A gradient descent algorithm built on approximate discrete gradients

Alessio Moreschini, Mattia Mattioni, Salvatore Monaco, Dorothée Normand-Cyrot

A gradient descent algorithm built on approximate discrete gradients

Alessio Moreschini[1], Mattia Mattioni[2], Salvatore Monaco[2], and Dorothée Normand-Cyrot[3]

***Abstract*—We propose an optimization method obtained by the approximation of a novel discretization approach for gradient dynamics recently proposed by the authors. It is shown that the proposed algorithm ensures convergence for all amplitudes of the step size, contrarily to classical implementations.**

***Index Terms*—Optimization; Nonlinear Systems; Modeling, Simulation and CAD Tools.**

I. Introduction

Over past years many researchers have been inspired by digital-based modeling to solve intelligent tasks through machines, computers, and microprocessors, such as noise cancellation, image reconstruction, financial forecasting, iterative learning [1]–[3]. Those problems are generally formulated in terms of optimization of a certain objective function to be minimized in a digital (discrete) environment. Recent developments and achievements in this optimization direction have brought to a substantial number methods intended to achieve a satisfactory solution.

The Gradient descent (GD) method is by far the most known optimization algorithm and used for convex optimization in many research areas. It is a relatively simple iterative method which exploit the gradient of the objective function to determine the suitable direction of searching of its minimum. More precisely, given an objective function, the search for an isolated minimum relies upon the definition of a dynamics, the so-called gradient dynamics, possessing an asymptotically stable equilibrium at the local extremum of the objective function. In practice, such dynamics are implemented digitally via a discrete-time algorithm. In its standard implementation, the search for an isolated minimum consists of an iterative procedure computing the sampled evolution of the gradient dynamics over time intervals of amplitude δ. In this setting, the GD provides the search direction for the next point with the step size δ determining how far we go in that particular direction. In this formulation, as also the intuition suggests, convergence is ensured when the step size (that is the frequency of the update) is chosen small enough [4], [5].

[1]Department of Electrical and Electronic Engineering, Imperial College London, London, SW7 2AZ, UK a.moreschini@imperial.ac.uk

[2]Dipartimento di Ingegneria Informatica, Automatica e Gestionale *A. Ruberti* (Sapienza University of Rome); Via Ariosto 25, 00185 Rome, Italy. {mattia.mattioni,salvatore.monaco}@uniroma1.it

[3]Laboratoire de Signaux et Systèmes (L2S, CNRS); 3, Rue Joliot Curie, 91192, Gif-sur-Yvette, France dorothee.normand-cyrot@centralesupelec.fr

Another well-known optimization algorithm is the Newton method. The idea behind the Newton method is that to find the isolated minimum of a given objective function, we approximate the objective function as a second-order truncation of its Taylor series, and then compute the minimum of that extension, which means taking its first derivative and setting it equal to zero, [6], [7]. The result of this procedure would be an update rule which iteratively tends to converge towards the isolated minimum of the objective function. However, a known drawback of the Newton method is that it includes the inverse of the Hessian of the objective function, which might lead to instability as the Hessian tends to zero. Accordingly, several modifications have been proposed with respect to its standard formulation depending on the specific context to cope with the arising criticalities, such as Gauss-Newton (see [8, Eq. 8]) and Regularized Newton (see [9, Eq. 31]). Yet, to the best of the Authors' knowledge, no solution ensuring convergence independently of the step size is available in the literature. Moreover, such a requirement is essential in several contexts such as deep learning [10], iterative learning control [11], [12], inverse map [13], observer design [14], and barrier functions [15], where the amplitude of the step size cannot be apriori fixed.

This work is inspired by the recent contribution of the authors [16], [17] where an exact sampled-data equivalent model to a gradient dynamics has been computed. It results that the discrete-time dynamics is implicitly defined via the discrete-gradient function [18]–[20]. Since such a dynamics represents exactly the dynamics of the gradient at the sampling instants (corresponding to the step size), one can naturally raise the question: is it possible to define an optimization algorithm based on a suitably computed approximation of the discrete equivalent dynamics? It is shown that, for a class of objective functions, the approximation of the discrete gradient at the first order provides a modified algorithm ensuring convergence independently on the step-size. The update rule is now realized by moving along a modified direction which inversely depends on the Hessian of the involved function. Several examples illustrate the benefits of the proposed method with respect to several choices of the step-size with respect to the number of steps required for converging.

The paper is organized as follows. In Section II we present the problem statement and the necessary background. In Section IV we state the main result proposing a new optimization algorithm based on the approximation of the discrete equivalent dynamics. In Section IV the result is applied by

means of two training examples. Finally, Section V concludes the paper.

