each basis function coefficient, and one for each point of interest in Q). One can easily solve linear programs with thousands of variables and constraints [5]. Thus, the problem of computing a stabilizing controller can be solved for finely discretized environments with thousands of basis functions. Note that in the above lemma, we are only checking feasibility, and thus the cost function in the optimization is arbitrary. For simplicity we write the cost as 0.

Remark III.2 (Robustness via Stability Margin)**.** *In Theorem III.1 the cost is set to* 0 *to highlight the feasibility constraints. However, in practice, an important consideration is robustness of the controller to errors in the model of the field evolution. Robustness of this type can be achieved by slightly altering the above optimization to maximize the* stability margin $\min_{\mathbf{q}_i \in Q} \sum_{j=1}^{n} \alpha_j K(\mathbf{q}_i, \beta_j)$*. We can do so by rephrasing the optimization as*

$$\begin{aligned} &\textit{maximize} && B \\ &\textit{subject to} && \sum_{j=1}^{n} \alpha_j K(\mathbf{q}_i, \beta_j) \ge B && \forall\, i \in \{1, \dots, m\} \\ &&& \frac{1}{v_{\max}(j)} \le \alpha_j \le \frac{1}{v_{\min}(j)}, && \forall\, j \in \{1, \dots, n\}. \end{aligned}$$

This controller is robust to errors in estimating the field production rate. If the robot's estimate of the production rate at a field point $\mathbf{q}_i \in Q$ is $\bar{p}(\mathbf{q}_i)$, and the true value is $p(\mathbf{q}_i) \leq \bar{p}(\mathbf{q}_i) + \epsilon$, then it can be easily verified (see [9]) that the field is stable provided that $\epsilon < B \cdot c(\mathbf{q}_i) / \big(\sum_{j=1}^{n} \alpha_j \int_0^1 \beta_j(\theta) d\theta\big)$, for each $\mathbf{q}_i \in Q$. □

B. Synthesis of an Optimal Controller

In this section we look at Problem II.6 (ii), which is to minimize the maximum value attained by the field over the finite region of interest Q. That is, for a given persistent task, our goal is to minimize the following cost function,

$$\mathcal{H}(v) = \max_{\mathbf{q} \in Q} \left(\limsup_{t \to +\infty} Z(\mathbf{q}, t) \right) \tag{10}$$

over all possible speed controllers v. At times we will refer to the maximum steady-state value for a point $\mathbf{q}$ using a speed controller v as

$$\mathcal{H}(\mathbf{q}, v) := \limsup_{t \to +\infty} Z(\mathbf{q}, t)$$

Our main result of this section, Theorem III.7, is that $\mathcal{H}(v)$ can be minimized through a linear program. However, we must establish some intermediate results. First we show that if v is a stabilizing controller, then for every initial condition there exists a finite time t^* such that $Z(\mathbf{q}, t) \leq \mathcal{H}(v)$ for all $t \geq t^*$. The proof of this result is contained in [9].

Proposition III.3 (Steady-State Field)**.** *Consider a feasible persistent task and a stabilizing speed controller. Then, there is a steady-state field*

$$\bar{Z} : Q \times [0, 1] \to \mathbb{R}_{\geq 0},$$

satisfying the following statements for each $\mathbf{q} \in Q$:

(i) *for every set of initial conditions $\theta(0)$ and $Z(\mathbf{q}, 0)$, there exists a time $t^* \geq 0$ such that $Z(\mathbf{q}, t) = \bar{Z}\big(\mathbf{q}, \theta(t)\big)$, for all $t \geq t^*$.*

(ii) *there exists $\theta \in [0, 1]$ such that $\bar{Z}(\mathbf{q}, \theta) = 0$.*

From the above result we see that from every initial condition, the field converges in finite time to a steady-state $\bar{Z}(\mathbf{q}, \theta)$. In steady-state, the field $Z(\mathbf{q}, t)$ at time t depends only on $\theta(t)$ (and is independent of $Z(\mathbf{q}, 0)$). Each time the robot is located at θ, the field is given by $\bar{Z}(\mathbf{q}, \theta)$. Moreover, the result tells us that in steady-state there is always a robot position at which the field is reduced to zero. Proposition III.3 relies on the following lemma. Recall that the cycle-time for a speed controller v is $T := \int_0^1 1/v(\theta) d\theta$.

