Memristor Neural Networks for Linear and Quadratic Programming Problems

Mauro Di Marco, Mauro Forti, Luca Pancioni, Giacomo Innocenti, and Alberto Tesi

***Abstract*—This article introduces a new class of memristor neural networks (NNs) for solving, in real-time, quadratic programming (QP) and linear programming (LP) problems. The networks, which are called memristor programming NNs (MPNNs), use a set of filamentary-type memristors with sharp memristance transitions for constraint satisfaction and an additional set of memristors with smooth memristance transitions for memorizing the result of a computation. The nonlinear dynamics and global optimization capabilities of MPNNs for QP and LP problems are thoroughly investigated via a recently introduced technique called the flux–charge analysis method. One main feature of MPNNs is that the processing is performed in the flux–charge domain rather than in the conventional voltage–current domain. This enables exploiting the unconventional features of memristors to obtain advantages over the traditional NNs for QP and LP problems operating in the voltage–current domain. One advantage is that operating in the flux–charge domain allows for reduced power consumption, since in an MPNN, voltages, currents, and, hence, power vanish when the quick analog transient is over. Moreover, an MPNN works in accordance with the fundamental principle of in-memory computing, that is, the nonlinearity of the memristor is used in the dynamic computation, but the same memristor is also used to memorize in a nonvolatile way the result of a computation.**

***Index Terms*—Global optimization, memristor, neural networks (NNs), nonsmooth analysis, programming problems, stability.**

I. Introduction

CURRENT data-intensive applications in the Internet of Things (IoT), cloud computing, and edge computing call for ever-increasing data processing capabilities and the availability of computing devices with lower power consumption [1]–[3]. Conventional computation machines, as the Turing–Von Neumann machine, are currently facing crucial challenges to tackle these computing power and energy-demanding applications called the heat and memory wall, in addition to the advent of Moore's law slowdown [4]–[6]. In particular, the Von Neumann bottleneck originates from the speed limitations due to the continuous data movements between the processor (CPU) and memory (RAM) with physically distinct locations.

The use of neural-network (NN) architectures imitating the analog and parallel computational abilities of the brain, in combination with new nanoscale components with unconventional functions and dynamics, are long-term visions aimed at overcoming the drawbacks of the Turing–Von Neumann machines and sustain the growth of the electronics industry at the end of Moore's law. In this scenario, a prominent role is played by the memristor, that is, the fourth basic passive circuit element, in addition to the resistor, inductor, and capacitor. This has been theoretically envisioned by Prof. Chua in 1971 [7] using an axiomatic approach on device modeling and symmetry arguments on the basic electric quantities. An ideal memristor is a circuit element defined by a nonlinear relationship between flux φ (the integral of voltage or voltage momentum) and charge q (the integral of current or current momentum). One distinctive signature is the pinched hysteresis loop displayed in the voltage–current plane when subject to a sinusoidal input. Memristors provide various advantages over state-of-the-art digital complementary metal–oxide–semiconductor (CMOS) components, such as scalability, small on-chip area, low-power dissipation, efficiency, and adaptability [8], [9]. They also enable implementing synapses and neurons in an extremely energy-efficient way. More important, memristors and memelements, in general, can perform both information processing and storing of computation outputs on the same physical device. This provides analog capabilities unavailable in standard circuit elements and enables on-chip memory, biologically inspired parallel computing, and *in-memory computing*, that is, the integration of storage and computation in the same physical location, which is crucial to overcome the Von Neumann bottleneck [10]–[14].

There is an extensive literature on the use of memristors as synapses in dynamic NNs. The reader is referred to [15]–[18] and the references therein, for an account on some relevant contributions. Such papers mainly consider NNs with an additive structure as Hopfield or cellular NNs (CNNs) [19], [20]. Several basic dynamic phenomena are investigated, as the convergence of solutions toward equilibrium points (EPs), in view of the applications to computational tasks related to content addressable memories and synchronization, or oscillatory and complex behaviors useful for implementing spatiotemporal real-time neural processors [21]–[25].

Manuscript received December 28, 2019; revised March 16, 2020; accepted May 18, 2020. Date of publication June 19, 2020; date of current version March 14, 2022. This work was supported by MUR (Ministero dell'Instruzione, dell'Università e della Ricerca) under Grant 2017LSCR4K_003. This article was recommended by Associate Editor Z. Zeng. *(Corresponding author: Mauro Forti.)*

Mauro Di Marco, Mauro Forti, and Luca Pancioni are with the Department of Information Engineering and Mathematics, University of Siena, 53100 Siena, Italy (e-mail: dimarco@dii.unisi.it; forti@diism.unisi.it; pancioni@diism.unisi.it).

