Derivative-Free Optimization Algorithms based on Non-Commutative Maps *

Jan Feiling[1,2], Amelie Zeller[1], and Christian Ebenbauer[1]

[1]Institute for Systems Theory and Automatic Control, University of Stuttgart, Germany
{jan.feiling@ist,st119167@stud,ce@ist}.uni-stuttgart.de
[2]Dr. Ing. h.c. F. Porsche AG, Stuttgart, Germany
jan.feiling@porsche.de

Abstract. A novel class of derivative-free optimization algorithms is developed. The main idea is to utilize certain non-commutative maps in order to approximate the gradient of the objective function. Convergence properties of the novel algorithms are established and simulation examples are presented.

1. INTRODUCTION

In the course of continuously increasing computational power, optimization algorithms and optimization-based control play more and more a central role in solving control and real-time decision making problems. In many tasks, the resulting optimization problems are very challenging, i.e., they are high-dimensional, non-convex, non-smooth, or of stochastic nature. Hence, improving existing optimization algorithms and developing novel algorithms is of central importance. An interesting class of algorithms are derivative-free algorithms, which typically need only evaluations of the objective function for optimization. Hence, derivative-free optimization algorithms are simple and appealing in many challenging applications, for example in real-time optimization, where only noisy measurements and no mathematical description of the objective function are available and therefore no derivative, i.e., no gradient can be computed directly. Consequently, derivative-free optimization and adaptive control techniques are often used in this applications [2, 3, 4, 5, 6, 7]. Moreover, in the field of machine learning, derivative-free optimization problems [8, 9, 10, 11] gain renewed interest, which are often tackled with so-called stochastic gradient approximation methods [11]. In this paper, we propose a novel class of derivative-free discrete-time optimization algorithms. The key characteristic of the proposed algorithms is the use of deterministic non-commutative maps to evaluate the objective function at certain points such that a gradient descent step is approximated. In comparison to [8, 9, 10, 11] our proposed method is not a randomized or stochastic algorithm. Interestingly and to the best of our knowledge, the use of such a deterministic non-commutative scheme in discrete-time optimization algorithms has not been studied so far in the optimization literature.

In the geometric control theory literature, however, the effect of non-commutative vector fields is of fundamental importance [12] and has been used in control design since decades. Recently, non-commutative vector fields have also been used in the design of continuous-time optimization algorithms. For example, Lie brackets, i.e., the infinitesimal characterization of non-commutativity between two vector fields, have been used to approximate gradients and to design extremum seeking algorithms for unconstrained and constrained optimization and adaptive control problems, see, e.g., [13, 14, 15, 16, 17]. Furthermore, in [18], Lie brackets have been exploited to tackle the problem of distributed optimization over directed graphs.

The proposed class of discrete-time derivative-free algorithms are inspired by these Lie bracket based continuous-time, explorative optimization methods, but our approach is not simply a discretization of continuous-time methods. For example, utilizing non-commutative maps based on Euler-integration steps lead to approximation results which are different from continuous-time Lie bracket approximation results (see, e.g., [19]), yet we show how they can be used for gradient approximation. Indeed, as shown in the paper, suitable integration schemes need to be employed in order to approximate the results known from continuous-time methods. One advantage which arises from utilizing non-commutative maps in gradient approximation is the robustness with respect to noisy ob-

---

*   This article is a slightly extended version of [1] with an extra Figure 6.

jective functions as we will see in simulations. This is not observed if finite differences are used for gradient approximations, like it was done in [20], where extremum seeking and a version of [11] is combined. Overall, the contributions of this work are as follows: Firstly, utilizing non-commutative maps, we develop novel discrete-time gradient approximation schemes and corresponding derivative-free optimization algorithms. Secondly, we perform a convergence analysis of the proposed algorithms. Finally, we study the proposed algorithm in different simulation examples. The sequel of the paper is structured as follows: In the next section we state the problem setup and give the main idea how the discrete-time derivative-free optimization algorithms are derived. Section III contains the main lemmas and theorems providing the gradient approximation and the asymptotic convergence behavior of the algorithms. In Section IV we analyze the algorithm in various simulation examples. Finally, we give a summary and outlook of further work.

Notation. $\mathbb{N}_0$ denote the natural numbers including zero, i.e., $\{0, 1, 2, \ldots\}$. C^n with $n \in \mathbb{N}_0$ stands for the set of n-times differentiable functions. The norm $|\cdot|$ denotes the Euclidean norm. A compact set denoted by $\mathcal{S}_{x^*}^{\delta}$ with $\delta \in (0, \infty)$ and $x^* \in \mathbb{R}^n$ is defined as $\{x \in \mathbb{R}^n : |x - x^*| \leq \delta\}$. The gradient of a function $J \in C^2 : \mathbb{R}^n \to \mathbb{R}$ is represented by $\nabla_x J(x) := (\partial J/\partial x(x))^\top$ and its Hessian by $\nabla_x^2 J(x) := \partial^2 J/\partial x^2(x)$. A function $f(\epsilon) : \mathbb{R} \to \mathbb{R}^n$ is said to be of order $\mathcal{O}(\epsilon)$, if there exist $k, \bar{\epsilon} \in \mathbb{R}$ such that $|f(\epsilon)| \leq k\epsilon$, for all $\epsilon \in [0, \bar{\epsilon}]$. The operator $k \bmod n$ denotes the modulo operation.

2. MAIN IDEA AND ALGORITHM

The aim of this section is to motivate and to develop the basic steps of the proposed derivative-free optimization algorithms for unconstrained multidimensional optimization problems of the form

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

where $J : \mathbb{R}^n \to \mathbb{R}$ is the so-called objective or cost function.