Lemma III.4 (Field Reduced to Zero)**.** *Consider a feasible persistent task and a stabilizing speed controller. For every $\mathbf{q} \in Q$ and every set of initial conditions $Z(\mathbf{q}, 0)$ and $\theta(0)$, there exists a time $t^* > T$ such that*

$$Z(\mathbf{q}, t^* + aT) = 0, \tag{11}$$

for all non-negative integers a.

Proof. Consider any $\mathbf{q} \in Q$, and initial conditions $Z(\mathbf{q}, 0)$ and $\theta(0)$, and suppose by way of contradiction that the speed controller is stable but $Z(\mathbf{q}, t) > 0$ for all $t > T$. From Lemma II.5, if the persistent task is stable, then $c(\mathbf{q})\tau(\mathbf{q}) > p(\mathbf{q})T$ for all $\mathbf{q}$. Thus, there exists $\epsilon > 0$ such that $c(\mathbf{q})\tau(\mathbf{q}) - p(\mathbf{q})T > \epsilon$ for all $\mathbf{q} \in Q$. From the proof of Lemma II.5, we have that

$$Z(\mathbf{q}, t + T) - Z(\mathbf{q}, t) = p(\mathbf{q})T - c(\mathbf{q})\tau(\mathbf{q}) = -\epsilon.$$

Therefore, given $Z(\mathbf{q}, 0)$, we have that $Z(\mathbf{q}, t^*) = 0$ for some finite $t^* > T$, a contradiction.

Next we will verify that if $Z(\mathbf{q}, t^*) = 0$ for some $t^* > T$, then $Z(\mathbf{q}, t^* + T) = 0$. To see this, note that the differential equation (1) is piecewise constant. Given a speed controller $v(\theta)$, the differential equation is time-invariant, and admits unique solutions.

Based on this, consider two initial conditions for (1),

$$Z_1(\mathbf{q}, 0) := Z(\mathbf{q}, t^* - T) \geq 0, \quad \theta_1(0) := \theta(t^* - T) = \theta(t^*),$$

and

$$Z_2(\mathbf{q}, 0) := Z(\mathbf{q}, t^*) = 0, \quad \theta_2(0) := \theta(t^*).$$

Since (1) is time-invariant, we have that $Z_1(\mathbf{q}, T) = Z(\mathbf{q}, t^*) = 0$, and $Z_2(\mathbf{q}, T) = Z(\mathbf{q}, t^* + T)$. In addition, by uniqueness of solutions, we also know that $Z_1(\mathbf{q}, 0) \geq Z_2(\mathbf{q}, 0)$ implies that $Z_1(\mathbf{q}, T) \geq Z_2(\mathbf{q}, T)$. Thus, we have that $Z(\mathbf{q}, t^*) = 0 \geq Z(\mathbf{q}, t^* + T)$, proving the desired result. □

With this lemma we can prove Proposition III.3.

Proof of Proposition III.3. In Lemma III.4 we have shown that for every set of initial conditions $Z(\mathbf{q}, 0)$, $\theta(0)$, there exists at time $t^* > T$ such that $Z(\mathbf{q}, t^* + aT) = 0$ for all non-negative integers a. Since T is the cycle-time for the robot, we also know that $\theta(t^* + aT) = \theta(t^*)$ for all a. Since (1) yields unique solutions, (11) uniquely defines $Z(\mathbf{q}, t)$ for all $t \geq t^*$, with

$$Z(\mathbf{q}, t + T) = Z(\mathbf{q}, t) \quad \text{for all } t \geq t^*.$$

Hence, we can define the steady-state profile $\bar{Z}(\mathbf{q}, \theta)$ as

$$\bar{Z}\big(\mathbf{q}, \theta(t)\big) := Z(\mathbf{q}, t) \quad \text{for all } t \in [t^*, t^* + T).$$

