Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

A Hybrid Control Algorithm for Gradient-Free Optimization using Conjugate Directions ⋆

Alessandro Melis * **Ricardo G. Sanfelice** ** **Lorenzo Marconi** *

* *CASY-DEI, University of Bologna, Bologna, Italy (e-mail: alessandro.melis4@unibo.it, lorenzo.marconi@unibo.it).*
** *Department of Electrical and Computer Engineering, University of California, Santa Cruz, CA 95064, USA (e-mail: ricardo@ucsc.edu)*

Abstract:
The problem of steering a particular class of n-dimensional continuous-time dynamical systems towards the minima of a function without gradient information is considered. We propose a hybrid controller, implementing a discrete-time Direct Search algorithm based on conjugate directions, able to solve the optimization problem for the resulting closed loop system in an almost global sense. Moreover, we propose a modified version by imposing a lower bound on the step size and able to achieve robust practical convergence to the optimum.

Keywords: Hybrid Systems, Direct Search Algorithms, Robustness Analysis, Optimization, Conjugate Directions

1. INTRODUCTION

In this paper we study the problem of steering a particular class of dynamical systems towards the set of minima of an objective function, assumed to not be known but whose measurements are available at fixed intervals of time. We consider continuous-time dynamical systems that can be steered, by a known input, between any two points of the state space. Examples of such systems are completely controllable linear time-invariant systems, as well as nonlinear systems whose reachable set after time $T > 0$, for all $T > 0$, is the whole state space, e.g. the Dubin's vehicle, see Shkel and Lumelsky (2001).

The problem at hand has been tackled in the literature with a variety of approaches, mostly related to source-seeking applications. In Burian et al. (1996) an approximated gradient descent method is implemented in order to steer an autonomous underwater vehicle to the deepest part of a pond, or locate hydrothermal vents. However no stability results are provided for the closed-loop formed by the exploration algorithm and the vehicle dynamics. Regarding source-seeking applications, a similar approach is used within a multi-agent framework in Bachmayer and Leonard (2002), where, instead, local gradient measurements are assumed, and stability of the closed-loop system is proved. In Azuma et al. (2012) a modified version of the simultaneous-perturbation stochastic approximation is proposed in order to recursively compute directions of exploration for a general randomly switching objective function, asked to be thrice differentiable. In Cochran and Krstic (2009) an extremum seeking controller is adopted assuming continuous availability of the measurements of a convex quadratic objective function.

The problem treated in this paper was solved in Mayhew et al. (2007) (see also Mayhew et al. (2008b) and Mayhew et al. (2008a)) for a convex objective function in a 2-dimensional search space by a hybrid controller, based on the Recursive Smith-Powell (RSP) algorithm. The latter is an optimization algorithm that, through a series of line minimizations, sequentially compute a set of conjugate directions. For convex quadratic functions, it ensures to reach a neighborhood of the minimizer in a finite amount of line minimizations. The classic RSP implementation, as in Mayhew et al. (2007), uses discrete line minimizations with fixed step size, able to achieve practical stability of a set of minimizers for the 2-dimensional convex quadratic case. In Coope and Price (1999) an extension of the RSP was proposed in the general context of continuously differentiable functions. By using a decreasing step size asymptotically converging to zero, this algorithm ensures asymptotic convergence to a stationary point. While some robustness results of the RSP algorithm where shown in Mayhew et al. (2007), no results are present regarding the algorithm in Coope and Price (1999), and in particular for the more general class of Direct Search methods.

In this paper, we study the class of Direct Search methods, to which the RSP algorithm belongs, which are optimization algorithms that minimize (or maximize) an objective function without using (or estimating) derivative information of any order of the objective function (see Lewis et al. (2000) for an overview). In particular, we propose a direct search algorithm combining the results of Coope and Price (1999), Kolda et al. (2003), and Garcia-Palomares and Rodriguez (2002) in order to achieve, contrary to the RSP algorithm, asymptotic convergence to the set of minima. Due to the inherent discrete dynamics of the algorithm and the continuous dynamics of the underlying dynamical system, on the wake of Mayhew et al. (2007), the controller is implemented by relying on the hybrid systems framework of Goebel et al. (2012). The proposed hybrid controller addresses the optimization problem of an n-dimensional continuously differentiable function with a set of global min-

⋆ Research partially supported by the European Project "AerIal RoBotic technologies for professiOnal seaRch aNd rescuE" (AirBorne), Call: H2020, ICT-25-2016/17, Grant Agreement no: 780960, by NSF Grants no. ECS-1710621 and CNS-1544396, by AFOSR Grants no. FA9550-16-1-0015, FA9550-19-1-0053, and Grant no. FA9550-19-1-0169, and by CITRIS and the Banatao Institute at the University of California.

ima, and possibly isolated local maxima, and guarantees almost global asymptotic stability of the set of minima. Moreover, we propose a robust algorithm, addressing n-dimensional objective functions (including the results of Mayhew et al. (2007) as a special case), highlighting that a trade-off between asymptotic convergence and robustness is mandatory.

