

# Robust Source-Seeking Hybrid Controllers for Autonomous Vehicles\*

Christopher G. Mayhew, Ricardo G. Sanfelice, and Andrew R. Teel †

**Abstract**— We consider the problem of steering an autonomous vehicle to locate a radiation source utilizing measurements of the radiation intensity only. We propose a control algorithm that locates the source through a sequence of line minimizations of the radiation intensity. We implement in a hybrid controller, with sample-and-hold and logic variables, a discretized version of the algorithm suitable for steering a point-mass vehicle. The algorithm confers global convergence and practical stability properties to the closed-loop hybrid system. We discuss these properties and characterize the region of convergence for the vehicle. Convergence and stability results are supplemented with simulations.

## I. INTRODUCTION

In this paper, we study the problem of steering a vehicle to the source of a radiation-like signal with minimal information, namely, only measurements of the radiation intensity. The radiation-like signal could be thought of as a potential function in the vehicle’s environment, taking a maximum or minimum value at the “source.” This setting is typical in the control of autonomous vehicles without relative position sensors where only measurements of a signal like light, sound, or temperature are available.

This problem has been addressed before in the literature, often in the context of autonomous underwater vehicles. Burian et al. [2] examined methods for conducting a gradient descent with a single vehicle, while Bachmayer and Leonard [1] proposed methods for steering a network of vehicles to the same effect. More recently, Silva et al. proposed a strategy which executes the simplex optimization method with a team of two vehicles in [15], [4]. In a related effort, Farrell et al. [5], [10] proposed a statistical method for locating the source of a chemical plume in the ocean. Lastly, Zhang et al. [18], [17] proposed an *extremum seeking* control strategy for the task.

In this paper, we take a hybrid systems approach to this problem and propose a control strategy for a single autonomous vehicle inspired by a minimization algorithm first introduced by C.S. Smith [16]. This algorithm is implemented with a hybrid controller that coordinates vehicle steering with the optimization algorithm to locate the source. The hybrid implementation introduces a “sample and hold” behavior to the nominal algorithm, conferring global convergence and stability properties and a margin of robustness to measurement noise to the closed-loop system. We characterize the region

of convergence of the closed-loop trajectories, and provide simulations of the algorithm in various situations.

## II. PROBLEM DESCRIPTION

We consider the problem of localizing a radiation source in a real-world environment with an autonomous vehicle. We assume the existence of a function  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  with a minimum at  $x^*$  describing the radiation intensity. Only measurements of this function are available for the control task as the function  $f$  and the position of the vehicle are unknown. This scenario is depicted in Figure 1.

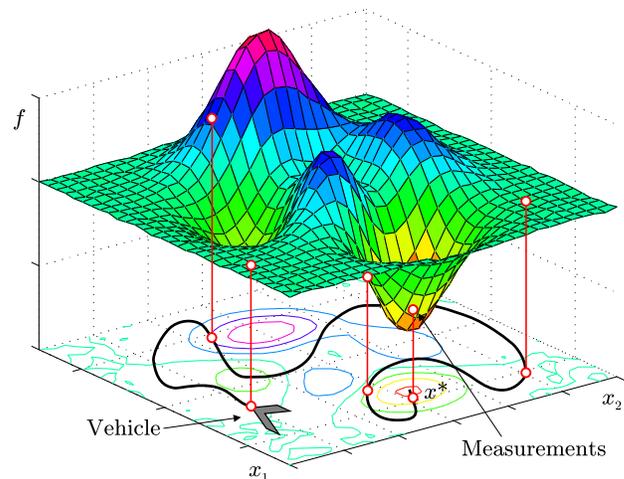


Fig. 1. The main task of the controller is to steer the vehicle to the destination,  $x^*$ , which minimizes a function  $f$  of the vehicle position, with only measurements of  $f$ , perhaps at several locations, denoted by  $\circ$ .

We consider a simplified kinematic point-mass model of the vehicle, allowing us to concentrate our discussion on the control algorithm itself. That is, we assume that the dynamics of the vehicle are given by  $\dot{x} = u$  with state  $x \in \mathbb{R}^2$  denoting its position and input  $u \in \mathbb{R}^2$ .

In this type of real-world scenario, measurement noise is typical. Hence, it is desired that the control strategy confers a margin of robustness to the closed-loop system.

## III. MAIN IDEA: A CONJUGATE DIRECTION ALGORITHM

To accomplish the convergence task motivated in Section II, we propose a control strategy which utilizes existing optimization techniques and adapts them to our setting. The optimization algorithm we use can be summarized in two stages:

- 1) Explore the environment by conducting a series of minimizations along a line, defined by a direction  $v$ .
- 2) Update the direction.

\*Research partially supported by the Army Research Office under Grant no. DAAD19-03-1-0144, the National Science Foundation under Grant no. CCR-0311084 and Grant no. ECS-0324679, and by the Air Force Office of Scientific Research under Grant no. FA9550-06-1-0134.