Finally, we need to verify that $\bar{Z}(\mathbf{q}, \theta)$ is independent of initial conditions. To see this, suppose by way of contradiction that we have two sets of initial conditions $\theta_1(0)$, $Z_1(\mathbf{q}, 0)$, and $\theta_2(0)$, $Z_2(\mathbf{q}, 0)$ which yield different steady-state fields $\bar{Z}_1(\mathbf{q}, \theta)$ and $\bar{Z}_2(\mathbf{q}, \theta)$. Now, if there exists a θ such that $\bar{Z}_1(\mathbf{q}, \theta) > \bar{Z}_2(\mathbf{q}, \theta)$, then by uniqueness of solutions $\bar{Z}_1(\mathbf{q}, \theta) \geq \bar{Z}_2(\mathbf{q}, \theta)$ for all θ. To see this, suppose by way of contradiction that there exist times $t_1 > 0$ and $t_2 > t_1$ such that $\bar{Z}_1(\mathbf{q}, \theta(t_1)) > \bar{Z}_2(\mathbf{q}, \theta(t_1))$ and $\bar{Z}_1(\mathbf{q}, \theta(t_2)) < \bar{Z}_2(\mathbf{q}, \theta(t_2))$. Then, by the continuity of Z, there exists a time $t_3 \in (t_1, t_2)$ such that $\bar{Z}_1(\mathbf{q}, \theta(t_3)) = \bar{Z}_2(\mathbf{q}, \theta(t_3))$. But, by the uniqueness of solutions Z, this implies that $\bar{Z}_1(\mathbf{q}, \theta(t)) = \bar{Z}_2(\mathbf{q}, \theta(t))$ for all $t \geq t_3$, a contradiction.

Thus, we have $\bar{Z}_1(\mathbf{q}, \theta) \geq \bar{Z}_2(\mathbf{q}, \theta)$ for all θ. From Lemma III.4, there exists a $\bar{\theta}$ for which $\bar{Z}_1(\mathbf{q}, \bar{\theta}) = 0$ implying that $\bar{Z}_2(\mathbf{q}, \bar{\theta}) = 0$. However, the value of Z_1 and

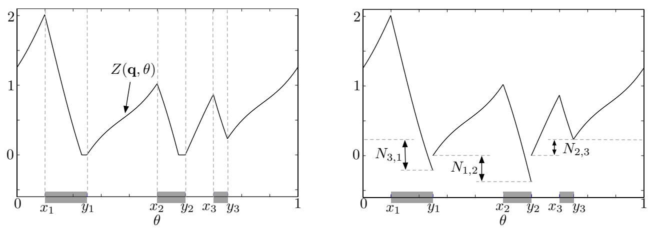


Fig. 3: The steady-state field $\bar{Z}(\mathbf{q},\theta)$ is shown on the left, with the production rate $p(\mathbf{q}) = 3$ and consumption rate $c(\mathbf{q}) = 8.5$. The set $F(\mathbf{q})$ consists of three intervals which are shaded on the θ-axis. The steady-state profile is increasing outside of $F(\mathbf{q})$ and decreasing inside $F(\mathbf{q})$. On the right, the maximum reduction from y values is shown, denoted by $N_{1,2}$, $N_{2,3}$, and $N_{3,1}$.

$\bar{Z}_2$ at $\bar{\theta}$ uniquely defines $\bar{Z}_1$ and $\bar{Z}_2$ for all θ, implying that $\bar{Z}_1(\mathbf{q},\theta) = \bar{Z}_2(\mathbf{q},\theta)$, a contradiction. □

From Proposition III.3 we have shown the existence of a steady-state field $\bar{Z}(\mathbf{q},\theta)$ that is independent of initial conditions $Z(\mathbf{q},0)$ and $\theta(0)$.

Now, consider a point $\mathbf{q} \in Q$ and a stabilizing speed controller $v(\theta)$, and let us solve for its steady-state field $\bar{Z}(\mathbf{q},\theta)$. To begin, let us write $F(\mathbf{q})$ (the set of θ values for which the point $\mathbf{q}$ is covered by the footprint) as a union of disjoint intervals

$$F(\mathbf{q}) = [x_1, y_1] \cup [x_2, y_2] \cup \cdots \cup [x_\ell, y_\ell], \tag{12}$$

where ℓ is a positive integer, and $y_k > x_k > y_{k-1}$ for each $k \in \{1,\ldots,\ell\}$.[1] Thus, on the intervals $[x_k, y_k]$ the point $\mathbf{q}$ is covered by the robot footprint, and on the intervals $[y_k, x_{k+1}]$, the point $\mathbf{q}$ in uncovered. As an example, in Figure 2, the set $F(\mathbf{q})$ consists of two intervals, and thus $\ell = 2$. An example of a speed controller and an example of a set $F(\mathbf{q})$ are shown in Figure 3.

