A STOCHASTIC APPROACH TO THE SYNCHRONIZATION OF COUPLED OSCILLATORS

UMBERTO BICCARI AND ENRIQUE ZUAZUA

ABSTRACT. This paper deals with an optimal control problem associated with the Kuramoto model describing the dynamical behavior of a network of coupled oscillators. Our aim is to design a suitable control function allowing us to steer the system to a synchronized configuration in which all the oscillators are aligned on the same phase. This control is computed via the minimization of a given cost functional associated with the dynamics considered. For this minimization, we propose a novel approach based on the combination of a standard Gradient Descent (GD) methodology with the recently-developed Random Batch Method (RBM) for the efficient numerical approximation of collective dynamics. Our simulations show that the employment of RBM improves the performances of the GD algorithm, reducing the computational complexity of the minimization process and allowing for a more efficient control calculation.

1. INTRODUCTION

Synchronization is a common phenomenon which has been observed in biological, chemical, physical and social systems for centuries and has attracted the interest of researcher in a diversified spectrum of scientific fields.

Common examples of synchronization phenomena often cited in review articles include groups of synchronously chirping crickets ([44]), fireflies flashing in unison ([5]), superconducting Josephson junction ([45]), or crowds of people walking together that will tend to synchronize their footsteps ([36]).

Roughly speaking, synchronization means that a network of several periodic processes with different natural frequencies reaches an equilibrium configuration sharing the same common frequency as a result of their mutual interaction.

This concept is closely related to the one of consensus for multi-agent systems, widely analyzed in many different frameworks including collective behavior of flocks and swarms, opinion formation, and distributed computing (see [2, 3, 23, 27, 28]). In broad terms, consensus means to reach an agreement regarding a certain quantity of interest that depends on the state of all agents.

Synchronization is also a key issue in power electrical engineering, for instance in the model and stability analysis of utility power grids ([6, 13, 30, 35]). Indeed, large networks of connected power plants need to be synchronized to the same frequency in order to work properly and prevent the occurrence of blackouts.

Synchronization phenomena are most often characterized by the so-called Kuramoto model ([21]), describing the dynamical behavior of a (large) network of oscillators in a *all-to-all* coupled configuration in which every oscillator is connected with all the others. This model extends the original studies by Winfree in the context of mutual synchronization in multi-oscillator systems based on a phase description ([46]). In particular, in Kuramoto's work, synchronization appears as an asymptotic pattern which is spontaneously reached by the system when the interactions among the oscillators are sufficiently strong.

In some more recent contribution, control theoretic methods have been employed to analyze the synchronization phenomenon. For instance, in [8] the authors design passivity-based controls for the synchronization

of multi-agent systems, with application to the general problem of multi-robot coordination. In [33], feedback control laws for the stabilization of coupled oscillators are designed and analyzed. In [29, 42], the authors propose methods for the suppression of synchrony in a globally coupled oscillator network, based on (possibly time-delayed) feedback schemes. Finally, [24] deals with the problem of desynchronizing a network of synchronized and globally coupled neurons using an input to a single neuron. This is done in the spirit of dynamic programming, by minimizing a certain cost function over the whole state space.

In this work, we address the synchronization problem for coupled oscillators through the construction of a suitable control function via an appropriate optimization process. To this end, we propose a novel approach which combines a standard Gradient Descent (GD) methodology with the recently-developed Random Batch Method (RBM, see [20]) for the efficient numerical approximation of collective dynamics. This methodology has the main advantages of allowing to significantly reduce the computational complexity of the optimization process, especially when considering oscillator networks of large size, yielding to an efficient control calculation.

At this regard, we shall mention that GD methodologies have already been applied in the context of the Kuramoto model. For instance, in [38], the author develop GD algorithms to efficiently solve optimization problems that aim to maximize phase synchronization via network modifications. Moreover, in [22], optimization and control theory techniques are applied to investigate the synchronization properties of a generalized Kuramoto model in which each oscillator lives on a compact, real Stiefel manifold. Nevertheless, to the best of our knowledge, the employment of stochastic techniques such as RBM to improve the efficiency of the GD strategy has never been proposed in the context of the Kuramoto model.

For completeness, let us stress that stochastic approaches have been widely considered, especially by the machine learning community, for treating minimization problems depending on very large data set. In this context, they have shown amazing performances in terms of the computational efficiency (see, for instance, [4] and the references therein). Nowadays, stochastic techniques are among the preeminent optimization methods in fields like empirical risk minimization ([34]), data mining ([39]) or artificial neural networks ([31]).

This contribution is organized as follows: in Section 2, we present the Kuramoto model and we discuss some of its more relevant properties. We also provide there a rigorous mathematical characterization of the synchronization phenomenon. In Section 3, we introduce the controlled Kuramoto model and we describe the GD methodology for the control computation. Moreover, we briefly present the RBM approach and its inclusion into the GD algorithm. Section 4 is devoted to the numerical simulations and to the comparison of the two optimization techniques considered in this paper. Finally, in Section 5 we summarize and discuss our results.

2. The mathematical model

From a mathematical viewpoint, synchronization phenomena are most often described through the so-called Kuramoto model, consisting of a population of $N \geq 2$ coupled oscillators whose dynamics are governed by the following system of non-linear first-order ordinary differential equations

$$(2.1) \quad \begin{cases} \dot{\theta}_i(t) = \omega_i + \dfrac{K}{N} \displaystyle\sum_{j=1}^{N} \sin\big(\theta_j(t) - \theta_i(t)\big), \quad i = 1, \ldots, N, \quad t > 0 \\ \theta_i(0) = \theta_i^0, \end{cases}$$

where $\theta_i(t)$, $i = 1, \ldots, N$, is the phase of the i-th oscillator, ω_i is its natural frequency and K is the coupling strength.

The frequencies ω_i are assumed to be distributed with a given probability density $f(\omega)$, unimodal and symmetric around the mean frequency

$$\Omega = \frac{1}{N} \sum_{i=1}^{N} \omega_i,$$

that is, $f(\Omega + \omega) = f(\Omega - \omega)$.

In this framework, each oscillator tries to run independently at its own frequency, while the coupling tends to synchronize it to all the others.

In the literature, many notions of synchronization have been considered. For identical oscillators (i.e., those in which $\omega_i = \widehat{\omega}$ for every $i = 1, \ldots, N$), one often studies whether the network can reach a configuration in which all the phases converge to the same value, that is

$$(2.2) \qquad \lim_{t \to +\infty} |\theta_i(t) - \theta_j(t)| = 0, \quad \text{for all} \ \ i, j = 1, \ldots, N.$$

For systems with heterogeneous dynamics, such as when the natural frequencies ω_i are not all identical (which is typical in real-world scenarios), this definition of synchronization is too restrictive (see [37]). In these cases, (2.2) is replaced by the alignment condition

$$(2.3) \qquad \lim_{t \to +\infty} |\dot{\theta}_i(t) - \dot{\theta}_j(t)| = 0, \quad \text{for all} \ \ i, j = 1, \ldots, N,$$

according to which synchronization occurs when the phase differences given by $|\theta_i(t) - \theta_j(t)|$ become constant asymptotically for all $i, j \in 1, \ldots, N$. This notion (2.3), which in some references is called *complete synchronization* (see for instance [15]), is the one that we will consider in this work.

In its original work [21], Kuramoto considered the continuum limit case where $N \to +\infty$ and showed that the coupling K has a key role in determining whether a network of oscillators can synchronize. In more detail, he showed that, when the coupling K is weak, the oscillators run incoherently, whereas beyond a certain threshold collective synchronization emerges.

Later on several research works provided specific bounds for the threshold of K ensuring synchronization (see, e.g., [1, 7, 9, 11, 12, 19]). In particular, in order to achieve (2.3) it is enough that

$$(2.4) \qquad K > K^* = |\omega_{max} - \omega_{min}|,$$

where $\omega_{min} < \omega_{max}$ are the minimum and maximum natural frequencies.

Notice, however, that (2.3) is an asymptotic characterization, meaning that is satisfied as $t \to +\infty$. In this work we are rather interested in the possibility of synchronizing the oscillators in a finite time horizon T. As we will discuss in the next section, this may be achieved by introducing a control into the Kuramoto model (2.1).