2.1. Non-Commutative Vector Fields and Continuous-time Gradient Approximation

Consider the scalar continuous-time system

$$\dot{x}(t) = f_1(x(t))u_1(t) + f_2(x(t))u_2(t) \tag{2}$$

with state $x(t) \in \mathbb{R}$, vector fields $f_1, f_2 \in C^2 : \mathbb{R} \to \mathbb{R}$ and periodic inputs $u_i(t) = u_i(t + T) \in \mathbb{R}$, $i = 1, 2$, with

period $T \in \mathbb{R}$. It is very well-known that, if for example periodic rectangular-shaped input signals such as

$$u_1(t) = \begin{cases} 1, & t \in [0, h) \\ 0, & t \in [h, 2h) \\ -1, & t \in [2h, 3h) \\ 0, & t \in [3h, 4h) \end{cases}, \tag{3}$$

$$u_2(t) = \begin{cases} 0, & t \in [0, h) \\ 1, & t \in [h, 2h) \\ 0, & t \in [2h, 3h) \\ -1, & t \in [3h, 4h) \end{cases} \tag{4}$$

are applied to system (2), as shown in Figure 1(a), then, for $h > 0$ small, a second order Taylor expansion reveals

$$x(4h) - x(0) = h^2[f_1, f_2](x(0)) + \mathcal{O}(h^3), \tag{5}$$

where $[f_1, f_2] = \frac{\partial f_2}{\partial x} f_1 - \frac{\partial f_1}{\partial x} f_2$ is the Lie bracket between the vector fields f_1, f_2. Scaling the vector fields f_1, f_2 by $\frac{1}{\sqrt{h}}$ gives $x(4h) - x(0) = h[f_1, f_2](x(0)) + \mathcal{O}(h^{3/2})$. Then, by successively repeating (5) with new initial values $x(lh)$, $l = 1, 2, \ldots$ and taking the limit $h \to 0$ leads to the approximate (Lie bracket) system

$$\dot{\bar{x}}(t) = [f_1, f_2](\bar{x}(t)). \tag{6}$$

Regarding the (scalar) optimization problem (1), one desirable methodology to find an extremum, would be a simple continuous-time gradient descent equation $\dot{x} = -\nabla_x J(x)$. Interestingly, by defining
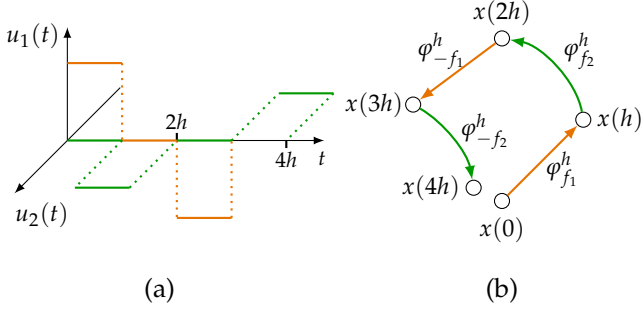
$$\mathfrak{f}_i(x) := f_i(J(x)), \tag{7}$$

$i = 1, 2$, and choosing $\mathfrak{f}_1, \mathfrak{f}_2 : \mathbb{R} \to \mathbb{R}$ as

$$\mathfrak{f}_1(x) = J(x), \qquad \mathfrak{f}_2(x) = 1, \quad \text{or} \tag{8}$$
$$\mathfrak{f}_1(x) = \sin(J(x)), \quad \mathfrak{f}_2(x) = \cos(J(x)), \tag{9}$$

we have $[\mathfrak{f}_1, \mathfrak{f}_2](x) = -\nabla_x J(x)$. Hence, by substitution f_i by $h^{-1/2}\mathfrak{f}_i$, $i = 1, 2$, in (5) and (2), respectively, we obtain by (2) a derivative-free approximation of a gradient flow. This observation has been utilized and generalized for the design and analysis of continuous-time extremum seeking algorithms, see [13, 15, 17] and references therein. Especially, [15] presents a broad class of generating vector fields $\mathfrak{f}_1, \mathfrak{f}_2$ for gradient approximation. Non-commutativity comes into play since Lie brackets naturally arise when studying the commutativity of flows. In particular, if we denote with $\varphi_{f_i}^t(x_0) : \mathbb{R}^n \to \mathbb{R}^n$ the flow map of the vector field $f_i : \mathbb{R}^n \to \mathbb{R}^n$ at time $t + t_0$ and initial condition $x(t_0) = x_0 \in \mathbb{R}^n$, then we can express equation (5) as

$$x(4h) = \left(\varphi_{-f_2}^h \circ \varphi_{-f_1}^h \circ \varphi_{f_2}^h \circ \varphi_{f_1}^h \right)(x_0), \tag{10}$$

Figure 1. Switching vector fields: (a) periodic inputs $u_1(t), u_2(t)$ depicted for one period $4h$; (b) non-commutative flow maps of vector fields $\pm f_1, \pm f_2$.

as shown in Figure 1(b). It is easy to see that $x(4h) = x(0)$ if and only if the flow maps $\varphi_{f_1}^h, \varphi_{f_2}^h$ commute, since the flow is a bijection with $(\varphi_{f_1}^h)^{-1} = \varphi_{-f_1}^h$ and hence $x(4h) = x(0)$ if and only if $(\varphi_{f_2}^h \circ \varphi_{f_1}^h)(x(0)) = (\varphi_{f_1}^h \circ \varphi_{f_2}^h)(x(0))$. Further, it can be shown that the flow maps commute if and only if the Lie bracket between f_1, f_2 vanishes [21], hence the Lie bracket is an infinitesimal measure for the commutativity of vector fields.