From differential equation (1) we can write

$$\bar{Z}(\mathbf{q}, x_k) = \bar{Z}(\mathbf{q}, y_{k-1}) + p(\mathbf{q}) \int_{y_{k-1}}^{x_k} \frac{d\theta}{v(\theta)} \tag{13}$$

$$\bar{Z}(\mathbf{q}, y_k) = \left(\bar{Z}(\mathbf{q}, x_k) + \big(p(\mathbf{q}) - c(\mathbf{q})\big) \int_{x_k}^{y_k} \frac{d\theta}{v(\theta)} \right)^+, \tag{14}$$

where for $z \in \mathbb{R}$, we define $(z)^+ := \max\{z, 0\}$. Combining equations (13) and (14) we see that

$$\bar{Z}(\mathbf{q}, y_k) = \Bigg(\bar{Z}(\mathbf{q}, y_{k-1}) + p(\mathbf{q}) \int_{y_{k-1}}^{y_k} \frac{d\theta}{v(\theta)} - c(\mathbf{q}) \int_{x_k}^{y_k} \frac{d\theta}{v(\theta)} \Bigg)^+. \tag{15}$$

[1]Note that the number of intervals ℓ, and the points $x_1,\ldots,x_\ell$ and $y_1,\ldots,y_\ell$ are a function of $\mathbf{q}$. However, for simplicity of notation, we will omit writing the explicit dependence.

For each $b \in \{1,\ldots,\ell\}$, let us define[2]

$$N_{k-b,k}(\mathbf{q}) := p(\mathbf{q}) \int_{y_{k-b}}^{y_k} \frac{d\theta}{v(\theta)} - c(\mathbf{q}) \sum_{w=0}^{b-1} \int_{x_{k-w}}^{y_{k-w}} \frac{d\theta}{v(\theta)}. \tag{16}$$

We can write $\bar{Z}(\mathbf{q}, y_k) = \big(\bar{Z}(\mathbf{q}, y_{k-1}) + N_{k-1,k}(\mathbf{q})\big)^+$, and thus from (15) we have

$$\bar{Z}(\mathbf{q}, y_k) \geq \bar{Z}(\mathbf{q}, y_{k-b}) + N_{k-b,k}(\mathbf{q}). \tag{17}$$

Moreover,

$$\bar{Z}(\mathbf{q}, y_k) = \bar{Z}(\mathbf{q}, y_{k-b}) + N_{k-b,k}(\mathbf{q}), \quad \text{if } Z(\mathbf{q}, y_{k-j}) > 0 \text{ for all } j \in \{1,\ldots,b-1\}. \tag{18}$$

Thus, we see that the quantity $N_{k-b,k}(\mathbf{q})$ gives the maximum reduction in the field between $\theta = y_{k-b}$ and $\theta = y_k$. An example for $b = 1$ is shown in Figure 3. With these definitions, we can characterize the steady-state field at the points y_k. The proof of this result is given in [9].

Lemma III.5 (Steady-State Field at Points y_k)**.** *Given a feasible persistent task and a stabilizing speed controller, consider a point $\mathbf{q} \in Q$ and the set $F(\mathbf{q}) = \cup_{k=1}^{\ell} [x_k, y_k]$. Then, for each $k \in \{1,\ldots,\ell\}$ we have*

$$\bar{Z}(\mathbf{q}, y_k) = \max_{b \in \{0,\ldots,\ell-1\}} N_{k-b,k}(\mathbf{q}),$$

where $N_{k-b,k}(\mathbf{q})$ is defined in (16) *and $N_{k,k}(\mathbf{q}) := 0$.*

The above lemma gives the value of the field in steady-state at each end point y_k. However, the maximum steady-state value is attained at an x_k rather than a y_k. For example, in Figure 3, the maximum is attained at the point x_1. However, the value at x_k can be easily computed from the value at y_{k-1} using (13):

$$\bar{Z}(\mathbf{q}, x_{k+1}) = \max_{b \in \{0,\ldots,\ell-1\}} N_{k-b,k}(\mathbf{q}) + p(\mathbf{q}) \int_{y_k}^{x_{k+1}} \frac{d\theta}{v(\theta)}.$$

From this we obtain the following result.

