

STOCHASTIC OPTIMIZATION METHODS FOR THE SIMULTANEOUS CONTROL OF PARAMETER-DEPENDENT SYSTEMS

UMBERTO BICCARI, ANA NAVARRO-QUILES, AND ENRIQUE ZUAZUA

ABSTRACT. We address the application of stochastic optimization methods for the simultaneous control of parameter-dependent systems. In particular, we focus on the classical Stochastic Gradient Descent (SGD) approach of Robbins and Monro, and on the recently developed Continuous Stochastic Gradient (CSG) algorithm. We consider the problem of computing simultaneous controls through the minimization of a cost functional defined as the superposition of individual costs for each realization of the system. We compare the performances of these stochastic approaches, in terms of their computational complexity, with those of the more classical Gradient Descent (GD) and Conjugate Gradient (CG) algorithms, and we discuss the advantages and disadvantages of each methodology. In agreement with well-established results in the machine learning context, we show how the SGD and CSG algorithms can significantly reduce the computational burden when treating control problems depending on a large amount of parameters. This is corroborated by numerical experiments.

1. INTRODUCTION

Stochastic Gradient Descent (SGD) is an iterative method for optimizing functions in the form of a sum of different observations in a data set:

$$(1.1) \quad \begin{aligned} \hat{u} &= \min_{u \in \mathbb{R}^N} f(u) \\ f(u) &:= \frac{1}{n} \sum_{i=1}^n f_i(u). \end{aligned}$$

It is a random variant of Gradient Descent (GD) optimization, in which the minimum of f is approximated by a gradient-based methodology employing, instead of the full realization of the system, an estimate thereof obtained through a batch of few components picked randomly.

This approach was originally proposed in the seminal work of Robbins and Monro [43]. It has recently received renewed attention, especially in the Machine Learning and Big Data communities, for treating minimization problems depending on very large data sets. In this context, it has shown impressive performance in terms of the computational efficiency (see, for instance, [3, 8, 9, 47]). Nowadays, SGD and its variants have become preeminent optimization methods in fields such as empirical risk minimization ([46, 48, 52]), data mining ([49]) or artificial neural networks ([45]).

Stochastic-based optimization approaches can also be tailored to address many challenges in different contexts of the experimental sciences. In fluid mechanics, these techniques are employed to deal with several practical issues including flow analysis and turbulence modeling (see [12]). In seismology, stochastic algorithms are used to improve existing methodologies to find high-resolution and high-fidelity models of the subsurface ([21]). In natural and social sciences, random algorithms such as the so-called Random Batch Method have been developed to reduce the computational cost when simulating large systems of interacting agents ([31]).

2010 *Mathematics Subject Classification.* 34H05, 49J15, 49M29, 90C15, 93E20.

Key words and phrases. Parameter-dependent systems, simultaneous controllability, stochastic optimization, computational cost.

This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement NO. 694126-DyCon). The work of U. B. and E. Z. was partially supported by the Grant MTM2017-92996-C2-1-R COSNET of MINECO (Spain) and by the Air Force Office of Scientific Research (AFOSR) under Award NO. FA9550-18-1-0242. The work of E.Z. is partially funded by the Alexander von Humboldt-Professorship program, the European Unions Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No.765579-ConFlex, the Grant ICON-ANR-16-ACHN-0014 of the French ANR and the Transregio 154 Project “Mathematical Modelling, Simulation and Optimization Using the Example of Gas Networks” of the German DFG.

In this context, we shall also mention the recent contribution [33] concerning optimal control strategies for guidance by repulsion models. Finally, we can mention applications in several engineering problems, including the management of power generation systems and smart grids ([1, 19, 20]) or the synchronization of coupled oscillators described by the Kuramoto model ([5]).

In this paper, we transfer this discussion to the framework of parameter-dependent optimal control problems. In particular, we examine the employment of stochastic algorithms for the computation of *simultaneous controls* (see [36]). As a matter of fact, we will see in what follows that such class of problems conforms with the setting of (1.1), thus rendering natural the choice of a stochastic approach in this context.

In many real-life applications, parameter-dependent models are commonly employed to describe physical phenomena which may have different realizations. In this setting, the parameters are associated, for instance, to material properties (such as the Lamé coefficients in models for elasticity, see [14, Chapter 3, Section 3.8]), to non-uniform heat conduction in heterogeneous bodies ([29, Chapter I, Section 1.2]) or to the viscosity properties of a fluid ([24, Chapter 1, Section 4]).

Control problems for parameter-dependent models most often represent a big computational challenge.

Indeed, their numerical resolution typically requires repeated simulations of the dynamics one aims to control, spanning within a (possibly large) range of distinct parameters. This amounts to solving a different equation for each new desired configuration, resulting in a complex computational problem, especially for parameter sets of large cardinality.

For this reason, many analytical and computational techniques have been developed in the past years in order to speed up the simulation of parameterized optimal control problems. Among others, we can mention Proper Orthogonal Decomposition (see, e.g., [2, 27, 34, 42]), other more general Reduced Basis approaches ([4, 18, 28, 32]), or the so-called *greedy* methodology ([13, 15, 35, 26]).

The purpose of this work is to investigate the employment of stochastic algorithms for the control of parameter-dependent systems, and to discuss their advantages and disadvantages with respect to some more classical deterministic ones, namely GD and Conjugate Gradient (CG). In particular, we shall establish in which situations one methodology is preferable with respect to the others, based on the amount of parameters in the system and on the accuracy we seek in the control computation. For this analysis, we will consider two specific stochastic approaches:

1. the standard SGD of Robbins and Monro [43] (see also [3, 6, 7, 8, 9]);
2. the recently developed Continuous Stochastic Gradient (CSG) algorithm [41].

More details on the aforementioned optimization techniques will be given later.

To test the efficiency of the optimization algorithms we propose and compare their performances, we shall focus on a couple of specific control problems for linear finite-dimensional ODE models. Notwithstanding that, our discussion and conclusions may be extended also to the non-linear setting and have a wide range of applicability in what concerns control problems.

In accordance with the well-established results in the Machine Learning and Big Data communities, our analysis will confirm that SGD and CSG are, among the considered methods, the only computationally viable for systems of large dimension and massive dependence on parameters.

This work is organized as follows. In Section 2, we present the problem we are going to analyze and the methodology we propose. In Section 3, we give a general overview of GD, CG, SGD and CSG in the context of the present work. Some fundamental results about the convergence of those algorithms are summed up. In addition, we present an analysis of the computational complexity which justifies why the stochastic approach is attractive when dealing with abundant amounts of parameters. We conclude that section presenting the algorithms that we will use for our numerical experiments. In Section 4, we will compare the GD, CG, SGD and CSG approaches for the simultaneous controllability of a specific example of linear control system and we will present numerical simulations confirming our previous theoretical discussion. Finally, Section 5 summarizes our main achievements and gathers some open problems and possible directions of future research.

2. PROBLEM FORMULATION

In this work, we will analyze the problem of *simultaneous control*, where one aims to design a unique control function capable to steer all the different realizations of a given parameter-dependent system to some prescribed final target (see [36] and the references therein).

To be more precise, we will consider the parameter-dependent finite-dimensional linear model

$$(2.1) \quad \begin{cases} x'_\nu(t) = \mathbf{A}_\nu x_\nu(t) + \mathbf{B}u(t), & 0 < t < T, \\ x_\nu(0) = x^0, \end{cases}$$

in which $x_\nu(t) \in L^2(0, T; \mathbb{R}^N)$, $N \geq 1$, denotes the state, the $N \times N$ matrix \mathbf{A}_ν describes the dynamics, and the function $u(t) \in L^2(0, T; \mathbb{R}^M)$, $1 \leq M \leq N$, is the M -component control acting on the system through the $N \times M$ matrix \mathbf{B} .

All along this paper, $\nu \in \{\nu_1, \nu_2, \dots, \nu_{|\mathcal{K}|}\} =: \mathcal{K}$, where $|\mathcal{K}|$ denotes the cardinality of \mathcal{K} , is a random parameter following a probability law μ , with $(\mathcal{K}, \mathcal{F}, \mu)$ the corresponding complete probability space. Moreover, the initial datum x^0 is assumed to be independent of ν .

The aim of simultaneous control is to find a unique *parameter-independent* control u such that, at time $T > 0$, the corresponding solution x_ν of (2.1) satisfies

$$(2.2) \quad x_\nu(T) = x^T, \quad \nu \in \mathcal{K} \quad \mu - \text{a. e.}$$

for some desired final target $x^T \in \mathbb{R}^N$. Here we will always assume x^T to be parameter-independent, although parameter-dependent targets x_ν^T could also be considered.

It has been pointed out for instance in [36, Remark 1.1-5] that (2.2) is guaranteed by the fact that

$$\mathbb{E} \left[\|x_\nu(T) - x^T\|_{\mathbb{R}^N}^2 \right] = \int_{\mathcal{K}} \|x_\nu(T) - x^T\|_{\mathbb{R}^N}^2 d\mu = 0,$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator. In view of this observation, we will address the simultaneous control of (2.1) from an optimal control viewpoint, that is, by solving the following minimization problem

$$(2.3) \quad \begin{aligned} \hat{u} &= \min_{u \in L^2(0, T; \mathbb{R}^M)} F_\nu(u) \\ F_\nu(u) &:= \frac{1}{2} \mathbb{E} \left[\|x_\nu(T) - x^T\|_{\mathbb{R}^N}^2 \right] + \frac{\beta}{2} \|u\|_{L^2(0, T; \mathbb{R}^M)}^2, \end{aligned}$$

subject to the dynamics given by (2.1).

In (2.3), $0 < \beta \ll 1$ is a suitable small penalization constant introduced to enhance the requirement of getting close to the target x^T while keeping the control u of small $L^2(0, T; \mathbb{R}^M)$ -norm. Moreover, notice that, since the expectation operator $\mathbb{E}[\cdot]$ is convex (see, e.g., [23, Theorems 1.6.1 and 1.6.2]), the functional F_ν is convex as well.