Notation: Throughout the paper all functions and vector fields are assumed smooth and complete over the respective definition spaces. $\mathbb{R}$ and $\mathbb{N}$ denote, respectively, the set of real and natural numbers including 0. For any vector $v \in \mathbb{R}^n$, $v^\top$ defines the transpose of v. Given a real-valued differentiable function $S(\cdot) : \mathbb{R}^n \to \mathbb{R}$, $\nabla S(\cdot)$ represents its vector gradient function when ∇ denotes the $\mathbb{R}^n$ vector of partial derivatives as $\nabla^2 S(\cdot)$ represents its Hessian matrix.

II. Recalls and Problem statement

In this paper we focus on the unconstrained optimization problem concerning the minimum seeking of the objective function $S(\cdot) : \mathbb{R}^n \to \mathbb{R}$, that is

$$\min_{x \in \mathbb{R}^n} S(x). \tag{1}$$

We assume the optimization problem with a non-empty solution set $\mathcal{X}_\star$ defined as

$$\mathcal{X}_\star = \{x \in \mathbb{R}^n \mid \nabla S(x) = 0 \text{ and } \nabla^2 S(x) \succ 0\},$$

containing all (local) minima of the objective function $S(x)$. In the following, we will say the optimization problem is locally solved if we compute $x_\star \in \mathcal{X}_\star$ from an initial condition $x_0 = x(0) \in \mathbb{R}^n$ contained in a neighborhood of $x_\star$, say

$$x_0 \in \mathbf{B}_\epsilon(x_\star) = \{x \in \mathbb{R}^n \mid ||x - x_\star|| < \epsilon, \text{ for } \epsilon > 0\}.$$

The objective function is assumed to be ℓ-*smooth* [7], i.e.

$$S(y) \leq S(x) + \nabla^\top S(x)(y - x) + \frac{\ell}{2}||y - x||^2 \tag{2}$$

for all $x, y \in \mathbb{R}^n$ and $\ell > 0$. For twice-differentiable objective functions, the condition (2) yields a boundary condition upon the Hessian matrix, that is

$$\nabla^2 S(x) \preceq \ell I, \tag{3}$$

which is typically a reasonable assumption in many optimization problems, see [7], [21], [22].

One popular and consolidated approach solving (1) is the gradient descent (GD) method [21]–[23]. The GD algorithm finds a local minimum $x_\star$ starting from an initial guess by iteratively proceeding along the negative gradient of the objective function. In particular, for a given initial guess $x_0 = x(0) \in \mathbb{R}^n$, the GD aims at iteratively updating the point $x_k = x(k)$, for $k \in \mathbb{N}$, using information upon the steepest descent of the objective function, i.e.

$$x_{k+1} = x_k - \delta \nabla S(x_k) \tag{4}$$

where $\delta > 0$ denotes the step-size, or learning rate as known in the machine learning context. Methods of the form (4) are first-order optimization method and their strength is the low computational complexity. However, their convergence is ensured only for δ sufficiently small. As a matter of fact, the method (4) leads to instabilities of the algorithm for large choices of the step size [4], [5], and thus generally one needs additional conditions to regulate the step-size, [24], [25].

Other popular algorithms for solving (1) are second-order optimization methods, usually known as Newton's methods, [8], [9], [26], [27]. These are methods which attempt to solve (1) by constructing an algorithm upon a sequence of second-order Taylor approximations of the objective function around the iterates, [26]. In the sequel we present the most known methods.

The first most known method is the Standard Newton (SN), which is given by the following (see e.g. [9, Eq. 29] and [27, Eq. 2])

$$x_{k+1} = x_k - \delta(\nabla^2 S(x_k))^{-1} \nabla S(x_k). \tag{5}$$

Although this method is generally very fast and improves the seeking of the minimum by means of the information upon the curvature of the objective function, its main drawback is that the minimum seeking becomes inefficient in the proximity to $x_\star$ due to the inverse of the Hessian. To overcome this inefficiency, numerous modifications have been proposed, where the most common are: the Gauss-Newton (GN) method (see [8, Eq. 8])

$$x_{k+1} = x_k - \delta \left(\mu + ||\nabla S(x_k)||^2\right)^{-1} \nabla S(x_k), \tag{6}$$

involving a freely tunable parameter μ, and the Regularized Newton (RN) method (see [9, Eq. 31])

$$x_{k+1} = x_k - \delta \left(||\nabla S(x_k)|| I + \nabla^2 S(x_k)\right)^{-1} \nabla S(x_k). \tag{7}$$

All the aforementioned methods suffers from the choice of the step-size parameter, and additional pre-computation conditions must be verified, [24], [25].