Lemma III.6 (Steady-State Upper Bound)**.** *Given a stabilizing speed controller v, the maximum steady-state field at $\mathbf{q} \in Q$ (defined in* (10)*) satisfies*

$$\mathcal{H}(\mathbf{q}, v) = \max_{\substack{k \in \{1,\ldots,\ell\} \\ b \in \{0,\ldots,\ell-1\}}} X_{k,b}(\mathbf{q}),$$

where

$$X_{k,b}(\mathbf{q}) = p(\mathbf{q}) \int_{y_{k-b}}^{x_{k+1}} \frac{d\theta}{v(\theta)} - c(\mathbf{q}) \sum_{w=0}^{b-1} \int_{x_{k-w}}^{y_{k-w}} \frac{d\theta}{v(\theta)},$$

and $F(\mathbf{q}) = \cup_{k=1}^{\ell} [x_k, y_k]$ with $y_k > x_k > y_{k-1}$ for each k.

The above lemma provides a closed form expression (albeit quite complex) for the largest steady-state value of the field. Thus, consider speed controllers of the form

$$v^{-1}(\theta) = \sum_{j=1}^{n} \alpha_j \beta_j(\theta),$$

[2]In this definition, and in what follows, addition and subtraction in the indices is performed modulo ℓ. Therefore, if $k = 1$, then $N_{k-1,k} = N_{\ell,1}$.

where $\beta_1, \dots, \beta_n$ are basis functions (e.g., the rectangular basis). For a finite field $Q = \{\mathbf{q}_1, \dots, \mathbf{q}_m\}$, the terms $N_{k-b,k}(\mathbf{q}_i)$ can be written as $X_{k,b}(\mathbf{q}_i) = \sum_{j=1}^{n} \alpha_j X_{k,b}(\mathbf{q}_i, \beta_j)$, where

$$X_{k,b}(\mathbf{q}_i, \beta_j) := p(\mathbf{q}) \int_{y_{k-b}}^{y_k} \beta_j(\theta) d\theta - c(\mathbf{q}) \sum_{w=0}^{b-1} \int_{x_{k-w}}^{y_{k-w}} \beta_j(\theta) d\theta. \quad (19)$$

With these definitions we can define a linear program for minimizing the maximum of the steady-state field. We will write $\ell(\mathbf{q})$ to denote the number of disjoint intervals on the curve γ over which the point $\mathbf{q}$ is covered, as defined in (12).

Theorem III.7 (Minimizing the Steady-State Field)**.** *Given a feasible persistent task, the solution to the following linear program yields a speed controller v of the form* (7) *that minimizes the maximum value of the steady-state field $\mathcal{H}(v)$.*

$$\begin{aligned} &\textit{minimize} \;\; V \\ &\textit{subject to} \;\; \sum_{j=1}^{n} \alpha_j X_{k,b}(\mathbf{q}_i, \beta_j) \le V && \forall\, i \in \{1, \dots, m\}, \\ & && k \in \{1, \dots, \ell(\mathbf{q}_i)\}, \\ & && b \in \{0, \dots, \ell(\mathbf{q}_i) - 1\}, \\ &\sum_{j=1}^{n} \alpha_j K(\mathbf{q}_i, \beta_j) > 0 && \forall\, i \in \{1, \dots, m\}, \\ &\frac{1}{v_{\max}(j)} \le \alpha_j \le \frac{1}{v_{\min}(j)}, && \forall\, j \in \{1, \dots, n\}. \end{aligned}$$

The optimization variables are α_j and V and the quantities $X_{k,b}(\mathbf{q}_i, \beta_j)$ and $K(\mathbf{q}_i, \beta_j)$ are defined in (19) *and* (8)*.*

From the above theorem, we can minimize the maximum value of the field using a linear program. This optimization has $n + 1$ variables (n basis function coefficients, and one upper bound B). The number of constraints is $m \sum_{i=1}^{m} \ell(\mathbf{q}_i)^2 + m + 2n$. In practice, $\ell(\mathbf{q}_i)$ is small compared to n and m, and is independent of n and m. Thus, for most instances, the linear program has $O(2n + m)$ constraints.