Nowadays, a large number of gradient-based methods are available for solving optimization problems in the form (2.3). An incomplete literature includes [22, 30, 38].

Nevertheless, when applied to parameter-dependent problems, all these methods present a main drawback. Indeed, their implementation requires, in each iteration, to compute the gradient of the functional giving the descent direction by solving the state equation (2.1) and the corresponding adjoint equation for all parameter values. This, of course, may rapidly increase the computational cost, especially when the dimension of \mathcal{K} is large, thus reducing the efficiency of the algorithm.

To bypass this issue, a natural approach is to employ a stochastic algorithm to reduce the number of gradient calculations and, consequently, the total computational complexity. In this paper, we will consider two of these methodologies:

1. The SGD algorithm (see [43]). This is a simplification of the classical GD in which, instead of computing ∇F_ν for all parameters $\nu \in \mathcal{K}$, in each iteration this gradient is estimated on the basis of a single randomly picked configuration. It has been shown that, if the cardinality $|\mathcal{K}|$ is sufficiently large, SGD becomes computationally less expensive than GD (see [8, 9]).
2. The newly developed CSG approach (see [41]). This is a variant of SGD, based on the idea of reusing previously obtained information to improve the efficiency of the algorithm. As it is for SGD, in each iteration of CSG the method requires the computation of only one component of the gradient

corresponding to a randomly picked realization of the model. The descent direction is then determined by a linear combination of this component with the ones computed in previous iterations. In this way, the CSG algorithm allows to approximate the full gradient of the objective functional with arbitrary precision in the course of the optimization process, while maintaining a low computational effort per iteration. As we will see, this eventually yields to a better convergence behavior with respect to SGD.

As we mentioned, the main interest of this paper is to analyze to which extents the SGD and CSG approaches may be successfully applied for solving optimal control problems in the form (2.3). In particular, we will discuss their efficiency - in terms of computational complexity - by comparing them with the classical GD and CG algorithm, and by illustrating the advantages and disadvantages of the four methodologies in our context.

This will be done both on a theoretical level and numerically, by employing the GD, CG, SGD and CSG methodologies to compute the simultaneous control for a couple of specific examples of our problem (2.1).

3. GD, CG, SGD AND CSG APPROACHES FOR THE OPTIMAL CONTROL PROBLEM (2.3)

In this section, we give an overall description of the GD, CG, SGD and CSG approaches for the resolution of the optimization problem (2.3). In particular, we recall the principal results about the convergence of these methods and on their computational complexity. Besides, we provide the specific algorithms we will employ for our numerical simulations.

3.1. The GD approach. Let us start with the GD procedure, consisting in minimizing an objective function by finding the direction in which it can be reduced faster. This optimal descent direction is known to be given by the gradient of the function. Hence, the GD algorithm translates in finding the minimizer \hat{u} in (2.3) as the limit $k \rightarrow +\infty$ of the following iterative process

$$(3.1) \quad u^{k+1} = u^k - \eta_k \nabla F_\nu(u^k),$$

where $\eta_k > 0$ is called the step-size or, in the machine learning context, the *learning rate*. We stress that the selection of a correct learning rate is crucial in terms of the algorithm performances. As a matter of fact, if η_k is not properly chosen, the iterative scheme (3.1) may actually not converge to the minimum of F_ν . In practice, a suitable learning rate depends on the regularity properties of the objective function. For more details, see e.g. [10, Section 9.2] or [39, Section 2.1.5].

Let us now compute the gradient of the functional F_ν giving the descent direction for the GD scheme. To this end, we shall first measure the rate of change of F_ν in any direction $\zeta \in L^2(0, T; \mathbb{R}^M)$ by calculating the directional derivative as follows:

$$(3.2) \quad D_\zeta F_\nu(u) = \left. \frac{d}{d\epsilon} F_\nu(u + \epsilon\zeta) \right|_{\epsilon=0} = \beta \int_0^T \langle u, \zeta \rangle_{\mathbb{R}^M} dt + \mathbb{E}[\langle x_\nu(T) - x^T, z_\nu(T) \rangle_{\mathbb{R}^N}],$$

where $z_\nu \in L^2(0, T; \mathbb{R}^N)$ is the solution of the following equation

$$(3.3) \quad \begin{cases} z'_\nu(t) = \mathbf{A}_\nu z_\nu(t) + \mathbf{B}\zeta, & 0 < t < T, \\ z_\nu(0) = 0. \end{cases}$$

Let now $p_\nu \in L^2(0, T; \mathbb{R}^N)$ be the solution of the adjoint problem

$$(3.4) \quad \begin{cases} p'_\nu(t) = -\mathbf{A}_\nu^\top p_\nu(t), & 0 < t < T, \\ p_\nu(T) = -(x_\nu(T) - x^T). \end{cases}$$

Multiplying (3.3) by p_ν and integrating by parts we obtain

$$\begin{aligned} 0 &= \int_0^T \mathbb{E}[\langle z'_\nu - \mathbf{A}_\nu z_\nu - \mathbf{B}\zeta, p_\nu \rangle_{\mathbb{R}^N}] dt \\ &= -\mathbb{E}[\langle x_\nu(T) - x^T, z_\nu(T) \rangle_{\mathbb{R}^N}] - \int_0^T \mathbb{E}[\langle \zeta, \mathbf{B}^\top p_\nu \rangle_{\mathbb{R}^M}] dt \\ &= -\mathbb{E}[\langle x_\nu(T) - x^T, z_\nu(T) \rangle_{\mathbb{R}^N}] - \int_0^T \langle \zeta, \mathbb{E}[\mathbf{B}^\top p_\nu] \rangle_{\mathbb{R}^M} dt. \end{aligned}$$

Therefore,

$$\mathbb{E}[\langle x_\nu(T) - x^T, z_\nu(T) \rangle_{\mathbb{R}^N}] = - \int_0^T \langle \zeta, \mathbb{E}[\mathbf{B}^\top p_\nu] \rangle_{\mathbb{R}^M} dt$$

and, replacing this last expression in (3.2), we obtain that, for any $\zeta \in L^2(0, T; \mathbb{R}^M)$,

$$D_\zeta F_\nu(u) = \int_0^T \langle \beta u - \mathbb{E}[\mathbf{B}^\top p_\nu], \zeta \rangle_{\mathbb{R}^M} dt.$$

In view of the above computations, the gradient of the functional F_ν is given by the expression

$$(3.5) \quad \nabla F_\nu(u) = \beta u - \mathbb{E}[\mathbf{B}^\top p_\nu] = \beta u - \frac{1}{|\mathcal{K}|} \sum_{\nu \in \mathcal{K}} \mathbf{B}^\top p_\nu.$$

Consequently, the GD scheme to solve the optimization problem (2.3) becomes

$$(3.6) \quad \mathbf{GD}: \quad u^{k+1} = u^k - \eta_k \left(\beta u^k - \frac{1}{|\mathcal{K}|} \sum_{\nu \in \mathcal{K}} \mathbf{B}^\top p_\nu^k \right).$$

We then see that applying (3.6) for minimizing the functional $F_\nu(u)$ requires to solve at each iteration the coupled system

$$(3.7) \quad \begin{cases} x'_\nu(t) = \mathbf{A}_\nu x_\nu(t) + \mathbf{B}u, & 0 < t < T, \\ p'_\nu(t) = -\mathbf{A}_\nu^\top p_\nu(t), & 0 < t < T, \\ x_\nu(0) = x^0, \quad p_\nu(T) = -(x_\nu(T) - x^T), \end{cases}$$

for all the parameters $\nu \in \mathcal{K}$ (that is, $|\mathcal{K}|$ times).

Concerning now the convergence rate, it is known that the regularity of the objective function and the choice of the step-size play a fundamental role in the efficiency of the GD algorithm.

In our case, since F_ν is convex, it follows for instance from [39, Theorem 2.1.15] (see also [40, Theorem 3.3]) that, if we take η_k constant small enough, we have

$$(3.8) \quad \|u^k - \hat{u}\|_{\mathbb{R}^N}^2 \leq \|u^0 - \hat{u}\|_{\mathbb{R}^N}^2 e^{-2\mathcal{C}_{GD}k},$$

where the positive constant \mathcal{C}_{GD} is given by

$$(3.9) \quad \mathcal{C}_{GD} = \ln \left(\frac{\rho + 1}{\rho - 1} \right),$$

ρ being the conditioning number of the controllability Grammian associated to (2.1) (see, e.g., [16, Definition 1.10] for the definition of the controllability Grammian).

From (3.8) we immediately conclude that, for computing the control \hat{u} up to some given tolerance $\varepsilon > 0$, i.e. for obtaining

$$\|u^k - \hat{u}\|_{\mathbb{R}^N}^2 < \varepsilon,$$

the GD algorithm requires

$$k = \mathcal{O} \left(\frac{\ln(\varepsilon^{-1})}{\mathcal{C}_{GD}} \right)$$

iterations. This means that, with a per-iteration cost proportional to $|\mathcal{K}|$ (due to the need to compute $\nabla F_\nu(u^k)$ for all $k \in \mathbb{N}$), the total cost required to obtain ε -optimality for the GD method is

$$(3.10) \quad \text{cost}_{GD} = \mathcal{O} \left(\frac{|\mathcal{K}| \ln(\varepsilon^{-1})}{\mathcal{C}_{GD}} \right).$$

3.2. The CG approach. Let us now describe the CG approach and comment its convergence properties. The starting point is, once again, to compute the gradient of the functional F_ν which, according to (3.5), is given by

$$\nabla F_\nu(u) = \beta u - \mathbb{E}[\mathbf{B}^\top p_\nu] = \beta u - \frac{1}{|\mathcal{K}|} \sum_{\nu \in \mathcal{K}} \mathbf{B}^\top p_\nu,$$

with $p_\nu \in L^2(0, T; \mathbb{R}^N)$ solution of the adjoint problem

$$\begin{cases} p'_\nu(t) = -\mathbf{A}_\nu^\top p_\nu(t), & 0 < t < T, \\ p_\nu(T) = -(x_\nu(T) - x^T) =: p_{T,\nu} \end{cases}$$

and $x_\nu \in L^2(0, T; \mathbb{R}^N)$ solution of the original system (2.1). Moreover, we remark that x_ν is given by the sum $x_\nu = y_\nu + z_\nu$, with

$$(3.11) \quad \begin{cases} y'_\nu(t) = \mathbf{A}_\nu y_\nu(t), & 0 < t < T, \\ y_\nu(0) = x_0 \end{cases}$$

and

$$(3.12) \quad \begin{cases} z'_\nu(t) = \mathbf{A}_\nu z_\nu(t) + \mathbf{B}u(t), & 0 < t < T, \\ z_\nu(0) = 0. \end{cases}$$