Giacomo Innocenti and Alberto Tesi are with the Department of Information Engineering, University of Florence, 50139 Firenze, Italy (e-mail: giacomo.innocenti@unifi.it; alberto.tesi@unifi.it).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2020.2997686.

Digital Object Identifier 10.1109/TCYB.2020.2997686

Solving linear programming (LP) and quadratic programming (QP) problems in real time is another category of computation problems of extreme importance in all engineering fields. Several classes of NNs with outstanding performances have been put forward to solve LP and QP problems. The first pioneering works on NNs for LP problems are due to Pyne [26] and Tank and Hopfield [27]. Later, Kennedy and Chua [28] greatly extended those works to nonlinear programming problems, while Forti *et al.* [29] further extended the approach to classes of nonsmooth optimization problems. In particular, Liu and Wang [30] proposed an effective NN for LP problems with hard-limiter saturation nonlinearities. Such approaches are based on a penalty function method, either approximate (smooth case) or exact (nonsmooth case). NNs for LP and QP problems that avoid the use of a penalty parameter have been devised by Rodríguez-Vázquez *et al.* [31]. Subsequently, several approaches have been proposed, leading to NNs with different architectures, each displaying special features and advantages. These include the NNs with a single-layer structure by Liu and Wang [32] or a multilayer structure by Gao and Liao [33]; the NNs introduced by Xia [34], Leung *et al.* [35], and Xia and Wang [36] using a primal–dual approach; and the NNs by Xia *et al.* based on a projection approach [37]. This brief review is far from being exhaustive. The reader is referred to [30], [34], [36], [38]–[46], and the references therein, for a more complete overview of the related relevant literature.

All the quoted approaches share the common feature that they map and solve the QP or LP problems via a dynamic NN evolving in the traditional voltage–current domain. In particular, the problems are solved during the evolution of the capacitor voltages, and the results of the computation are the asymptotic values of such voltages. Since a small capacitance cannot hold a charge for a sufficiently long time, extra memory is in practice required, implying additional circuitry and leading to problems as the Von Neumann bottleneck.

The goal of this article is to introduce new memristor NN architectures for solving real-time QP and LP problems. We refer henceforth to these networks as memristor programming NNs (MPNNs). MPNNs exploit some nonconventional features of memristors for improving their performance with respect to the traditional memristorless NN for solving analogous optimization problems. The advantages include the implementation of an in-memory computing scheme, thus providing a concrete way to overcome the Von Neumann bottleneck and reduce power dissipation.

The main contributions can be summarized as follows.

1) This article introduces a new class of MPNNs, where two different sets of memristors are used in the neurons and in the constraint neurons. Memristors in the constraint neurons are used to implement the diode-like nonlinear characteristic needed to impose constraint satisfaction in the penalty function method. They are characterized by a conductance (memductance) that sharply switches between two largely different values, as it happens for the class of filamentary-type memristors [47]. Memristors in the neurons are used to memorize the result of the computation as the asymptotic value of their resistance (memristance). They have a smooth dependence of the memristance on the input as it happens for memristors based on diffusion mechanisms as the celebrated HP memristor [48].
2) We use a recent method [49]–[51], called the flux–charge analysis method (FCAM), in order to: a) map QP and LP problems onto the new MPNN architecture and b) perform a thorough analysis of the nonlinear dynamics and optimization capabilities. In particular, we have the following.
   a) New nonlinear dynamic phenomena due to the presence of memristors are highlighted, such as the basic property of foliation of the state space in invariant manifolds, that follows from the existence of invariants of motion for the MPNN dynamics.
   b) By using FCAM in combination with tools from nonsmooth analysis, and a nonsmooth form of the Łojasiewicz inequality, conditions are given ensuring that each solution converges either exponentially or in finite time to an optimal solution. The robustness of the global convergence is also addressed.
3) Examples and numerical simulations are provided to verify and illustrate the effectiveness, advantages, and optimization capabilities of MPNNs.