†{mayhew,rsanfelice,teel}@ece.ucsb.edu, Center for Control Engineering and Computation, Electrical and Computer Engineering Department, University of California, Santa Barbara, CA 93106-9560.

This general idea is common among several different optimization algorithms, but the differences appear mainly in how a new direction is chosen. The algorithm which we propose uses the notion of conjugate vectors to choose a new direction, without gradient information. These techniques appeared in the numerical optimization literature in the early 1960's and include the work by Rosenbrock [13], Smith [16], and Powell [11], among others. While these optimization techniques have seen widespread use, to the best of our knowledge, exploiting these ideas in problems like the one described in Section II is novel.

Let  $V^n = \{v \in \mathbb{R}^n \mid \|v\|_2 = 1\}$ . Given a function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , a point  $x^0 \in \mathbb{R}^n$ , and a non-zero vector  $v \in V^n$ , a *line minimization* from  $x^0$  in the direction of  $v$  consists of finding the value  $\lambda \in \mathbb{R}$  such that  $x^0 + \lambda v$  is a minimum of  $f$  along  $v$  from  $x^0$ . We denote the operation of performing a line minimization from a point  $x^0 \in \mathbb{R}^n$  along a direction  $v \in \mathbb{R}^n$  on a function  $f$  and returning the value  $\lambda$  as a mapping  $\mu_f : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$ . Our algorithm is as follows.

*Recursive Smith-Powell (RSP) Algorithm*

Given  $x^0 \in \mathbb{R}^2$  and  $v \in V^2$ ,

- Step 1)** Calculate  $x^1 = x^0 + \mu_f(x^0, v)v$
- Step 2)** Calculate<sup>1</sup>  $x^2 = x^1 + \mu_f(x^1, v_\perp)v_\perp$
- Step 3)** Calculate  $x^3 = x^2 + \mu_f(x^2, v)v$
- Step 4)** If  $\mu_f(x^1, v_\perp) \neq 0$  and  $\mu_f(x^2, v) \neq 0$ , Update the direction  $v$  with the vector  $v^+ := (x^3 - x^1)/\|x^3 - x^1\|$ . Otherwise, update the direction  $v$  with a vector linearly independent from the current one, so that a large set of directions are eventually explored.
- Step 5)** Set  $x^0 = x^3$  and go to **Step 1**).

We now state convergence properties of *RSP*. Let

$$\mathcal{Q}^n = \{f : \mathbb{R}^n \rightarrow \mathbb{R} \mid f = \sigma \circ f_P\}, \quad (1)$$

where  $f_P = f^* + (x - x^*)^\top P(x - x^*)$  is a quadratic function with  $P = P^\top > 0$ , and  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$  is strictly increasing and continuous.

**Theorem 3.1:** *Given  $f \in \mathcal{Q}^2$ , for any initial point  $x^0 \in \mathbb{R}^2$  and for any initial direction  $v \in V^2$ , the RSP algorithm converges to  $x^*$ , the global minimum of  $f$  in no more than four line minimizations.*

Figure 2 illustrates the steps of this algorithm for a quadratic function, where the level sets are ellipses. It also suggests the result stated in Theorem 3.1.

Similar results have been stated in the literature for  $n$ -dimensional quadratic functions. As it is shown in [6, Section 4.2], see also [16], the number of line minimizations needed for convergence to the minimum with an  $n$ -dimensional quadratic function is  $n(n + 1)/2$ . Note that the result in Theorem 3.1 is for  $n = 2$  and it states that one more line minimization is required; see Figure 2. This is due to the fact that in order to conduct two parallel line minimizations in the direction of  $v$ , we use another line minimization in the direction of  $v_\perp$  to move the vehicle from  $x^1$  to  $x^2$ . However,

<sup>1</sup> $v_\perp \in V^2$  denotes the vector orthogonal to  $v$ .

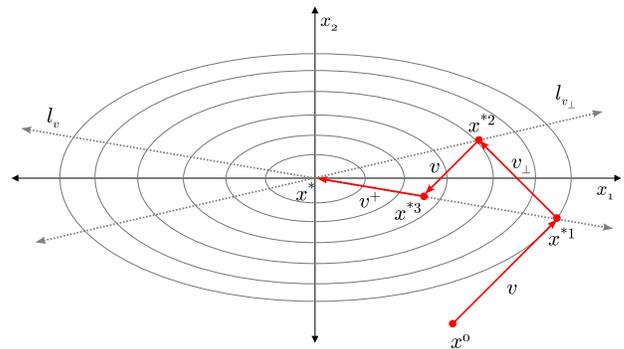


Fig. 2. A typical trajectory generated by the *RSP* algorithm on a quadratic-like function. The initial condition is given by  $x^0 \in \mathbb{R}^2$  and the initial direction by  $v$ . Convergence to the destination is achieved after four line minimizations. The ellipses (gray) denote level sets of the quadratic function.

any other strategy applied at  $x^1$  that moves the algorithm in a linearly independent direction to  $v$  would suffice.