On the basis of the foregoing considerations and invoking differential arguments, one can see that for initial conditions sufficiently close to $x_\star$ the evolutions described by the following equation

$$\dot{x} = -\rho \nabla S(x), \qquad \rho > 0. \tag{8}$$

locally converge toward $x_\star$ with a decay rate modulated by the choice of ρ, and with the variation of the objective described by

$$\dot{S}(x) = -\rho ||\nabla S(x)||^2 \leq 0. \tag{9}$$

This variation of the objective function, along the direction governed by (8), can be represented through the *discrete gradient function* [18], that is a vector-valued function $\bar{\nabla} S|_v^w : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ which satisfies

$$S(y) - S(x) = (y - x)^\top \bar{\nabla} S|_x^y, \tag{10}$$

for all $x, y \in \mathbb{R}^n$ with $\bar{\nabla} S|_x^x = \nabla S(x)$.

Now the question is, how to deduce from the gradient dynamics (8) an iterative procedure that computes y from x ensuring decreasing along the objective function towards $x_\star$ while preserving the discrete gradient constraint (10)? To solve this problem, it comes to our aid the result recently proposed by the authors in [16], [17] where the problem of the characterization of a discrete-time model [19] describing the

sampled evolution of a gradient dynamics has been addressed. The result is formally restated below.

Theorem 2.1 ([16]): Given the differential gradient dynamics (8), then for all $\delta \in]0, T[$, its sampled-data equivalent dynamics admits the implicit representation

$$x_{k+1} - x_k = F^\delta(x_k) := -\delta \mathcal{I}^\delta(x_k) \bar{\nabla} S|_{x_k}^{x_{k+1}} \tag{11}$$

with $x_{k+1} = x((k+1)\delta)$, $x_k = x(k\delta)$, and non-singular square matrix

$$\mathcal{I}^\delta(x) = \rho M^\delta(x)(I - \delta\rho Q^\delta(x))^{-1}, \tag{12}$$

where

$$M^\delta(x) = \frac{1}{\delta}\int_0^\delta J[e^{-s\rho\nabla S}x]ds,$$
$$Q^\delta(x) = \Big(\int_0^1 s\int_0^1 J[\nabla S](x + \tau(sF^\delta(x)))d\tau ds\Big)M^\delta(x).$$

The proposed discrete dynamics (11), that is implicitly defined due to the discrete gradient function, match the variation of the function $S(\cdot)$ at the sampling instants, i.e.

$$S(x_{k+1}) - S(x_k) = -\delta \bar{\nabla}^\top S|_{x_k}^{x_{k+1}} \mathcal{I}^\delta(x_k) \bar{\nabla} S|_{x_k}^{x_{k+1}} \leq 0. \tag{13}$$

As a matter of fact, the abovementioned result would suggest to exploit (11) as an iterative procedure to solve the optimization problem (1). However, since only power series expansion of this dynamics can be computed (see [16], [17]), the crucial question is whether we can use satisfactory approximations of the dynamics (11) to solve the optimization problem (1).

In this paper we propose an iterative procedure obtained by performing suitable approximations of the implicit dynamics (11). The resulting algorithm is explicitly defined, and for objective functions satisfying the ℓ-smooth condition with $\ell > 0$, converges towards $x_\star$ for all finite fixed step-size. In addition, if also the Polyak-Łojasiewicz (PL) condition [22] is satisfied, i.e. for $0 < \lambda \leq 1$

$$\frac{1}{2}||\nabla S(x)||^2 \geq \frac{1}{2\lambda\ell}(S(x) - S(x_\star)), \tag{14}$$

we show that the proposed method arises with a linear convergence rate.

III. Main result

The proposed iteration method is presented in the following statement.

Theorem 3.1: Consider the problem (1) with non-empty solution set $\mathcal{X}_\star$ and assume $S(\cdot) : \mathbb{R}^n \to \mathbb{R}$ to be ℓ-smooth and twice-differentiable. Then, for all $x_0 \in \mathbf{B}_\epsilon(x_\star)$, the one-step numerical procedure

$$x_{k+1} = x_k - \delta\left(I + \frac{\delta}{2}\nabla^2 S(x_k)\right)^{-1}\nabla S(x_k), \tag{15}$$

locally solves the optimization problem (1).