One main novelty of the proposed MPNNs is that the *analog computation and optimization is performed in the flux–charge domain* rather than in the traditional voltage–current domain as it happens for NNs introduced so far to solve QP and LP problems. Computation in the flux–charge domain permits to achieve a number of advantages as summarized next.

1) It is shown that at the end of the analog transient, all capacitor voltages, as well as all other voltages and currents in the MPNN, vanish, that is, in the steady state, an MPNN turns off and power consumption vanishes. Nonetheless, memristors are able to memorize in a *nonvolatile way* the result of computation in the asymptotic value of their memristance. An MPNN consumes power only during the short analog transient with potential advantages in terms of power consumption over NNs computing in the voltage–current domain, where voltages, current, and power do not vanish in the steady state.
2) An MPNN works in accordance with the *fundamental principle of in-memory computing* [10], [13], [14]. Indeed, the role of memristors is two-fold: their nonlinearity is used for analog computation purposes, moreover, memristors are also used to store in a nonvolatile way the result of computation. Again, this is basically different from NNs computing in the voltage–current domain, where the result of computation is the asymptotic value of the voltage on capacitors.

The structure of this article is outlined as follows. In Section II, we recall some facts about FCAM needed in this article. The optimization problem considered here is formalized in Section III, while the MPNN architecture to solve this problem is introduced in Section IV. The main results on dynamics, optimization capabilities, and

robustness of convergence are given in Sections V–VII. Section VIII provides some application examples. Finally, the main conclusions drawn in this article are collected in Section IX.

II. FCAM

FCAM is a recently developed method to effectively analyze the dynamics of nonlinear memristor circuits in the flux–charge domain [49]–[51]. Consider a two-terminal (one-port) circuit element and let v and i be the voltage and current at its terminals. Let $\varphi(t) = \int_{-\infty}^{t} v(\tau)d\tau$ be the flux (or voltage momentum) and $q(t) = \int_{-\infty}^{t} i(\tau)d\tau$ be the charge (or current momentum).

Suppose we are interested in analyzing the dynamics of a memristor circuit starting from a finite initial instant t_0. Let us introduce the incremental flux

$$\varphi(t; t_0) = \varphi(t) - \varphi(t_0) = \int_{t_0}^{t} v(\tau)d\tau$$

and the incremental charge

$$q(t; t_0) = q(t) - q(t_0) = \int_{t_0}^{t} i(\tau)d\tau$$

for any $t \geq t_0$. FCAM is based on using the Kirchhoff flux law (KφL) and Kirchhoff charge law (KqL) for the incremental flux and charge in combination with the constitutive relations (CRs) of circuit elements, that is, the links that each element establishes between the incremental flux and charge at its terminals. KφL and KqL, which are the natural counterparts of the Kirchhoff voltage and current laws in the voltage–current domain, simply express the physical law of conservation of the incremental flux and charge.

The building blocks of MPNNs are resistors, capacitors, memristors, and operational amplifiers (Section IV). Next, we recall their CRs in the flux–charge domain. The reader is referred to [49] for further details.

A (linear) resistor defined by Ohm's law $v = Ri$ satisfies an analogous law $\varphi(t; t_0) = Rq(t; t_0)$ in the flux–charge domain. An ideal capacitor $q = Cv$ has the CR in the flux–charge domain

$$q(t; t_0) = C\dot{\varphi}(t; t_0) - Cv(t_0)$$

where $v(t_0)$ is the initial condition (IC) at t_0 for the state variable v of C (the dot means the time derivative).

Consider now a flux-controlled memristor $q(t) = h(\varphi(t))$ [7]. The CR in the flux–charge domain is

$$q(t; t_0) = h(\varphi(t; t_0) + \varphi(t_0)) - h(\varphi(t_0))$$

where $\varphi(t_0)$ is the IC at t_0 for the state variable φ of the memristor. Dually, for a charge-controlled memristor $\varphi(t) = f(q(t))$, the CR in the flux–charge domain is

$$\varphi(t; t_0) = f(q(t; t_0) + q(t_0)) - f(q(t_0))$$

where $q(t_0)$ is the IC at t_0 for the state variable q.

Finally, consider an ideal operational amplifier (oa) operating in the linear region and satisfying $v_1 = 0$ and $i_1 = 0$. In the flux–charge domain, the CRs are $\varphi_1(t; t_0) = 0$ and $q_1(t; t_0) = 0$.