3. Optimal control of the Kuramoto model

As we mentioned in Section 2, in this work we are interested in the finite-time synchronization of the Kuramoto model. In particular, we aim at designing a control capable to steer the Kuramoto dynamics (2.1) to synchronization in a final time horizon T. In other words, we are going to consider the controlled system

$$(3.1) \qquad \begin{cases} \dot{\theta}_i(t) = \omega_i + \dfrac{Ku(t)}{N} \displaystyle\sum_{j=1}^{N} \sin\left(\theta_j(t) - \theta_i(t)\right), \quad i = 1, \ldots, N, \quad t > 0 \\ \theta_i(0) = \theta_i^0, \end{cases}$$

and we want to compute a control function u such that the synchronized configuration (2.3) is achieved at time T, i.e.,

$$(3.2) \qquad |\dot{\theta}_i(T) - \dot{\theta}_j(T)| = 0, \quad \text{for all} \ i, j = 1, \ldots, N.$$

From the practical applications viewpoint, this problem may be assimilated for instance to the necessity of synchronizing all the components of an electric grid after a black-out. In this interpretation, the different elements in the grid are represented by the oscillators in (3.1), and T is the time horizon we provide for the black-start, being therefore an external input to our problem. The objective is then to complete restoring the network in a finite (possibly small) time T, which can be done by introducing a control in the system.

To compute this optimal control allowing us to reach the synchronized configuration (3.2) we will adopt a classical optimization approach based on the resolution of the following optimization problem

$$\widehat{u} = \min_{u \in L^2(0,T;\mathbb{R})} J(u)$$

$$(3.3) \qquad J(u) = \frac{1}{2} \sum_{i,j=1}^{N} \sin^2\left(\theta_j(T) - \theta_i(T)\right) + \frac{\beta}{2} \|u\|_{L^2(0,T;\mathbb{R})}^2,$$

3

subject to the dynamics (3.1). Here, with $L^2(0,T;\mathbb{R})$ we denoted the space of all functions $u : (0,T) \to \mathbb{R}$ for which the following norm is finite:

$$ (3.4) \qquad \|u\|_{L^2(0,T;\mathbb{R})} := \left(\int_0^T |u(t)|^2 \, dt \right)^{\frac{1}{2}}. $$

In what follows, we will use the abridged notation $\|u\|_2 := \|u\|_{L^2(0,T;\mathbb{R})}$.

In the cost functional (3.3), the first term enhances the fact that all the oscillators have to synchronize at time T. In particular, the optimal control $\widehat{u}$ will yield to a dynamics in which

$$ (3.5) \qquad \sin(\theta_j(T) - \theta_i(T)) = 0 \quad \Rightarrow \quad \theta_j(T) - \theta_i(T) = k\pi, \ k \in \mathbb{Z}. $$

This is consistent with (3.2). For completeness, let us also stress that, in the case of identical oscillators, it has been shown for instance in [14] that, at least asymptotically, the two notions (3.2) and (3.5) coincide.

The second term in (3.3) is introduced to avoid controls with a too large size. In it, $\beta > 0$ is a (usually small) penalization parameter which allows to tune the norm of the optimal control $\widehat{u}$. Roughly speaking, the smaller is β the larger will be $\widehat{u}$. A more detailed discussion on this point will be presented in Section 4.

Through the minimization of $J(u)$, we will obtain a unique scalar control function $\widehat{u} : (0,T) \to \mathbb{R}$, $\widehat{u} > 1$, for all the oscillators included in the network. In other words, we are going to define a unique control law which is capable to act globally on the entire oscillator network in order to reach the desired synchronized configuration. This is a different approach than the ones presented in [8, 24, 29, 33, 42] which we mentioned above and are based on designing feedback laws or controlling only some specific components of the model, using the coupling to deal with the uncontrolled ones.

One advantage of the control strategy that we propose is that, requiring only one control computation, from the computational viewpoint is more efficient than a feedback approach which necessitates repeated measurements of the state. Moreover, let us notice that, in (3.1), the control acts as a multiplicative force which increases the coupling among the oscillators, thus enhancing their synchronization properties. In particular, as we will see in our numerical simulations, this will allow to reach synchronization also in situations where K violates the condition (2.4) and the uncontrolled dynamics runs incoherently towards a desynchronized configuration.

Nevertheless, our proposed methodology may have the disadvantage of being less flexible than the others we mentioned above. In particular, it does not allow to control only a specific component of the network and this may be a limitation in certain practical applications.

Let us stress that the above considerations are merely heuristic and should be corroborated by a deeper analysis based, for instance, on sharp numerical experiments. Notwithstanding that, in the present work we will not address this specific issue, since our main interest is not to compare the performances of different control strategies but rather to present an efficient way to tackle the control problem (3.1).

In the optimization literature, several different techniques have been proposed for minimizing the functional $J(u)$ (see, e.g., [26]). In this work, we focus on the standard GD method, which looks for the minimum u as the limit $k \to +\infty$ of the following iterative process

$$ (3.6) \qquad u^{k+1} = u^k - \eta_k \nabla J(u^k), $$

where $\eta_k > 0$ is called the step-size or, in the machine learning context, the learning rate. The step size is typically selected to be a constant depending on certain key parameters of the optimization problem, or following an adaptive strategy. See, e.g., [25, Section 1.2.3] for more details.

This gradient technique is most often chosen because it is easy to implement and not very memory demanding.

Nevertheless, when applied to the optimal control of collective dynamics, the GD methodology has a main drawback. Indeed, as we shall see in Section 3.1, at each iteration k the optimization scheme (3.6) requires to solve (3.1), that is, a N-dimensional non-linear dynamical system. This may rapidly become computationally very expensive, especially when the number N of oscillators in our system is large.

In order to reduce this computational burden, in this work we propose a novel methodology which combines the standard GD algorithm with the so-called *Random Batch Method* (RBM).

RBM is a recently developed approach which has been introduced in [20] for the numerical simulation of high-dimensional collective behavior problems. This method uses small but random batches for particle

interactions, lowering the computational cost $\mathcal{O}(N^2)$ per time step to $\mathcal{O}(N)$, for systems with N particles with binary interactions. Therefore, as our numerical simulations will confirm, embedding RBM into the GD iterative scheme yields to a less expensive algorithm and, consequently, to a more efficient control computation. In what follows, we will call this approach GD-RBM algorithm, to differentiate it from the standard GD one.

3.1. **The Gradient Descent approach.** Let us now describe in detail the GD approach to minimize the functional (3.3), and discuss its convergence properties.

In order to fully define the iterative scheme (3.6), we need to compute the gradient $\nabla J(u)$. Since we are dealing with a non-linear control problem, we will do this via the so-called *Pontryagin maximum principle* (see [41, Chapter 4, Section 4.8] or [40, Chapter 7]).

To this end, let us first rewrite the dynamics (3.1) in a vectorial form as follows

$$(3.7) \qquad \begin{cases} \dot{\Theta}(t) = \Omega + F\big(\Theta(t), u(t)\big), & t > 0 \\ \Theta(0) = \Theta^0, \end{cases}$$

with $\Theta := (\theta_1, \dots, \theta_N)^\top$, $\Theta^0 := (\theta_1^0, \dots, \theta_N^0)^\top$ and $\Omega := (\omega_1, \dots, \omega_N)^\top$, and where F is the vector field given by

$$(3.8) \qquad F = (F_1, \dots, F_N), \quad F_i := \frac{Ku(t)}{N} \sum_{j=1}^{N} \sin\big(\theta_j(t) - \theta_i(t)\big), \quad i = 1, \dots, N.$$

In the control literature, (3.7) is usually called the *primal system*.

Using the notation just introduced, we can then see that $J(u)$ can be rewritten in the form

$$(3.9) \qquad J(u) = \int_0^T L(u(t))\,dt + \phi(\Theta(T)),$$

with

$$L(u(t)) = \frac{\beta}{2}|u(t)|^2 \quad \text{and} \quad \phi(\Theta(T)) = \frac{1}{2}\sum_{j=1}^{N} \sin^2\big(\theta_j(T) - \theta_i(T)\big).$$

Let us stress that (3.9) is in the standard form to apply the Pontryagin maximum principle. Through this approach, we can obtain the following expression for the gradient of $J(u)$

$$(3.10) \qquad \nabla J(u) = \beta u + (\mathcal{D}_u F)^\top p,$$

where $\mathcal{D}_u F$ indicates the Jacobian of the vector field F, computed with respect to the variable u.

In (3.10), we denoted with $p = (p_1, \dots, p_N)$ the solution of the *adjoint equation* associated with (3.1), which is given by

$$(3.11) \qquad \begin{cases} -\dot{p} = (\mathcal{D}_\Theta F)^\top p \\ p(T) = \nabla_{\Theta(T)} \phi(\Theta(T)), \end{cases}$$

where $\mathcal{D}_\Theta F$ stands again for the Jacobian of the vector field F, this time computed with respect to the variable Θ.