The core of *RSP* is the calculation of the direction  $v^+$  in **Step 4**). In the case of  $f \in \mathcal{Q}^2$ , this step computes the so-called *conjugate direction* to  $v$ . Then, Theorem 3.1 follows directly from results relating conjugate directions and line minimizations in [11].

**Definition 3.2:** (conjugate directions) Let  $P = P^\top > 0$ . Nonzero vectors  $u_1, u_2 \in \mathbb{R}^n$  are said to be *conjugate* (with respect to  $P$ ) if  $u_1^\top P u_2 = 0$ .

Vaguely, two directions are conjugate (with respect to  $f \in \mathcal{Q}^2$  defined by the  $P$ ) if the minimization in one of the directions does not “spoil” the minimization in the other direction.

**Remark 3.3:** When computing a line minimization of  $f$  from  $x^0$  in the direction of  $v$  (as in **Step 1**)), it can be shown that there exists a line passing through  $x^*$  where  $f$  is minimized in the direction of  $v$ . The equation of this line, parameterized by  $r \in \mathbb{R}$ , is given by

$$l_v(r) = rQ(x^* - x^0) + x^*, \quad (2)$$

where  $Q = \frac{vv^\top P}{v^\top P v} - I$ . Note that  $Q^2 = -Q$ , so  $Q$  is a projection matrix. By some abuse of notation, we let  $l_v = \{x \in \mathbb{R}^2 \mid x = l_v(r), r \in \mathbb{R}\}$ . In the 2-dimensional case, it can be shown that  $Q$  has rank one, uniquely defining  $l_v$ . Then, by conducting two parallel line minimizations in the direction of  $v$ , one can find two points on the line  $l_v$  and calculate the direction to  $x^*$ . This idea is illustrated in Figure 2, where  $x^1$  and  $x^3$  are two points on the line  $l_v$ , and  $v^+ = x^3 - x^1$  is a vector pointing towards the minimizer from  $x^3$ . This new direction, which passes through the minimizer, and the direction  $v$  are conjugate.

#### IV. ALGORITHM IMPLEMENTATION FOR VEHICLES

Implementing the *Recursive Smith-Powell Algorithm* algorithm on a vehicle requires a controller capable of coordinating vehicle steering with the optimization algorithm. In this section, we show that such a synergy is made possible with hybrid control.

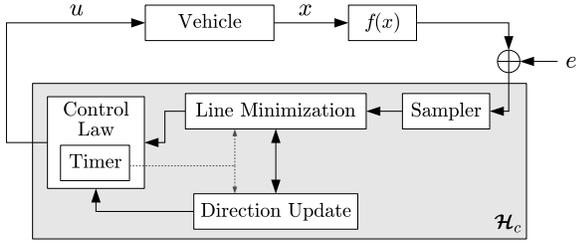


Fig. 3. Block diagram of the proposed hybrid controller. The controller can be thought to have several interacting components which ultimately steer the vehicle to the minimum of  $f$ .

Our hybrid controller  $\mathcal{H}_c$  is depicted in Figure 3. We describe the control structure with interconnected components:

- 1) Sampler: takes measurements of  $f$ ;
- 2) Line Minimization: organizes the logic for conducting a line minimization;
- 3) Direction Update: calculates the next search direction by using the *RSP* algorithm;
- 4) Control Law & Timer: steers the vehicle between measurements, keeps track of time for open-loop control law, triggers jumps in the Line Minimization and Direction Update.

Before we present the implementation of  $\mathcal{H}_c$ , we discuss a procedure for conducting a discretized version of a line minimization with vehicles using a sample-and-hold steering methodology.

#### A. Procedure for Conducting a Line Minimization

In the literature of numerical optimization, various methods exist for conducting a line minimization. In general, the basic idea is to compute a *bracket* (see [12], [6]).

*Definition 4.1:* Given a function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $x^0 \in \mathbb{R}^n$ , and  $v \in V^n$ , a *bracket* is defined as an interval  $(\lambda_1, \lambda_3) \subset \mathbb{R}$  such that  $\exists \lambda_2 \in (\lambda_1, \lambda_3)$  with the property that  $f(x^0 + \lambda_2 v) < f(x^0 + \lambda_1 v)$  and  $f(x^0 + \lambda_2 v) < f(x^0 + \lambda_3 v)$ .

With the concept of a bracket, a line minimization consists of finding a bracket and estimating the minimum within the bracket. A method for carrying out this task is illustrated in Figure 4. Here, given an initial point  $x^0$  and direction  $v$ , the algorithm takes steps of  $d \in \mathbb{R}$  in the direction  $v$  from  $x^0$  (blocks 1a & 1b) and evaluates the change in  $f$  over each step (blocks 2a & 2b). A bracket is located when the current step produces an increase in  $f$  after the previous step produced a decrease.