2.2. Non-Commutative Maps

In principle, one could just numerically integrate (2) with appropriate vector fields and inputs in order to get a derivative-free discrete-time algorithm. However, rectangular-shaped inputs as shown in Figure 1(a) often lead to bad numerical behavior in combination with standard numerical integration schemes. Hence, in continuous-time algorithms [13, 15], the rectangular-shaped inputs are often replaced by sinusoidal inputs in order to avoid these problems. A key idea of the proposed discrete-time optimization algorithm is to incorporate non-commutative maps more directly into a tailored integration scheme, which approximates a gradient descent step with the help of composite maps of the form (10). Hereby, two challenges arise: i) How to efficiently obtain the maps? ii) What are suitable discrete-time input functions in an *n*-dimensional optimization problem?

i) Maps. We construct the composition of flow maps by suitable numerical integration methods, where we consider two approaches. Firstly, we utilize the *Euler*-integration method. It does not lead to a Lie bracket approximation as in (5), since the Euler-integration method is a first order method while Lie brackets are second order effects. However, we show that the Euler-integration method still can be used for gradient approximation. 1Secondly, we utilize the so-called *Heun*-integration method, also known as the trapezoidal-integration method, which preserve properties like (5) (see Lemma 2 in Section 3).

Accordingly, for an ODE $\dot{x} = g(x)$ with $x \in \mathbb{R}^n$ and vector field $g : \mathbb{R}^n \to \mathbb{R}^n$ we use the Euler- and Heun-integration steps given by

$$E_g^h(x_k) := x_k + hg(x_k), \quad \text{and} \tag{11}$$

$$H_g^h(x_k) := x_k + \frac{h}{2}\left(g(x_k) + g(x_k + hg(x_k))\right), \tag{12}$$

as an approximation of the flow map φ_g^h.

ii) Inputs. As depicted in Figure 1(a), we consider periodic rectangular-shaped inputs. These inputs can simply be interpreted as switching $\mathfrak{f}_1, \mathfrak{f}_2 : \mathbb{R}^n \to \mathbb{R}$ at every time step (interval) k (h). Hence, a switched vector field is defined as

$$g_k(x) = \begin{cases} \mathfrak{f}_1(x)e_l & \text{if } k \bmod 4 = 0 \\ \mathfrak{f}_2(x)e_l & \text{if } k \bmod 4 = 1 \\ -\mathfrak{f}_1(x)e_l & \text{if } k \bmod 4 = 2 \\ -\mathfrak{f}_2(x)e_l & \text{if } k \bmod 4 = 3 \end{cases}, \tag{13}$$

where $l = k/4 \bmod n + 1$ with e_l the l-th unit vector in $\mathbb{R}^n$. Functions $\mathfrak{f}_1, \mathfrak{f}_2 : \mathbb{R}^n \to \mathbb{R}$, as for example in (8) and (9) are components of vector fields $g_k(x) = \mathfrak{f}_i(x)e_l$, but for the sake of convenience we sometimes call $\mathfrak{f}_1, \mathfrak{f}_2$ vector fields and use the notation $[\mathfrak{f}_1, \mathfrak{f}_2] = \nabla_x \mathfrak{f}_2 \mathfrak{f}_1 - \nabla_x \mathfrak{f}_1 \mathfrak{f}_2$. Regarding (13), one can think of executing four steps in every coordinate direction successively, hence the switching rule represents a coordinate wise approximation scheme. To keep the main idea clear, we consider only this single switching function and refer to further work where we will investigate more complex switching policies.

2.3. Derivative-free Optimization Algorithms

Combining the integration steps (11) and (12), respectively, with the sequential switching rule (13) and scaling the vector fields $\mathfrak{f}_1, \mathfrak{f}_2$ with $h^{-1/2}$, as discussed in the previous section, results in our discrete-time derivative-free optimization algorithms:

$$x_{k+1} = M_{g_k}^{\sqrt{h}}(x_k) = \begin{cases} E_{g_k}^{\sqrt{h}}(x_k) & \text{(14a)} \\ H_{g_k}^{\sqrt{h}}(x_k). & \text{(14b)} \end{cases}$$

As shown in the next section, every four steps (scalar case)

$$x_{k+4} = (M_{g_{k+3}}^{\sqrt{h}} \circ M_{g_{k+2}}^{\sqrt{h}} \circ M_{g_{k+1}}^{\sqrt{h}} \circ M_{g_k}^{\sqrt{h}})(x_k) \tag{15}$$

a gradient descent step under suitable conditions on the vector fields $\mathfrak{f}_1, \mathfrak{f}_2$ is approximated. Summarizing, two algorithms presented in (14) and also in Algorithm 1, were derived by utilizing the effect of non-commutative maps, introduced by a sequential switching rule of vector fields and proper integration methods. In the next section we analyze both schemes by showing the gradient approximation property and prove asymptotic convergence to a neighborhood of an extremum.

3

Algorithm 1 Derivative-free optimization algorithm with non-commutative maps (14b)

1: **Required:** $x_0, h, \mathfrak{f}_1, \mathfrak{f}_2$, stop criterion
2: **Init:** $k = 0$
3: **while** stop criterion is not fulfilled **do**
4:      $l \leftarrow n \bmod k/4 + 1$
5:      $e_l \leftarrow [0_l, 1, 0_{n-1-l}]^\top$
6:      $g_k(x) \leftarrow \begin{cases} \mathfrak{f}_1(x)e_l & \text{if } k \bmod 4 = 0 \\ \mathfrak{f}_2(x)e_l & \text{if } k \bmod 4 = 1 \\ -\mathfrak{f}_1(x)e_l & \text{if } k \bmod 4 = 2 \\ -\mathfrak{f}_2(x)e_l & \text{if } k \bmod 4 = 3 \end{cases}$
7:      $c_1 \leftarrow g_k(x_k)$
8:      $c_2 \leftarrow g_k(x_k + \sqrt{h}c_1)$
9:      $x_{k+1} \leftarrow x_k + \frac{\sqrt{h}}{2}(c_1 + c_2)$
10:      $k \leftarrow k + 1$
11: **end while**
12: **return** $[x_0, x_1, \ldots]$