IV. Simulations

The optimization framework was implemented in MATLAB®, and the linear programs were solved using the freely available SeDuMi (Self-Dual-Minimization) toolbox. To give the readers some feel for the efficiency of the approach, we report the time to solve each optimization on a laptop computer with a 2.66 GHz dual core processor and 4 GB of RAM.

Figure 4 shows a simulation for one robot performing a persistent monitoring task with $|Q| = 10$ points. The speed controller resulting from the optimization in Section III-B is shown in Figure 5a. For the optimization, the minimum speed was set to $v_{\min} = 0.1$ and the maximum speed to $v_{\max} = 1$, for all θ. A total of 150 rectangular basis functions were used. The optimization was solved in less than $1/10$ of a second. The field $Z(\mathbf{q}, t)$ for the red (shaded) point in Figure 4, is shown as a function of time in Figure 5b. One can see that the field converges in finite time to a periodic cycle. In addition, the field goes to zero during each cycle. The periodic cycle is the steady-state as characterized in Section III-B.

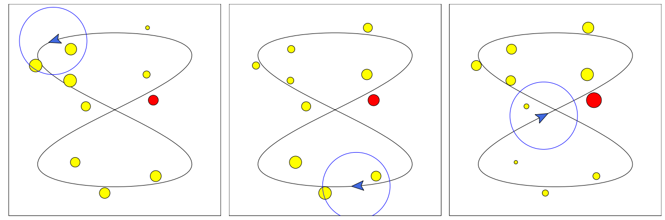
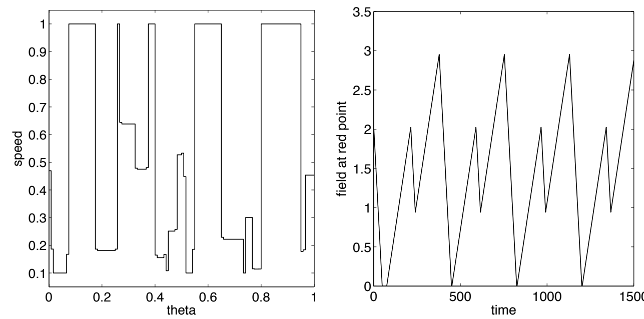


Fig. 4: An example with a field consisting of 10 points. The field $Z(\mathbf{q}, t)$ at each point is indicated by the area of the disk centered at the point. The vehicle footprint is a disk. The time sequence of the three snapshots goes from left to right. The vehicle is moving counter-clockwise around the top half of the figure eight and clockwise around the bottom half. The time evolution of the red field point is shown in Figure 5.



(a) Optimal speed controller. (b) Field $Z(\mathbf{q}, t)$ at red (dark) point.

Fig. 5: The optimal speed controller corresponding to the curve and field shown in Figure 4. The minimum speed is $v_{\min} = 0.1$ and the maximum speed is $v_{\max} = 1$. The field $Z(\mathbf{q}, t)$ is shown for the red (dark) point in Figure 4. It converges in finite time to a steady state.

In Figure 1 a simulation is shown for the case when the entire continuous environment must be monitored. In this case the environment was discretized into a 40×40 grid. The speed constraints were given by $v_{\min} = 0.1$ and $v_{\max} = 1$. A total of 300 rectangular basis functions were used. The consumption function (or rate of reduction) was given by $c(\mathbf{q}) = 1$ for all $\mathbf{q} \in Q$. The production function (or rate of increase) $p(\mathbf{q})$ was given by a bi-modal Gaussian. The optimization was solved in approximately 10 seconds.