Proof: The procedure (15) is obtained approximating $\mathcal{I}^\delta(x)$ and $\bar{\nabla} S|_{x_k}^{x_{k+1}}$ into the right-hand side of (11). In particular, from [16] one has

$$\mathcal{I}^\delta(x) = I - \frac{\delta^2}{12}\nabla^2 S(x)\nabla^2 S(x) + \mathcal{O}(\delta^4)$$
$$\bar{\nabla} S|_{x_k}^{x_{k+1}} = \nabla S(x_k) + \frac{1}{2}\nabla^2 S(x_k)(x_{k+1} - x_k) + \mathcal{O}(\|x_{k+1} - x_k\|^2),$$

and, substituting them into (11), one gets

$$x_{k+1} - x_k = -\delta(\nabla S(x_k) + \frac{1}{2}\nabla^2 S(x_k)(x_{k+1} - x_k)) + \mathcal{O}(\delta^3).$$

By truncating the above implicit equation in $\mathcal{O}(\delta^3)$, one obtains the explicit representation (15) because

$$\begin{aligned} x_{k+1} &= x_k - \delta\nabla S(x_k) - \frac{\delta}{2}\nabla^2 S(x_k)(x_{k+1} - x_k) \\ &= x_k - \delta\left(I + \frac{\delta}{2}\nabla^2 S(x_k)\right)^{-1}\nabla S(x_k). \end{aligned}$$

Then, $S(x)$ has a local minimum $x_\star \in \mathcal{X}_\star$ and thus its derivative vanishes at $x_\star$, i.e $\nabla S(x_\star) = 0$, and this implies that $x_\star$ is an equilibrium of (15) since $x_\star$ is also contained in the set

$$\left\{x \in \mathbb{R}^n \mid \left(I + \frac{\delta}{2}\nabla^2 S(x)\right)^{-1}\nabla S(x) = 0\right\}.$$

Then, substituting the approximate model (15) into the ℓ-smooth condition (2), one gets the variation inequality

$$\begin{aligned} S(x_{k+1}) - S(x_k) &\leq \nabla^\top S(x_k)(x_{k+1} - x_k) + \frac{\ell}{2}||(x_{k+1} - x_k)||^2 \\ &= -\delta\nabla^\top S(x_k)\left(I + \frac{\delta}{2}\nabla^2 S(x_k)\right)^{-1}\nabla S(x_k) \\ &\quad + \frac{\delta^2\ell}{2}\left|\left|\left(I + \frac{\delta}{2}\nabla^2 S(x_k)\right)^{-1}\nabla S(x_k)\right|\right|^2. \end{aligned}$$

Because $S(x)$ is assumed twice-differentiable, from (3) one directly deduces

$$\left(1 + \frac{\delta\ell}{2}\right)^{-1} I \preceq \left(I + \frac{\delta}{2}\nabla^2 S(x)\right)^{-1},$$

so that at the boundary case, for all $\ell > 0$ and $\delta > 0$, the variation inequality for all $x_k \neq x_\star$ reads

$$\begin{aligned} S(x_{k+1}) - S(x_k) &< -\delta\nabla^\top S(x_k)\left(1 + \frac{\delta\ell}{2}\right)^{-1}\nabla S(x_k) \\ &\quad + \frac{\delta^2\ell}{2}\left|\left|\left(1 + \frac{\delta\ell}{2}\right)^{-1}\nabla S(x_k)\right|\right|^2 \\ &= -\frac{4\delta}{(2 + \delta\ell)^2}||\nabla S(x_k)||^2 < 0, \end{aligned} \tag{16}$$

and since $S(x_{k+1}) - S(x_k) = 0$ only for $x_k = x_\star$, this provides convergence of the sequence (15) toward $x_\star$ for all $x_0 \in \mathbf{B}_\epsilon(x_\star)$.

◁

The representation of the proposed model (11) defines an approximation of the model occurred over the discrete gradient

that does not match the evolution of $S(x)$ with respect to the differential model (8) for any value of δ. Unlike Newton's methods in (5), (6), and (7), the proposed model (15) for $\delta \to 0$ converges to the standard GD method (4), namely

$$\lim_{\delta \to 0} \left(I + \frac{\delta}{2} \nabla^2 S(x) \right)^{-1} \nabla S(x) = \nabla S(x).$$

In the following, exploiting the assumption made in [22] for Gradient descent dynamics of the form (4), we show that the proposed method (15) comes with a linear rate of convergence for all step-size $\delta \in]0, +\infty[$ when assuming the objective function satisfying also the PL condition (14). This result is stated below.

Corollary 3.1: Assume that the problem (1) has a non-empty solution set $\mathcal{X}_\star$ with an objective function $S(\cdot) : \mathbb{R}^n \to \mathbb{R}$ that is twice-differentiable, ℓ-smooth with $\ell > 0$, and for $0 < \lambda \leq 1$ satisfies (14). Then, for all $x_0 \in \mathbf{B}_\epsilon(x_\star)$, (15) converges for any step-size $\delta \in]0, +\infty[$, with a linear convergence rate

$$S(x_k) - S(x_\star) \leq \mathbf{\Omega}^k \left(S(x_0) - S(x_\star) \right), \tag{17}$$

and decay term

$$\mathbf{\Omega} = \left(1 - \frac{8\delta\ell\lambda}{(2+\delta\ell)^2} \right), \tag{18}$$

verifying $|\mathbf{\Omega}| < 1$.