Notation: The set $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}_{\geq 0} := [0, \infty)$, $\mathbb{R}_{\geq 1} := [1, \infty)$, and $\mathbb{N}$ the set of natural numbers. We let e denote Euler's number. We denote by $|\cdot|$ the absolute value of a scalar quantity and $\|\cdot\|$ the vector 2-norm. For a scalar function $f : \mathbb{R}^n \to \mathbb{R}$, we denote as $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ the gradient of f. Given a nonempty set $\mathcal{A} \subset \mathbb{R}^n$ and $\varepsilon > 0$, we denote as $\varepsilon\mathbb{B}(\mathcal{A})$ the set $\{x \in \mathbb{R}^n : \|x\|_{\mathcal{A}} < \varepsilon\}$, where $\|x\|_{\mathcal{A}} := \inf_{y \in \mathcal{A}} \|x - y\|$. A set valued mapping F from $\mathbb{R}^n$ to $\mathbb{R}^m$ is denoted as $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$. Define a hybrid system in $\mathbb{R}^n$ as the 4-tuple $\mathcal{H} = (C, F, D, G)$, with $C \subset \mathbb{R}^n$ the flow set, $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ the flow map, $D \subset \mathbb{R}^n$ the jump set, and $G : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ the jump map. Solutions to hybrid systems are defined on *hybrid time domains* (see Goebel et al. (2012) for more details) parameterized by a continuous time variable $t \in \mathbb{R}_{\geq 0}$ and a discrete time variable $j \in \mathbb{N}$, keeping track, respectively of the continuous and discrete evolution. We denote as $\operatorname{dom} x \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$ the hybrid time domain corresponding to the solution x. We say that for a hybrid system $\mathcal{H}$ with state $x \in \mathbb{R}^n$, the set $\mathcal{A} \subset \mathbb{R}^n$ is: *stable* if for all $\epsilon > 0$, there exists $\delta_\epsilon > 0$ such that $x(0,0) \in \delta_\epsilon\mathbb{B}(\mathcal{A})$ implies $x(t,j) \in \epsilon\mathbb{B}(\mathcal{A})$ for all $(t,j) \in \operatorname{dom} x$; *globally attractive* if $(t,j) \mapsto \|x(t,j)\|_{\mathcal{A}}$ is bounded and $\lim_{t+j\to\infty} \|x(t,j)\|_{\mathcal{A}} = 0$, with $(t,j) \in \operatorname{dom} x$; *almost globally attractive* when it is globally attractive from all initial conditions apart from a set of measure zero; *almost globally asymptotically stable* if it is both stable and almost globally attractive; *semiglobally practically asymptotically stable* on the parameter $\theta \in \Theta \subset \mathbb{R}^m$, with $m > 0$, if, assuming $\mathcal{H}$ complete and dependent on θ, for any $\epsilon_1 > \epsilon_2 > 0$ and there exist $\delta > 0$ and $\Theta^\star \subset \Theta$ such that for all $\theta \in \Theta^\star$, $x(0,0) \in \delta\mathbb{B}(\mathcal{A})$ implies $x(t,j) \in \epsilon_1\mathbb{B}(\mathcal{A})$ for all $(t,j) \in \operatorname{dom} x$ and $\lim_{t+j\to\infty} \|x(t,j)\|_{\epsilon_2\mathbb{B}(\mathcal{A})} = 0$.

2. PROBLEM FORMULATION

In this paper we tackle the following optimization problem:

Problem 1. Minimize a function $f : \mathbb{R}^n \to \mathbb{R}$, namely

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1}$$

subject to the dynamics

$$\dot{\xi} = \varphi(\xi, u) \qquad \xi = \operatorname{col}(x, \zeta) \in \mathbb{R}^{n+l}, u \in \mathbb{R}^m. \tag{2}$$

The state variables x represent the variables involved in the optimization problem, while ζ represents other possible states.

For simplicity we consider $\varphi : \mathbb{R}^{n+l} \times \mathbb{R}^m \to \mathbb{R}^{n+l}$ to be continuously differentiable in ξ and u. Moreover, given $\tau^\star > 0$, we assume that for each x_0 and x_f in $\mathbb{R}^n$, with $x_0 \neq x_f$, there exists $t \mapsto u(t)$ such that the solution to $\dot{\xi} = \varphi(\xi, u(t))$ from $\xi_0 = (x_0, \cdot)$ reaches $\xi_f = (x_f, \cdot)$ after $\tau^\star$ seconds. We assume that for each bounded input $u(t)$ for all $t \geq 0$, $\zeta(t)$ is bounded for all $t \geq 0$. The class of systems represented by (2), includes, for example, point-mass vehicles ($\xi = x$, with x representing the position) and Dubin's vehicles ($\xi = \operatorname{col}(x^T, \zeta)$, with x and ζ representing position and orientation, respectively).

Moreover, we make the following assumptions on f:

(A0) f is continuously differentiable, lower bounded and it is not assumed to be known, but sampled measurements of it are available every $\tau^\star > 0$, with $\tau^\star$ a tunable parameter;
(A1) the set $\{x \in \mathbb{R}^n : \nabla f(x) = 0\}$ of critical points of f is such that every local minimum is also a global minimum (i.e. all local minima share the same objective function value), every local maximum is an isolated point and f is analytic at every local maximum, and there are no saddle points;
(A2) the sublevel sets of f, namely the sets $\mathcal{L}_f(c) := \{x \in \mathbb{R}^n : f(x) \leq c\}$, are compact for all $c \in \mathbb{R}$.

Assumptions (A0) and (A2) are standard for Direct Search methods, see Coope and Price (1999), Kolda et al. (2003) and Garcia-Palomares and Rodriguez (2002). Assumption (A0) can be relaxed by considering f to be locally Lipschitz, as shown in Kolda et al. (2003) and Popovic and Teel (2004), which requires the use of generalized gradients for analysis.

The reason for the particular structure of the set of critical points assumed in (A1) stems from the fact that our goal is to prove and guarantee convergence to the set of minima. While the assumptions on the value of the local minima is considered to simplify the structure of the problem, without the other assumptions on local maxima and saddle points, Direct Search algorithms, and our proposed controller derived from it, only guarantee convergence to the set of critical points.

Notice that, contrary to Mayhew et al. (2007), no convexity assumptions have been made on the cost function.

3. THE RSP AND THE PROPOSED ALGORITHM

3.1 Background

Throughout the paper we call *line minimization* any procedure that, given a function, a direction and a point, explores the line defined by the direction applied to the point, and returns the position of the minimum, or point in a neighborhood of it, of the function along the line.