Taking into account the expression (3.8) of the vector field F, we can then readily check that the iterative scheme (3.6) becomes

$$(3.12) \qquad u^{k+1} = u^k - \eta_k \left[\beta u^k + \frac{K}{N} \sum_{i=1}^{N} p_i \left(\sum_{j=1}^{N} \sin(\theta_j - \theta_i) \right) \right],$$

with

$$(3.13) \quad \begin{cases} -\dot{p}_i = -\dfrac{Kup_i}{N} \displaystyle\sum_{i\neq j=1}^{N} \cos\left(\theta_j - \theta_i\right) + \dfrac{Ku}{N} \displaystyle\sum_{i\neq j=1}^{N} p_j \cos\left(\theta_j - \theta_i\right), \quad i = 1,\ldots,N, \quad t > 0 \\[4mm] p_i(T) = \dfrac{1}{2} \displaystyle\sum_{i\neq j=1}^{N} \sin\left(2\theta_i(T) - 2\theta_j(T)\right). \end{cases}$$

In view of the above computations, the GD algorithm for the minimization of the cost functional $J(u)$ can be explicitly formulated as follows:

GD algorithm.

  input Θ^0: initial condition of the primal system (3.7)

          u^0: initial guess for the control u

          $k \leftarrow 0$: iteration counter

          k_{max}: maximum number of iterations allowed

          tol: tolerance

  while STOP-CRIT and $k < k_{max}$ **do**

    $k \leftarrow k + 1$

    for $j = 1$ to N **do**

      Solve the the primal system (3.7)

      Solve the the adjoint system (3.13)

    end for

    Update the control through the scheme (3.12)

  end while

  return $u^{k+1} = \widehat{u}$: minimum of the functional $J(u)$.

In particular, we see that the control computation through the above algorithm requires, at each iteration k, to solve $2N$ non-linear differential equations (N for the variables θ_i and N for p_i). If we introduce the time-mesh of N_t points

$$0 = t_0 < t_1 < \ldots < t_{N_t} = T, \quad t_m = t_0 + m\frac{T}{N_t}, \quad m = 1,\ldots,N_t,$$

at each time-step t_m this operation has a computational cost of $\mathcal{O}(N^2)$ and the total computational complexity for the simulation of (3.1) and (3.13) will then be $\mathcal{O}(N_t N^2)$. If N is large, that is, if the number of oscillators in the network is considerable, this will rapidly become very expensive.

3.2. **The Random Batch Method.** In order to reduce the computational burden of GD for the optimization process (3.3), we propose a modification of this algorithm which includes the aforementioned *Random Batch Method* (RBM) for the numerical simulation of the ODE systems (3.1) and (3.13).

This technique, presented in [20] for interacting particle systems, is based on the following simple idea: at each time step $t_m = m \cdot dt$ in the mesh we employ to solve the dynamics, we divide randomly the N particles into n small batches with size $2 \leq P < N$, denoted by C_q, $q = 1,\ldots,n$, that is

$$\begin{aligned} C_q &= \{i_{q_1},\ldots,i_{q_P}\} \subset \{1,\ldots,N\}, && \text{for all } q = 1,\ldots,n \\ C_q \cap C_r &= \emptyset, && \text{for all } q,r = 1,\ldots,n \\ \bigcup_{q=1}^{n} C_q &= \{1,\ldots,N\}. \end{aligned}$$

Notice that the last batch C_n may have size smaller than P if $nP \neq N$.

Once this partition of $\{1,\ldots,N\}$ has been performed, we solve the dynamics by interacting only particles within the same batch. This gives the following algorithm for the numerical approximation of (3.1) and (3.13):

RBM algorithm.

    for $m = 1$ to $N_t = T/dt$ **do**

        Divide randomly $\{1, \ldots, N\}$ into n batches C_q, $q = 1, \ldots, n$

        for $q = 1$ to n **do**

            Update θ_i $(i \in C_q)$ by solving the ODE

$$\begin{cases} \dot{\theta}_i = \omega_i + \dfrac{Ku}{P} \sum_{j \in C_q} \sin\left(\theta_j - \theta_i\right) \\ \theta_i(0) = \theta_i^0. \end{cases}$$

        Update p_i $(i \in C_q)$ by solving the ODE

$$\begin{cases} -\dot{p}_i = -\dfrac{Kup_i}{P} \sum_{j \in C_q} \cos\left(\theta_j - \theta_i\right) + \dfrac{Ku}{P} \sum_{i \neq j \in C_q} p_j \cos\left(\theta_j - \theta_i\right) \\ p_i(T) = \dfrac{1}{2} \sum_{i \neq j \in C_q} \sin\left(2\theta_i(T) - 2\theta_j(T)\right). \end{cases}$$

      end for

    end for

Regarding the complexity, note that random division into n batches of can be implemented using random permutation. In Matlab, this can be done by using the function *randperm(N)*. Then, the first P elements are considered to be in the first batch, the second P elements are in the second batch, and so on. According to the discussion presented in [20], at each time step t_m this procedure yields to a cost of $\mathcal{O}(PN)$ for approximating the dynamics with RBM.

If one is to simulate up to time T, the total number of time steps is N_t as in the algorithm above. Then, the total computational complexity for the simulation of (3.1) and (3.13) is $\mathcal{O}(PN_tN)$. Notice that, since $P < N$, this is always smaller than $\mathcal{O}(N_tN^2)$.

Summarizing, with the GD and GD-RBM methodologies we obtain the following per-iteration costs:

- GD $\longrightarrow$ $\mathrm{cost}_{GD} = \mathcal{C}_{GD} N_t N^2$.
- GD-RBM $\longrightarrow$ $\mathrm{cost}_{GD-RBM} = \mathcal{C}_{GD-RBM} P N_t N$.

Therefore, independently of the value of N, employing RBM to simulate the dynamics in each iteration of GD yields improvements in terms of the computational cost.

For completeness, we shall mention that the above considerations are simply heuristic and would require a deeper analysis. As a matter of fact, to have a rigorous validation of the reduction in the computational complexity when using RBM one should have more precise information on the two constants $\mathcal{C}_{GD}$ and $\mathcal{C}_{GD-RBM}$, and be sure that the difference among them does not overwhelm the help that the batching procedure is providing.

At this regard, let us stress that the RBM method has been developed only recently in [20] and, at present time, there is not a well-established qualitative analysis on its computational cost, going in more detail than what we mentioned above. The evidence that in our case of the Kuramoto model (3.1) the GD-RBM method allows for a more efficient control computation, in particular for large oscillator networks, will then be given through the numerical simulations in Section 4.

3.3. **Convergence analysis.** To complete this section, let us briefly comment about the convergence properties of the GD methodology. It is nowadays classically known that the convergence rate of the GD algorithm is determined by the regularity of the objective function. In our case, since $J(u)$ is L-smooth, that is

$$\|\nabla J(u) - \nabla J(v)\|_2 \leq L\|u - v\|_2,$$

it can be proven that

(3.14) $$\|\nabla J(u^k)\|_2 \to 0 \quad \text{as } k \to +\infty$$

and

(3.15)
$$\|J(u^k) - J(\widehat{u})\|_2 = \mathcal{O}\left(\frac{1}{k}\right),$$

where, we recall, $\widehat{u}$ denotes the minimum of $J(u)$ and the norm $\|\cdot\|_{L^2(0,T;\mathbb{R})}$ has been defined in (3.4).

In particular, (3.15) implies that for achieving ε-optimality, i.e. for obtaining $\|J(u^k) - J(\widehat{u})\|_2 < \varepsilon$, the GD algorithm requires $k = \mathcal{O}(\varepsilon^{-1})$ iterations.

Combining this with the per-iteration costs we gave at the end of Section 3.2, we can thus obtain the following total computational costs

- GD: $\mathcal{O}\left(\frac{N_t N^2}{\varepsilon}\right)$
- GD-RBM: $\mathcal{O}\left(\frac{P N_t N}{\varepsilon}\right)$,

and we can conclude that the GD-RBM approach will be more efficient than the standard GD one to solve our optimization problem. This is enhanced for large values of N and will be confirmed by our numerical simulations.

4. NUMERICAL SIMULATIONS

We present here our numerical results for the control of N coupled oscillators described by the Kuramoto model (3.1), following the strategy previously described. This section is divided into two parts:

1. In a first moment, we will show that the optimization problem (3.3) indeed allows to compute an effective control function which is capable to steer the Kuramoto model (3.1) to a synchronized configuration. This will be done both for a strong coupling $K > K^*$ (see (2.4)) and for a weak coupling $K < K^*$. Besides, we will also briefly analyze the role of the parameter β in the optimization process. Finally, we will show the efficacy of our control strategy in the more realistic cases of a sparse interaction network and for a second-order Kuramoto model with damping.