Proof: By using the PL condition (14), namely

$$-||\nabla S(x_k)||^2 \leq -2\lambda\ell(S(x_k) - S(x_\star)),$$

and substituting it into the convergence rate (16) one gets in the boundary case the following variation inequality

$$\begin{aligned} S(x_{k+1}) &\leq S(x_k) - \frac{4\delta}{(2+\delta\ell)^2} ||\nabla S(x_k)||^2 \\ &\leq S(x_k) - \frac{4\delta}{(2+\delta\ell)^2} (2\lambda\ell(S(x_k) - S(x_\star))) \\ &= S(x_k) + S(x_\star) - S(x_\star) - \frac{8\delta\ell\lambda}{(2+\delta\ell)^2} (S(x_k) - S(x_\star)) \\ &\leq S(x_\star) + \left(1 - \frac{8\delta\ell\lambda}{(2+\delta\ell)^2} \right) (S(x_k) - S(x_\star)) \end{aligned}$$

which leads in the whole time horizon $k \in [0, \infty[$ the upper bound upon the variation of the objective function

$$\begin{aligned} S(x_k) - S(x_\star) &\leq \left(1 - \frac{8\delta\ell\lambda}{(2+\delta\ell)^2} \right) (S(x_{k-1}) - S(x_\star)) \\ &= \left(1 - \frac{8\delta\ell\lambda}{(2+\delta\ell)^2} \right)^k (S(x_0) - S(x_\star)). \end{aligned}$$

Finally, due to the definition space of λ and ℓ, i.e. $\lambda \in]0, 1]$ and $\ell \in]0, \infty[$, for any step-size $\delta \in]0, \infty[$ one gets

$$|\mathbf{\Omega}| = \left| 1 - \frac{8\delta\ell\lambda}{(2+\delta\ell)^2} \right| < 1,$$

which implies that the right-hand side of (17) is approaching zero since $\mathbf{\Omega}^k \to 0$ as $k \to \infty$. $\triangleleft$

Differently from the result in [22], the proposed model shows a linear convergence rate provided by the decay term

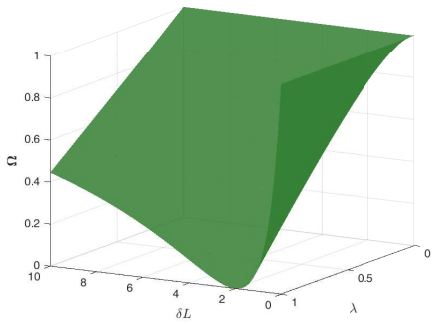

Fig. 1: Surface depicting $\mathbf{\Omega}$ with $\lambda \in]0, 1]$ and $\delta\ell \in]0, \infty[$.

$\mathbf{\Omega}$ for all $\ell, \delta \in]0, \infty[$ and $\lambda \in]0, 1]$ as depicted in Fig. 1. As a matter of fact, for x_0 sufficiently close to $x_\star$, the method (15) always provides finite convergence without involving other methods to regulate the step-size, as in [24], [25].

IV. Examples

In this section we investigate two training standard problems in machine learning like logistic regression and least squares to provide numerical evidence of the proposed method, based upon the proposed method in Theorem 3.1 showing linear convergence rate in Corollary 3.1. The implementation of the proposed method (15) is reported in Algorithm 1 considering

Algorithm 1:

Result: x_k
initialize x_0 and $k \leftarrow 0$;
define step-size δ and tolerance ε;
while $|S(x_{k+1}) - S(x_k)| > \varepsilon$ **do**
  $\mathbf{g}_k \leftarrow -\left(I + \frac{\delta}{2}\nabla^2 S(x_k)\right)^{-1} \nabla S(x_k)$;
  $x_{k+1} \leftarrow x_k + \delta \mathbf{g}_k$;
  $k \leftarrow k + 1$;
end

an initial guess x_0 and arbitrary tolerance ε, needed to determine whether the sequence x_k has arrived close to the minimum point $x_\star$. Simulations have been performed using Matlab on an Intel(R) Core(TM) i7-8550U CPU and 16.0 GB RAM.