We label a line minimization as *exact* when the minimum along the explored direction is exactly reached, and as *discrete* when the line minimization is an iterative procedure that explores at each iteration a new point at distance $\Delta > 0$ (fixed or changing at each iteration), called *step size*, from the previously explored one. A discrete line minimization terminates when the function value of the newly explored point did not decrease enough with respect to the function value at the last explored point.

Given a set $\mathcal{G} \subset \mathbb{R}^n$ of linearly independent directions spanning $\mathbb{R}^n$, the classic RSP sequentially computes exact line minimizations along the directions in $\mathcal{G}$ in order to minimize the cost function. Moreover, every n line minimizations, a new search direction $d_{new} \in \mathbb{R}^n$ is computed by exploiting the $Parallel\ Subspace\ Property$ (see Theorem 4.2.1 in Fletcher (2000)) and the set $\mathcal{G}$ is updated accordingly. For a convex quadratic function with Hessian matrix H, the newly computed direction d_{new} is conjugate, by the Parallel Subspace Property, to the last $n-1$ directions in $\mathcal{G}$, i.e. such that $d_{new}^T H d = 0$ for each $d \in \mathcal{G}$. This property of conjugacy of directions for a convex quadratic function implies that the line minimization along one direction is independent of the line minimizations along the other directions in $\mathcal{G}$. Thus, given a set of n conjugate directions for a convex quadratic function from $\mathbb{R}^n$ to $\mathbb{R}$, the minimum will be reached after n line minimizations, each along a different conjugate direction. By recursively computing a set of conjugate directions, the RSP algorithm reaches the minimum of a convex quadratic function, starting from a set of linearly independent directions, in at most n^2 line minimizations. This property is usually denoted as *quadratic termination property*.

3.2 Proposed algorithm

The algorithm proposed in this paper, shown in Alg. 1, is inspired by Garcia-Palomares and Rodriguez (2002) and improves the results in Mayhew et al. (2007) by guaranteeing, under the less restrictive assumptions (A0) and (A1), asymptotic convergence to the set of minima. The main differences with the RSP considered in Mayhew et al. (2007) are reported in the following. In particular:

1) A different step size Δ_i is associated to each direction $d_j \in \mathcal{G}$ in order to guarantee more freedom of exploration. As such, when a new direction is computed (lines 28-32) also a new step size is associated to the new direction (line 27).
2) A variable global step size Φ is considered, such that $\lambda_s\Phi \leq \Delta_j \leq \lambda_t\Phi$ for all $j \in \{0, 1, ..., n-1\}$, with $0 < \lambda_s < 1 < \lambda_t$. If no improvement is found along any direction, the global step size Φ is reduced to $\mu\Phi$, with $\mu \in (0, 1/\lambda_t)$ (lines 14-21).
3) In case no improvement is made along a direction (lines 8-12), meaning that $\alpha_{kj} = 0$, the corresponding step size is reduced. This is the key step guaranteeing asymptotic convergence to the minima of the cost function.
4) The new direction, computed via the Parallel Subspace Property, is "accepted" only if it keeps the directions in $\mathcal{G}$ linearly independent (lines 28), otherwise the previous set of directions is retained.

Remark 2. The idea of reducing the step size when no improvement is found stems from Theorem 3.3 in Kolda et al. (2003), where it is reported that the norm of the gradient of the cost function, at points where no improvement was found along any direction, is bounded by a class $\mathcal{K}$ function of the step size. Thus, reducing the step size at those iterations implies reducing the norm of the gradient, hence approaching a stationary point (or minimum in our case). •

The line minimization procedure explores a direction d_{kj} from a starting point x_{kj} and returns the distance α_{kj} traveled from x_{kj} to the found minimum of f along d_{kj}. The main differences in the line minimization procedure with respect to the RSP in Mayhew et al. (2007) are the following:

1) Newly explored points are accepted only if a *sufficient decrease* condition is satisfied (lines 2 and 12), namely the function has decreased at least $\rho(\Delta)$ along the direction d, where $\rho : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is defined below.
2) When a new iteration is accepted, the step size is, possibly, increased (lines 5 and 15) if the step size does not violate the upper bound imposed by the global step size.

Remark 3. The sufficient decrease condition (lines 2 and 12) guarantees that the Armijo condition, needed for the algorithm to converge, is satisfied (see Section 3.7.1 in Kolda et al. (2003) for more details). The function ρ in the sufficient decrease condition is a strictly increasing function of Δ, that at $\Delta = 0$ is smooth (from the right) but non-analytic, and such that $\rho(\Delta) = o(\Delta^n)$ for $\Delta \to 0$ for all $n \in \mathbb{N}$. The properties of ρ imply