---

3. MAIN RESULTS

In the sequel, basic convergence and approximation properties of both discrete-time derivative-free optimization algorithms presented in (14) are analyzed. Note, convergence rate results of derivative-free algorithms are almost absent in literature [8]. The algorithms are applicable for broad classes of objective functions J, including non-differentiable objective functions, given a choice of proper algorithmic parameters $x_0, h, \mathfrak{f}_1, \mathfrak{f}_2$, as we observe in simulations, since only function evaluations of J are necessary. However, for the approximation and convergence analysis we assume:

Assumption 1. The objective function $J : \mathbb{R}^n \to \mathbb{R}$ in (1) and the vector fields $\mathfrak{f}_1, \mathfrak{f}_2 : \mathbb{R}^n \to \mathbb{R}$ in (7) are class C^2 functions. ▲

First, we state two lemmas, which show the approximated optimization direction after $4n$ steps, induced by the non-commutative maps of our proposed algorithms (14):

Lemma 1. Let Assumption 1 hold and consider algorithm (14a). Then the evolution of $x_k \in \mathbb{R}^n$ at step $k + 4n$ is given by

$$x_{k+4n} = x_k + h\left([\mathfrak{f}_1, \mathfrak{f}_2](x_k) - \nabla_x \mathfrak{f}_1(x_k)\mathfrak{f}_1(x_k)\right.$$
$$\left. - \nabla_x \mathfrak{f}_2(x_k)\mathfrak{f}_2(x_k)\right) + \mathcal{O}(h^{3/2}). \quad (16)$$

for all $k \in \mathbb{N}_0$ ▲

Lemma 2. Let Assumption 1 hold and consider algorithm (14b). Then the evolution of $x_k \in \mathbb{R}^n$ at step $k + 4n$ is given by

$$x_{k+4n} = x_k + h[\mathfrak{f}_1, \mathfrak{f}_2](x_k) + \mathcal{O}(h^{3/2}). \quad (17)$$

for all $k \in \mathbb{N}_0$ ▲

The proofs of Lemma 1 and Lemma 2 are given in Appendix A.1 and Appendix A.2, respectively. We observe that the Heun-integration steps $H_{g_k}^{\sqrt{h}}(x_k)$ preserve the Lie bracket approximation property, as shown in (17). However, as mention in Section 1, successively applying the Euler-integration steps $E_{g_k}^{\sqrt{h}}(x_k)$ reveals the Lie bracket but additional terms of the same order $\mathcal{O}(h)$ are present, as shown in (16). Interestingly, choosing proper vector fields $\mathfrak{f}_1, \mathfrak{f}_2$, *both* algorithms approximate a gradient descent method:

Theorem 1. Let Assumption 1 hold and consider the algorithms in (14) with the pair of vector fields (9). Then the evolution of $x_k \in \mathbb{R}^n$ at step $k + 4n$ is given by

$$x_{k+4n} = x_k - h\nabla_x J(x_k) + \mathcal{O}(h^{3/2}) \quad (18)$$

for all $k \in \mathbb{N}_0$ ▲

Proof. This result follows directly from Lemma 1 and Lemma 2 and simple calculations. □

Remark 1. The pair of vector fields (8) with algorithm (14a), as commonly employed in continuous-time methods [13], yield an approximation scheme of the form

$$x_{k+4n} = x_k - h\nabla_x J(x_k)(1 + J(x_k)) + \mathcal{O}(h^{3/2}). \quad (19)$$

Hence, this scheme is not a gradient descent step but it is still suitable for optimization if, for example, the objective function is positive semi-definite. ▲

For the convergence analysis of (14), we impose the following additional assumptions on the cost function $J : \mathbb{R}^n \to \mathbb{R}$ in (1):

Assumption 2. $J(x)$ is radially unbounded and there exists a $x^* \in \mathbb{R}^n$ such that $\nabla_x J(x)^\top (x - x^*) > 0$ for all $x \in \mathbb{R}^n \setminus \{x^*\}$. ▲

Remark 2. Assumption A2 implies that x^* is the unique global minimizer and there exists no other local minimum. ▲

The following theorem shows asymptotic convergence to a neighborhood of the minimum x^* of $J(x)$:
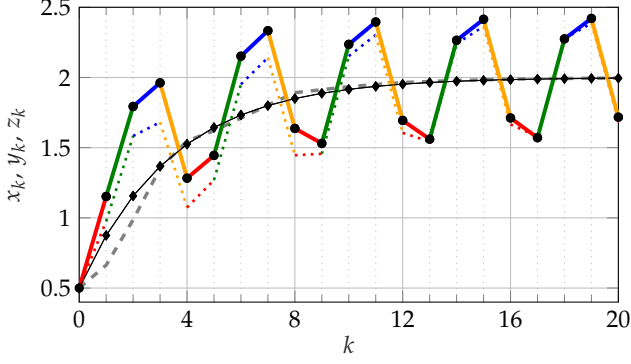
Theorem 2. Let Assumption 1 and Assumption 2 hold. Consider the discrete-time derivative-free optimization algorithm (14) and let the vector fields $\mathfrak{f}_1, \mathfrak{f}_2$ satisfy for (14a) and (14b) respectively:

$$-\nabla_x J(x) = \left([\mathfrak{f}_1, \mathfrak{f}_2] - \nabla_x \mathfrak{f}_1 \mathfrak{f}_1 - \nabla_x \mathfrak{f}_2 \mathfrak{f}_2\right)(x), \quad (20)$$
$$-\nabla_x J(x) = [\mathfrak{f}_1, \mathfrak{f}_2](x). \quad (21)$$

Then for all $0 < \delta_2 < \delta_1$ there exists an $h^* > 0$ such that for all $h \in (0, h^*)$ and all initial conditions $x_0 \in \mathcal{S}_{x^*}^{\delta_1}$, x_k converges to $\mathcal{S}_{x^*}^{\delta_2}$. ▲