While there are many ways of answering the question, “Did  $f$  decrease,” (in blocks 2a & 2b) and many ways of estimating the minimum within a bracket (block 4), we propose a *discretized* approach, which, as we will see shortly, is applicable to vehicles.

#### B. Discretized Line Minimization Vehicle Implementation

For implementation on a digital controller, we assume that only samples of measurements of  $f$  are available. To accommodate the line minimization algorithm to such a scenario, we propose a sample-and-hold implementation of the

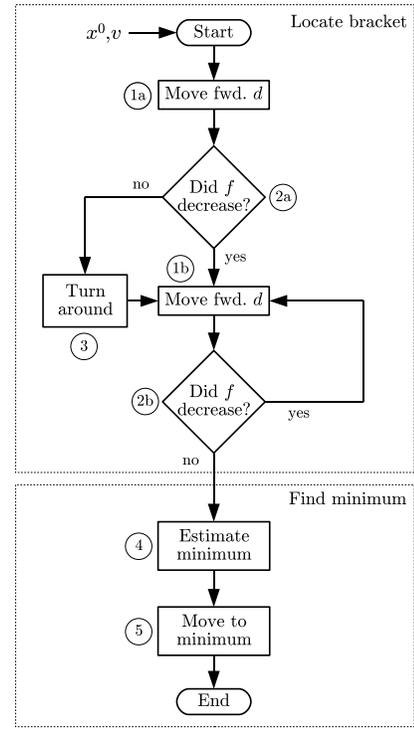


Fig. 4. A discretized approach to locating a bracket when conducting a line minimization. The change in  $f$  is evaluated over an interval of length  $d$ . A decrease means that the minimum lies ahead. A bracket is found when there is an increase in  $f$  directly following a decrease (forming a valley).

line minimization which we refer to as a *discretized line minimization*. To execute a discretized line minimization with an autonomous vehicle, the vehicle must be steered between measurements along the line defined by  $x^0$  and  $v$ . Since we are not assuming the availability of state measurements, open-loop control will be used for steering. Following Figure 4, there are three cases when open-loop control is needed during a line minimization:

- 1) (Block 1a & 1b) Driving the vehicle forward to the next measurement;
- 2) (Block 3) Turning the vehicle around to the previous measurement when the opposite direction must be explored;
- 3) (Block 5) When a bracket is found, turning the vehicle to the estimated minimum, ready to proceed in the next direction (according to the algorithm).

Figure 5 illustrates this entire process. Here, the vehicle begins at  $x^0$  (with  $\lambda = 0$ ). The vehicle samples  $f$  at its initial position, then moves to the next measurement and compares a new sample of  $f$  with the previous sample. It first detects an increase in  $f$  and then turns around to explore the opposite direction. A bracket is located at the next increase in  $f$ .

#### C. Hybrid Control Implementation

As shown in Figure 3, our hybrid controller  $\mathcal{H}_c$  has one input, the value of  $f$  obtained from a sensor through the sampler, and one control output to steer the vehicle.

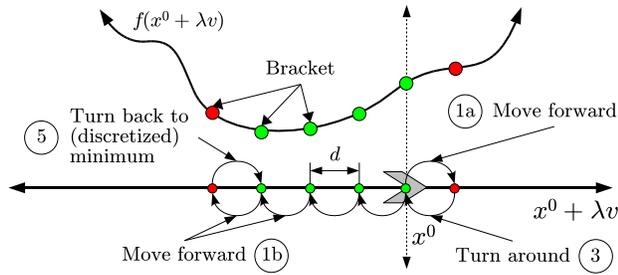


Fig. 5. A discretized line minimization carried out by a vehicle. Open-loop control is used to drive the vehicle between measurements until a bracket containing a local minimum is located. The vehicle is driven to the estimated (discretized) minimum.

We denote the state of the controller as  $x_c$ , which has several components. We describe these states, explain to which module in Figure 3 they belong, and describe their continuous and discrete evolution in the hybrid controller,  $\mathcal{H}_c$ .

### 1) Control Law & Timer

- $\tau, \tau^* \in \mathbb{R}$ : a timer and timer limit;

### 2) Line Minimization

- $\lambda \in \mathbb{R}$ : distance traveled by the line minimization;
- $z \in \mathbb{R}$ : keeps the previous measurement;
- $p \in \{-1, 1\}$ : determines sign of exploration;
- $q \in \{0, 1, 2\}$ : defines the state of the line minimization;
  - $q = 0$  if the line minimization is in its first step;
  - $q = 1$  if the vehicle is going towards the minimum;
  - $q = 2$  if the line minimization is completed;
- $m \in \{0, 1\}$ ;

### 3) RSP Direction Update

- $\alpha \in \mathbb{R}^2$ : stores the last vector traveled;
- $k \in \{0, 1, 2\}$ : defines the state of RSP;
- $v \in V^2$ : the current direction of exploration.