Then, we can readily check that, for all $\nu \in \mathcal{K}$, ∇F_ν can be rewritten in the form

$$(3.13) \quad \nabla F_\nu(u) = \underbrace{(\beta I + \mathbb{E}[\mathcal{L}_{T,\nu}^* \mathcal{L}_{T,\nu}])}_{\mathbb{A}} u + \underbrace{\mathbb{E}[\mathcal{L}_{T,\nu}^*(y_\nu(T) - x^T)]}_{-b},$$

where the operators $\mathcal{L}_{T,\nu}$ and $\mathcal{L}_{T,\nu}^*$ are defined as

$$(3.14) \quad \begin{array}{ccc} \mathcal{L}_{T,\nu} : U & \longrightarrow & \mathbb{R}^N \\ u & \longmapsto & z_\nu(T) \end{array} \quad \text{and} \quad \begin{array}{ccc} \mathcal{L}_{T,\nu}^* : \mathbb{R}^N & \longrightarrow & U \\ p_{T,\nu} & \longmapsto & \mathbf{B}^\top p_\nu, \end{array}$$

and where we used the abridged notation $U := L^2(0, T; \mathbb{R}^M)$.

Since, clearly, the minimizer \hat{u} of F_ν has to satisfy $\nabla F_\nu(\hat{u}) = 0$, we see from (3.13) that computing \hat{u} is equivalent to solve the linear system

$$(3.15) \quad \mathbb{A}u = b$$

The CG methodology amounts to solve (3.15) through the iterative procedure of Algorithm 1.

Algorithm 1 CG algorithm for solving the linear system (3.15)

```

input  $d^0 = r^0 = b - \mathbb{A}u^0$ 
for  $k \geq 1$  do
   $\alpha_k = \frac{(r^k)^\top r^k}{(d^k)^\top \mathbb{A}d^k}$ 
   $x^{k+1} = x^k + \alpha_k d^k$ 
   $r^{k+1} = r^k - \alpha_k d^k$ 
   $\gamma_{k+1} = \frac{(r^{k+1})^\top r^{k+1}}{(r^k)^\top r^k}$ 
   $d^{k+1} = r^{k+1} + \gamma_{k+1} d^k$ 
end for

```

We then see that, as for GD, applying Algorithm 1 for minimizing $F_\nu(u)$ requires to solve at each iteration the coupled system (3.7) for all $\nu \in \mathcal{K}$ (hence $|\mathcal{K}|$ times).

Concerning now the convergence rate, we know for instance from [44, Theorem 6.29] (see also [44, Equation 6.128]) that

$$(3.16) \quad \|u^k - \hat{u}\|_{\mathbb{R}^N}^2 \leq 4 \|u^0 - \hat{u}\|_{\mathbb{R}^N}^2 e^{-2C_{CG}k},$$

where the positive constant \mathcal{C}_{CG} is given by

$$(3.17) \quad \mathcal{C}_{CG} = \ln \left(\frac{\sqrt{\rho} + 1}{\sqrt{\rho} - 1} \right),$$

ρ being once again the conditioning number of the controllability Grammian associated to the control system (2.1).

From (3.8) we immediately conclude that, for computing the control \hat{u} up to some given tolerance $\varepsilon > 0$, i.e. for obtaining

$$\|u^k - \hat{u}\|_{\mathbb{R}^N}^2 < \varepsilon,$$

the CG algorithm requires

$$k = \mathcal{O} \left(\frac{\ln(\varepsilon^{-1})}{\mathcal{C}_{CG}} \right)$$

iterations. This means that the total cost to obtain ε -optimality for the CG method is

$$(3.18) \quad \text{cost}_{CG} = \mathcal{O} \left(\frac{|\mathcal{K}| \ln(\varepsilon^{-1})}{\mathcal{C}_{CG}} \right).$$

As a final remark, let us stress that the convergence properties of the CG algorithm are known to be better than the GD ones. This is due to two main reasons.

First of all, taking into account that, by definition of conditioning number, we always have $\rho > 1$, we can immediately notice that the constant \mathcal{C}_{CG} in (3.17) is always larger than \mathcal{C}_{GD} given in (3.9). Hence, even if both GD and CG algorithm converge exponentially, this convergence will actually be faster in the case of GD.

In addition to that, a well known property of CG is the so-called *finite termination* (see, e.g., [25, Remark 2.4]). This means that, if we apply CG to solve a N -dimensional problem, the algorithm will converge in at most N -iterations. Practical implementations of CG may partially lose this finite termination property due to round-off errors. Nevertheless, this iterative method still provides monotonically improving approximations to the exact solution, which usually reach the required tolerance after a small (compared to the problem size) number of iterations. We refer to [44, Section 6.11.3] for more details on this specific issue.

3.3. The SGD approach. Let us now describe the SGD algorithm. As we mentioned, the main difference of this approach with respect to the classical GD one is that, in the iterative scheme (3.1), we do not employ all the components of $\nabla F_\nu(u)$. Instead, we pick a parameter ν_k i.i.d. from \mathcal{K} and we use the corresponding gradient as descent direction. Hence, the SGD recursion process for optimizing F_ν is given by

$$(3.19) \quad u^{k+1} = u^k - \eta_k \nabla F_{\nu_k}(u^k),$$

where $(\eta_k)_{k \geq 1}$ is a deterministic sequence of positive scalars which we still refer to as the *learning rates sequence*. Moreover, in view of the computations in Section 3.1, the descent direction ∇F_{ν_k} can be computed as

$$\nabla F_{\nu_k}(u^k) = \beta u^k - \mathbf{B}^\top p_{\nu_k}^k,$$

with $p_{\nu_k}^k$ solution of the adjoint equation (3.4). Hence, the complete SGD scheme to solve the optimization problem (2.3) is given by

$$(3.20) \quad \text{SGD: } u^{k+1} = u^k - \eta_k (\beta u^k - \mathbf{B}^\top p_{\nu_k}^k).$$

We then see that applying (3.20) for minimizing the functional $F_\nu(u)$ requires, at each iteration k , only one resolution of the coupled system (3.7). Because of that, each iteration of this stochastic approach results very cheap.

Concerning now the convergence properties of SGD, some preliminary observations have to be made:

1. First of all, we have to stress that in the SGD method the iterate sequence is not determined uniquely by the function F_ν , the starting point u^0 , and the learning rates sequence $(\eta_k)_{k \geq 1}$, as it would be in a deterministic optimization algorithm. Rather, $(u^k)_{k \geq 1}$ is a stochastic process whose behavior is determined by the random sequence $(\nu_k)_{k \geq 1} \subset \mathcal{K}$. In particular, this will imply that the convergence properties of the algorithm have to be defined in terms of stochastic quantities, namely

$\mathbb{E}[\|u^{k+1} - \hat{u}\|_{\mathbb{R}^N}^2]$ (see, e.g., [3, 9]), or in the context of *almost sure convergence* (see [6, Section 4.5]).

2. Because of the randomness of the stochastic process defined by (3.19), in certain iterations, the direction $-\nabla F_{\nu_k}(u_k)$ might not be one of descent from u_k (in the sense of yielding a negative directional derivative for F_{ν} from u^k). Notwithstanding that, if it is a descent direction in expectation, then the sequence $(u_k)_{k>1}$ can be guided toward a minimizer of F_{ν} . Also for this reason, the convergence rate of SGD will always be at most linear.
3. As for the deterministic case above, the choice of a good learning rate sequence is crucial for the general performances of the algorithm. In the stochastic framework, this may become a quite delicate issue. In fact, as it is illustrated for instance in [9, Section 4.2], the convergence of SGD is guaranteed as long as the stochastic directions and step-sizes are chosen such that the second moment $\mathbb{E}[\|\nabla F_{\nu}(u^k)\|^2]$ is bounded above by a deterministic quantity. There is therefore an interplay between the step-sizes and bounds on the variance of the stochastic directions which, we recall, is defined by

$$\text{var}[\nabla F_{\nu}(u^k)] = \mathbb{E}[\|\nabla F_{\nu}(u^k)\|^2] - \|\mathbb{E}[\nabla F_{\nu}(u^k)]\|^2.$$

In particular, when the gradient computation is noisy (namely, when the value of $\sigma := \|\nabla F_{\nu}(u^k)\|$ is large), for obtaining optimal convergence rates for the SGD iteration one has to properly reduce the value of η_k . There is nowadays an extended literature concerning how to select an appropriate learning rate. A standard approach (see [43]) is to choose $\{\eta_k\}_{k \geq 1}$ as a decreasing sequence such that

$$(3.21) \quad \sum_{k=1}^{\infty} \eta_k = +\infty \quad \text{and} \quad \sum_{k=1}^{\infty} \eta_k^2 < \infty.$$

Nevertheless, this choice may be inconvenient in practical applications, since the step-length may become very small, thus producing a very slow progresses in the actualization of the objective functional and deteriorating the convergence behavior. To avoid this situation, one possibility is to select the learning rate through an *adaptive* approach (see, e.g., [9, Section 4.2]): SGD is run with a fixed step-size, η_k small enough with respect to the parameters defining the regularity of the objective functional and the bounds on σ . If progress appears to stall, a smaller η_k is selected and the process is repeated. This is the approach we will use in our simulations. For completeness, let us stress that choosing a constant learning rate is not a good option for SGD. Indeed, as it has been proved for instance in [9, Theorem 4.6], if SGD is run with a fixed step-size $\eta_k = \bar{\eta}$, even if the value of $\bar{\eta}$ is small we may not reach convergence because of the noise introduced by the stochastic process.