Algorithm 2 : Proposed RSP

1 **Data:** Suppose that we are given a set $\mathcal{G}_0 := \{d_{00}, ..., d_{0(n-1)}\}$ of linearly independent directions in $\mathbb{R}^n$, the set of initial step-sizes $\Delta_{0,n-1} := \{\Delta_{00}, ..., \Delta_{0(n-1)}\}$, each corresponding to a direction in $\mathcal{G}$, $\Phi_0 > 0$ a global step size, $0 < \lambda_s < 1 < \lambda_t$, $\theta \in (0,1)$, $\gamma \geq 1$, $\mu \in (0, 1/\lambda_t)$, $\delta_{\text{det}} > 0$. Let the initial position $x_\circ \in \mathbb{R}^n$ be given.
2 **Initialization:** Let α be such that $x_\circ + \alpha d_{0(n-1)}$ is the minimizer of (1) resulting from a line minimization with step-size $\Delta_{0(n-1)}$, along the direction $d_{0(n-1)}$ from $x_\circ$.
3 $x_{00} \longleftarrow x_\circ + \alpha d_{0(n-1)}$
4 **for** $k \in \mathbb{N}$ **do**
5   **for** $j \in \{0, 1, ..., n-1\}$ **do**
6   Compute a line minimization with step size Δ_{kj} along d_{kj} from x_{kj} to obtain α_{kj}
7     $x_{k(j+1)} \longleftarrow x_{kj} + \alpha_{kj} d_{kj}$
8   **if** $\alpha_{kj} = 0$ **then**
9   **if** $\theta\Delta_{kj} \leq \lambda_s \Phi_k$ **then**
10         $\Delta_{kj} \longleftarrow \theta\Delta_{kj}$
11   **end**
12   **end**
13   **end**
14   **if** $\alpha_{kj} = 0 \ \forall j \in \{0, 1, ..., n-1\}$ **then**
15     $\Phi_{k+1} = \mu\Phi_k$
16   **for** $j \in \{0, 1, ..., n-1\}$ **do**
17   **if** $\Delta_{kj} > \mu\lambda_t\Phi_k$ **then**
18         $\Delta_{kj} = \mu\lambda_t\Phi_k$
19   **end**
20   **end**
21   **end**
22   $\bar{z} \longleftarrow x_{k0} + \sum_{j=0}^{n-1} \alpha_{kj} d_{kj}$
23   **for** $j \in \{0, 1, ..., n-2\}$ **do**
24     $d_{(k+1)j} \longleftarrow d_{k(j+1)}$
25     $\Delta_{(k+1)j} \longleftarrow \Delta_{k(j+1)}$
26   **end**
27   $\Delta_{(k+1)(n-1)} \longleftarrow \max_{j \in \{0,1,...,n-2\}} \Delta_{(k+1)j}$
28   **if** $|\det(\text{col}(d_{(k+1)0}^\top, d_{(k+1)1}^\top, ..., d_{(k+1)(n-2)}^\top, (\bar{z} - x_{k0})^\top))| \geq \delta_{det}$ **then**
29     $d_{(k+1)(n-1)} \longleftarrow \bar{z} - x_{k0} \ \ (= \sum_{j=0}^{n-1} \alpha_{kj} d_{kj})$
30   **else**
31     $d_{(k+1)(n-1)} = d_{k0}$
32   **end**
33   Compute a line minimization with step size $\Delta_{(k+1)(n-1)}$ along $d_{(k+1)(n-1)}$ from $\bar{z}$ to obtain $\alpha_{(k+1)(n-1)}$
34   $x_{(k+1)0} \longleftarrow \bar{z} + \alpha_{(k+1)(n-1)} d_{(k+1)(n-1)}$
35 **end**

Line minimization procedure

1 **Initialization:** $i = 0$, $x_{kj0} = x_{kj}$, $\Delta_{kj0} = \Delta_{kj}$
2 **while** $f(x_{kji} + \Delta_{kji} d_{kj}) \leq f(x_{kji}) - \rho(\Delta_{kji})$ **do**
3   $x_{kj(i+1)} \longleftarrow x_{kji} + \Delta_{kji} d_{kj}$
4   **if** $\gamma\Delta_{kji} \leq \lambda_t \Phi_k$ **then**
5     $\Delta_{kj(i+1)} \longleftarrow \gamma\Delta_{kji}$
6   **else**
7     $\Delta_{kj(i+1)} \longleftarrow \lambda_t \Phi_k$
8   **end**
9   $i \longleftarrow i + 1$
10 **end**
11 **if** $i = 0$ **then**
12   **while** $f(x_{kji} - \Delta_{kji} d_{kj}) \leq f(x_{kji}) - \rho(\Delta_{kji})$ **do**
13     $x_{kj(i+1)} \longleftarrow x_{kji} - \Delta_{kji} d_{kj}$
14   **if** $\gamma\Delta_{kji} \leq \lambda_t \Phi_k$ **then**
15       $\Delta_{kj(i+1)} \longleftarrow \gamma\Delta_{kji}$
16   **else**
17       $\Delta_{kj(i+1)} \longleftarrow \lambda_t \Phi_k$
18   **end**
19     $i \longleftarrow i + 1$
20   **end**
21 **end**
22 $\alpha_{kj} \longleftarrow i\Delta_{kji}$, $x_{kj} \longleftarrow x_{kji}$, $\Delta_{kj} \longleftarrow \Delta_{kji}$

Alg. 1: New RSP algorithm with line minimization procedure.

that, under assumption (A1), if $\bar{x} \in \mathbb{R}^n$ is a local maxima for f, there exists $\bar{\Delta} > 0$ such that for all $d \in \mathcal{G}$ and $\Delta \in (0, \bar{\Delta}]$, $f(\bar{x} + \Delta d) < f(\bar{x}) - \rho(\Delta)$. •

Define the set of global minima of f as $\mathcal{A}^\star := \{x^\star \in \mathbb{R}^n : f(x^\star) \leq f(x) \ \forall x \in \mathbb{R}^n\}$ and define $i_{kj}^\star$ as the number of steps computed in the line minimization procedure in Alg. 1 at iteration k along direction d_j. We can conclude the following convergence result for the algorithm in Alg. 1.

Theorem 4. Consider the class of cost functions fulfilling (A0)-(A2). Then, for any initial condition $x_\circ \in \mathbb{R}^n$, the sequence of iterates x_{kji} generated by the RSP algorithm and the line minimization procedure in Alg. 1 is such that

$$\lim_{k\to\infty} \lim_{i\to i^*_{kj}} \|x_{kji}\|_{\mathcal{A}^\star} = 0 \quad \forall j \in \{0,1,...,n-1\}. \tag{3}$$

□

The proof of Theorem 4 is based on standard arguments for the proof of convergence to stationary points of f in Direct Search algorithms. In particular, under assumptions (A0) and (A2), convergence of the sequence of global step size Φ_k to 0 is shown first, which, together with the sufficient decrease condition, guarantees convergence of x_{kji} to a stationary point. Under assumption (A2), and due to the particular structure of ρ, convergence to the set of minima is shown. The detailed proof of Theorem 4 will be published elsewhere.