We combine the logic of the RSP algorithm in Section III with the discretized line minimization logic discussed in Section IV-B in  $\mathcal{H}_c$ . The modeling, notation, and concept of solution used for  $\mathcal{H}_c$  and the resulting hybrid closed-loop system follow the framework for hybrid systems in [7], [8] where solutions are given on *hybrid time domains* and the dynamics of a hybrid system with state  $\xi$  are given by a flow map  $F$ , a flow set  $C$ , a jump map  $G$ , and a jump set  $D$ . The function  $F$  governs continuous evolution of the state when  $\xi \in C$  and  $G$  governs discrete jumps of the state when  $\xi \in D$ . In this framework, a solution  $\xi$  to a hybrid system on a hybrid time domain  $\text{dom } \xi$  is parameterized by a continuous variable  $t$  which keeps track of the continuous dynamics and a discrete variable  $j$  which keeps track of the discrete dynamics. Then,  $\xi(t, j)$  is the value of the solution at time  $(t, j) \in \text{dom } \xi$ .

The output of the controller for the point-mass vehicle is given by  $u = \gamma p v$  where  $\gamma > 0$  is the velocity constant. With this control law, the vehicle is steered between measurements.

The only controller state that changes during flows is the timer  $\tau$ . Then, the continuous dynamics of the controller are

given by

$$\dot{\tau} = 1 \quad x_c \in C := \{x_c \mid \tau \leq \tau^*\}. \quad (3)$$

Jumps of  $\mathcal{H}_c$  are triggered when the timer expires, that is,  $\tau \geq \tau^*$ . Then,  $D := \{x_c \mid \tau \geq \tau^*\}$ . At every jump, the timer state and constant are updated by

$$\tau^+ = 0, \quad \tau^{*+} = T(x_c, f(x)), \quad (4)$$

where  $T$  is a continuous function that computes the total time needed by the open-loop control law to steer the vehicle to the next measurement. For the point-mass vehicle, this function is given by the constant  $d/\gamma$  so that the vehicle is steered a distance  $d$  in between jumps. Additionally, at every jump, the logic involved in the Line Minimization and Direction Update modules is executed. We embed this logic into an outer-semicontinuous<sup>2</sup> set-valued map  $G_c$  that updates the states  $\lambda, z, p, q, m, \alpha, k, v$ , and is constructed from the maps  $g_i(x_c, f(x))$  and the sets  $D_i, i = 1, 2, \dots, 6$ . These are given below. We omit the update law of the state variables that remain constant at jumps.

#### 1) Continue a positive line search:

$$D_1 = \{z \geq f(x) \text{ and } p = 1 \text{ and } q \in \{0, 1\} \text{ and } m = 0\}$$

$$g_1: z^+ = f(x), q^+ = 1, \lambda^+ = \lambda + p$$

#### 2) Correct overshoot:

$$D_2 = \{z \leq f(x) \text{ and } q \in \{0, 1\} \text{ and } m = 0\}$$

$$g_2: p^+ = -p, q^+ = q + 1, m^+ = 1$$

#### 3) Start negative line search:

$$D_3 = \{m = 1 \text{ and } p = -1 \text{ and } q = 1\}$$

$$g_3: z^+ = f(x), m^+ = 0, \lambda^+ = 0$$

#### 4) Continue a negative line search:

$$D_4 = \{z \geq f(x) \text{ and } p = -1 \text{ and } q = 1 \text{ and } m = 0\}$$

$$g_4: z^+ = f(x), \lambda^+ = \lambda + p$$

#### 5) Update direction and start positive line search:

$$D_5 = \{\tau \geq \tau^* \text{ and } q = 2\}$$

$$q^+ = 0, p^+ = 1, \lambda^+ = 0, m^+ = 0,$$

$$\alpha^+ = \lambda v$$

$$k^+ = (k + 1) \bmod 3$$

$$g_5: v^+ = \begin{cases} R_{\pi/2} v & k = 0 \\ R_{-\pi/2} v & k = 1 \\ \Phi(\alpha, \lambda, v) & k = 2 \end{cases}$$

$$z^+ = f(x)$$

$$\text{where } \Phi(\alpha, \lambda, v) = \begin{cases} \lambda v + \alpha & \alpha \neq 0 \text{ and } \lambda \neq 0 \\ \Pi(v) & \text{otherwise} \end{cases}$$

#### 6) Handle other conditions:

$$D_6: \text{any state } x_c \notin \bigcup_{i=1, \dots, 5} D_i$$

$$g_6: q^+ = 2$$

In 1),  $g_1$  describes how the state evolves when a beneficial measurement is found during a line minimization. Simply, it records the new measurement and increments  $\lambda$ . In 2), ‘‘Correct overshoot’’ is addressing the case where the vehicle either needs to ‘‘Turn around’’ to the previous measurement and start a search in the negative direction or ‘‘Move to minimum,’’

<sup>2</sup>A set-valued mapping  $G$  defined on an open set  $O$  is *outer semicontinuous* if for each sequence  $x_i \in O$  converging to a point  $x \in O$  and each sequence  $y_i \in G(x_i)$  converging to a point  $y$ , it holds that  $y \in G(x)$ .

in the sense of Figure 4. The way  $D_i$  and  $g_i$  are designed causes the state to jump according to the directed graph in Figure 6 (after the timer expires).