4. If the learning rate is properly chosen, by means of standard martingale techniques we can show (see for instance [9, Section 4.5]) that the SGD converges almost surely

$$u^k \xrightarrow{a.s.} \hat{u}, \quad \text{as } k \rightarrow +\infty.$$

In practical applications, this means that for solving the optimization problem (2.3) it is enough to run the SGD algorithm only once and we will have probability one of converging to the minimum \hat{u} . Said in other terms, although it is possible that a single launch of SGD fails to compute the optimal control \hat{u} , this is an extremely rare event belonging to a set of zero measure.

Concerning the convergence rate, it has been proved in several contributions (see, e.g., [3, Theorem 1]) that no matter the regularity of F_{ν} , because of the noise introduced by the random selection of the descent direction the convergence of (3.19) is always at most linear:

$$\mathbb{E}[\|u^k - \hat{u}\|_{\mathbb{R}^N}^2] = \mathcal{O}(k^{-1}).$$

This implies that, in order to converge to some given tolerance ε , that is,

$$\mathbb{E}[\|u^k - \hat{u}\|_{\mathbb{R}^N}^2] < \varepsilon$$

a single launch of the SGD algorithm will require $\mathcal{O}(\varepsilon^{-1})$ iterations.

However, it is crucial to note that this time the per-iteration cost does not depend on the sample set size $|\mathcal{K}|$. This means that the total work required to obtain ε -optimality for SGD is only proportional to ε^{-1} .

Admittedly, this can be larger than (3.10) or (3.18) for small or moderate values of $|\mathcal{K}|$, while the comparison favors SGD when dealing with a large amount of parameters. This can be quantified by saying that, in the case of our functional F_ν , SGD becomes competitive with respect to GD and CG when considering sample sets of size

$$(3.22) \quad |\mathcal{K}|(\varepsilon) = \mathcal{O}\left(\frac{1}{\varepsilon \ln(\varepsilon^{-1})}\right).$$

Hence, given a tolerance ε , (3.22) provides an approximated threshold for $|\mathcal{K}|$ giving us a hint on when SGD is expected to outperform GD and CSG for the control computation.

3.4. The CSG approach. Finally, let us now describe the CSG algorithm. As we mentioned, the main difference of this approach with respect to SGD one is that, in the iterative scheme, we do not employ anymore only a randomly picked gradient component but we reuse previously obtained information to improve the efficiency of the algorithm. In particular, in each iteration the descent direction of the CSG algorithm is chosen as follows:

Step 1. As in SGD, we select a parameter ν_k i.i.d. from \mathcal{K} and we compute the corresponding gradient.

Step 2. We combine linearly this gradient with the ones obtained in the previous iterations. The result of this linear combination is our descent direction.

In more detail, the CSG recursion process for optimizing F_ν is given by

$$(3.23) \quad u^{k+1} = u^k - \eta_k G^k, \quad G^k = \sum_{\ell=1}^k \alpha_\ell \nabla F_{\nu_\ell}(u^\ell),$$

where the weights $\{\alpha_\ell\}_{\ell=1}^k$ are obtained as in [41, Section 2.1].

Moreover, as for GD and SGD the gradient components $\{\nabla F_{\nu_\ell}(u^\ell)\}_{\ell=1}^k$ are computed through the adjoint methodology, giving

$$\nabla F_{\nu_\ell}(u^\ell) = \beta u^\ell - \mathbf{B}^\top p_{\nu_\ell}^\ell,$$

with $p_{\nu_\ell}^\ell$ solution of (3.4). Hence, the complete CSG scheme to solve the optimization problem (2.3) is given by

$$(3.24) \quad \text{CSG: } u^{k+1} = u^k - \eta_k \sum_{\ell=1}^k \alpha_\ell (\beta u^\ell - \mathbf{B}^\top p_{\nu_\ell}^\ell).$$

Besides, we stress that applying (3.24) for minimizing the functional $F_\nu(u)$ still requires, at each iteration k , only one resolution of the coupled system (3.7). Because of that, also in this case, as it was for SGD, each iteration of CSG results very cheap.

Concerning now the convergence properties of CSG, as for SGD the choice of a good learning rate sequence is crucial for the general performances of the algorithm.

Also in this case a possible choice would be to use a reducing learning rate sequence satisfying (3.21) (see [41, Remark 7]). This guarantees the convergence in expectation of the CSG algorithm, as it has been shown in [41, Theorem 18 and Theorem 19].

Notwithstanding that, as we already mentioned in the case of SGD, for practical applications the condition (3.21) on the step-length is inconvenient, since η_k rapidly becomes very small and the algorithm thus progresses only slowly.

At this regard, let us mention that an important property of CSG is that, as the optimization process evolves, the approximated gradient G^k converges almost surely to the full gradient of the objective functional, that is,

$$G^k \xrightarrow{a.s.} \nabla F_\nu, \quad \text{as } k \rightarrow +\infty.$$

See [41, Corollary 14]. This fact translates in a less noisy algorithm (if compared with SGD), and eventually yields to better convergence properties. In particular, for CSG convergence may be guaranteed also choosing a fixed learning rate sequence $\eta_k = \bar{\eta}$ (see [41, Theorem 20]). This is the choice of η_k we will adopt in our simulations.

Finally, let us mention that the results of [41] only prove the convergence in expectation of CSG, but they do not provide a converging rate. For this reason, the comparison of the convergence properties of CSG with respect to the other algorithms considered in this paper will be carried out directly at the numerical level.

3.5. GD,CG, SGD and CSG algorithms presentation. We present in this section the GD, CG, SGD and CSG algorithms we will employ in our numerical simulations.

Algorithm 2 GD algorithm for the optimal control problem (2.3)

input x^0, x^T : initial condition and final target of the primal system
 u^0 : initial guess for the minimizer
 $k \leftarrow 0$: iteration counter
 k_{max} : maximum number of iterations allowed
 tol : tolerance
while STOP-CRIT and $k < k_{max}$ **do**
 for $j = 1$ to $|\mathcal{K}|$ **do**
 Solve the coupled system (3.7)
 end for
 $u^{k+1} \leftarrow u^k - \eta_k \left(\beta u^k + \frac{1}{|\mathcal{K}|} \sum_{\nu \in \mathcal{K}} \mathbf{B}^\top p_\nu^k \right)$
 $k \leftarrow k + 1$
end while
return $u^{k+1} = \hat{u}$: minimum of the functional $F_\nu(u)$.

Algorithm 3 CG algorithm for the optimal control problem (2.3)

input x^0, x^T : initial condition and final target of the primal system
 u^0 : initial guess for the minimizer
 $k \leftarrow 0$: iteration counter
 k_{max} : maximum number of iterations allowed
 tol : tolerance
 $d^0 = r^0 = b - \mathbb{A}u^0$
while STOP-CRIT and $k < k_{max}$ **do**
 Solve the linear system (3.15) with the algorithm 1
end while
return $u^{k+1} = \hat{u}$: minimum of the functional $F_\nu(u)$.

Algorithm 4 SGD algorithm for the optimal control problem (2.3)

input x^0 and x^T : initial condition and final target of the primal system
 u^0 : initial guess for the minimizer
 $k \leftarrow 0$: iteration counter
 k_{max} : maximum number of iterations allowed
 tol : tolerance
while STOP-CRIT and $k < k_{max}$ **do**
 Select $\nu_k \in \{\nu_1, \dots, \nu_{|\mathcal{K}|}\}$ i.i.d. and compute ∇F_{ν_k} solving the coupled system (3.7)
 $u^{k+1} \leftarrow u^k - \eta_k \left(\beta u^k + \mathbf{B}^\top p_{\nu_k}^k \right)$
 $k \leftarrow k + 1$
end while
return $u^{k+1} = \hat{u}$: minimum of the functional $F_\nu(u)$.

Algorithm 5 CSG algorithm for the optimal control problem (2.3)

input x^0 and x^T : initial condition and final target of the primal system
 u^0 : initial guess for the minimizer
 $k \leftarrow 0$: iteration counter
 k_{max} : maximum number of iterations allowed
 tol : tolerance
while STOP-CRIT and $k < k_{max}$ **do**
 Select $\nu_k \in \{\nu_1, \dots, \nu_{|\mathcal{K}|}\}$ i.i.d. and compute ∇F_{ν_k} solving the coupled system (3.7)
 Compute the weights α_ℓ , $\ell = 1, \dots, k$ using the methodology of [41]
 $u^{k+1} \leftarrow u^k - \eta_k \sum_{\ell=1}^k \alpha_\ell (\beta u^\ell + \mathbf{B}^\top p_{\nu_\ell}^\ell)$
 $k \leftarrow k + 1$
end while
return $u^{k+1} = \hat{u}$: minimum of the functional $F_\nu(u)$.

3.6. Practical considerations on the implementation of GD and CG. In Sections 3.1 and 3.2 we presented explicit convergence rates for GD and CG. In particular, we saw that, in both cases, exponential convergence is expected.

Nevertheless, we have to stress that these convergence rates are given in terms of some explicit constants \mathcal{C}_{GD} and \mathcal{C}_{CG} (see (3.9) and (3.17)), which depend on the conditioning ρ of the problem we are considering.

If we analyze the behavior of these constants with respect to ρ (taking into account that, by definition, we always have $\rho > 1$), we immediately notice that both \mathcal{C}_{GD} and \mathcal{C}_{CG} are positive decreasing functions of ρ and they converge to zero as $\rho \rightarrow +\infty$. This implies that a bad conditioning in a minimization problem affects the actual convergence of GD and CG, which may deteriorate and violate (3.8) and (3.16).

This is a well-known computational limitation of gradient optimization methods. In particular, it is nowadays classical that the GD algorithm is very sensitive to the problem conditioning and, if ρ is large, the convergence properties may deteriorate up to a linear rate. An illustrative example of this phenomenon is provided in [37].