4. HYBRID CONTROLLER

In this section we design a hybrid controller $\mathcal{H}_c$ implementing the new RSP to solve Problem 1 under the assumptions (A0)-(A2), and steer the state of (2) towards the set of minima of f.

The reason for resorting to the hybrid systems framework is to provide results regarding the stability and robustness of the proposed algorithm when applied to continuous-time dynamical systems, also in the presence of measurement noise. In particular, the resulting hybrid controller is based on the framework for hybrid systems in Goebel et al. (2012), and its dynamics are given by a flow map F_c when the state belongs to the flow set C, and a jump map G_c when the state belongs to the jump set D. In particular, the algorithm in Alg.1 defines the jump map G_c, which is set valued in order to satisfy the *hybrid basic conditions* (Assumption 6.5 in Goebel et al. (2012)) and lead to a closed-loop system $\mathcal{H}_{cl}$, given by the interconnection of $\mathcal{H}_c$ and (2), that is nominally well-posed (see Definition 6.2 in Goebel et al. (2012)), a property needed for the application of invariance principles in the proofs of the results in the next section.

The state of the controller is defined as $x_c = \text{col}(\tau, \Delta_0, ..., \Delta_{n-1}, d_0, ..., d_{n-1}, \Phi, \lambda, \alpha^\top, \bar{\alpha}, p, m, \ell, q, z, \Delta, v^\top)$, and it ranges in $\mathcal{X}_c := \mathbb{R}_{\geq 0} \times \mathcal{X}_{c/\tau}$, with $\mathcal{X}_{c/\tau} := \mathbb{R}^n_{\geq 0} \times \mathbb{R}^{n\times n} \times \mathbb{R}_{>0} \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}_{\geq 0} \times \{-1,1\} \times \{0,1\} \times \{0,1,...,n\} \times \{0,1,2\} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}^n$.

The state variable τ is a timer that resets every $\tau^\star > 0$ seconds and regulates when new cost function evaluations are available. Its hybrid dynamics are given by

$$\dot{\tau} = 1 \qquad (\xi, x_c) \in C := \{(\xi, x_c) \in \mathbb{R}^{n+l} \times \mathcal{X}_c : \tau \leq \tau^\star\}, \tag{4}$$

during flow, and

$$\tau^+ = 0 \qquad (\xi, x_c) \in D := \{(\xi, x_c) \in \mathbb{R}^{n+l} \times \mathcal{X}_c : \tau \geq \tau^\star\}, \tag{5}$$

at jumps.

The states $d_j \in \mathbb{R}^n$ and $\Delta_j \in \mathbb{R}_{\geq 0}$, $j = 0, 1, ..., n$, represent, in Alg. 1, the search directions and the step sizes corresponding to each direction. The state variable $\lambda \in \mathbb{R}$, which keeps track of the distance traveled along the currently explored direction, and the state variable $\alpha \in \mathbb{R}^n$, which stores the total traveled vector from direction d_0, are related to the distance traveled along each direction, which is the variable α_{kj} introduced in Alg. 1.

The state $\Phi \in \mathbb{R}_{\geq 0}$ represents the global step size and $\bar{\alpha} \in \mathbb{R}_{\geq 0}$ the total distance traveled during each cycle of directions exploration.

The positive or negative exploration along the current direction is determined by the state $p \in \{-1, 1\}$, and the variable $m \in \{0, 1\}$ indicates whether a turn has already been performed along the current direction.

To define in which operating point of the proposed RSP algorithm the controller is, the state variables $\ell \in \{0, 1, ..., n\}$ and $q \in \{0, 1, 2\}$ have been introduced. The variable ℓ represents the state of the RSP, namely which direction is currently being explored. Notice that it has $n + 1$ components since the direction d_{n-1} is explored twice to be able to exploit the *Parallel Subspace Property*. The variable q, defining the state of the line minimization, assumes these values:

- $q = 0$: when a positive line minimization is performed;
- $q = 1$: when a negative line minimization is performed;
- $q = 2$: when a line minimization is completed.

The state variable $z \in \mathbb{R}$ is a memory state that keeps track of the best minimum value of f found satisfying the sufficient decrease condition.

Two more states have been added for convenience: $\Delta \in \mathbb{R}$ and $v \in \mathbb{R}^n$, which store the currently explored search direction and its corresponding step size, respectively.

The structure of $\mathcal{H}_c$ is given by

$$\mathcal{H}_c\ :\ \begin{cases} \dot{x}_c = F_c := \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}^\top & (x, x_c) \in C \\ x_c^+ \in G_c(x_c, f(x)) := \begin{bmatrix} 0 \\ G_{c/\tau}(x_c, f(x)) \end{bmatrix} & (x, x_c) \in D \\ u = K(x, x_c, \tau^\star), \end{cases} \tag{6}$$

with sets $C,\ D$ defined in (4) and (5). The flow map F_c is a single-valued constant function with all components equal to zero except for the timer. The jump map $G_c : \mathcal{X}_c \times \mathbb{R} \to \mathcal{X}_c$ is a set-valued map, composed by the timer discrete dynamics and the map $G_{c/\tau} : \mathcal{X}_c \times \mathbb{R} \to \mathcal{X}_{c/\tau}$, representing the hybrid implementation of Alg. 1. The output of $\mathcal{H}_c$ is a function $K : \mathbb{R}^n \times \mathcal{X}_c \times \mathbb{R}_{>0} \to \mathbb{R}^m$ that steers the ξ-subsystem in (2) between two points, similar to the input steering from x_0 to x_f as explained below (2).