In Figure 6, the robot footprint is a square that is oriented with the robot's current heading. In addition, for this simulation the maximum speed is a function of θ, and is given by $v_{\max}(\theta) = 1/\big(1.5 - (\theta - .5)^2\big)$. The production function $p(\mathbf{q})$ for the simulation in Figure 6 is shown in Figure 7, along with the optimal speed controller. From Figure 7b, one can see that the maximum speed stays below the bound $v_{\max}(\theta)$, and that the robot moves at maximum speed for much of the trajectory.

V. Conclusions

In this paper we proposed the notion of persistent tasks for robots. A persistent task is one in which a robot has to continually move to keep an accumulation function as low as possible everywhere in the environment. We specifically considered the case in which robots are confined to pre-specified, closed paths, along which their speed must be

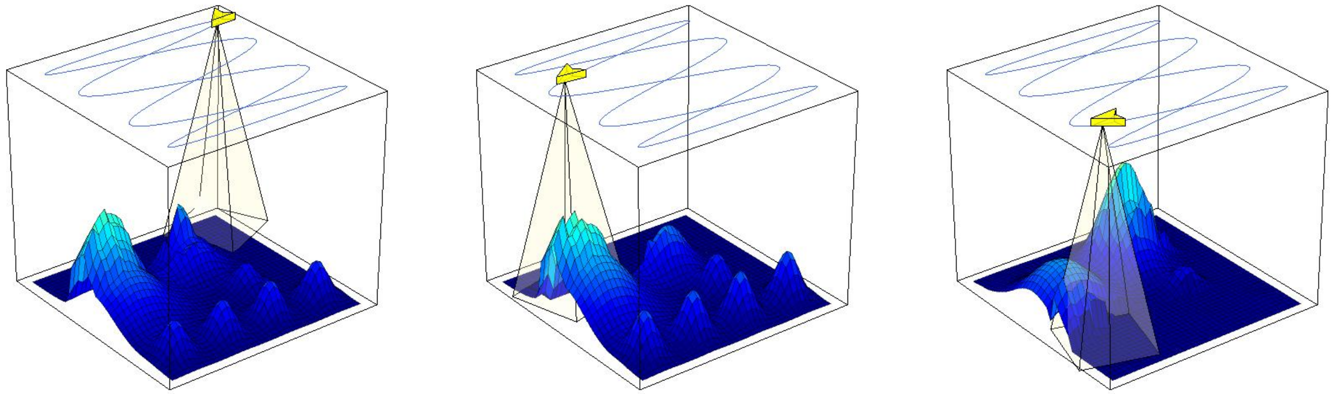
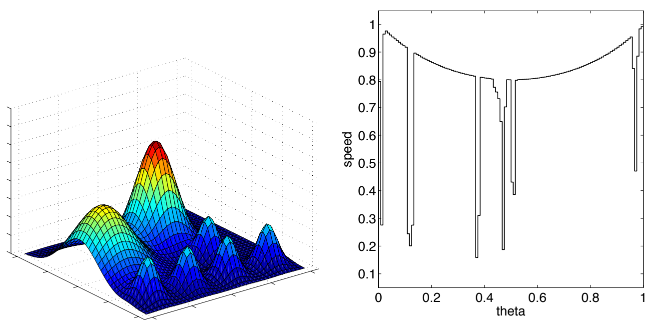


Fig. 6: In this example the robot has a square footprint that is oriented with its heading. The robot's trajectory is given by zig-zagging trajectory drawn on the top of the cube. The time sequence of the three snapshots goes from left to right.



(a) The production function $p(\mathbf{q})$. (b) The optimal speed controller.

Fig. 7: The production function $p(\mathbf{q})$ is shown on the left for the simulation in Figure 6. The optimal speed controller for Figure 6 is shown on the right. The minimum speed is $v_{\min} = 0.1$ and the maximum speed is given by $v_{\max}(\theta) = 1/\big(1.5 - (\theta - .5)^2\big)$.

controlled. We formulated an LP whose solution gives an optimal speed controller. One interesting direction in which to expand this work is to consider planning paths and full trajectories for robots to carry out persistent tasks. Another direction of extension is to have a robot solve the LP for its controller on-line. This would be useful if, for example the production rate is not know before hand, but can be sensed over the sensor footprint of the robot.