2. Once the effectiveness of the control strategy we propose has been corroborated, we will compare the GD and GD-RBM algorithms for the minimization of $J(u)$. In particular, we will show how the RBM approach allows to significantly reduce the computational complexity of the GD algorithm for the calculation of the control u, especially when considering oscillator networks of large dimension.

The oscillators are chosen such that their natural frequencies are given following the normal probability law

(4.1)
$$f(\omega) = \frac{1}{5\sigma\sqrt{2\pi}} e^{-\frac{\omega^2}{2\sigma}},$$

with $\sigma = 0.1$. This means that the values of ω_{min} and ω_{max} are given respectively by

$$\omega_{\min} = \min_{\omega \in \mathbb{R}} f(\omega) = 0$$

$$\omega_{\max} = \max_{\omega \in \mathbb{R}} f(\omega) = f(0) = \frac{2}{\sqrt{2\pi}}$$

and the coupling gain K which is necessary for synchronization in the absence of a control has to satisfy (see (2.4))

$$K > |\omega_{max} - \omega_{min}| = \frac{2}{\sqrt{2\pi}}.$$

The initial datum θ^0 is chosen following a normal distribution as well, in such a way that $|\theta_i^0 - \theta_j^0| < 2\pi$ for all $i, j = 1, \ldots, N$. Let us stress that this choice of θ^0 allows the synchronization of the uncontrolled model, as it has been shown for instance in [10].

Without loss of generality, we considered the time horizon $T = 3s$ for completing the synchronization. That is, we want all the oscillators in our model to reach the configuration (3.2) in three seconds.

Finally, we used an explicit Euler scheme for solving the direct and adjoint dynamics (3.1) and (3.13) during the minimization of $J(u)$, and we chose as a stopping criterion

(4.2)
$$e_k := \frac{\|\nabla J(u^k)\|_2}{\|u^k\|_2} < \varepsilon,$$

with $\varepsilon = 10^{-4}$, and where the notation $\|\cdot\|_2$ stands again for the $L^2(0,T\,\mathbb{R})$-norm defined in (3.4). Let us stress that the stopping criterion (4.2) is consistent with (3.14) and (3.15).

4.1. **Computation of the optimal control.** In this section, we show that through the optimization problem (3.3) we are able to compute an effective control function which is capable to steer the Kuramoto model (3.1) to a synchronized configuration in a given time horizon T.

We performed the simulations in Matlab R2018a on a laptop with Intel Core $i5-7200UCPU@2.50GHz \times 4$ processor and 7.7 GiB RAM.

We start by considering a simple scenario of $N = 10$ oscillators in an all-to-all coupled configuration and with a coupling gain $K > K^*$. Moreover, we set the penalization parameter β in (3.3) to take the value $\beta = 10^{-7}$

When using the GD-RBM approach, the family of $N = 10$ oscillators has been separated into $n = 5$ batches of size $P = 2$.

In Figure 1-top, we show the evolution of the uncontrolled dynamics, which corresponds to taking $u \equiv 1$ in (3.1). As we can see, the oscillators are evolving towards a synchronized configuration, which is consistent with our choice of the coupling gain K. Nevertheless, synchronization is not reached in the short time horizon we are providing. At this regard, let us remark that, for the uncontrolled Kuramoto model with a sufficiently strong coupling gain, synchronization is expected to be reached only asymptotically, i.e. when $t \to +\infty$.

In Figure 1-bottom, we show the evolution of the same dynamics, this time under the action of the control function u computed through the minimization of $J(u)$. The subplot on the left corresponds to the simulations done with the GD approach, while the one on the right is done employing GD-RBM. We can clearly see how, in both cases, the oscillators are all synchronized at the final time $T = 3s$. This means that both algorithms managed to compute an effective control.
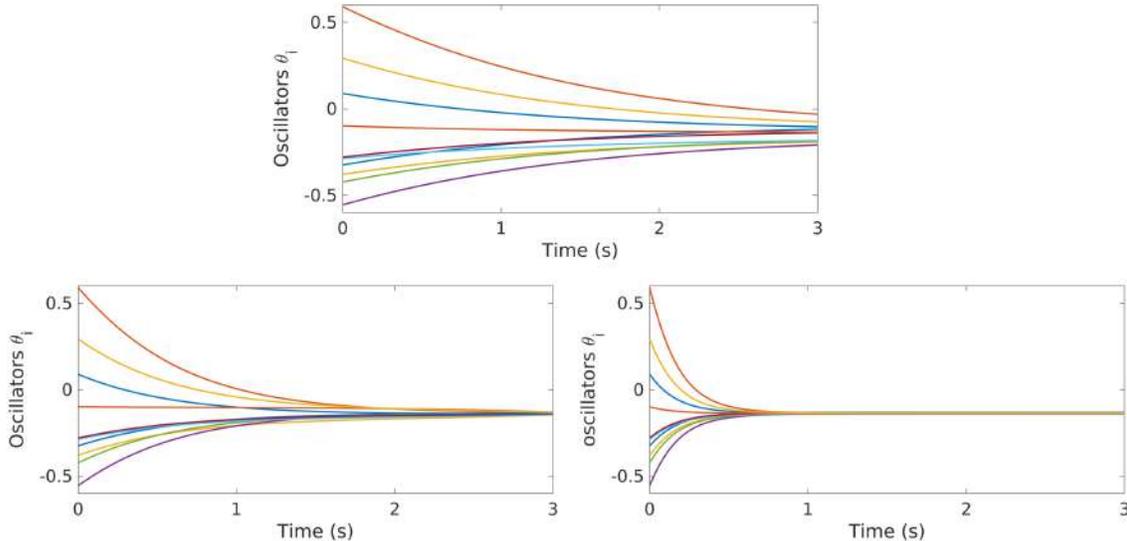

FIGURE 1. Top: evolution of the free dynamics of the Kuramoto model (3.1) with $N = 10$ oscillators. Bottom: evolution of the controlled dynamics of the Kuramoto model (3.1) with $N = 10$ oscillators. The control function $\widehat{u}$ is obtained with the GD (left) and the GD-RBM (right) approach.

In Figure 2, we show the convergence of the error in logarithmic scale when applying both the GD and GD-RBM approach. We can appreciate how, in the case of GD-RBM, this convergence is not monotonic as it is for the GD algorithm. This, however, is not surprising due to the stochastic nature of the RBM methodology.
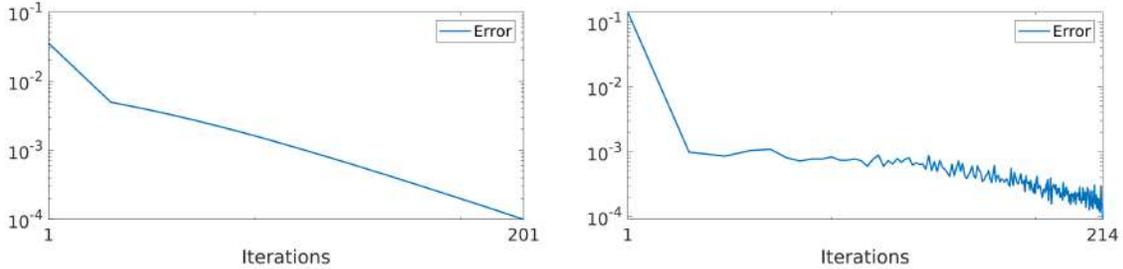
FIGURE 2. Convergence of the error e_k (see (4.2)) in logarithmic scale with the GD (left) and GD-RBM (right) algorithm.

In Figure 3, we display the behavior of the control function $\widehat{u}$ computed via the GD-RBM algorithm. We can see how, at the beginning of the time interval we are considering, this control is close to one and it is increasing with a small slope. On the other hand, this growth becomes more pronounced as we get closer to the final time $T = 3s$.
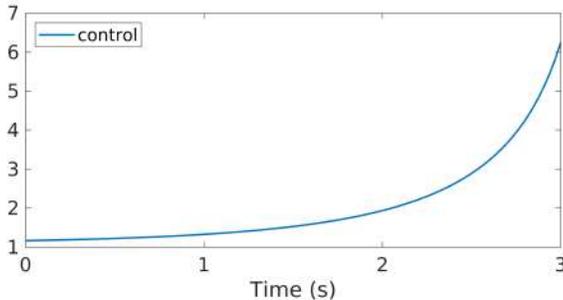

FIGURE 3. Control function $\widehat{u}$ obtained through the GD-RBM algorithm applied to the Kuramoto model (3.1) with $N = 10$ oscillators.