The construction of $G_{c/\tau}$ is outlined next.

- Continue a positive line search: if $f(x) \leq z - \rho(\Delta)$, $p = 1$, $q \in \{0, 1\}$, $m = 0$, then $G_{c/\tau}$ implements $z^+ = f(x)$, $q^+ = 1$.
- Correct overshoot: if $f(x) \geq z - \rho(\Delta)$, $q \in \{0, 1\}$, $m = 0$, then $G_{c/\tau}$ implements $p^+ = -p$, $m^+ = 1$, $q^+ = q + 1$.
- Start a negative line search: if $m = 1$, $p = -1$, $q = 1$, then $G_{c/\tau}$ implements $z^+ = f(x)$, $m^+ = 0$, $\lambda^+ = 0$.
- Continue a negative line search: if $f(x) \leq z - \rho(\Delta)$, $p = -1$, $q = 1$, $m = 0$, then $G_{c/\tau}$ implements $z^+ = f(x)$.
- Update the direction and start a positive line search: If $q = 2$, then $G_{c/\tau}$ implements $q^+ = 0,\ p^+ = 1,\ \lambda^+ = 0,\ m^+ = 0$. Moreover if $v = d_j$, for some $j < n$, then $v^+ = d_{j+1}$, and if $\ell = n$, then $v^+ = d_{\text{new}}$ and $d_0^+ = d_{\text{new}}$

if linear independence of $\mathcal{G}$ is preserved, with $d_{\text{new}} \in \mathbb{R}^3$ computed exploiting the Parallel Subspace Property.

Due to reasons of space, the full expression of $G_{c/\tau}$ is not provided but it is available in the simulation case [1].

5. STABILITY ANALYSIS

Define the hybrid closed-loop $\mathcal{H}_{cl}$ as the interconnection of the dynamics (2) and the controller $\mathcal{H}_c$ developed in the previous section, namely

$$\mathcal{H}_{cl} \; : \begin{cases} \left.\begin{aligned} \dot{\xi} &= \varphi(\xi, K(x, x_c, \tau^\star)) \\ \dot{x}_c &= F_c \end{aligned}\right\} & (\xi, x_c) \in C \\ \left.\begin{aligned} \xi^+ &= \xi \\ x_c^+ &\in G_c(x_c, f(x)) \end{aligned}\right\} & (\xi, x_c) \in D \end{cases} \tag{7}$$

Define $\mathcal{A}_{dis} := \{-1, 1\} \times \{0, 1\} \times \{0, 1, ..., n\} \times \{0, 1, 2\}$. We consider the stabilization problem with respect to the sets $\mathcal{A} \subset \mathcal{A}_e \subset \mathbb{R}^{n+l} \times \mathcal{X}_c$, defined as

$$\begin{aligned} \mathcal{A} := \mathcal{A}^\star \times \mathbb{R}^l \times [0, \tau^\star] \times \{0^n\} \times \mathbb{R}^{n\times n} \times \{0\} \times \{0\} \times \\ \{0^n\} \times \{0\} \times \mathcal{A}_{dis} \times \{f(\mathcal{A}^\star)\} \times \{0\} \times \mathbb{R}^n, \end{aligned} \tag{8}$$

$$\begin{aligned} \mathcal{A}_e := \mathbb{R}^{n+l} \times [0, \tau^\star] \times \{\{0^n\} \times \mathbb{R}^{n\times n} \times \{0\} \cup \mathbb{R}^n \times \\ \{0^{n\times n}\} \times \mathbb{R}_{\geq 0}\} \times \{0\} \times \{0^n\} \times \{0\} \times \mathcal{A}_{dis} \\ \times \mathbb{R} \times \{\{0\} \times \mathbb{R}^n \cup \mathbb{R} \times \{0^n\}\}. \end{aligned} \tag{9}$$

The set $\mathcal{A}$ represents the desired equilibrium set, namely the subset of $\mathbb{R}^{n+l} \times \mathcal{X}_c$ such that if $(\xi(0,0), x_c(0,0)) \in \mathcal{A}$, then $x(t,j) \in \mathcal{A}^\star$ for all $(t,j) \in \text{dom}(\xi, x_c)$. Notice that invariance of $\mathcal{A}$ is guaranteed by all the step size variables being zero on $\mathcal{A}$. However, the set of equilibria of (7) is a superset of $\mathcal{A}$, and it is defined as the set with all the step size variables and/or directions equal to zero. The reason is that for $\Phi = 0$ and/or $d_j = 0$ for all $j \in \{0, 1, ..., n-1\}$, every $x \in \mathbb{R}^n$ is an equilibrium point for (7). The set of equilibria of (7) is exactly $\mathcal{A}_e$.

Theorem 5. *Let assumptions (A0)-(A2) hold and the parameters of the algorithm Alg. 1 satisfy $\tau^\star > 0$, $\delta_{det} > 0$, $0 < \lambda_s < 1 < \lambda_t$, $\mu \in (0, 1/\lambda_t)$, $\theta \in (0,1)$ and $\gamma \geq 1$. Then, for the closed-loop system $\mathcal{H}_{cl}$, the set $\mathcal{A}$ in (8) is*

- *stable;*
- *almost globally attractive;*

hence, it is almost globally asymptotically stable. Furthermore, the set $\mathcal{A}_e$ in (9) is globally attractive for $\mathcal{H}_{cl}$. □

The proof of Theorem 5 and of the next theorem are based on Lyapunov arguments and invariance principles, applied considering the Lyapunov candidate function $V(\xi, x_c) := z - f(\mathcal{A}^\star)$. From Theorem 5 and the structure of $\mathcal{A}$ and $\mathcal{A}_e$, it follows in particular that, for any initialization such that $\det(\text{col}(d_0, d_1, ..., d_{n-1})) \neq 0$ and $\Phi \neq 0$, boundedness of the closed-loop trajectories and asymptotic convergence to the set $\mathcal{A}$ are guaranteed.