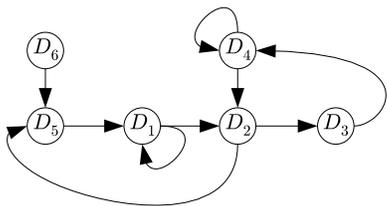


Fig. 6. A directed graph showing the flow of  $x_c$ , the state of  $\mathcal{H}_c$ , through the jump set partitions  $D_i$ .

In 5),  $\Pi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  is a function which forces the *RSP* algorithm to explore a large set of direction when conjugate information is not available. For a given function  $\Gamma : X \rightarrow X$ , we denote  $\Gamma^n$  as the composition of  $\Gamma$   $n$  times, that is,

$$\Gamma^n = \underbrace{\Gamma \circ \dots \circ \Gamma}_{n \text{ times}}.$$

Then,  $\Pi$  is such that  $\forall u^0 \in V^2$ ,  $\{u \in V^2 \mid u = \Pi^m(u^0), m \in \mathbb{Z}_{>0}\}^3$  is dense in  $V^2$ . One could easily implement this function as a rotation matrix that rotates the vector through a rational angle (in radians). The simulations in the sequel use this implementation.

## V. CONVERGENCE AND STABILITY RESULTS

We denote the closed-loop hybrid system shown in Figure 3 as  $\mathcal{H}_{cl}$ . The following global convergence result holds for the trajectories of  $\mathcal{H}_{cl}$ .

**Theorem 5.1: (global practical convergence)** For every  $f \in \mathcal{Q}^2$ , initial condition of  $\mathcal{H}_{cl}$ , and  $\epsilon > 0$  there exists  $d^*$  such that for all  $d \in (0, d^*]$ , the  $x$  trajectories of  $\mathcal{H}_{cl}$  converge to  $\epsilon\mathbb{B}(x^*)$ <sup>4</sup>.

The theorem states that despite our discretized implementation of the line minimization, the hybrid controller  $\mathcal{H}_c$  steers the vehicle to a region around  $x^*$ . This result can be shown using invariance principles for hybrid systems in [14]. In fact, the hybrid controller  $\mathcal{H}_c$  is constructed such that the variable  $z$  is monotonically non-increasing along solutions. With the Lyapunov function  $V(x, x_c) = z^2$  and results in [14], given  $d > 0$ , it can be shown that  $z$  converges to

$$Z = \{z \in \mathbb{R} \mid z = f(x) \leq f(x \pm dv), x \in \mathbb{R}^2, v \in V^2\}. \quad (5)$$

The set to which the  $x$  trajectories of  $\mathcal{H}_{cl}$  converge is given by

$$X = \{x \in \mathbb{R}^2 \mid x \in d\mathbb{B}(x_0), f(x_0) \in Z\}. \quad (6)$$

Additionally, the closed-loop system has the following stability property.

**Theorem 5.2: (practical stability)** For every  $f \in \mathcal{Q}^2$ ,  $v^0 \in V^2$ , and  $\epsilon > 0$ , there exist  $\delta > 0$  and  $d^* > 0$  such that for

<sup>3</sup> $\mathbb{Z}_{>0}$  denotes the set of integers greater than zero, i.e.,  $\{1, 2, \dots\}$

<sup>4</sup>By  $\delta\mathbb{B}(x)$ , we denote the closed  $\delta$ -ball centered at  $x$  in  $\mathbb{R}^2$ .

all  $d \in (0, d^*]$ , initial conditions of  $\mathcal{H}_c$  with  $v(0, 0) = v^0$ , and initial vehicle position  $x^0 \in \delta\mathbb{B}(x^*)$ , every  $x$  trajectory of  $\mathcal{H}_{cl}$  starting from  $x^0$  satisfies  $x(t, j) \in \epsilon\mathbb{B}(x^*)$  for all  $(t, j) \in \text{dom } x$ .

These results for the closed loop system are closely related to the convergence of *RSP* under discretization. In fact, when *RSP* is implemented with discretized line minimizations, it converges to the set

$$\begin{aligned} R &= \{x \in \mathbb{R}^2 \mid f(x) \in Z\} \\ &= \{x \in \mathbb{R}^2 \mid f(x + dv) \geq f(x), v \in V^2\} \\ &= \bigcap_{v \in V^2} L_v, \end{aligned} \quad (7)$$

where, for a given  $v \in V^2$ ,

$$L_v = \{x \in \mathbb{R}^2 \mid f(x \pm dv) \geq f(x)\}, \quad (8)$$

which we refer to as the *turning tube* for the direction  $v$ . During a discretized line minimization in the direction  $v$ , the *RSP* algorithm drives the vehicle into the turning tube for the direction  $v$  and turns to proceed in the next direction. An execution of the *RSP* algorithm is depicted with turning lines and turning tubes using discretized line minimizations in Figure 7.