Notice that, in (3.1), $\widehat{u}$ enters as a multiplicative control which modifies the strength of the coupling K. Hence, according to the profile displayed in Figure 2, the control function $\widehat{u}$ we computed is initially letting the system evolving following its natural dynamics. Then, as the time evolves towards the horizon $T = 3s$, $\widehat{u}$ enhances the coupling strength K in order to reach the desired synchronized configuration (3.2).

Finally, notice also that the control $\widehat{u}$ is always positive. This is actually not surprising, if one takes into account the following observation.

In the Kuramoto model, in order to reach synchronization the coupling strength K needs to be positive. Otherwise, the system would converge to a desynchronized configuration (see [18]). Moreover, according to the model (3.1), if we start from $K > 0$, in order to maintain this coupling positive $\widehat{u}$ has to remain positive as well.

Recall that $\widehat{u}$ is computed minimizing the functional (3.3), in which the second term is a measurement of the level of synchronization in the model. Hence, since negative values of $\widehat{u}$ would lead to desynchronization and to the corresponding increasing of the functional, these values remain automatically excluded during the minimization process.

Let us now discuss briefly the role of the penalization parameter β in the computation of the optimal control. To this end, we have run simulations with different values of $\beta = 10^{-2}, 10^{-3}, 10^{-4}$ and 10^{-7}.

As we already mentioned in Section 3, in the cost functional (3.3) β is a (usually small) penalization parameter which allows to tune the norm of the optimal control $\widehat{u}$, that is, the amount of energy that the control introduces into the system. Roughly speaking, the smaller is β the larger will be $\widehat{u}$.

This is clearly seen in Figure 4. In particular, we can appreciate how, for $\beta = 10^{-2}$, the computed control remains smaller than in the other cases.
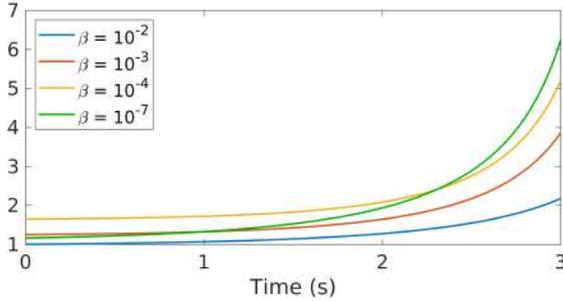
10

FIGURE 4. Control function $\widehat{u}$ obtained through the GD-RBM algorithm applied to the Kuramoto model (3.1) with $N = 10$ oscillators and different values of β.

We already mentioned above that the effect of the control in (3.1) is to enhance synchronization by modifying the strength of the coupling K. Hence, we can expect that, if $\widehat{u}$ is small (in particular, if it remains close to one), the synchronization properties of the Kuramoto model (3.1) will be worst than when applying a larger control. At this regard, let us recall that the level of synchronization in (3.1) can be analyzed in terms of the quantity

$$r(t) := \left| \frac{1}{N} \sum_{j=1}^{N} e^{i\theta_j(t)} \right|,$$

measuring the coherence of the oscillator population (see [1]). In particular we always have $0 \leq r(t) \leq 1$ and synchronization arises when r reaches the value one.

In Figure 5, we show the behavior of $r(t)$ with respect to the parameter β. On the one hand, in all the cases displayed we can clearly see that $r(T) = 1$. This means that all the computed controls will be effective to steer the system (3.1) to its synchronized configuration at time T. On the other hand, we can also notice how, when decreasing β, the function $r(t)$ reaches the value one faster, meaning that the corresponding control is expected to yield to better synchronization properties.
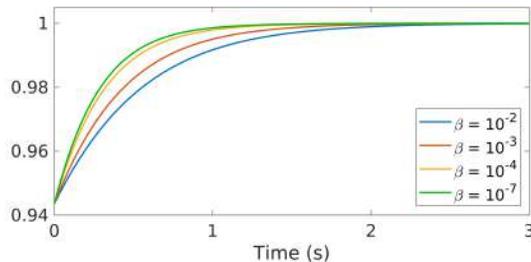

FIGURE 5. Behavior of the synchronization function $r(t)$ corresponding to the controlled dynamics (3.1) for different values of the parameter β.

Let us now conclude this section by showing that the control strategy that we propose in this paper is effective also in situations in which the coupling gain among the oscillators is too weak to ensure synchronization for the uncontrolled dynamics of the Kuramoto model. This corresponds to taking $K < K^*$ (see (2.4)). In particular, we will consider the case $K < 0$ in which the system is known to converge to a desynchronized configuration (see [18]).

For simplicity, in these simulations we only employed the GD-RBM algorithm, since using the GD approach we would obtain analogous results.

We can see in Figure 6 how, in this case of a negative coupling gain, the uncontrolled dynamics is diverging as t increases. On the other hand, when applying the control $\widehat{u}$, the system is once again steered to a synchronized configuration.
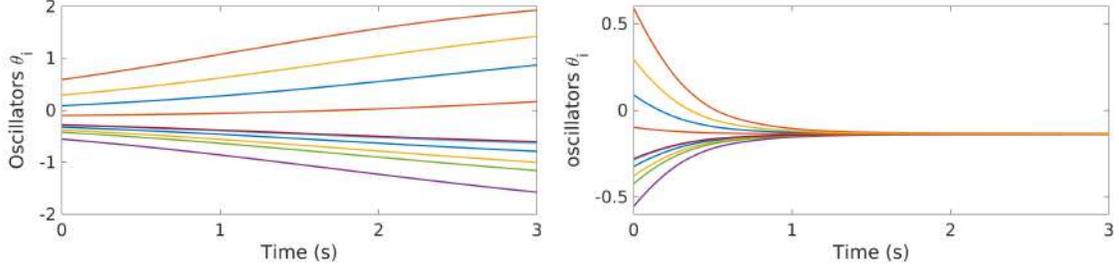
FIGURE 6. Evolution of the uncontrolled (left) and controlled (right) dynamics of the Kuramoto model (3.1) with $N = 10$ oscillators and $K < 0$.

At this regard, it is also interesting to observe that, this time, the control function we obtained is always negative (see Figure 7). This fact is not surprising, if we recall that in (3.1) the control acts by modifying the coupling gain K so that the oscillators are all synchronized at time T and that, for the uncontrolled dynamics, synchronization requires $K > 0$.
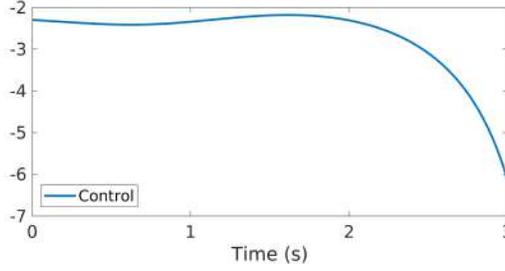

FIGURE 7. Control function $\widehat{u}$ obtained through the GD-RBM algorithm applied to the Kuramoto model (3.1) with $N = 10$ oscillators and $K < 0$.

Let us now complement our analysis by briefly showing the efficacy of our control strategy in a couple of more complex and realistic situations.

4.1.1. *The case of a sparse interaction network.* We start by considering the case of a sparse interaction network in our Kuramoto model (3.1). In other words, we are considering here the following system

(4.3)
$$\begin{cases} \dot{\theta}_i(t) = \omega_i + \dfrac{Ku(t)}{N} \sum_{j=1}^{N} a_{i,j} \sin\left(\theta_j(t) - \theta_i(t)\right), & i = 1, \ldots, N, \quad t > 0 \\ \theta_i(0) = \theta_i^0, \end{cases}$$

with

$$a_{i,j} = \begin{cases} 1, & \text{if } \theta_i \text{ is connected with } \theta_j \\ 0, & \text{if } \theta_i \text{ is not connected with } \theta_j \end{cases}$$

A schematic representation of the network considered in our simulations is given in Figure 8, in which the blue dots correspond to $a_{i,j} = 1$.

The simulations have been performed with the same initial datum and time horizon we considered in our previous experiments. Moreover, we addressed here only the case of a strong coupling gain $K > K^*$. The minimization of the functional $J(u)$ has been performed with the GD algorithm.