A. Training Example 1

Consider a toy example assuming an objective function $S : \mathbb{R} \to \mathbb{R}$ with the unconstrained optimization problem (see, Example 1.4.3 in [28])

$$\min_{x \in \mathbb{R}} S(x) = \log(\exp(x) + 1) - \frac{x}{2} + \frac{x^2}{2} \tag{19}$$

The function (19) is ℓ-smooth and satisfies (14), and the proposed model takes the form $x_{k+1} = x_k + \delta \mathbf{g}(x_k)$ with

$$\mathbf{g}(x) = \frac{(\exp(x) + 1)\left(2x + \exp(x) + 2x\exp(x) - 1\right)}{2(1 + 2\exp(x) + \exp(2x)) + \delta(1 + 3\exp(x) + \exp(2x))}.$$

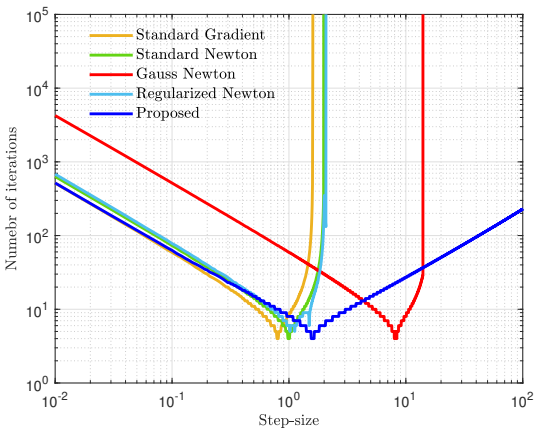

Fig. 2: Number of iterations for minimizing (19)

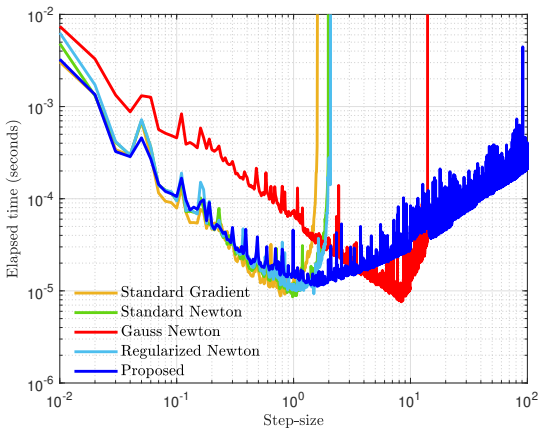

Fig. 3: Elapsed time for minimizing (19)

Simulations: Simulations reported in Fig. 2 and Fig. 3 compare the behaviour of the proposed second-order method (15) with the gradient descent in (4), Standard Newton in (5), Gauss Newton in (6), and Regularized Newton in (7). In particular, Fig. 2 shows the number of iterations required by each method to solve (19) for different step-sizes, and Fig. 3 shows the computational cost, that is the time required by each algorithm to solve (19). In this scenario, we consider tolerance $\varepsilon = 10^{-8}$, initial condition $x_0 = -0.5$, and step-size $\delta \in [10^{-2}, 10^{2}]$. Figures highlight that, unlike the proposed method (15), all the other involved optimization methods explode after a certain value of the step-size and no longer solve the minimization problem (19) for values of δ sufficiently large. In particular, the standard gradient method (4) for small step-sizes seems the fastest, as it converges with fewer iterations than the other methods, but it is also the first method to explode and fail to solve (19) for $\delta > 1.5$. The Standard Newton (5) and Regularized Newton (7) shows roughly the same performance. Although for small step-sizes the methods are relatively slow compared to the standard gradient, both methods solve the problem for slightly larger values of the step-size, i.e. $\delta \approx 2$. The Gauss Newton (6) for small step-sizes is the slower method but it converges to the minimum for a larger range of δ than the other methods. Differently form the others, the proposed method is the only one that does not explode after a certain value of δ and solve (19) for larger step-sizes, such as $\delta \approx 10^2$. Moreover, for small step-sizes the algorithm is fast as the standard gradient and achieves the minimum of the objective function with similar performance and similar time of convergence, as seen in Fig. 3. Finally, the best performance of the proposed method for (19) is achieved with $\delta \approx 1.6$.

B. Training Example 2

For sake of illustration, we consider now an objective function $S : \mathbb{R}^n \to \mathbb{R}$, of the form

$$\min_{x \in \mathbb{R}^n} S(x) = \frac{1}{2} \sum_{i=1}^{n} (\alpha_i x_i - \mathrm{b}_i)^2, \tag{20}$$

generally used as loss function describing the error of the difference between $\mathrm{b}_i \in \mathbb{R}$ (playing the role of given data) and $x_i \in \mathbb{R}$ (playing the role of predicted data) with model parameter $\alpha_i x_i \in \mathbb{R}$. The function (20) belongs to the class of objective functions satisfying (14). In particular, straightforward computations yield

$$\nabla S = \begin{pmatrix} \alpha_1^2 x_1 - \alpha_1 \mathrm{b}_1 \\ \vdots \\ \alpha_n^2 x_n - \alpha_n \mathrm{b}_n \end{pmatrix}, \nabla^2 S = \begin{pmatrix} \alpha_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \alpha_n^2 \end{pmatrix}$$