The situation is less critical in the case of CG, because of the considerations we presented at the end of Section 3.2. In particular, the fact that the constant \mathcal{C}_{CG} depends on the square root of ρ (see (3.17)) implies that the CG algorithm is less sensible to the conditioning of the problem. Furthermore, recall that CG enjoys the finite termination property, which helps in achieving convergence in a relatively small number of iterations.

As a final remark, let us stress that a control problem is usually quite bad conditioned, meaning that ρ is typically very large (see [11, Remark 4.2] for some explicit estimates in the CG context). As a consequence of this bad conditioning, and according to the discussion above, we will see in our numerical simulations that the convergence rate of GD is significantly reduced with respect to the expected one given by (3.8), thus making this algorithm not very efficient in the context of our simultaneous controllability problem.

4. EXPERIMENTAL RESULTS: COMPARISON OF THE ALGORITHMS

This section is devoted to some numerical experiments. The main goal of these simulations is to confirm our previous theoretical discussion by comparing GD, CG, SGD and CSG for the simultaneous controllability of a specific linear parameter-dependent ODE system in the form (2.1). In particular, we shall establish when the stochastic approaches become convenient with respect to the deterministic ones.

To this end, we will consider a couple of specific examples of linear parameter-dependent problems.

Example 1. Our first example is linearized cart-inverted pendulum model (see Figure 1):

$$(4.1) \quad \begin{pmatrix} \dot{x}_\nu \\ \dot{v}_\nu \\ \dot{\theta}_\nu \\ \dot{\omega}_\nu \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & -\frac{\nu}{M} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{\nu+M}{M} & 0 & 0 \end{pmatrix} \begin{pmatrix} x_\nu \\ v_\nu \\ \theta_\nu \\ \omega_\nu \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \end{pmatrix} u.$$

It describes the dynamical behavior of a system composed of a cart of mass M and a rigid pendulum of length ℓ . Both M and ℓ will be considered to assume fixed values. The pendulum is anchored to the cart and

at the free extremity it is placed a (small) variable mass described by the parameter ν . The cart moves on a horizontal plane. The states $x_\nu(t)$ and $v_\nu(t)$ describe its position and velocity, respectively. During the motion of the cart the pendulum deviates from the initial vertical position by an angle $\theta_\nu(t)$, with an angular velocity $\omega_\nu(t)$. Starting from an initial state $(x^i, v^i, 0, 0)$, our goal will be to compute a parameter-independent control function u steering all the realizations of the system in time T to the final state $(x^f, 0, 0, 0)$, in which the cart is at rest and the pendulum is in the vertical position.

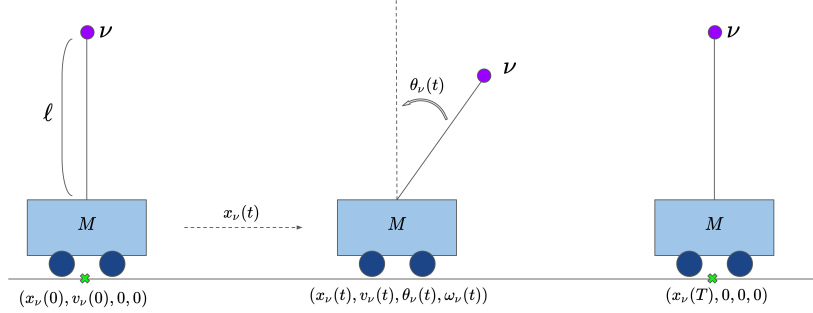


FIGURE 1. Cart-inverted pendulum system.

Example 2. Our second example is the linear parameter-dependent system associated with the following matrices:

$$(4.2) \quad \mathbf{A}_\nu = \begin{pmatrix} 0 & 1 & 0 & \dots & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 1 & 0 \\ -\nu & 0 & \dots & \dots & \dots & 0 \end{pmatrix} \in \mathbb{R}^{N \times N}, \quad \mathbf{B} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ \vdots \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^N.$$

Notice that the matrix \mathbf{A}_ν in (4.2) corresponds to the Brunovsky canonical form (see, e.g., [50, Section 2.2.3]) of the linear ODE

$$x^{(N)}(t) + \nu x(t) = 0,$$

where with $x^{(N)}(t)$ we indicate the N -th derivative of the function $x(t)$.

Let us stress that both models (4.1) and (4.2) are indeed simultaneously controllable. At this regard, we recall that the simultaneous controllability property is equivalent to the classical controllability of the augmented system

$$(4.3) \quad \dot{\mathbf{x}} = \mathcal{A}\mathbf{x} + \mathcal{B}\mathbf{u}$$

with $\mathbf{x} = (x_{\nu_1}, \dots, x_{\nu_{|\mathcal{K}|}})^T \in \mathbb{R}^{N|\mathcal{K}|}$, $\mathbf{u} = (u, \dots, u)^T \in L^2(0, T; \mathbb{R}^{N|\mathcal{K}|})$, and where the matrices \mathcal{A} and \mathcal{B} are given by

$$\mathcal{A} = \begin{pmatrix} \mathbf{A}_{\nu_1} & & 0 \\ & \ddots & \\ 0 & & \mathbf{A}_{\nu_{|\mathcal{K}|}} \end{pmatrix} \in \mathbb{R}^{N|\mathcal{K}| \times N|\mathcal{K}|} \quad \text{and} \quad \mathcal{B} = \begin{pmatrix} \mathbf{B} \\ \vdots \\ \mathbf{B} \end{pmatrix} \in \mathbb{R}^{N|\mathcal{K}| \times 1}.$$

To check that (4.3) is controllable in both cases we consider is a direct application of the Hautus test ([51, Proposition 1.5.5]). We leave the details to the reader.

4.1. Numerical simulations - Example 1. We present here the numerical experiments corresponding to the linear system (4.1).

In these simulations, we have chosen the initial state $(-1, 1, 0, 0)^\top$ and the final target $(0, 0, 0, 0)^\top$. The time horizon is set to be $T = 1s$. The mass of the cart and the length of the pendulum are fixed taking the values $M = 10$ and $\ell = 1$, while the mass of the pendulum is described by the parameter $\nu \in \mathcal{K} = \{\nu_1, \dots, \nu_{|\mathcal{K}|}\}$ with $\nu_1 = 0.1$ and $\nu_{|\mathcal{K}|} = 1$. Finally, we set the tolerance to the value $\varepsilon = 10^{-4}$.

To test the efficiency of each algorithm with respect to the cardinality of \mathcal{K} , we will perform simulations for increasing values of $|\mathcal{K}|$. Our numerical results will show that there is a threshold in the number of parameters above which the deterministic algorithms are not viable to treat the simultaneous controllability of (2.1).

For completeness, let us recall that our simulations with the SGD and CSG approach are done with only one run of the algorithm. Indeed, according to the discussion in Section 3.3, this is enough to obtain almost sure convergence and the computation of the optimal control \hat{u} .

The simulations have been performed in Matlab R2018a on a laptop with Intel Core *i5-7200UCPU@2.50GHz* \times 4 processor and 7.7 GiB RAM.

Before comparing the performances of the four algorithms, let us show that the optimization problem (2.3) indeed provides an effective simultaneous control for (4.1).

To this end, in Figure 2, we display the evolution of the free (that is, when $u \equiv 0$) and controlled dynamics associated to (4.1). In order to increase the visibility of our plots, we consider here only the case of $|\mathcal{K}| = 10$ parameters in our system. Moreover, the simulations displayed in Figure 2 have been performed with the CG algorithm, although the other three approaches described in this paper provide the same result.

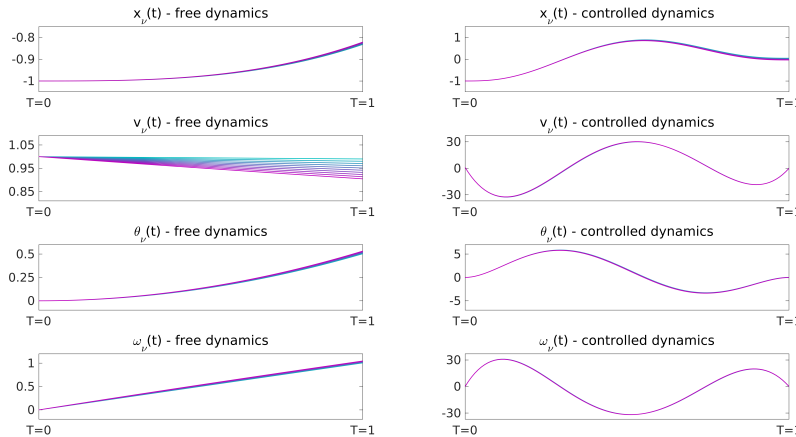


FIGURE 2. Evolution of the free (left) and controlled (right) dynamics of (4.1).

We can clearly see how the introduction of a control allows to steer all the different realization of (4.1) to the desired target configuration at time T . Hence, through the minimization of F_ν we managed to compute an effective optimal control.

Let us now analyze and discuss the behavior of the four algorithms we are considering with respect to the amount of parameters included in our model. To this end, we have run simulations with GD, CG, SGD and CSG for increasing values of the cardinality of \mathcal{K} , namely $|\mathcal{K}| = 2, 10, 100, 250$ and 500 . The results of our numerical experiments are collected in Table 1 and displayed in Figure 3.

Let us also stress that, for GD, we performed the simulations only until a cardinality $|\mathcal{K}| = 100$, for which we already have a substantial computational cost.

It is evident the role that the cardinality of \mathcal{K} has on the performances of the four algorithms. In particular, we can make the following observations:

1. The GD algorithm is the one showing the worst performances. Even for very small parameter sets ($|\mathcal{K}| = 2$), its computational time is higher than the other three algorithms. This fact, which may

$ \mathcal{K} $	GD		CG		SGD		CSG	
	Iter.	Time	Iter.	Time	Iter.	Time	Iter.	Time
2	1868	45.1s	12	1.1s	2195	33.1s	930	18.6s
10	1869	150.1s	13	2.6s	2106	31.4s	923	17.4s
100	1870	1799.5s	12	17.7s	2102	28.9s	929	17.4s
250			13	50.3s	2080	28.2s	928	17.9s
500			13	101.3s	2099	32.9s	927	21.5s

TABLE 1. Number of iterations and computational time to converge to the tolerance $\varepsilon = 10^{-4}$ of the GD, CG, SGD and CSG algorithms applied to the problem (4.1) with different values of $|\mathcal{K}|$.