We refer the reader to [49] and [52] for the equivalent circuits in the (φ, q)-domain of the resistor, capacitor, flux-controlled or charge-controlled memristor, and oa.

Remark 1: Note that a flux-controlled memristor has an algebraic CR in the flux–charge domain depending however on the IC at t_0 for the state variable. In the voltage–current domain, the same memristor has a CR in differential form $i(t) = W(\varphi(t))v(t)$, $\dot{\varphi}(t) = v(t)$, where $W(\varphi) = f'(\varphi)$ is the memductance in Ohm^{-1} (the prime denotes the derivative of f with respect to its argument). It is well known that one main signature of the memristor is the pinched hysteresis loop in the v-i plane in response to a sinusoidal voltage signal. Dual considerations hold for a charge-controlled memristor.

III. QP AND LP PROBLEMS

Consider the following optimization problem.

Minimize the scalar function

$$\psi(x) = \frac{1}{2}x^{\mathrm{T}}Gx + a^{\mathrm{T}}x : \mathbb{R}^q \to \mathbb{R} \tag{1}$$

subject to the affine constraints

$$\gamma_j(x) = \langle B_j, x\rangle - c_j \geq 0,\ j = 1, 2, \ldots, p \tag{2}$$

where $\langle\cdot,\cdot\rangle$ denotes the scalar product, p and q are positive integers, $x, a \in \mathbb{R}^q$, $0 \neq B_j \in \mathbb{R}^q$, and $c_j \in \mathbb{R}$, $j = 1, 2, \ldots, p$. Moreover, $G \in \mathbb{R}^{q\times q}$ is a symmetric positive-definite matrix for a QP problem, while $G = 0$ for an LP problem. The superscript T means the transpose.

The *feasibility region* is the closed convex polyhedron defined by the affine constraints

$$\mathcal{P} = \left\{x \in \mathbb{R}^q : \langle B_j, x\rangle - c_j \geq 0, j = 1, 2, \ldots, p\right\} \subset \mathbb{R}^q.$$

Henceforth, we assume that $\mathcal{P}$ is bounded, moreover, we have $\mathrm{int}(\mathcal{P}) \neq \emptyset$, where int denotes the interior. Let

$$\mathcal{M} = \arg\min_{x\in\mathcal{P}} \psi(x) \neq \emptyset$$

be the set of global minimizers of ψ in $\mathcal{P}$. It is known that for LP problems, $\mathcal{M}$ is a closed convex set, while for QP problems, $\mathcal{M} = \{x^*\}$ is a singleton.

IV. MEMRISTOR NN FOR QP AND LP PROBLEMS

The dynamic canonical nonlinear programming network proposed by Kennedy and Chua [28] is one of the first and more relevant NNs for solving the QP and LP problems. The network satisfies the system of differential equations

$$C\dot{v}_i = -a_i - \sum_{j=1}^{q} g_{ij}v_j - \sum_{\ell=1}^{p} b_{\ell i}s\left(\sum_{k=1}^{q} b_{ik}v_k\right) \tag{3}$$

for $i = 1, 2, \ldots, q$, where C is the neuron capacitance, v_i, $i = 1, 2, \ldots, q$, are the capacitor voltages, and $s(\cdot)$ is a nonlinearity with a suitable shape used to impose constraint satisfaction.

Kennedy–Chua (KC) NN is an extension of the LP NN originally proposed by Tank and Hopfield [27]. KCNN exploits a penalty function method, that is, the dynamic equations are the negative gradient of an energy function given by $(1/2)v^{\mathrm{T}}Gv + a^{\mathrm{T}}v$ plus a barrier term forcing constraint satisfaction. Later, Forti *et al.* [29] introduced a nonsmooth version