In Figure 9, we show the evolution of the uncontrolled and controlled dynamics. As we can see, while in the absence of a control the oscillators are evolving towards a desynchronized configuration, when applying the control function we computed the system still reaches synchronization at time T.
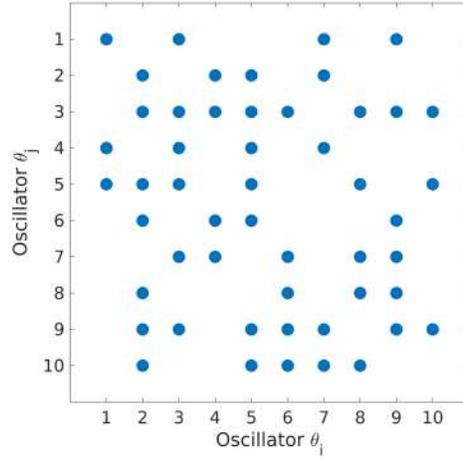
12

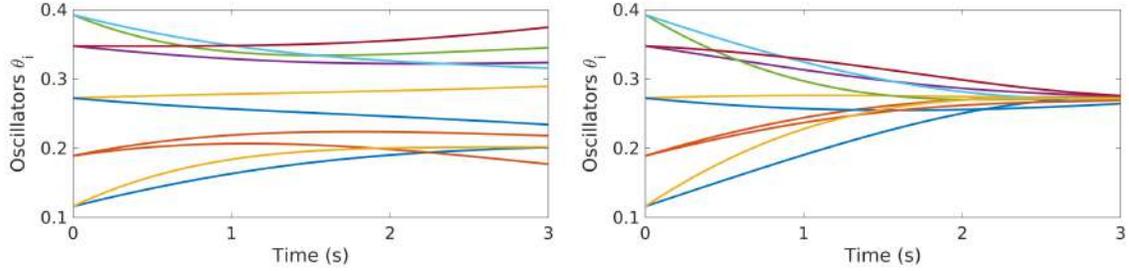FIGURE 8. Sparse interaction scheme for the Kuramoto model (4.3).

FIGURE 9. Evolution of the uncontrolled (left) and controlled (right) dynamics of the Kuramoto model (4.3) with $N = 10$ oscillators, $K > K^*$ and interactions as in Figure 8.

The control function obtained for these numerical experiments is plotted in Figure 10. We can observe how, differently from what is shown in Figures 3 and 4, this time $\widehat{u}$ reaches larger values. This is not surprising, if we consider that now our model has a lower level of interactions and if we recall our previous discussion on how our control affects the Kuramoto dynamics.
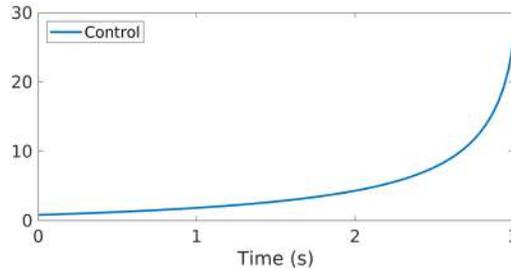

FIGURE 10. Control function $\widehat{u}$ for the Kuramoto model (4.3) with $N = 10$ oscillators, $K > K^*$ and interactions as in Figure 8.

13

4.1.2. *A second-order model with damping.* We consider here the second-order Kuramoto model with damping

$$\begin{cases} \ddot{\theta}_i(t) + \dot{\theta}_i(t) = \omega_i + \dfrac{Ku(t)}{N} \displaystyle\sum_{j=1}^{N} \sin\big(\theta_j(t) - \theta_i(t)\big), \quad i = 1, \ldots, N, \quad t > 0 \\ \theta_i(0) = \theta_i^0, \quad \dot{\theta}_i(0) = \theta_i^1, \end{cases}$$

which can be rewritten as the following first-order system

(4.4)
$$\begin{cases} \dot{\theta}_i(t) = \xi_i(t), & i = 1, \ldots, N, \quad t > 0 \\ \dot{\xi}_i(t) = -\xi_i(t) + \omega_i + \dfrac{Ku(t)}{N} \displaystyle\sum_{j=1}^{N} \sin\big(\theta_j(t) - \theta_i(t)\big), & i = 1, \ldots, N, \quad t > 0 \\ \theta_i(0) = \theta_i^0, \quad \xi_i(0) = \theta_i^1. \end{cases}$$

In the context of power grids, this model has been firstly introduced in the work [13]. Later on, in [32], it has been extended to consider more complex scenarios in which the dynamics of the voltage amplitude is taken into account.

Also in this case, we are interested in computing a control capable to steer the system to the synchronized configuration (3.2). This can be done once again by solving the optimal control problem (3.3), this time under the dynamics (4.4).

The simulations have been performed with the same initial datum $\Theta^0 = (\theta_i^0)_{i=1}^N$ we considered in our previous experiments and with $\Theta^1 = (\theta_i^1)_{i=1}^N = (0, 0, \ldots, 0)^\top$. The time horizon is once again $T = 3s$. Moreover, we addressed here both the cases of a strong coupling gain $K > K^*$ and of a negative one $K < 0$. The minimization of the functional $J(u)$ has been performed with the GD algorithm.

At this regard, we shall mention that, in [43], the GD methodology has been applied to obtain synchronization in a sparse network of Kuramoto oscillators with damping under the action of an additive control, i.e. the following model

$$\begin{cases} \dot{\theta}_i(t) = \xi_i(t), & i = 1, \ldots, N, \quad t > 0 \\ \dot{\xi}_i(t) = -\xi_i(t) + \omega_i + \dfrac{K}{N} \displaystyle\sum_{j=1}^{N} a_{i,j} \sin\big(\theta_j(t) - \theta_i(t)\big) + u_i(t), & i = 1, \ldots, N, \quad t > 0 \\ \theta_i(0) = \theta_i^0, \quad \xi_i(0) = \theta_i^1. \end{cases}$$

The advantages and disadvantages of this additive control action with respect to the multiplicative one we propose have been discussed in Section 3. In particular, our control strategy allows us to deal also with negative coupling gain $K < 0$, while in [43] only $K > 0$ has been considered.

As a matter of fact, in Figure 11, we show the evolution of the uncontrolled and controlled dynamics, for $K > K^*$ and $K < 0$, respectively. Also in this case, the proposed control strategy allows us to compute an effective control function $\widehat{u}$ which steers the system to a synchronized configuration in time T.

Finally, the control functions obtained for these numerical experiments are plotted in Figure 12. Also in these cases we can observe different behaviors than what is shown in Figures 3 and 4. In particular, this time $\widehat{u}$ changes sign in the time horizon $(0, T)$. At this regard, let us mention that for the second-order Kuramoto model (4.4), due to the presence of the damping term, our previous considerations on the sign of the control do not apply anymore. Hence, it is not surprising to obtain a behavior as the one displayed.

4.2. **Comparison of GD and GD-RBM.** We conclude this section on the numerical experiments by comparing the performances of the GD and GD-RBM algorithms for the computation of the optimal control $\widehat{u}$.

To this end, we run simulations for increasing values of N, namely $N = 10, 50, 100, 250, 1000$. As before, we chose a time horizon $T = 3s$ and a penalization parameter $\beta = 10^{-7}$. Moreover, we considered the case of a large coupling gain $K > K^*$.

In what follows, we focus only on the simple case of the first-order Kuramoto model (3.1) with an all-to-all interaction network. We will briefly comment about possible extensions to the more realistic scenarios described in Sections 4.1.1 and 4.1.2 in the the last part of this paper, devoted to conclusions and open problems.
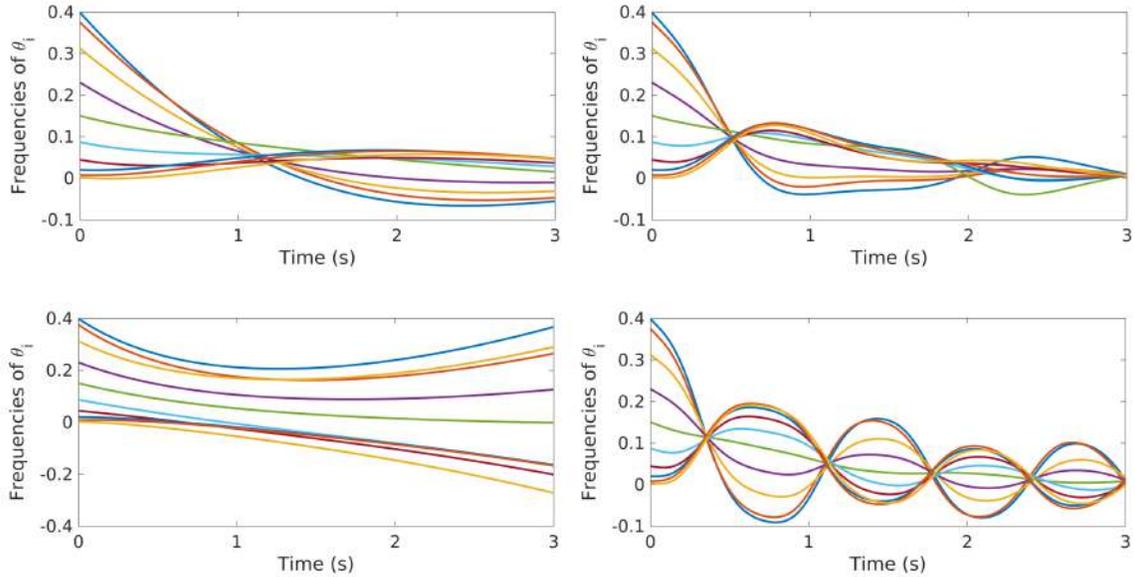
14