When  $f \in \mathcal{Q}^2$ , we can describe  $L_v$  precisely as

$$L_v = \left\{x \in \mathbb{R}^2 \mid x = l_v(r) + \delta v, r, \delta \in \mathbb{R}, |\delta| \leq \frac{d}{2}\right\}, \quad (9)$$

where  $l_v(r)$  is the *turning line* for the direction  $v$ , defined in (2). Because each turning tube has a width of  $d$  along  $v$  about  $x^*$ , we have that

$$R \subset \frac{d}{2}\mathbb{B}(x^*). \quad (10)$$

This follows from the fact that  $R$  is formed from the intersection of all turning tubes as in (7).

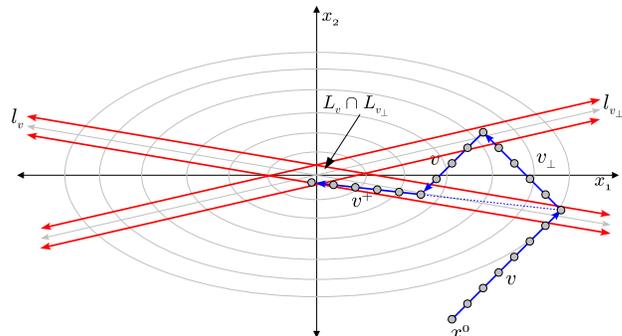


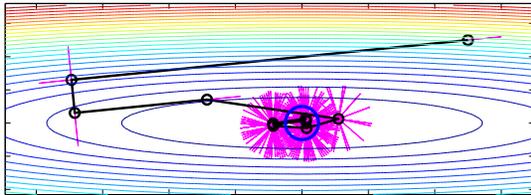
Fig. 7. Turning lines (gray) and turning tubes (outlined in red) are plotted for the direction  $v$  and its orthogonal. The region  $R$  is partially formed by the intersection of  $L_v$  with  $L_{v_\perp}$ . The ‘inflation’ of the turning lines by  $d/2$  define the turning tubes. A typical algorithm path (solid blue) is shown when discretization (gray circles) is introduced.

**Remark 5.3:** From the regularity properties of the data of  $\mathcal{H}_{cl}$ , the closed-loop system is nominally robust to external perturbations [9]. Moreover, the sample-and-hold implementation of the control algorithm confers to the closed-loop system a margin of robustness to measurement noise. Furthermore,

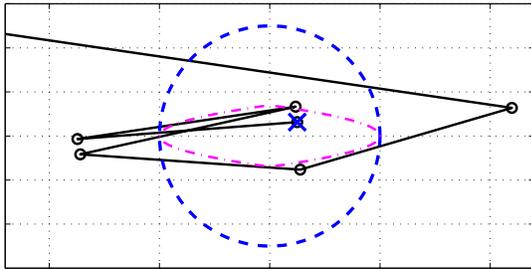
the convergence properties of  $\mathcal{H}_{cl}$  in Theorem 5.1 imply robustness to slowly-varying changes in  $f$ ; see [3].

## VI. SIMULATION RESULTS

Figure 8 illustrates the algorithm approaching  $x^*$  in four (discretized) line minimizations on a quadratic function and eventually reaching  $R$  through exploration of many directions. The function  $\Pi$  is implemented by rotating the current direction by a rational angle close to  $\pi/6$ .



(a) Convergence on a quadratic function.



(b) Zoomed view of  $R \subset (d/2)\mathbb{B}(x^*)$ .

Fig. 8. Convergence to  $x^*$  on a quadratic function  $f(x) = 5x_1^2 + 0.1x_2^2$  with  $d = 1$ . (a) A vehicle is steered to a neighborhood of  $x^* = 0$  with RSP. The vehicle trajectory (magenta) “overshoots” the minimum along discretized line minimizations. The algorithm path (black -o-) is shown on top of the vehicle path and converges to  $R$ , a subset of  $(d/2)\mathbb{B}(x^*)$  (solid blue). Level sets of  $f$  are also illustrated. (b) A zoomed view of  $R$  (magenta ---), a subset of  $(d/2)\mathbb{B}(x^*)$  (blue dashed) with algorithm path (black -o-). Vehicle trajectories and level sets are not shown for clarity.