In the case in which the cost function measurements are affected by noise, it is possible to show that general Direct Search Algorithms based on line minimizations and asymptotic step size reduction, and in particular the algorithm in Alg. 1, are not robust to any bounded random noise, even if stability has been shown and convergence results are attainable for a proper choice of initial conditions.

Robustness to measurement noise for the hybrid closed-loop system $\mathcal{H}_{cl}$ is recovered by imposing a lower bound $\underline{\Phi} > 0$ on the global step size Φ, implemented by restricting $\text{dom}\,\Phi$ to $[\underline{\Phi} + \epsilon, +\infty)$, with $\epsilon > 0$, and modifying $G_{c/\tau}$ by adding $\Phi^+ = \underline{\Phi} + \epsilon$ if $\mu\Phi \leq \underline{\Phi} + \epsilon$. Moreover, given $\delta_{\text{det}} > 0$, we restrict the domain of all the directions d_j to be such that $\det(\text{col}(d_0, d_1, ..., d_{n-1})) \geq \delta_{\text{det}}$. Without loss of generality, we will denote the desired equilibrium set within the restricted domain for the directions as $\mathcal{A}$.

Theorem 6. Let assumptions (A0)-(A2) hold, the parameters of the algorithm Alg. 1 satisfy $\tau^\star > 0$, $0 < \lambda_s < 1 < \lambda_t$, $\delta_{det} > 0$, $\mu \in (0, 1/\lambda_t)$, $\theta \in (0,1)$ and $\gamma \geq 1$. Then, for all measurement noise signals $n_s : \mathbb{R} \to \mathbb{R}$, with $|n_s(t)| \leq \bar{n}_s$, there exists $\underline{\Phi} > 0$, with the update of Φ modified such that $\Phi(t,j) \geq \underline{\Phi}$, for all $(t,j) \in \text{dom}\,\Phi$, the set $\mathcal{A}$ is semiglobally practically asymptotically stable on $\underline{\Phi} > 0$ for $\mathcal{H}_{cl}$. □

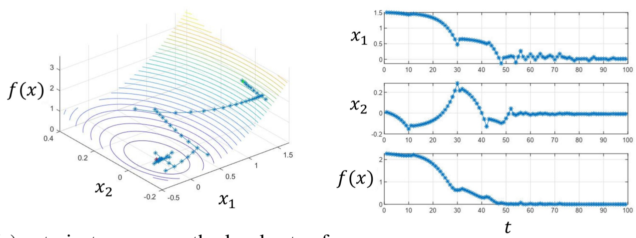
6. SIMULATIONS RESULTS

In this section we show the results of different simulations of the proposed hybrid controller to the minimization of a convex quadratic function.

Fig. 1 illustrates the level sets of the quadratic convex function

$$f(x) = x_1^2 + 5x_2^2, \tag{10}$$

where $x = \text{col}(x_1, x_2)$. The trajectory of a point-mass vehicle, steered by the proposed hybrid controller in order to minimize (10), is superimposed to the level sets of (10), showing the value of $f(x)$ at each corresponding point of the trajectory. The control input was chosen as $K(x, x_c, \tau^\star) = p\Delta v/\tau^\star$. The tunable parameters of the controller were defined as $\gamma = 1.2$, $\theta = 0.5$, $\delta_{\text{det}} = 0.001$, $\mu = 0.15$, $\lambda_s = 0.001$, $\lambda_t = 5$. It can be noticed as in both Fig. 1(a) and Fig. 1(b), the distance



(a) x trajectory versus the level sets of a quadratic convex function.

(b) x and $f(x)$ trajectories.

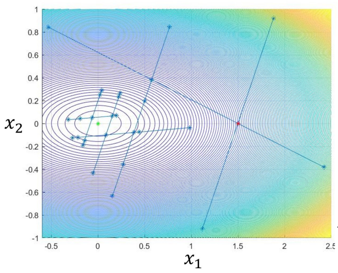
Fig. 1. Plot of the trajectories of x and $f(x)$, where $f(x) = x_1^2 + 5x_2^2$. (a) Shows the vehicle path (blue with '*' where jump occurs) on the level sets of f. The initial point is indicated with a green '*' and the unique minimizer $(0,0,0)$ with a red '*'. (b) Shows the evolution of x and $f(x)$ as function of time.

to the minimizer tends asymptotically to zero as the step size converges to zero.

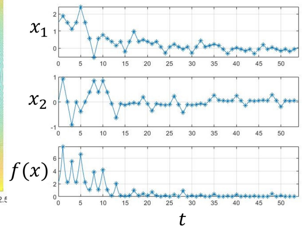
In Fig. 2 we show a comparison of the x-trajectories of $\mathcal{H}_{cl}$ for a point-mass vehicle in case measurement noise affecting (10) are considered. In Fig. 2(a)-2(b) no lower bound on Φ and no measurement noise is assumed, the x-trajectory indeed behaves similarly to the one in Fig. 1, converging asymptotically to the minimum $x^\star = (0,0)$. In Fig. 2(c)-2(d) no lower bound on Φ is assumed, but a measurement noise $n_s(t,j)$, upper

[1] GitHub Repository: https://github.com/AleMell/Hybrid-RSP

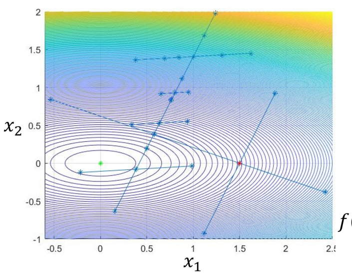
bounded by $\bar{n}_s = 0.04$ on f is considered. We highlight how the effect of the noise tricks the hybrid controller into steering the x-subsystem away from the minimum. This behavior can be seen in the plot of $f(x)$ in Fig. 2(d) after about 20 seconds of simulation. In Fig. 2(e)-2(f) the same measurement noise is assumed, but $\underline{\Phi} = 40$ is chosen, implying $\Delta_j \geq 0.4$ for all time. The imposed lower bound on Φ compensates the effects of the measurement noise, stabilizing the state x in a neighborhood of the minimum.