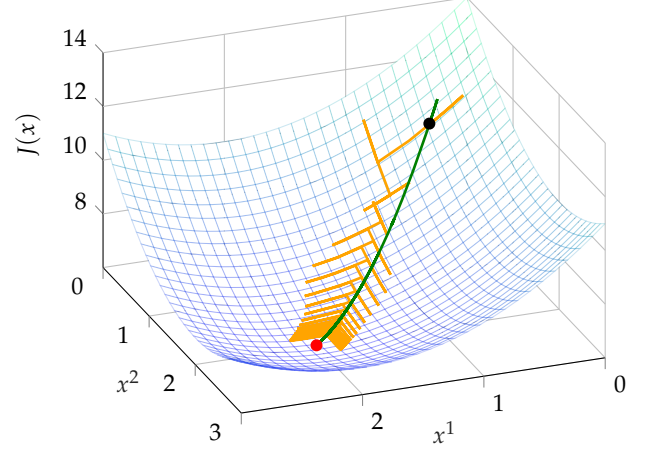**Figure 2.** Convergence of x_k in (14) for setup $\mathcal{P}$ with $h = 0.5$ to a neighborhood of $x^* = 2$. The flow maps as in (15) for $M = E$ are highlighted with $E_{\mathfrak{f}_1}^{\sqrt{h}}$ (—), $E_{\mathfrak{f}_2}^{\sqrt{h}}$ (—), $E_{-\mathfrak{f}_1}^{\sqrt{h}}$ (—), $E_{-\mathfrak{f}_2}^{\sqrt{h}}$ (—) and the same color scheme (dotted) for (15), $M = H$. This is compared with the trajectory resulting from a gradient descent algorithm with the exact gradient (—). By filtering x_k, a good asymptotic convergence behavior of y_k to x^* can be observed (- -).
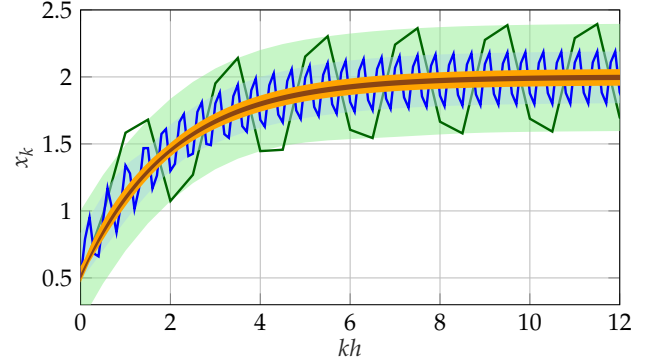
The proof of Theorem 2 is given in Appendix A.3. Note, that Theorem 2 states the semi-global uniform practical asymptotic stability property [13] of $\mathcal{S}_{x^*}^{\delta_2}$ of (14) under the given assumptions.

4. SIMULATIONS

In this section we study our discrete-time derivative-free optimization algorithms (14) in various simulation examples and provide some tuning rules. To this end, we consider the cost function $J(x) = (x - 2)^2 + 6$. If not otherwise specified we use the vector fields (9) and initialize the algorithms with $x_0 = x(0) = 0.5$. For the sake of convenience we define this setup as $\mathcal{P} := (J, x_0, \mathfrak{f}_1, \mathfrak{f}_2)$. Note that the results obtained with such a quadratic cost function also provide insight into the results that would be obtained with an arbitrary function of class C^2 near a minimum with positive definite Hessian. We start by analyzing (14), applied to the basic setup $\mathcal{P}$, which is depicted for (a large) step size $h = 0.5$ in Figure 2, where the switching of the vector fields $\mathfrak{f}_1, \mathfrak{f}_2$ is highlighted. As it can be observed, the trajectories of x_k are converging into a neighborhood of x^*, which is of order $\mathcal{O}(\sqrt{h})$ (see proof of Theorem 2). To eliminate the steady-state oscillatory behavior, we propose the filter $y_{k+1} = y_k + \frac{1}{4n}\sum_{i=k-4n}^{k-1} x_{i+1} - x_i$ where $y_0 = x_0$ and $x_l = x_0$, $l < 0$, which shows a good asymptotic convergence behavior of the algorithms (14) to the extremum x^* as illustrated in Figure 2. Obviously, y_k is a windowed filter with length $4n$, which is not fed back into the algorithm. This behavior can also be observed in a multidimensional optimization problem as depicted in Figure 3.

Figure 3. A two-dimensional optimization problem with $J(x) = (x^1 - 2)^2 + (x^2 - 2)^2 + 6$. Trajectories x_k (—) of (14a) and y_k (—) converge to the extremum $x^* = [2, 2]$ (•) initialized with $x_0 = y_0 = [0.5, 0.5]$ (•).
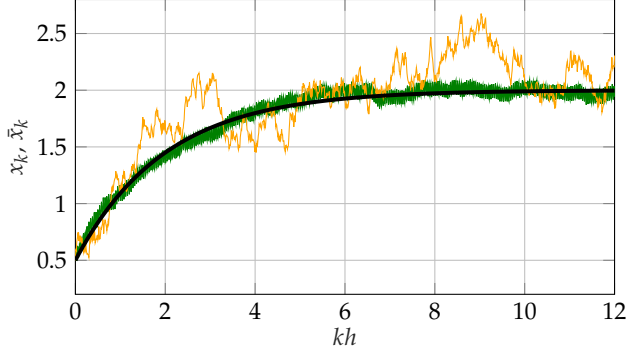

Figure 4. Choosing smaller step sizes for algorithm (14a) or (14b) yield to a smaller neighborhood of convergence. Trajectories of x_k for setup $\mathcal{P}$ with step sizes $h = 0.5$ (—), $h = 0.1$ (—), $h = 0.01$ (—), $h = 0.001$ (—).