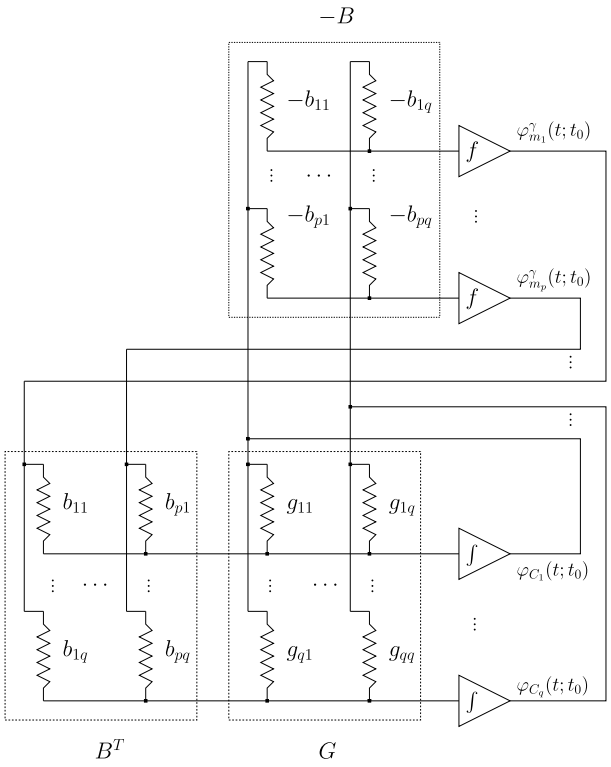

Fig. 1. Architecture of an MPNN.

of KCNN, where $s(\cdot)$ is replaced by a discontinuous function, that is able to implement an exact penalty function method.

Next, we introduce a new class of NNs with memristors that satisfy a system of differential equations analogous to (3) where v is replaced by the vector φ of fluxes of certain memristors. Such NNs, called MPNNs, map and solve the QP or LP problems by *computing in the flux–charge domain* and are shown to feature potential advantages over the traditional NNs operating in the voltage–current domain.

A. MPNN Architecture

The proposed MPNN architecture is composed of a set of neurons and a set of constraint neurons interconnected with each other as shown in Fig. 1. Each neuron is implemented via a summing integrator oa. A flux-controlled memristor M_φ with CR $q_{m_i}(t) = h(\varphi_{m_i}(t))$ is connected at the oa output to memorize the result of computation in a nonvolatile way (Fig. 2). Each constraint neuron is implemented via a summing oa with a charge-controlled memristor M_q with CR $\varphi^{\gamma}_{m_i}(t) = f(q^{\gamma}_{m_i}(t))$ in feedback (Fig. 3). Note that each neuron and constraint neuron has an inverting oa to implement negative interconnections.

Let us describe in detail the neurons and constraint neurons in the flux–charge domain using FCAM (see Section II).

The equivalent circuit of the ith neuron in the flux–charge domain is shown in Fig. 2. This has been obtained using the equivalent circuits in the same domain for a resistor, capacitor, flux-controlled memristor, and oa, as in [49] and [52].

We assume that the capacitor C is initially charged at $v_{i0} = v_i(t_0)$, while M_φ is initially discharged, that is, $\varphi_{m_{i0}} = \varphi_{m_i}(t_0) = 0$. By using K$\varphi$L, K$q$L, and CRs of circuit elements, it is seen that the neuron satisfies

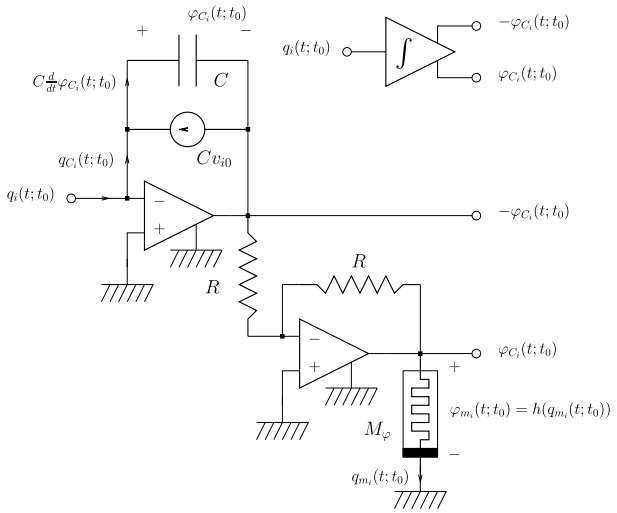

Fig. 2. Neuron and equivalent circuit in the flux–charge domain.

$$\begin{aligned}\varphi_{m_i}(t;t_0) &= \varphi_{C_i}(t;t_0)\\ q_i(t;t_0) &= C\dot{\varphi}_{C_i}(t;t_0) - Cv_{i0}\\ q_{m_i}(t;t_0) &= h\big(\varphi_{C_i}(t;t_0)\big).\end{aligned}$$