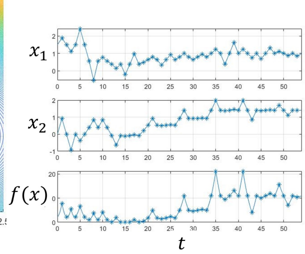
(a) x trajectory versus the level sets (10) assuming no lower bound on Φ and no measurement noise.



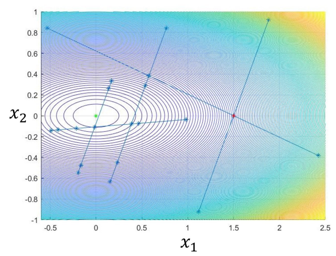
(b) x and $f(x)$ trajectories.



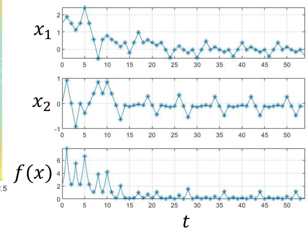
(c) x trajectory versus the level sets (10) assuming no lower bound on Φ but measurement noise added to f.



(d) x and $f(x)$ trajectories.



(e) x trajectory versus the level sets (10) assuming a lower bound on Φ and measurement noise added to f.



(f) x and $f(x)$ trajectories.

Fig. 2. Comparison of the plots of the trajectories of $x(t, j)$ and $f(x(t, j))$, where $f(x) = x_1^2 + 5x_2^2$, under different assumptions on measurement noise and $\underline{\Phi}$. (a),(c) and (e) show the vehicle path (blue with '*' where jump occurs) on the level sets of f. The initial point is indicated with a red '*' and the unique minimizer $(0, 0, 0)$ with a green '*'. (b),(d) and (f) show the evolution of x and $f(x)$ as function of time.

7. CONCLUSION

This paper presents an extension of the results in Mayhew et al. (2007). In particular, an hybrid controller based on a modified RSP algorithm, which optimizes an objective function without gradient information, and that is able to achieve almost global asymptotic stability of the closed loop composed by the controller and a particular class of continuous-time dynamical systems is proposed. As direct search methods are not robust to measurement noise, a modified practical scheme is proposed, showing how a trade-off between asymptotic convergence and robustness is inevitable for this class of algorithms. Simulations results are provided to validate the proposed approach. Future developments include the extension of the proposed controller to the multiagent scenario, in order to efficiently exploit the parallel subspace property, and to more general objective functions, e.g. to nonsmooth functions, as well as the relaxation of the regularity of V and of the controllability assumptions on the vehicle dynamics.

REFERENCES

Azuma, S.i., Sakar, M.S., and Pappas, G.J. (2012). Stochastic Source Seeking by Mobile Robots. *IEEE Transactions on Automatic Control*, 57(9), 2308–2321.

Bachmayer, R. and Leonard, N.E. (2002). Vehicle Networks for Gradient Descent in a Sampled Environment. In *Proceedings of the 41 Conference on Decision and Control*.

Burian, E., Yoerger, D., Bradley, A., and Singh, H. (1996). Gradient Search with Autonomous Underwater Vehicles Using Scalar Measurements. In *Proceedings on Symposium on Autonomous Underwater Vehicle Technology*.

Cochran, J. and Krstic, M. (2009). Nonholonomic Source Seeking With Tuning of Angular Velocity. *IEEE Transactions on Automatic Control*, 54(4), 717–731.

Coope, I.D. and Price, C.J. (1999). A Direct Search Conjugate Directions Algorithm for Unconstrained Minimization. Technical Report November.

Fletcher, R. (2000). *Practical Methods of Optimization*. Second edition.

Garcia-Palomares, U. and Rodriguez, J. (2002). New Sequential and Parallel Derivative-free Algorithms for Unconstrained Minimization. *SIAM Journal on Optimization*, 13(1), 79–96.

Goebel, R., Sanfelice, R.G., and Teel, A.R. (2012). *Hybrid Dynamical Systems Modelling, Stability and Robustness*.

Kolda, T.G., Lewis, R.M., and Torczon, V. (2003). Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods. *SIAM Review*, 45(3), 385–482.

Lewis, R.M., Torczon, V., Trosset, M.W., and William, C. (2000). Direct Search Methods : Then and Now. *Journal of Computational and Applied Mathematics*, 124, 191–207.

Mayhew, C.G., Sanfelice, R.G., and Teel, A.R. (2007). Robust Source-Seeking Hybrid Controllers for Autonomous Vehicles. *American Control Conference*.

Mayhew, C.G., Sanfelice, R.G., and Teel, A.R. (2008a). Robust hybrid source-seeking algorithms based on directional derivatives and their approximations. *Proceedings of the IEEE Conference on Decision and Control*, (2), 1735–1740.

Mayhew, C.G., Sanfelice, R.G., and Teel, A.R. (2008b). Robust source-seeking hybrid controllers for nonholonomic vehicles. *Proceedings of the American Control Conference*, 2722–2727.

Popovic, D. and Teel, A.R. (2004). Direct Search Methods for Nonsmooth Optimization. *Conference on Decision and Control*, 3173–3178.

Shkel, A.M. and Lumelsky, V. (2001). Classification of the Dubins set. *Robotics and Autonomous Systems*, 34, 179–202.