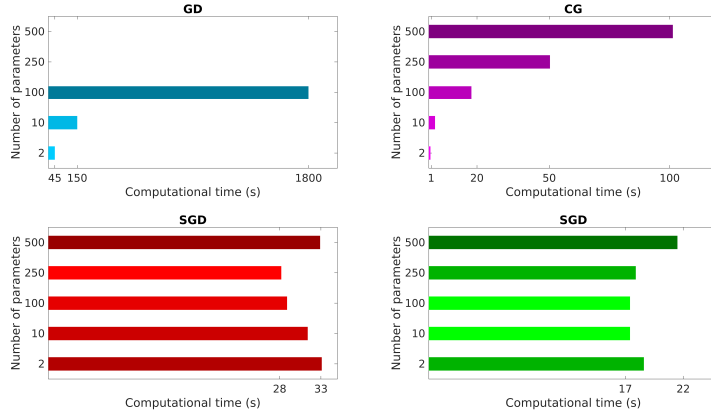


FIGURE 3. Computational time to converge to the tolerance $\varepsilon = 10^{-4}$ of the GD, CG, SGD and CSG algorithms applied to the problem (4.1) with different values of $|\mathcal{K}|$. For GD we do not include the cases $|\mathcal{K}| = 250$ and $|\mathcal{K}| = 500$ because of the excessive computational cost.

appear in counter-trend with the expected exponential convergence of GD (see (3.8)), is actually not surprising if we recall the considerations of Section 3.6.

2. The CG algorithm is the one requiring the lower number of iterations to converge. On the one hand, this confirms what we already observed in Section 3.6, i.e. that CG is less sensible to the bad conditioning of the problem and is able to maintain its exponential convergence rate (see (3.16)). On the other hand, this implies that the GD algorithm is the best approach among the one considered when dealing with a low and moderate amount of parameters, since it is capable to compensate the increasing per-iteration cost with the very limited amount of iterations it requires to achieve ε -optimality. As a matter of fact, we can see in Table 1 that SGD and CSG start outperforming CG when considering a parameter set of cardinality around $|\mathcal{K}| = 250$. Below this threshold, CG shows up to be the best behaving algorithm, in term of its total computational time, to address the optimization problem (2.3).
3. The stochastic approaches SGD and CSG appear to be essentially insensitive to the cardinality of the parameter set. This fact is not surprising if we consider that, no matter how many parameters enter in our control problem, with SGD and CSG each iteration of the optimization process always requires only one resolution of the coupled system (3.7).
4. The CSG algorithm always outperforms SGD in terms of the number of iterations it requires to converge and, consequently, of the total computational time. This can be explained by recalling that, in the CSG algorithm, the approximated gradient G_k is close to the full gradient ∇F_ν of the objective

functional when $k \rightarrow +\infty$. Hence, the optimization process is less noisy than SGD, yielding to a better convergence behavior. This can be appreciated in Figure 4, where we compare the convergence behavior of 50 different launches of SGD and CSG. As we can see, SGD presents more enhanced oscillations, while in each run of CSG the error remains very close to the mean error.

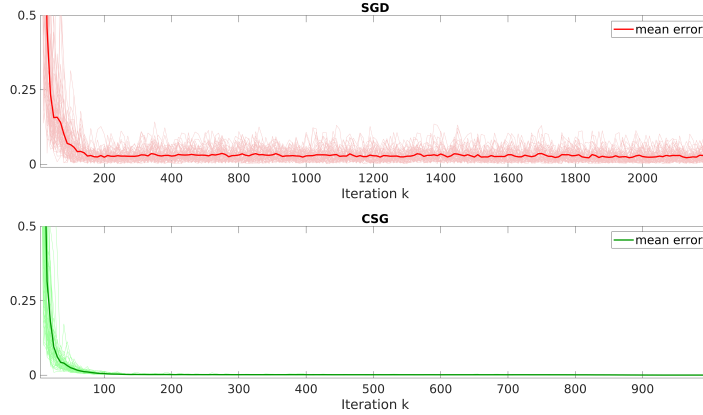


FIGURE 4. Convergence of the error for SGD and CSG. The plots correspond to 50 launches of the two algorithms with a tolerance $\varepsilon = 10^{-4}$.

Figure 5 shows a comparison among the computational times of the four algorithms when considering increasing values of $|\mathcal{K}|$. Due to the differences in the time-scales, here the plots are in logarithmic scale. We can appreciate the following facts:

1. The computational time of GD is always higher than the other three algorithms, thus making it the less effective approach among the ones we considered.
2. The computational times of SGD and CSG remain essentially constant with respect to the cardinality of the parameter set. This confirms the fact that $|\mathcal{K}|$ has not a relevant role in the convergence properties of those algorithms.
3. We distinguish a threshold for $|\mathcal{K}|$ above which the efficiency of CG is overcome by the stochastic approaches. This is the point in which the exponential convergence rate of CG is not able anymore to compensate the high per-iteration cost of the algorithm.

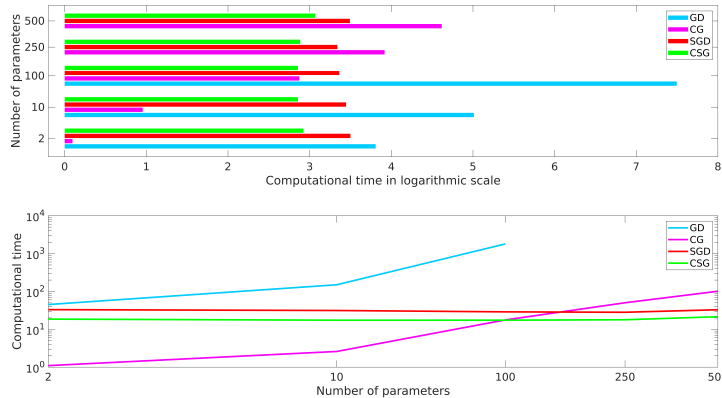


FIGURE 5. Computational time (in logarithmic scale) to converge to the tolerance $\varepsilon = 10^{-4}$ of the GD, CG, SGD and CSG algorithms applied to the problem (4.1) with different values of $|\mathcal{K}|$.

All these considerations are aligned with our previous discussion and corroborate the fact that, when dealing with large parameter sets, a stochastic approach is preferable to a deterministic one to address the simultaneous controllability of parameter-dependent systems. In particular, in the specific case of (4.1), we can conclude that CSG is the best performing algorithm among the ones we have analyzed.

4.2. Numerical simulations - Example 2. Let us now consider the linear system (4.2). In this case, to test the performances of the four optimization algorithms we are considering, we will analyze two aspects:

1. In a first moment, we will fix the dimension of our problem to $N = 4$ and we will run simulations for increasing values of the cardinality of \mathcal{K} , namely $|\mathcal{K}| = 2, 10, 100, 250, 500$ and 1000.
2. In a second moment, we will fix the cardinality of the parameter set to $|\mathcal{K}| = 100$ and we will run simulations for problems of increasing dimension, namely $N = 10, 100, 500, 750$ and 1000.

As initial datum and target we consider the vectors $x^0 = (1, 1, \dots, 1)^\top \in \mathbb{R}^N$ and $x_T = (0, 0, \dots, 0)^\top \in \mathbb{R}^N$, respectively. The time horizon is set to be $T = 1s$ and the parameter set \mathcal{K} is a $|\mathcal{K}|$ points discretization of the interval $[1, 6]$. Moreover, we will fix the tolerance to be $\varepsilon = 10^{-4}$.

As before, let us start by showing that the optimization problem (2.3) indeed provides an effective simultaneous control for (4.2). To this end, in Figure 6, we display the evolution of the free and controlled dynamics for $|\mathcal{K}| = 10$ parameters. Once again, we can appreciate how the introduction of a control into the system allows to steer the solution to the desired target configuration.

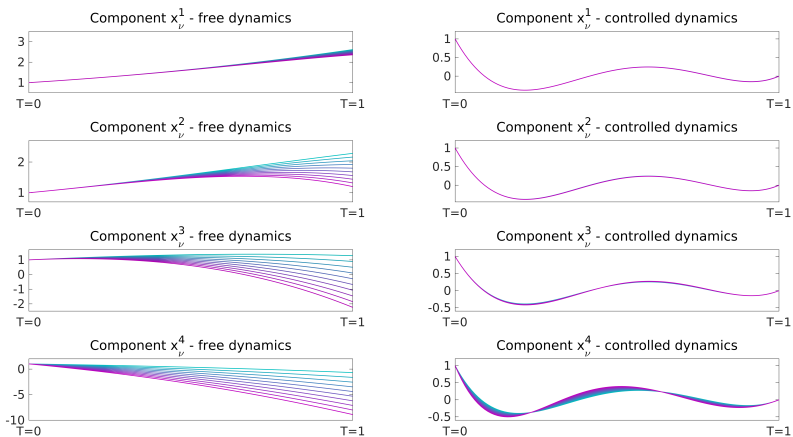


FIGURE 6. Evolution of the free (left) and controlled (right) dynamics of (4.2).

Let us now analyze and discuss the behavior of the four algorithms we are considering with respect to the amount of parameters included in our model. The results of our numerical experiments are collected in Table 2 and displayed in Figures 7 and 8. Let us also stress that, also in this case, for GD we performed the simulations only until $|\mathcal{K}| = 100$, for which we already have a substantial computational cost.

Once again, it is evident the role that the cardinality of \mathcal{K} has on the performances of the four algorithms. In particular, we observe how the GD algorithm is still the one showing the worst performances since it has to cope not only with a high per-iteration cost but also with the bad conditioning of the problem. On the other hand, the CG algorithm, being the one which requires the lower number of iterations to converge, shows up to be once again the best approach to deal with a low and moderate amount of parameters. Only when the parameter set reaches a large cardinality the advantages of the stochastic approaches appear.