As known from continuous-time methods, see for example [13], increasing the (switching) frequency, and hence reducing the step size h, leads to a convergence into a smaller neighborhood around x^* as depicted for different h's in Figure 4. Clearly, for decreasing the step size, more iterations are needed, hence more function evaluations.

As mentioned in Section 1, an advantage of the proposed class of algorithms is robustness with respect to noisy objective functions. Therefore, consider the setup $\mathcal{P}$, but with a noisy cost function $\bar{J} : \mathbb{R} \to \mathbb{R}$ such that

$$\bar{J}(x_k) \sim \mathcal{N}(J(x_k), 0.04), \qquad (22)$$

where $\mathcal{N}(J(x_k), 0.04)$ is a normal distribution with mean $J(x_k)$ and standard derivation $\sqrt{0.04} = 0.2$. In Figure 5, algorithm (14b) is compared with the well known difference quotient method to approximate the gradient

Figure 5. Study of noisy objective function: trajectory x_k of algorithm (14b) (━), trajectory $\bar{x}_k$ of gradient descent with gradient approximation via difference quotient (━), trajectory of gradient descent with exact gradient (━).

$D(J(\bar{x})) := h^{-1}(J(\bar{x}_k) - J(\bar{x}_k + h))$ used in a gradient descent algorithm $\bar{x}_{k+1} = \bar{x}_k - hD(J(\bar{x}))$ with step size $h = 0.01$ and initial condition $\bar{x}_0 = x_0$. Apparently, approximating the gradient via the difference quotient method is much more sensitive regarding noisy objectives as algorithms (14). This is reasoned by the averaging behavior of the proposed algorithms, implied by utilizing non-commutative maps.

Concluding, we illustrated both algorithms (14) on simple examples and we introduced a filtering method. Secondly, we showed the impact of decreasing h and the performance under noisy measurements. Regarding the convergence behavior, algorithm (14a) and (14b) show similar convergence behavior for the considered vector fields (9) and switching function (13). Nevertheless, algorithm (14b) needs two times more function evaluations than (14a).

5. CONCLUSION

In this work we introduced a methodology for the design of novel discrete-time derivative-free optimization algorithms. Based on non-commutative maps, designed by two different numerical integration methods and a sequential switching rule, it was shown how to approximate a gradient descent step. Furthermore, practical convergence and stability towards an extremum of the objective function was proven. The algorithms are illustrated using numerical examples.

The introduced class of algorithms raise several new research questions, which will be part of future work. For example, what are more efficient switching rules in multidimensional optimization problems compared to the introduced coordinate wise switching rule. Moreover, it is interesting to characterize a general class of non-commutative maps which show gradient approximation

properties or to study convergence rates or accelerated gradient schemes.

References

[1] J. Feiling, A. Zeller, and C. Ebenbauer, "Derivative-free optimization algorithms based on non-commutative maps," *IEEE Control Systems Letters, submitted*, 2018.

[2] K. B. Ariyur and M. Krstic, *Real-time optimization by extremum-seeking control*. John Wiley & Sons, 2003.

[3] M. Guay and T. Zhang, "Adaptive extremum seeking control of nonlinear dynamic systems with parametric uncertainties," *Automatica*, vol. 39, no. 7, pp. 1283–1293, 2003.

[4] Y. Tan, W. Moase, C. Manzie, D. Nešić, and I. Mareels, "Extremum seeking from 1922 to 2010," in *Control Conference (CCC), 2010 29th Chinese*. IEEE, 2010, pp. 14–26.

[5] K. T. Atta, A. Johansson, and T. Gustafsson, "Extremum seeking control based on phasor estimation," *Systems & Control Letters*, vol. 85, pp. 37–45, 2015.

[6] J. Poveda, M. Benosman, and A. Teel, "Distributed extremum seeking in multi-agent systems with arbitrary switching graphs," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 735–740, 2017.

[7] J. Poveda and A. Teel, "A robust event-triggered approach for fast sampled-data extremization and learning," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 4949–4964, 2017.

[8] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Foundations of Computational Mathematics*, vol. 17, no. 2, pp. 527–566, 2017.

[9] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono, "Optimal rates for zero-order convex optimization: The power of two function evaluations," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2788–2806, 2015.

[10] V. Torczon, "On the convergence of pattern search algorithms," *SIAM Journal on optimization*, vol. 7, no. 1, pp. 1–25, 1997.

[11] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 332–341, 1992.

[12] R. Brockett, "The early days of geometric nonlinear control," *Automatica*, vol. 50, no. 9, pp. 2203–2224, 2014.

[13] H.-B. Dürr, M. S. Stanković, C. Ebenbauer, and K. H. Johansson, "Lie bracket approximation of extremum seeking systems," *Automatica*, vol. 49, no. 6, pp. 1538–1552, 2013.

[14] H.-B. Dürr, M. S. Stanković, K. H. Johansson, and C. Ebenbauer, "Extremum seeking on submanifolds in the Euclidian space," *Automatica*, vol. 50, no. 10, pp. 2591–2596, 2014.

[15] V. Grushkovskaya, A. Zuyev, and C. Ebenbauer, "On a class of generating vector fields for the extremum seeking problem: Lie bracket approximation and stability properties," *Automatica*, vol. 94, pp. 151 – 160, 2018.

[16] H.-B. Dürr, C. Zeng, and C. Ebenbauer, "Saddle point seeking for convex optimization problems," *IFAC Proceedings Volumes*, vol. 46, no. 23, pp. 540–545, 2013.

[17] A. Scheinker and M. Krstić, "Extremum seeking with bounded update rates," *Systems & Control Letters*, vol. 63, pp. 25–31, 2014.