with $x_\star^\top = \begin{pmatrix} \alpha_1^{-1}\mathrm{b}_1 & \vdots & \alpha_n^{-1}\mathrm{b}_n \end{pmatrix}^\top$ and thus from the ℓ-smooth condition and (14) one gets

$$\ell \in \left] 0, \max_x \left(\sum_{i=1}^{n} \frac{(x_i \alpha_i - \mathrm{b}_i)^2}{(x_i - \alpha_i^{-1}\mathrm{b}_i)^2} \right) \right], \qquad \lambda \leq \sum_{i=1}^{n} \frac{\alpha_i^2}{\ell}.$$

The upper-boundary condition of ℓ provides λ is contained in its definition set $\lambda \in]0, 1]$. Finally, the proposed model yields the structure $x_{k+1} = x_k + \delta \mathbf{g}(x_k)$ with

$$\mathbf{g}(x) = \begin{pmatrix} \frac{2\alpha_1(\alpha_1 x_1 - \mathrm{b}_1)}{2+\delta\alpha_1^2} & \cdots & \frac{2\alpha_n(\alpha_1 x_n - \mathrm{b}_n)}{2+\delta\alpha_n^2} \end{pmatrix}^\top. \tag{21}$$

Simulations. Simulations reported in Fig. 4 and Fig. 5 compare the behaviour of the proposed second-order method (15) with the gradient descent in (4), Standard Newton in (5), Gauss Newton in (6), and Regularized Newton in (7). In the simulations we fix $n = 200$ and random initial conditions, $\alpha_i = 5$, $\mathrm{b}_i = 1$, tolerance $\varepsilon = 10^{-8}$ and $\delta \in [10^{-2}, 10^{2}]$.

V. Conclusions

In this paper we took the idea from the approximation of sampled-data equivalent gradient dynamics in [16], [17] and proposed an easily implementable algorithm to solve a class of optimization problems. Under standard assumption on the objective function, we proved that the proposed method solves the minimization problem for any arbitrarily chosen step-size. Two numerical examples illustrate the advantage of the proposed algorithm when compared with some popular optimization methods. This first achievement in the optimization context paves the way for further investigations upon the

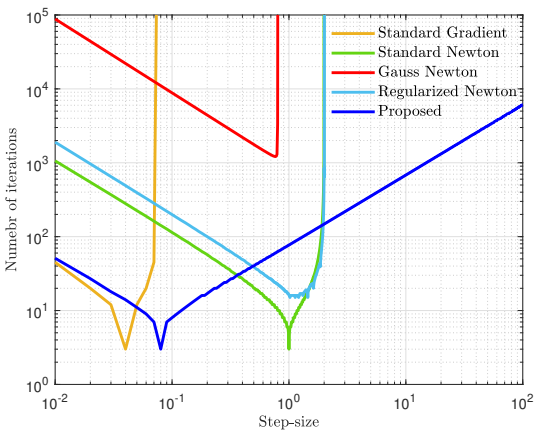

Fig. 4: Number of iterations for minimizing (20)

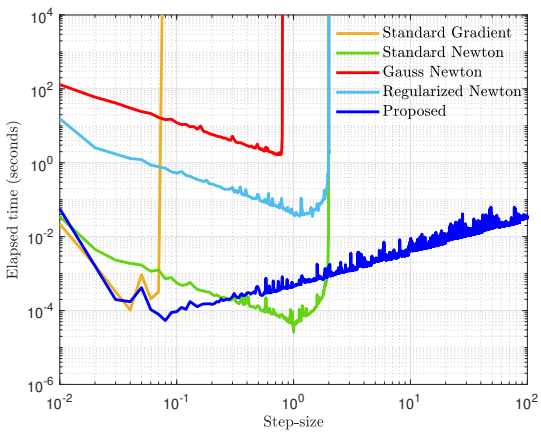

Fig. 5: Elapsed time for minimizing (20)

convergence properties due to higher-order approximations of the gradient dynamics of the form (11). Future works are aimed at applying this method in real-time optimal control involving, for instance, sampled-data MPC [29], [30].

References

[1] A. Ghosh, "Comparative study of financial time series prediction by artificial neural network with gradient descent learning," *arXiv preprint arXiv:1111.4930*, 2011.

[2] T. Tirer and R. Giryes, "Image restoration by iterative denoising and backward projections," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1220–1234, 2018.

[3] C. Shang and F. You, "Data analytics and machine learning for smart process manufacturing: recent advances and perspectives in the big data era," *Engineering*, vol. 5, no. 6, pp. 1010–1016, 2019.

[4] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, 2013, pp. 1310–1318.