FIGURE 11. Evolution of the uncontrolled (left) and controlled (right) dynamics of the Kuramoto model (4.4) with $N = 10$ oscillators and $K > K^*$ (top) and $K < 0$ (bottom).
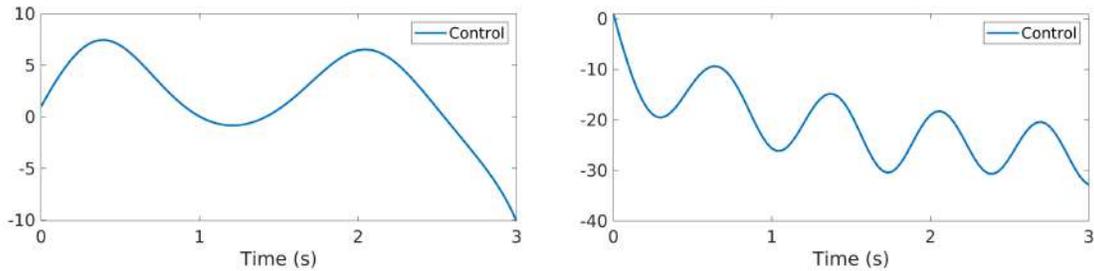

FIGURE 12. Control function $\widehat{u}$ for the Kuramoto model (4.4) with $N = 10$ oscillators and $K > K^*$ (left) and $K < 0$ (right).

In Table 1 (see also Figure 13) we collected the computational times required by the two methodologies to solve the optimization problem (3.3). At this regard, let us stress that the values contained in the table do not represent the time required by the control to synchronize the network which, we recall, is a fixed external input in our algorithms.

Our simulations show how, for low values of N, the two approaches show similar behaviors. Nevertheless, when increasing the number of oscillators in our system, the advantages of the GD-RBM methodology with respect to GD become evident. In particular, the growth of the computational time for GD-RBM is significantly less pronounced than for GD. As a matter of fact, in the case of $N = 1000$ oscillators, we decided not to perform the simulations with the GD algorithm since its behavior with smaller values of N already suggested that this experiment would be computationally too expensive.

On the other hand, even with $N = 1000$ oscillators in the system, the GD-RBM approach turns out to be able to compute an effective control for the Kuramoto model (3.1) (see Figure 14) in about 29 seconds.

15

	GD	GD-RBM
N	Time (sec.)	Time (sec.)
10	5.3	2.4
50	11.5	5.5
100	33.7	9.3
250	128.5	12.7
1000		29.1

TABLE 1. Computational times required by the GD and GD-RBM algorithm to compute the optimal control $\widehat{u}$ with increasing values of N.
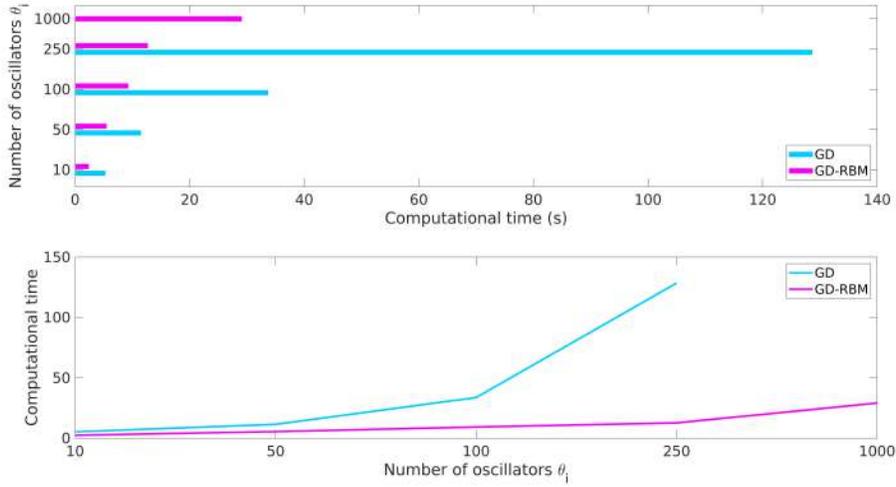

FIGURE 13. Computational times required by the GD and GD-RBM algorithm to compute the optimal control $\widehat{u}$ with increasing values of N.
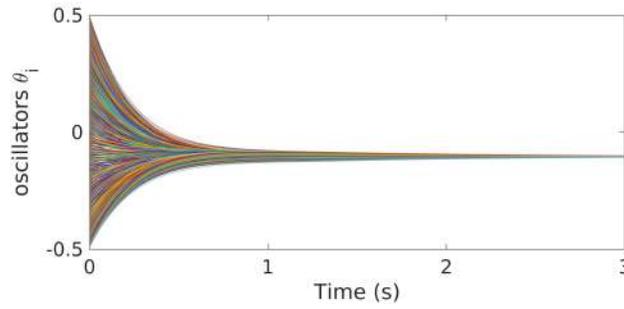

FIGURE 14. Evolution of the controlled dynamics of the Kuramoto model (3.1) with $N = 1000$ oscillators. The control has been computed with the GD-RBM algorithm.

5. CONCLUSIONS

This paper deals with the synchronization of coupled oscillators described by the Kuramoto model. In particular, we design a unique scalar control function $u(t)$ capable of steering a N-dimensional network of oscillators to a synchronized configuration in a finite time horizon. This is done following a standard optimal control approach and obtaining the function $u(t)$ via the minimization of a suitable cost functional.

With this approach, we computed a control which acts as a multiplicative force enhancing the coupling between the oscillators in the network, thus favoring the synchronization in the time horizon provided.

To carry out this minimization process, we used a Gradient Descent (GD) methodology, commonly employed in the optimal control community, which we have coupled with the novel Random Batch Method (RBM) for a more efficient numerical resolution of the Kuramoto dynamics. The main purpose of this work has been to show how the introduction of RBM into GD may yield considerable improvement in term of the computational complexity, in particular for large oscillator networks.

Our simulations results have shown the following main facts:

- The proposed control strategy is indeed effective to reach a synchronized configuration in a finite time horizon. Moreover, it allows to deal efficiently with large networks of oscillators (namely, $N = 1000$) and with the case of a low coupling gain in the network, when the uncontrolled dynamics is not expected to reach a synchronized configuration.
- For large values of N, the inclusion of RBM into the GD algorithm significantly reduces the computational burden for the control computation, thus allowing to deal with high-dimensional oscillators networks in a more efficient way.

In conclusion, the study conducted in this paper suggests that the proposed methodology, based on the combination of the standard GD optimization algorithm with the novel RBM method for the numerical resolution of multi-agent dynamics, may help in significantly reduce the computational complexity for the control computation of the Kuramoto model, in particular in the case of a high-dimensional system.

At this regard, we shall stress that the analysis in this paper has been developed mostly in a simplified framework of a network with all-to-all coupling, in which the oscillators are all interacting among themselves. The following interesting questions remain unaddressed:

- To study whether our methodology remains valid in more complex scenarios of networks with sparse interaction topologies, perturbations due to disconnections, or rewiring. In particular, it would be relevant to determine if the reduction in computational complexity that we obtained through the GD-RBM algorithm is related with the density of the network or, instead, this approach allows to deal efficiently also with the case of a low number of interactions. A starting point for this analysis would be to determine whether the GD-RBM methodology can be successfully applied in the scenarios we addressed in Sections 4.1.1 and 4.1.2.
- To analyze what happens if, instead of selecting the oscillators uniformly at random during the batching process in RBM, we organize them in groups with similar frequencies. At this regard, it would be important to understand if the methodology we propose is still effective or, instead, some modifications need to be introduced.
- To analyze if our methodology may be applied also in different framework than the ones considered in this paper. For instance, a similar formalism than the one we have proposed has been developed for computing rare-events in oscillator networks due to noise. See for instance [17, 16]. In these mentioned contributions, the objective functional is the probability for a rare event to happen. The actions are more complicated than the simple L^2-norm, but the batch techniques we employed may be useful in this context as well.