[18] S. Michalowsky, B. Gharesifard, and C. Ebenbauer, "A Lie bracket approximation approach to distributed optimization over directed graphs," *Automatica*, 2018 (provisionally accepted).

[19] C. Altafini, "Nonintegrable discrete-time driftless control systems: geometric phases beyond the area rule," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 4692–4697.

[20] D. Popovic, M. Jankovic, S. Magner, and A. R. Teel, "Extremum seeking methods for optimization of variable cam timing engine operation," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 3, pp. 398–407, 2006.

[21] H. Nijmeijer and A. Van der Schaft, *Nonlinear dynamical control systems*. Springer, 1990, vol. 175.

A. APPENDIX

A.1. Proof of Lemma 1

We consider the derivative-free optimization algorithm (14a) and assume without loss of generality $k \bmod 4 = 0$ and $k/4 \bmod n = 0$. Due to our assumption on k, $g_k(x_k) = \mathfrak{f}_1(x_k)e_1$. Since, the switching policy (13) is based on a coordinate-wise implementation, it is sufficient to analyze the evolution of x_k in only one dimension. Hence, it holds

$$x_{k+1} = x_k + \sqrt{h}\mathfrak{f}_1(x_k)e_1. \tag{23}$$

For the sake of readability in the sequel we define $\mathfrak{f}_i^k := \mathfrak{f}_i(x_k)e_1$ and $F_i^k := (\frac{\partial \mathfrak{f}_i}{\partial x}(x_k))^\top e_1$ and neglect e_1. Therefore, the next step is given by

$$x_{k+2} = x_k + \sqrt{h}(\mathfrak{f}_1^k + \mathfrak{f}_2(x_k + \sqrt{h}\mathfrak{f}_1^k)). \tag{24}$$

Performing a Taylor expansion on $\mathfrak{f}_2(x_k + \sqrt{h}\mathfrak{f}_1^k)$ in (24) reveals

$$x_{k+2} = x_k + \sqrt{h}(\mathfrak{f}_1^k + \mathfrak{f}_2^k) + hF_2^k\mathfrak{f}_1^k + \mathcal{O}(h^{3/2}), \tag{25}$$

where all higher order terms are pushed in $\mathcal{O}(h^{3/2})$. Repeating this procedure, as presented above, x_k evolutes as follows:

$$x_{k+3} = x_k + \sqrt{h}\mathfrak{f}_2^k + h(F_2^k\mathfrak{f}_1^k - F_1^k\mathfrak{f}_1^k - F_1^k\mathfrak{f}_2^k) + \mathcal{O}(h^{3/2}), \tag{26}$$

$$x_{k+4} = x_k + h(F_2^k\mathfrak{f}_1^k - F_1^k\mathfrak{f}_1^k - F_1^k\mathfrak{f}_2^k - F_2^k\mathfrak{f}_2^k) + \mathcal{O}(h^{3/2}). \tag{27}$$

Eventually, repeating the same procedure for dimensions $\{2, 3, \ldots, n\}$ delivers the same expression (27) for each dimension, hence (16) can be directly concluded. □

A.2. Proof of Lemma 2

As in the proof of Lemma 1 we repeat the same procedure and only state the main equations. For notations and assumptions (w.l.o.g) reread Appendix A.1. Therefore it holds

$$x_{k+1} = x_k + \frac{\sqrt{h}}{2}\left(\mathfrak{f}_1^k + \mathfrak{f}_1(x_k + \sqrt{h}\mathfrak{f}_1^k)\right). \tag{28}$$

Performing a Taylor expansion on $\mathfrak{f}_1(x_k + \sqrt{h}\mathfrak{f}_1^k)$ in (28) reveals

$$x_{k+1} = x_k + \sqrt{h}\mathfrak{f}_1^k + \frac{h}{2}F_1^k\mathfrak{f}_1^k + \mathcal{O}(h^{3/2}). \tag{29}$$

Repeat the procedure, as presented above, x_k evolutes as follows:

$$x_{k+2} = x_k + \sqrt{h}(\mathfrak{f}_1^k + \mathfrak{f}_2^k) + \frac{h}{2}\left(F_1^k\mathfrak{f}_1^k + F_2^k(2\mathfrak{f}_1^k + \mathfrak{f}_2^k)\right) + \mathcal{O}(h^{3/2}), \tag{30}$$

$$x_{k+3} = x_k + \sqrt{h}\mathfrak{f}_2^k + \mathcal{O}(h^{3/2}) + \frac{h}{2}(-2F_1^k\mathfrak{f}_2^k + 2F_2^k\mathfrak{f}_1^k + F_2^k\mathfrak{f}_2^k), \tag{31}$$

$$x_{k+4} = x_k + h(F_2^k\mathfrak{f}_1^k - F_1^k\mathfrak{f}_2^k) + \mathcal{O}(h^{3/2}). \tag{32}$$

Eventually, repeating the same procedure for dimensions $\{2, 3, \ldots, n\}$ delivers the same expression (32) for each dimension, hence (17) can be directly concluded. □

A.3. Proof of Theorem 2

Let $0 \leq \delta_3 \leq \delta_2 \leq \delta_1 \leq \delta_0$. The proof is separated in: 1) Define a Lyapunov-like function $V(x)$, such that $V(x_{k+4n}) - V(x_k) < 0$ for $\mathcal{S}_{x^*}^{\delta_0} \setminus \mathcal{S}_{x^*}^{\delta_3}$, 2) Practical invariance of $\mathcal{S}_{x^*}^{\delta_1}$, 3) Convergence of x_k to $\mathcal{S}_{x^*}^{\delta_2}$ with $x_0 \in \mathcal{S}_{x^*}^{\delta_1}$.