Figure 9 depicts the algorithm steering the point-mass to a neighborhood of the minimum of  $f_R(x) = \sqrt{10(x_2 - x_1^2)^2 + 5(1 - x_1)^2}$ , a variant of the Rosenbrock function (see [6, Ch. 1]). Here, the algorithm doesn’t converge to the minimum in four line minimizations; however, the calculation of conjugate directions guides the algorithm along narrow valleys of  $f_R$ .

## REFERENCES

- [1] R. Bachmayer and N.E. Leonard. Vehicle networks for gradient descent in a sampled environment. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 1, pages 112–117 vol.1, 2002.
- [2] E. Burian, D. Yoerger, A. Bradley, and H. Singh. Gradient search with autonomous underwater vehicles using scalar measurements. In *Proceedings of the 1996 Symposium on Autonomous Underwater Vehicle Technology*, pages 86–98, 1996.
- [3] C. Cai, A. R. Teel, and R. Goebel. Results on existence of smooth Lyapunov functions for (pre-)asymptotically stable hybrid systems with non-open basins of attraction. In *Proceedings of the 26th ACC*, 2006.

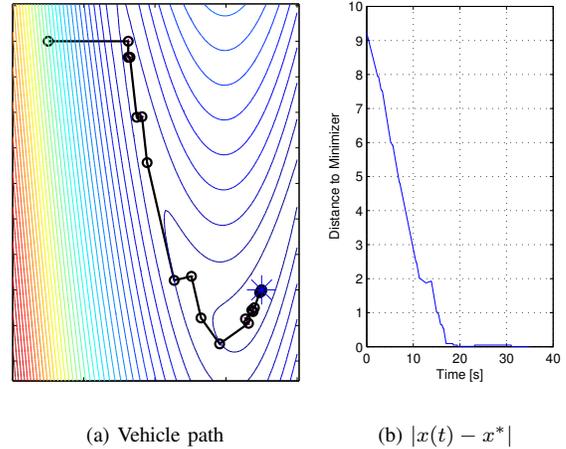


Fig. 9. The RSP algorithm drives the point-mass vehicle to a neighborhood of the minimum of  $f_R$ .  $d = 0.05$ ,  $\gamma = 1$ . (a) Shows the vehicle path (black) and the minimum of  $f_R$  denoted by a blue  $*$ . (b) Shows the time taken by algorithm to reach minimum at  $\gamma = 1\text{m/s}$ .

- [4] Joao Borges de Sousa, Karl H. Johansson, Jorge Silva, and Alberto Speranzon. A verified hierarchical control architecture for coordinated multi-vehicle operations. *International Journal of Adaptive Control and Signal Processing*, 21:159–188, 2007.
- [5] J. A. Farrell, S. Pang, and W. Li. Chemical plume tracing via an autonomous underwater vehicle. *IEEE Journal Of Oceanic Engineering*, 30(2):428–442, April 2005.
- [6] R. Fletcher. *Practical methods of optimization; (2nd ed.)*. Wiley-Interscience, New York, NY, USA, 1987.
- [7] R. Goebel, J.P. Hespanha, A.R. Teel, C. Cai, and R.G. Sanfelice. Hybrid systems: Generalized solutions and robust stability. In *6th IFAC NOLCOS*, pages 1–12, 2004.
- [8] R. Goebel and A.R. Teel. Solutions to hybrid inclusions via set and graphical convergence with stability theory applications. *Automatica*, 42(4):573–587, April 2006.
- [9] R. Goebel and A.R. Teel. Solutions to hybrid inclusions via set and graphical convergence with stability theory applications. *Automatica*, 42(4):573–587, 2006.
- [10] S. Pang and J. A. Farrell. Chemical plume source localization. *IEEE Transactions On Systems Man And Cybernetics*, 36(5):1068–1080, October 2006.
- [11] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162, 1964.
- [12] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C (2nd ed.): the art of scientific computing*. Cambridge University Press, New York, NY, USA, 1992.
- [13] H. H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184, 1960.
- [14] R.G. Sanfelice, R. Goebel, and A.R. Teel. Results on convergence in hybrid systems via detectability and an invariance principle. In *Proc. 24th IEEE American Control Conference*, pages 551–556, 2005.
- [15] Jorge Silva, Alberto Speranzon, Joao Borges de Sousa, and Karl Henrik Johansson. Hierarchical search strategy for a team of autonomous vehicles. In *Proceedings of IAV 2004, the 5th IFAC Symposium on Intelligent Autonomous Vehicles*, 2004.
- [16] C.S. Smith. The automatic computation of maximum likelihood estimates. *SIAM Review*, 4:343, 1962.
- [17] C. Zhang, D. Arnold, N. Ghods, A. Siranosian, and M. Krstic. Source seeking with nonholonomic unicycle without position measurement – Part I: Tuning of forward velocity. In *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006.
- [18] Chunlei Zhang, A. Siranosian, and M. Krstic. Extremum seeking for moderately unstable systems and for autonomous vehicle target tracking without position measurements. In *Proceedings of the 25th American Control Conference*, 2006.