All these are key open problem which will be considered in future works.

Acknowledgment

References

[1] Acebrón, J. A., Bonilla, L. L., Vicente, C. J. P., Ritort, F., and Spigler, R. The Kuramoto model: A simple paradigm for synchronization phenomena. *Rev. Modern Phys. 77*, 1 (2005), 137.

[2] Ben-Naim, E. Opinion dynamics: rise and fall of political parties. *Europhys. Lett. 69*, 5 (2005), 671.

[3] Biccari, U., Ko, D., and Zuazua, E. Dynamics and control for multi-agent networked systems: A finite-difference approach. *Math. Models Methods Appl. Sci. 29*, 04 (2019), 755--790.

[4] Bottou, L., Curtis, F., and Nocedal, J. Optimization methods for large-scale machine learning. *SIAM Rev. 60*, 2 (2018).

[5] Buck, J. Synchronous rhythmic flashing of fireflies II. *Quart. Rev. Bio. 63*, 3 (1988), 265--289.

[6] CHASSIN, D. P., AND POSSE, C. Evaluating north american electric grid reliability using the Barabási-Albert network model. *Phys. A 355*, 2-4 (2005), 667--677.

[7] CHOPRA, N., AND SPONG, M. W. On synchronization of Kuramoto oscillators. In *Proceedings of the 44th IEEE Conference on Decision and Control* (2005), IEEE, pp. 3916--3922.

[8] CHOPRA, N., AND SPONG, M. W. Passivity-based control of multi-agent systems. In *Adv. Robot Control.* Springer, 2006, pp. 107--134.

[9] CHOPRA, N., AND SPONG, M. W. On exponential synchronization of Kuramoto oscillators. *IEEE Trans. Autom. Control 54*, 2 (2009), 353--357.

[10] DONG, J.-G., AND XUE, X. Synchronization analysis of Kuramoto oscillators. *Commun. Math. Sci. 11*, 2 (2013), 465--480.

[11] DÖRFLER, F., AND BULLO, F. Synchronization of power networks: Network reduction and effective resistance. *IFAC Proceedings Volumes 43*, 19 (2010), 197--202.

[12] DÖRFLER, F., CHERTKOV, M., AND BULLO, F. Synchronization in complex oscillator networks and smart grids. *Proc. Nat. Acad. Sci. 110*, 6 (2013), 2005--2010.

[13] FILATRELLA, G., NIELSEN, A. H., AND PEDERSEN, N. F. Analysis of a power grid using a Kuramoto-like model. *Europ. Phys. J. B 61*, 4 (2008), 485--491.

[14] HA, S.-Y., KIM, H. K., AND PARK, J. Remarks on the complete synchronization of Kuramoto oscillators. *Nonlinearity 28*, 5 (2015), 1441.

[15] HA, S.-Y., KIM, H. K., AND RYOO, S. W. Emergence of phase-locked states for the Kuramoto model in a large coupling regime. *Commun. Math. Sci. 14*, 4 (2016), 1073--1091.

[16] HINDES, J., JACQUOD, P., AND SCHWARTZ, I. B. Network desynchronization by non-gaussian fluctuations. *Phys. Rev. E 100*, 5 (2019), 052314.

[17] HINDES, J., AND SCHWARTZ, I. B. Rare slips in fluctuating synchronized oscillator networks. *Chaos 28*, 7 (2018), 071106.

[18] HONG, H., AND STROGATZ, S. H. Kuramoto model of coupled oscillators with positive and negative coupling parameters: an example of conformist and contrarian oscillators. *Phys. Rev. Lett. 106*, 5 (2011), 054102.

[19] JADBABAIE, A., MOTEE, N., AND BARAHONA, M. On the stability of the Kuramoto model of coupled nonlinear oscillators. In *Proc. 2004 American Control Conference* (2004), vol. 5, IEEE, pp. 4296--4301.

[20] JIN, S., LI, L., AND LIU, J.-G. Random Batch Methods (RBM) for interacting particle systems. *J. Comput. Phys. 400* (2020), 108877.

[21] KURAMOTO, Y. Self-entrainment of a population of coupled non-linear oscillators. In *International symposium on mathematical problems in theoretical physics* (1975), Springer, pp. 420--422.

[22] MARKDAHL, J., THUNBERG, J., AND GONCALVES, J. High-dimensional Kuramoto models on Stiefel manifolds synchronize complex networks almost globally. *Automatica 113* (2020), 108736.

[23] MEHYAR, M., SPANOS, D., PONGSAJAPAN, J., LOW, S. H., AND MURRAY, R. M. Distributed averaging on asynchronous communication networks. In *Proceedings of the 44th IEEE Conference on Decision and Control* (2005), IEEE, pp. 7446--7451.

[24] NABI, A., AND MOEHLIS, J. Single input optimal control for globally coupled neuron networks. *J. Neural Eng. 8*, 6 (2011), 065008.

[25] NESTEROV, Y. *Applied optimization.* Introductory Lectures on Convex Optimization: A Basic Course. Kluwer Academic Publishers Boston, Dordrecht, London, 2004.

[26] NOCEDAL, J., AND WRIGHT, S. *Numerical optimization.* Springer Science & Business Media, 2006.

[27] OLFATI-SABER, R. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. Autom. Control 51*, 3 (2006), 401--420.

[28] OLFATI-SABER, R., FAX, J. A., AND MURRAY, R. M. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE 95*, 1 (2007), 215--233.

[29] ROSENBLUM, M., AND PIKOVSKY, A. Delayed feedback control of collective synchrony: An approach to suppression of pathological brain rhythms. *Phys. Rev. E 70*, 4 (2004), 041904.

[30] SACHTJEN, M., CARRERAS, B., AND LYNCH, V. Disturbances in a power transmission system. *Phys. Rev. E 61*, 5 (2000), 4877.

[31] SCHMIDHUBER, J. Deep learning in neural networks: an overview. *Neural networks 61* (2015), 85--117.

[32] SCHMIETENDORF, K., PEINKE, J., FRIEDRICH, R., AND KAMPS, O. Self-organized synchronization and voltage stability in networks of synchronous machines. *Eur. Phys. J. Spec. Top. 223*, 12 (2014), 2577--2592.

[33] SEPULCHRE, R., PALEY, D., AND LEONARD, N. Collective motion and oscillator synchronization. In *Cooperative control.* Springer, 2005, pp. 189--205.

[34] 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.

[35] STROGATZ, S. H. Exploring complex networks. *Nature 410*, 6825 (2001), 268--276.

[36] STROGATZ, S. H., ABRAMS, D. M., MCROBIE, A., ECKHARDT, B., AND OTT, E. Crowd synchrony on the millennium bridge. *Nature 438*, 7064 (2005), 43--44.

[37] SUN, J., BOLLT, E. M., AND NISHIKAWA, T. Master stability functions for coupled nearly identical dynamical systems. *Europhys. Lett. 85*, 6 (2009), 60011.

[38] TAYLOR, D., SKARDAL, P. S., AND SUN, J. Synchronization of heterogeneous oscillators under network modifications: Perturbation and optimization of the synchrony alignment function. *SIAM J. Appl. Math. 76*, 5 (2016), 1984--2008.

[39] TOSCHER, A., AND JAHRER, M. Collaborative filtering applied to educational data mining. *KDD cup* (2010).

[40] Trélat, E. *Contrôle optimal: théorie & applications*. Vuibert, Collection Mathématiques Concrètes, 2005.

[41] Tröltzsch, F. *Optimal control of partial differential equations: theory, methods, and applications*. American Mathematical Soc., 2010.

[42] Tukhlina, N., Rosenblum, M., Pikovsky, A., and Kurths, J. Feedback suppression of neural synchrony by vanishing stimulation. *Phys. Rev. E 75*, 1 (2007), 011918.

[43] Tumash, L., Olmi, S., and Schöll, E. Stability and control of power grids with diluted network topology. *Chaos: An Interdisciplinary Journal of Nonlinear Science 29*, 12 (2019), 123105.

[44] Walker, T. J. Acoustic synchrony: two mechanisms in the snowy tree cricket. *Science 166*, 3907 (1969), 891--894.

[45] Wiesenfeld, K., Colet, P., and Strogatz, S. H. Frequency locking in Josephson arrays: connection with the Kuramoto model. *Phys. Rev. E 57*, 2 (1998), 1563.

[46] Winfree, A. T. Biological rhythms and the behavior of populations of coupled oscillators. *J. Theor. Bio. 16*, 1 (1967), 15--42.

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

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