1) Consider the algorithms (14) and their evolution as given in (16) and (17) under conditions (20), (21) on vector fields $\mathfrak{f}_1, \mathfrak{f}_2$ as stated in Theorem 2. W.l.o.g., let the Lagrange remainder of the Taylor expansion (16) and (17), respectively, be of the form $h^{3/2} R_k = \mathcal{O}(h^{3/2})$, where $R_k \in \mathbb{R}^n$ depends on the vector fields $\mathfrak{f}_1, \mathfrak{f}_2$ and their Jacobians and Hessians. Hence, (14) is given by

$$x_{k+4n} = x_k - h \nabla_x J(x_k) + h^{3/2} R_k. \tag{33}$$

Consider the Lyapunov-like candidate function $V(x_k) = J(x_k) - J(x^*)$, then it holds for all $k \geq 0$,

$$V(x_{k+4n}) - V(x_k) =$$
$$= J\left(x_k - h\nabla_x J(x_k) + h^{3/2} R_k\right) - J(x_k)$$
$$= -h \nabla_x J(x_k)^\top \nabla_x J(x_k)$$
$$+ h^{3/2} \nabla_x J(x_k)^\top R_k + \frac{h^2}{2} \gamma^\top H_k \gamma, \tag{34}$$

where the last equality is gained by a second order Taylor expansion at x_k, with the Lagrange reminder $\frac{h^2}{2}\gamma^\top H_k \gamma$, where $\gamma = \nabla_x J(x_k) + h^{3/2} R_k$ and $H_k \in \mathbb{R}^{n \times n}$. R_k and H_k depend on J, $\mathfrak{f}_1, \mathfrak{f}_2$ and their Jacobians and Hessians, Hence, R_k, H_k, and $\nabla_x J(x)$ are bounded on every compact set $\mathcal{S}_{x^*}^{\delta_0}$, due to Assumption 1. Therefore, for every δ_0 there exists a $0 \leq R_{\delta_0} \in \mathbb{R}$ such that for all $x_k \in \mathcal{S}_{x^*}^{\delta_0}$ hold

$$V(x_{k+4n}) - V(x_k) \leq -h|\nabla_x J(x_k)|^2 + h^{3/2} R_{\delta_0}. \tag{35}$$

Then under Assumption 2, there exists a $h_1 > 0$, such that for all $h \in (0, h_1)$ hold

$$V(x_{k+4n}) - V(x_k) \leq -\epsilon \quad \text{on } \mathcal{L} = \mathcal{L}_0 \setminus \mathcal{L}_3, \tag{36}$$

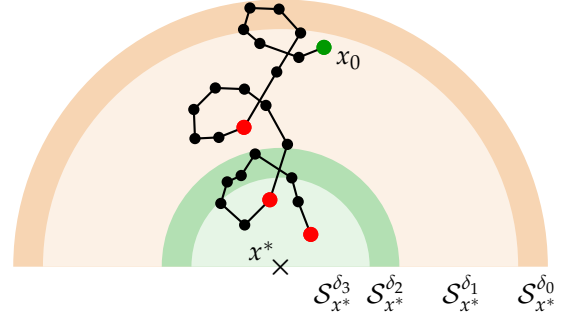with $\epsilon > 0$, $\mathcal{S}_{x^*}^{\delta_0} \setminus \mathcal{S}_{x^*}^{\delta_3} \subseteq \mathcal{L}$ and sub-level sets of $V(x)$, $\mathcal{L}_0 \supseteq \mathcal{S}_{x^*}^{\delta_0}$ and $\mathcal{L}_3 \subseteq \mathcal{S}_{x^*}^{\delta_3}$.

2) By Assumption 1 and the first order Taylor expansions, it holds that for all $k \geq 0$, $0 < l < 4n$, there exist an upper bound on the Lagrange reminders $0 \leq \tilde{R}_{k+l} \in \mathbb{R}$ in $\mathcal{S}_{x^*}^{\delta_0}$ such that

$$V(x_{k+l}) = V(x_k + \sqrt{h} \sum_{i=k}^{k+l-1} g_i(x_i))$$
$$\leq V(x_k) + \sqrt{h} \tilde{R}_{k+l} \tag{37}$$

Hence, in combination with 1) there exist a $h_2 > 0$ such that for all $h \in (0, h_2)$, $x_0 \in \mathcal{S}_{x^*}^{\delta_1} \Rightarrow x_k \in \mathcal{S}_{x^*}^{\delta_0}$ for all $k \geq 0$.

3) Due to (36) with $h_1 > 0$, it holds that for every initial value $x_0 \in \mathcal{S}_{x^*}^{\delta_1}$, there exist a maximum number of iterations $\bar{k} > 0$, such that for all $h \in (0, \min\{h_1, h_2\})$, it holds $x_{\bar{k}} \in \mathcal{S}_{x^*}^{\delta_3}$. Furthermore, as in 2), one can show that there exists a $h_3 > 0$, such that for all $h \in (0, h_3)$ and $x_0 \in \mathcal{S}_{x^*}^{\delta_3}, x_k \in \mathcal{S}_{x^*}^{\delta_2}$ for all $k \geq 0$. Thus, for all $h \in (0, \min\{h_1, h_2, h_3\})$, if $x_0 \in \mathcal{S}_{x^*}^{\delta_1}$, then x_k converges to $\mathcal{S}_{x^*}^{\delta_2}$. Figure 6 illustrates the proof and the introduced compact sets $\mathcal{S}_{x^*}$. $\square$

Figure 6. Convergence of $x_k \in \mathbb{R}^2$ (•) with initial value x_0 (•) $\in \mathcal{S}_{x^*}^{\delta_1} \subseteq \mathcal{S}_{x^*}^{\delta_0}$ to $\mathcal{S}_{x^*}^{\delta_2}$. For all $k > \tilde{k}$, x_k stays in $\mathcal{S}_{x^*}^{\delta_2}$. Note, every $4n$-th step is marked with •.