REFERENCES

[1] E. M. Arkin, S. P. Fekete, and J. S. B. Mitchell, "Approximation algorithms for lawn mowing and milling," *Computational Geom.: Theory and App.*, vol. 17, no. 1-2, pp. 25–50, 2000.

[2] S. L. Smith and D. Rus, "Multi-robot monitoring in dynamic environments with guaranteed currency of observations," in *Proc CDC*, Atlanta, GA, Dec. 2010, pp. 514–521.

[3] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *International Journal of Robotics Research*, vol. 5, no. 3, pp. 72–89, 1986.

[4] R. N. Smith, M. Schwager, S. L. Smith, B. H. Jones, D. Rus, and G. S. Sukhatme, "Persistent ocean monitoring with underwater gliders: Adapting spatiotemporal sampling resolution," *Journal of Field Robotics*, 2010, Submitted.

[5] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed. Addison Wesley, 1984.

[6] E. W. Cheney, *Introduction to Approximation Theory*, 2nd ed. AMS Chelsea Publishing, 2000.

[7] R. Sanner and J. Slotine, "Gaussian networks for direct adaptive control," *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 837–863, 1992.

[8] T. Poggio and S. Smale, "The mathematics of learning: Dealing with data," *Notices of the American Mathematical Society*, vol. 50, no. 5, pp. 537–544, 2003.

[9] S. L. Smith, M. Schwager, and D. Rus, "Persistent robotic tasks: Monitoring and sweeping in changing environments," *IEEE Trans Robotics*, Jul. 2010, submitted. Available at `http://arxiv.org/abs/1102.0603`.

[10] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 710–724, June 2008.

[11] J. Cortés, "Distributed Kriged Kalman filter for spatial estimation," *IEEE Trans Automatic Ctrl*, vol. 54, no. 12, pp. 2816–2827, 2009.

[12] F. Zhang and N. E. Leonard, "Cooperative filters and control for cooperative exploration," *IEEE Trans Automatic Ctrl*, vol. 55, no. 3, pp. 650–663, 2010.

[13] J. L. Ny and G. J. Pappas, "On trajectory optimization for active sensing in gaussian process models," in *IEEE Conf. on Decision and Control and Chinese Control Conference*, Shanghai, China, December 2009, pp. 6282–6292.

[14] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, "Information based adaptive robotic exploration," in *Proc IROS*, 2002, pp. 540–545.

[15] P. Yang, R. A. Freeman, and K. M. Lynch, "Mulit-agent coordination by decentralized estimation and control," *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 248–2496, December 2008.

[16] H. Choset, "Coverage for robotics – A survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1-4, pp. 113–126, 2001.

[17] I. Rekleitis, V. Lee-Shue, A. P. New, and H. Choset, "Limited communication multi-robot team based coverage," in *Proc ICRA*, New Orleans, LA, Apr. 2004, pp. 3462–3468.

[18] Y. Chevaleyre, "Theoretical analysis of the multi-agent patrolling problem," in *IEEE/WIC/ACM Int. Conf. Intelligent Agent Technology*, Beijing, China, Sep. 2004, pp. 302–308.

[19] Y. Elmaliach, N. Agmon, and G. A. Kaminka, "Multi-robot area patrol under frequency constraints," in *Proc ICRA*, Roma, Italy, Apr. 2007, pp. 385–390.

[20] P. F. Hokayem, D. Stipanović, and M. W. Spong, "On persistent coverage control," in *Proc CDC*, New Orleans, LA, Dec. 2007, pp. 6130–6135.

[21] N. Nigram and I. Kroo, "Persistent surveillance using multiple unmannded air vehicles," in *IEEE Aerospace Conf.*, Big Sky, MT, May 2008, pp. 1–14.

[22] B. Bethke, J. P. How, and J. Vian, "Group health management of UAV teams with applications to persistent surveillance," in *Proc ACC*, Seattle, WA, Jun. 2008, pp. 3145–3150.

[23] B. Bethke, J. Redding, J. P. How, M. A. Vavrina, and J. Vian, "Agent capability in persistent mission planning using approximate dynamic programming," in *Proc ACC*, Baltimore, MD, Jun. 2010, pp. 1623–1628.