[5] O. Gannot, "A frequency-domain analysis of inexact gradient methods," *Mathematical Programming*, pp. 1–42, 2021.

[6] A. Galántai, "The theory of newton's method," *Journal of Computational and Applied Mathematics*, vol. 124, no. 1-2, pp. 25–44, 2000.

[7] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[8] H. H. Tan and K. H. Lim, "Review of second-order optimization techniques in artificial neural networks backpropagation," in *IOP Conference Series: Materials Science and Engineering*, vol. 495, no. 1. IOP Publishing, 2019, p. 012003.

[9] R. A. Polyak, "Regularized newton method for unconstrained convex optimization," *Mathematical programming*, vol. 120, no. 1, pp. 125–145, 2009.

[10] K. Nar and S. S. Sastry, "Step size matters in deep learning," in *NeurIPS*, 2018.

[11] F. H. Kong and I. R. Manchester, "Contraction analysis of nonlinear noncausal iterative learning control," *Systems & Control Letters*, vol. 136, p. 104599, 2020.

[12] G. Turrisi, M. Capotondi, C. R. Gaz, V. Modugno, G. Oriolo, and A. De Luca, "On-line learning for planning and control of underactuated robots with uncertain dynamics," *IEEE Robotics and Automation Letters*, 2021.

[13] L. Menini, C. Possieri, and A. Tornambè, "A newton-like algorithm to compute the inverse of a nonlinear map that converges in finite time," *Automatica*, vol. 89, pp. 411–414, 2018.

[14] D. Astolfi and C. Possieri, "Design of local observers for autonomous nonlinear systems not in observability canonical form," *Automatica*, vol. 103, pp. 443–449, 2019.

[15] D. Astolfi, P. Bernard, R. Postoyan, and L. Marconi, "Redesign of discrete-time nonlinear observers with state estimate constrained in prescribed convex set," *IFAC-PapersOnLine*, vol. 52, no. 16, pp. 454–459, 2019.

[16] S. Monaco, D. Normand-Cyrot, M. Mattioni, and A. Moreschini, "Nonlinear hamiltonian systems under sampling," *IEEE Transactions on Automatic Control*, pp. 1–1, 2022.

[17] A. Moreschini, S. Monaco, and D. Normand-Cyrot, "Gradient and hamiltonian dynamics under sampling," *IFAC-PapersOnLine*, vol. 52, no. 16, pp. 472–477, 2019.

[18] R. I. McLachlan, G. R. W. Quispel, and N. Robidoux, "Geometric integration using discrete gradients," *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 357, no. 1754, pp. 1021–1045, 1999.

[19] A. Moreschini, M. Mattioni, S. Monaco, and D. Normand-Cyrot, "Discrete port-controlled hamiltonian dynamics and average passivation," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 1430–1435.

[20] M. Mattioni, A. Moreschini, S. Monaco, and D. Normand-Cyrot, "Quaternion-based attitude stabilization via discrete-time ida-pbc," *IEEE Control Systems Letters*, 2022.

[21] D. Bertsekas, *Nonlinear Programming: 3rd Edition*. Athena Scientific, 2016.

[22] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2016, pp. 795–811.

[23] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[24] B. Zhou, L. Gao, and Y.-H. Dai, "Gradient methods with adaptive step-sizes," *Computational Optimization and Applications*, vol. 35, no. 1, pp. 69–86, 2006.

[25] G. Frassoldati, L. Zanni, and G. Zanghirati, "On adaptive step-size selections in gradient methods," in *Proceedings in Applied Mathematics and Mechanics*, vol. 7, no. 1. Wiley Online Library, 2007, pp. 1 061 903–1 061 904.

[26] S. Ketabchi, H. Moosaei, M. Parandegan, and H. Navidi, "Computing minimum norm solution of linear systems of equations by the generalized newton method," *Numerical Algebra, Control & Optimization*, vol. 7, no. 2, p. 113, 2017.

[27] P. Rebentrost, M. Schuld, L. Wossnig, F. Petruccione, and S. Lloyd, "Quantum gradient descent and newton's method for constrained polynomial optimization," *New Journal of Physics*, vol. 21, no. 7, p. 073023, 2019.

[28] N. Doikov, "New second-order and tensor methods in convex optimization," Ph.D. dissertation, PhD thesis, Université catholique de Louvain, 2021.

[29] M. Elobaid, M. Mattioni, S. Monaco, and D. Normand-Cyrot, "Sampled-data tracking under model predictive control and multi-rate planning," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 3620–3625, 2020.

[30] ——, "Station-keeping of 1 2 halo orbits under sampled-data model predictive control," *Journal of Guidance, Control, and Dynamics*, pp. 1–10, 2022.