Let us now conclude this section by analyzing the influence of the dimension N on the optimal control problem we are considering. To this end, we will fix the value $|\mathcal{K}| = 100$ and we will run simulations for $N = 10, 100, 500, 750$ and 1000. In what follows, we will focus only on the two best performing algorithms, namely CG in the deterministic setting and CSG in the stochastic one. The results of our simulations are collected in Table 3 and displayed in Figure 9.

$ \mathcal{K} $	GD		CG		SGD		CSG	
	Iter.	Time	Iter.	Time	Iter.	Time	Iter.	Time
2	1223	23.6s	9	1.1s	1547	19.5s	1252	18.6s
10	1455	93.5s	12	2.1s	1220	14.8s	1453	21.8s
100	1489	905.7s	9	12.3s	1836	21.8s	1485	23.9s
250			10	29.1s	1408	17.9s	1468	23.7s
500			9	52.3s	1946	25.1s	1514	27.1s
1000			11	131.9s	1946	26.3s	1514	34.1s

TABLE 2. Number of iterations and computational time to converge to the tolerance $\varepsilon = 10^{-4}$ of the GD, CG, SGD and CSG algorithms applied to the problem (4.2) with different values of $|\mathcal{K}|$.

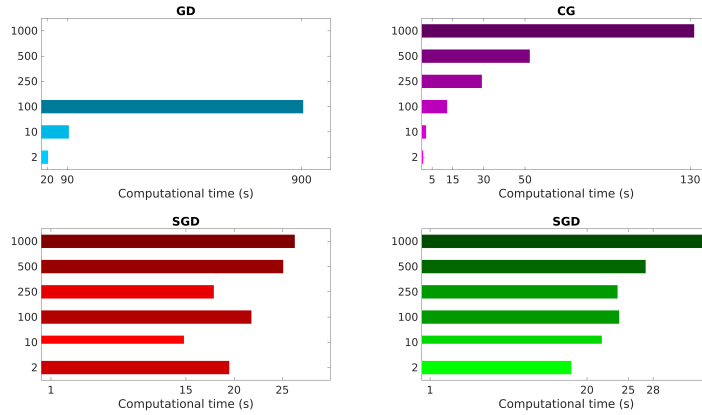


FIGURE 7. Computational time to converge to the tolerance $\varepsilon = 10^{-4}$ of the GD, CG, SGD and CSG algorithms applied to the problem (4.2) with different values of $|\mathcal{K}|$. For GD we do not include the cases $|\mathcal{K}| = 250$, $|\mathcal{K}| = 500$ and $|\mathcal{K}| = 1000$ because of the excessive computational cost.

N	CG		CSG	
	Iter.	Time	Iter.	Time
10	8	11.4s	1704	28.6s
100	14	21.3s	1329	37.8s
500	14	47.8s	1201	37.8s
750	16	139.2s	1384	88.8s
1000	16	267.1s	1421	209.1s

TABLE 3. Number of iterations and computational time to converge to the tolerance $\varepsilon = 10^{-4}$ of the CG and CSG algorithms applied to the problem (4.2) with different values of N .

We can observe some similarities with our previous analysis corresponding to the data collected in Table 2. In particular, our numerical experiments show that there is a threshold for N determining which one of the two algorithms we considered is the most efficient. As it is expectable, CG shows up to be a better approach for low and moderate values of N , but it is outperformed by CSG when N takes a large value.

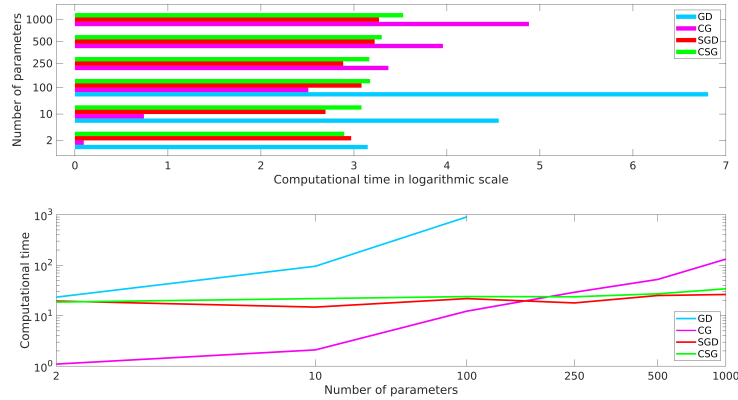


FIGURE 8. Computational time (in logarithmic scale) to converge to the tolerance $\varepsilon = 10^{-4}$ of the GD, CG, SGD and CSG algorithms applied to the problem (4.2) with different values of $|\mathcal{K}|$.

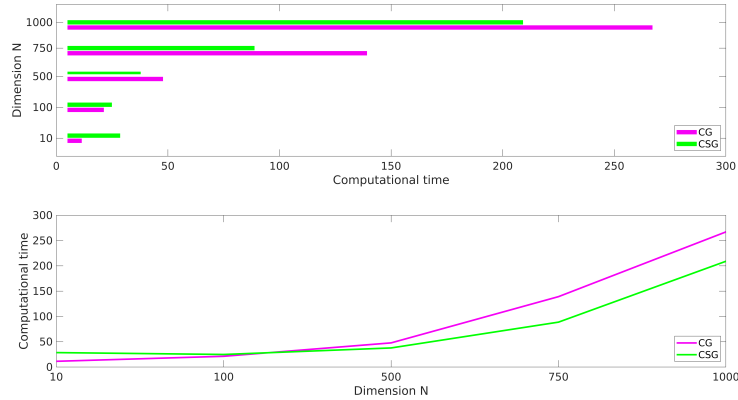


FIGURE 9. Computational time (in logarithmic scale) to converge to the tolerance $\varepsilon = 10^{-4}$ of the GD, CG, SGD and CSG algorithms applied to the problem (4.2) with different values of N .

Notice however that, in contrast with the behavior we observed and commented in our previous simulations, in these last experiments that we performed the computational time of CSG is also affected by N . This is natural since, whereas in each iteration of the algorithm we are still solving only one randomly picked realization of the dynamics, the underneath direct and adjoint systems are increasing in their dimension, which makes their numerical approximation each time more costly. For this reason, although it is still evident the gain of the stochastic approach on the deterministic one as N grows, the difference among the computational times is not as pronounced as in the cases we examined in precedence.

5. CONCLUSIONS AND OPEN PROBLEMS

In this paper, we compared the GD, CG, SGD and CSG algorithms for the minimization of a quadratic functional associated to the simultaneous controllability of linear parameter-dependent models.

The main difference between the four methodologies is that the former two require, at each gradient iteration, the resolution of the dynamics (forward and backward) for all realizations of the parameters. On the other hand, every iteration of SGD and CSG selects randomly, with uniform distribution, a single parameter from \mathcal{K} and uses only the dynamics generated by this choice.

For this reason, one may naively expect that the SGD and CSG methodologies are always computationally cheaper than the GD and CG ones. Our analysis exhibits that this is not necessarily the case and that the performances of each approach are strongly dependent on the amount of parameters in the model considered.

As a matter of fact, our numerical simulations presented in Section 4 shows the two following features:

1. The GD approach is the worst one in terms of the computational complexity. This is so because, although the algorithm is expected to have exponential convergence, in practical applications this rate may dramatically deteriorate as a consequence of the bad conditioning of the problem one aims to solve.
2. The choice of SGD and CSG instead of CG is preferable only when dealing with parameter sets of large cardinality $|\mathcal{K}|$. Indeed, the convergence of these stochastic approaches, independently of the characteristics of the problem treated, is generally worst than CG due to the noise introduced by the random selection of the parameter in each iteration. In view of that, although CG has a higher cost per iteration with respect to SGD and CSG, its faster convergence rate is able to compensate this gap when dealing with problems depending on a small or moderate amount of parameters.

We now conclude our discussion by introducing some suggestions of future research related to the topics addressed in this paper.

1. *Comparison with the greedy methodology.* In the context of the control of parameter-dependent linear equations, in the recent years several authors proposed the employment of the so-called *greedy methodology*. See, for instance, [26, 35]. This approach aims to approximate the dynamics and controls by identifying the most meaningful realizations of the parameters. In certain situations, it showed up to be a very efficient way to largely reduce the overall computational complexity. Therefore, a comparison of the greedy and stochastic approaches applied to parameter-dependent models, with the final aim of clearly determine in which situation one technique is preferable with respect to the other, becomes a very relevant issue.
2. *Simultaneous controllability of PDE models.* The study in this contribution is focused on the scenario of simultaneous controllability of finite-dimensional ODE systems. It would be interesting to extend our analysis to the infinite-dimensional PDE context. This, however, may be a cumbersome task. As a matter of fact, in the PDE setting, simultaneous controllability is a quite delicate issue because of the appearance of peculiar phenomena which are not detected at the ODE level. For some PDE systems, simultaneous controllability may be understood by looking at the spectral properties of the model. Roughly speaking, one needs all the eigenvalues to have multiplicity one in order to be able to observe every eigenmode independently. This fact generally yields restrictions to the validity of simultaneous controllability. For instance, in [17], the authors considered a star-shaped networks of vibrating strings, modeled as a system of wave equations on a graph, and showed that simultaneous controllability holds through a control acting on the junction point, provided that the length of all the edges are mutually irrational. This kind of restrictions appears also at the numerical level, and need to be carefully handled when developing efficient optimization algorithms. Therefore, the analysis of simultaneous controllability properties for a wide class of PDE models, both in the continuous and in the discrete framework, becomes a very interesting issue which deserves a deeper investigation.

ACKNOWLEDGMENTS

The authors wish to acknowledge Léon Bottou (Facebook AI Research) and Víctor Hernández-Santamaría for interesting discussions and clarification on some key points of this article. A special thanks goes to Lukas Pflug (Friedrich-Alexander-Universität, Erlangen-Nürnberg) for his valuable help in the implementation of the CSG algorithm.