We suppose that the memristor CR is given by

$$h\big(\varphi_{m_i}\big) = \frac{1}{2}\alpha\varphi^2_{m_i} + \beta\varphi_{m_i}$$

where $\alpha, \beta > 0$, hence the memductance (the reciprocal of memristance)

$$M\big(\varphi_{m_i}\big) = h'\big(\varphi_{m_i}\big) = \alpha\varphi_{m_i} + \beta$$

has a *linear dependence* on the flux.[1]

By measuring the memristance, we can indirectly obtain the flux, which is the variable used in the MPNN computation (see Section VIII). There are standard procedures to measure the memristance, which are typically based on applying a small sensing voltage or current pulse of some preset waveforms, across the memristor terminals. The memristor can be modeled as a linear resistor obeying Ohm's law when the sensing signal amplitude is sufficiently small. More details can be found in [9], while automated techniques for memristance measurements are described in [53].

The equivalent circuit of a constraint neuron in the flux–charge domain is shown in Fig. 3, where M_q is assumed to be initially charged at $q^{\gamma}_{m_{i0}} = q^{\gamma}_{m_i}(t_0)$. The constraint neuron is described by the static relationships

$$\begin{aligned}q^{\gamma}_i(t;t_0) &= q^{\gamma}_{m_i}(t;t_0)\\ \varphi^{\gamma}_{m_i}(t;t_0) &= f\big(q^{\gamma}_i(t;t_0) + q^{\gamma}_{m_{i0}}\big) - f\big(q^{\gamma}_{m_{i0}}\big).\end{aligned}$$

We assume henceforth that the nonlinearity $f(q^{\gamma}_{m_i})$ of the memristor has a (reverse) diode-like shape as in Fig. 4(a).

[1] Actually, the important property to be enjoyed by the memristors used in neurons is that the measurement of the memristance enables to indirectly measure the flux. This is true when there is a smooth monotone increasing (e.g., linear) dependence of memristance or memductance on flux, as it happens for some classes of practical memristors based on a diffusive mechanism as the celebrated memristor implemented at HP [48, p. 81].

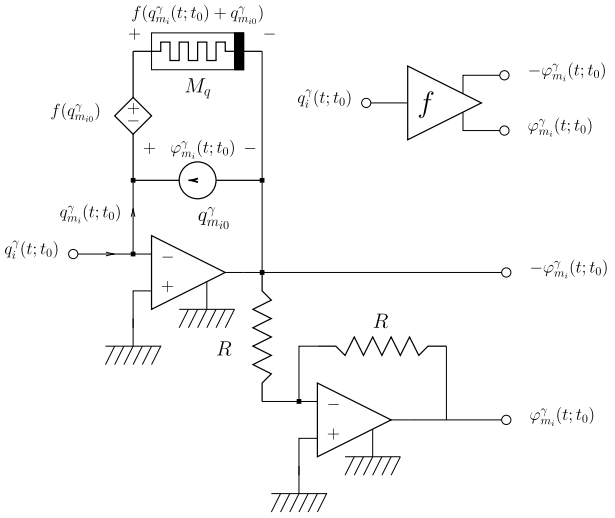

Fig. 3. Constraint neuron and equivalent circuit in the flux–charge domain.

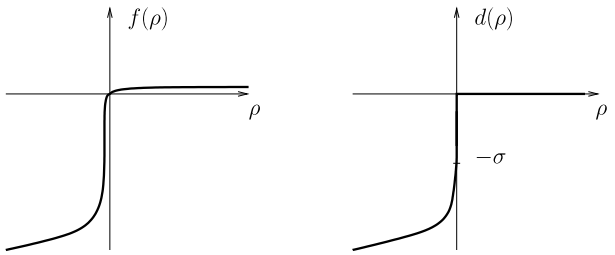

Fig. 4. (a) Stiff nonlinearity $f(\cdot)$ of a memristor in a constraint neuron and (b) discontinuous approximation $d(\cdot)$ of $f(\cdot)$.