REFERENCES

- [1] ANDERSON, R. N., BOULANGER, A., POWELL, W. B., AND SCOTT, W. Adaptive stochastic control for the smart grid. *IEEE Proc.* 99, 6 (2011), 1098–1115.
- [2] ATWELL, J. A., AND KING, B. B. Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations. *Math. Comput. Model.* 33, 1-3 (2001), 1–19.

- [3] BACH, F. R., AND MOULINES, E. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems* (2011), pp. 451–459.
- [4] BADER, E., KÄRCHER, M., GREPL, M. A., AND VEROY, K. Certified reduced basis methods for parametrized distributed elliptic optimal control problems with control constraints. *SIAM J. Sci. Comput.* 38, 6 (2016), A3921–A3946.
- [5] BICCARI, U., AND ZUAZUA, E. A stochastic approach to the synchronization of coupled oscillators. *arXiv preprint 2002.04472* (2020).
- [6] BOTTOU, L. Online learning and stochastic approximations. *On-line learning in neural networks* 17, 9 (1998), 142.
- [7] BOTTOU, L. Large-scale machine learning with stochastic gradient descent. In *Proc. COMPSTAT'2010*. Springer, 2010, pp. 177–186.
- [8] BOTTOU, L., AND BOUSQUET, O. The tradeoffs of large scale learning. In *Advances in neural information processing systems* (2008), pp. 161–168.
- [9] BOTTOU, L., CURTIS, F. E., AND NOCEDAL, J. Optimization methods for large-scale machine learning. *SIAM Rev.* 60, 2 (2018), 223–311.
- [10] BOYD, S., AND VANDENBERGHE, L. *Convex Optimization*. Cambridge University Press, Cambridge, 2004.
- [11] BOYER, F. On the penalised HUM approach and its applications to the numerical approximation of null-controls for parabolic problems. *ESAIM Proc.* 41 (2013), 15–58.
- [12] BRUNTON, S. L., NOACK, B. R., AND KOUMOUTSAKOS, P. Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* 52 (2020), 477–508.
- [13] BUFFA, A., MADAY, Y., PATERA, A. T., PRUDHOMME, C., AND TURINICI, G. A priori convergence of the greedy algorithm for the parametrized reduced basis method. *ESAIM: Math. Model. Numer. Anal.* 46, 3 (2012), 595–603.
- [14] CIARLET, P. G. *Mathematical Elasticity: Volume I: three-dimensional elasticity*. North-Holland, 1988.
- [15] COHEN, A., AND DEVORE, R. Kolmogorov widths under holomorphic mappings. *IMA J. Numer. Anal.* 36, 1 (2015), 1–12.
- [16] CORON, J.-M. *Control and nonlinearity*. American Mathematical Soc., 2007.
- [17] DÁGER, R., AND ZUAZUA, E. Controllability of star-shaped networks of strings. *C. R. Acad. Sci. Ser. I Math.* 332, 7 (2001), 621–626.
- [18] DEDÈ, L. Reduced basis method and a posteriori error estimation for parametrized linear-quadratic optimal control problems. *SIAM J. Sci. Comput.* 32, 2 (2010), 997–1019.
- [19] DENG, R., YANG, Z., CHEN, J., AND CHOW, M.-Y. Load scheduling with price uncertainty and temporally-coupled constraints in smart grids. *IEEE Trans. Power Syst.* 29, 6 (2014), 2823–2834.
- [20] DENTCHEVA, D., AND RÖMISCH, W. Optimal power generation under uncertainty via stochastic programming. In *Stochastic programming methods and technical applications*. Springer, 1998, pp. 22–56.
- [21] DÍAZ, E., AND GUITTON, A. Fast full waveform inversion with random shot decimation. In *SEG Technical Program Expanded Abstracts 2011*. Society of Exploration Geophysicists, 2011, pp. 2804–2808.
- [22] DUCHI, J., HAZAN, E., AND SINGER, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* 12, Jul (2011), 2121–2159.
- [23] DURRETT, R. *Probability: theory and examples*. Cambridge university press, Fourth edition, 2010.
- [24] FALKOVICH, G. *Fluid mechanics*. Cambridge University Press, 2011.
- [25] GLOWINSKI, R. *Variational methods for the numerical solution of nonlinear elliptic problems*. SIAM, 2015.
- [26] HERNÁNDEZ-SANTAMARÍA, V., LAZAR, M., AND ZUAZUA, E. Greedy optimal control for elliptic problems and its application to turnpike problems. *Numer. Math.* 141, 2 (2019), 455–493.
- [27] ITO, K., AND RAVINDRAN, S. S. A reduced basis method for control problems governed by PDEs. In *Control and estimation of distributed parameter systems*. Springer, 1998, pp. 153–168.
- [28] ITO, K., AND RAVINDRAN, S. S. Reduced basis method for optimal control of unsteady viscous flows. *Int. J. Comput. Fluid D.* 15, 2 (2001), 97–113.
- [29] JAEGER, J. C., AND CARSLAW, H. S. *Conduction of heat in solids*. Clarendon P, 1959.
- [30] JAMESON, A. *Gradient Based Optimization Methods*. MAE Technical Report No. 2057, Princeton University, 1995.
- [31] JIN, S., LI, L., AND LIU, J.-G. Random Batch Methods (RBM) for interacting particle systems. *J. Comput. Phys.* 400 (2020), 108877.
- [32] KÄRCHER, M., AND GREPL, M. A. A posteriori error estimation for reduced order solutions of parametrized parabolic optimal control problems. *ESAIM: Math. Model. Numer. Anal.* 48, 6 (2014), 1615–1638.
- [33] KO, D., AND ZUAZUA, E. Model predictive control with random batch methods for a guiding problem. *preprint* (2020).
- [34] KUNISCH, K., AND VOLKWEIN, S. Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition. *J. Optim. Theory Appl.* 102, 2 (1999), 345–371.
- [35] LAZAR, M., AND ZUAZUA, E. Greedy controllability of finite dimensional linear systems. *Automatica* 74 (2016), 327–340.
- [36] LOHÉAC, J., AND ZUAZUA, E. From averaged to simultaneous controllability. *Ann. Fac. Sci. Toulouse Math.* 25, 4 (2016), 785–828.
- [37] MEZA, J. C. Steepest descent. *WIRES Comput. Stat.* 2, 6 (2010), 719–722.
- [38] NESTEROV, Y. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. In *Doklady AN USSR* (1983), vol. 269, pp. 543–547.
- [39] NESTEROV, Y. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Science + Business Media, New York, 2004.
- [40] NOCEDAL, J., AND WRIGHT, S. *Numerical optimization*. Springer Science & Business Media, 2006.

- [41] PFLUG, L., BERNHARDT, N., GRIESHAMMER, M., AND STINGL, M. A new stochastic gradient method for the efficient solution of structural optimization problems with infinitely many state problems. *preprint* (2020).
- [42] RAVINDRAN, S. S. A reduced-order approach for optimal control of fluids using proper orthogonal decomposition. *Int. J. Numer. Methods Fluids* 34, 5 (2000), 425–448.
- [43] ROBBINS, H., AND MONRO, S. A stochastic approximation method. *Ann. Math. Stat.* 22, 3 (1951), 400–407.
- [44] SAAD, Y. *Iterative methods for sparse linear systems*, vol. 82. siam, 2003.
- [45] SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural networks* 61 (2015), 85–117.
- [46] SCHÖLKOPF, B., AND SMOLA, A. J. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.
- [47] SHALEV-SHWARTZ, S., AND SREBRO, N. SVM optimization: inverse dependence on training set size. In *Proceedings of the 25th international conference on Machine learning* (2008), ACM, pp. 928–935.
- [48] SHALEV-SHWARTZ, S., AND ZHANG, T. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *International conference on machine learning* (2014), pp. 64–72.
- [49] TÖSCHER, A., AND JAHRER, M. Collaborative filtering applied to educational data mining. *KDD cup* (2010).
- [50] TRÉLAT, E. *Contrôle optimal: théorie & applications*. Vuibert, 2005.
- [51] TUCSNAK, M., AND WEISS, G. *Observation and control for operator semigroups*. Springer Science & Business Media, 2009.
- [52] VAPNIK, V. Principles of risk minimization for learning theory. In *Advances in neural information processing systems* (1992), Morgan Kaufmann, pp. 831–838.

UMBERTO BICCARI, CHAIR OF COMPUTATIONAL MATHEMATICS, FUNDACIÓN DEUSTO, AVDA. DE LAS UNIVERSIDADES 24, 48007 BILBAO, BASQUE COUNTRY, SPAIN.

UMBERTO BICCARI, UNIVERSIDAD DE DEUSTO, AVDA UNIVERSIDADES 24, 48007 BILBAO, BASQUE COUNTRY, SPAIN.
E-mail address: `umberto.biccari@deusto.es,u.biccari@gmail.com`

DEPARTMENT OF STATISTICS AND OPERATIONAL RESEARCH, UNIVERSITAT DE VALÈNCIA, DR. MOLINER 50, 46100, BURJASSOT, SPAIN.

E-mail address: `ana.navarro@uv.es`

ENRIQUE ZUAZUA, CHAIR IN APPLIED ANALYSIS, ALEXANDER VON HUMBOLDT-PROFESSORSHIP, DEPARTMENT OF MATHEMATICS FRIEDRICH-ALEXANDER-UNIVERSITÄT ERLANGEN-NÜRNBERG, 91058 ERLANGEN, GERMANY.

ENRIQUE ZUAZUA, CHAIR OF COMPUTATIONAL MATHEMATICS, FUNDACIÓN DEUSTO, AVDA. DE LAS UNIVERSIDADES 24, 48007 BILBAO, BASQUE COUNTRY, SPAIN.

ENRIQUE ZUAZUA, DEPARTAMENTO DE MATEMÁTICAS, UNIVERSIDAD AUTÓNOMA DE MADRID, 28049 MADRID, SPAIN.
E-mail address: `enrique.zuazua@fau.de`