More precisely, $f(q^{\gamma}_{m_i})$ is a C^1 monotonically increasing function with a very large positive slope for $q^{\gamma}_{m_i} \leq -q_\epsilon$ and a very small positive slope for $q^{\gamma}_{m_i} \geq q_\epsilon$, where q_ϵ is a small positive charge. Hence, the memristance

$$W^{\gamma}\left(q^{\gamma}_{m_i}\right) = f'\left(q^{\gamma}_{m_i}\right)$$

is very high for $q^{\gamma}_{m_i} \leq -q_\epsilon$, whereas it assumes very low values for $q^{\gamma}_{m_i} \geq q_\epsilon$. Such a nonlinearity is typical of filamentary-type memristors [47]. Depending on the input, such memristors can quickly switch from an open circuit (very low conductance G_{off}) when the filament is not formed, to a short circuit (very high conductance G_{on}) after filament forming. In the literature, very high ratios $G_{\text{on}}/G_{\text{off}}$ reaching several orders of magnitude are reported for filamentary-type memristors (see [54] and [55]). We also note that the monotone increasing property of f is in accordance with the passivity of the memristor [7, Th. 1].

The linear interconnection structure (see Fig. 1) is such that we can write for the neurons

$$q_i(t;t_0) = -\sum_{j=1}^{q} g_{ij}\varphi_{C_j}(t;t_0) - \sum_{\ell=1}^{p} b_{\ell i}\varphi^{\gamma}_{m_\ell}(t;t_0)$$

for $i = 1, 2, \ldots, q$. Moreover, for the constraint amplifiers

$$q^{\gamma}_i(t;t_0) = \sum_{k=1}^{q} b_{ik}\varphi_{C_k}(t;t_0), \;\; i = 1, 2, \ldots, p.$$

Putting together the equations for the neurons and interconnections, and recalling that $\varphi_{m_{i0}} = 0$, hence $\varphi_{m_i}(t;t_0) = \varphi_{m_i}(t)$, we obtain that an MPNN satisfies

$$C\dot{\varphi}_{m_i}(t) = Cv_{i0} + \sum_{\ell=1}^{p} b_{\ell i} f\left(q^{\gamma}_{m_{\ell 0}}\right) - \sum_{j=1}^{q} g_{ij}\varphi_{m_j}(t) - \sum_{\ell=1}^{p} b_{\ell i} f\left(\sum_{k=1}^{q} b_{ik}\varphi_{m_k}(t) + q^{\gamma}_{m_{i0}}\right) \tag{4}$$

for $i = 1, 2, \ldots, q$. This is a system of q differential equations in the same number of state variables given by the memristor fluxes $\varphi_{m_i}(t)$, $i = 1, 2, \ldots, q$. The ICs are by construction $\varphi_{m_i}(t_0) = 0$, $i = 1, 2, \ldots, q$.

According to FCAM [49], the equations of the MPNN in the voltage–current domain can be simply obtained by differentiating in time (4), namely

$$C\dot{v}_i(t) = -\sum_{j=1}^{q} g_{ij}v_j(t) - \sum_{\ell=1}^{p}\sum_{k=1}^{q} b_{\ell i}b_{ik}f'\left(q^{\gamma}_{m_i}(t)\right)v_k(t) \tag{5}$$

for $i = 1, 2, \ldots, q$ and

$$\dot{q}^{\gamma}_{m_i}(t) = \sum_{k=1}^{q} b_{ik}v_k(t) \tag{6}$$

for $i = 1, 2, \ldots, p$, where we considered that $q^{\gamma}_{m_i}(t) = \sum_{k=1}^{q} b_{ik}\varphi_{m_k}(t) + q^{\gamma}_{m_{i0}}$ and $\dot{\varphi}_{m_i}(t) = \dot{\varphi}_i(t) = v_i(t)$.

This is a system of $q+p$ differential equations in the same number of state variables in the voltage–current domain given by the capacitor voltages $v = (v_1, v_2, \ldots, v_q)^{\mathrm{T}} \in \mathbb{R}^q$ and the constraint memristor charges $q^{\gamma}_m = (q^{\gamma}_{m_1}, q^{\gamma}_{m_2}, \ldots, q^{\gamma}_{m_p})^{\mathrm{T}} \in \mathbb{R}^p$. The ICs of the state variables are $v_i(t_0) = v_{i0}$, $i = 1, 2, \ldots, q$, and $q^{\gamma}_{m_j}(t_0) = q^{\gamma}_{m_{j0}}$, $j = 1, 2, \ldots, p$.

V. MPNN for QP and LP Problems

Our goal is to design an MPNN such that it maps and solves QP and LP programming problems in the flux–charge domain. This entails addressing a number of relevant dynamical issues including the following ones.

1) To prove that under suitable assumptions, (4) satisfied by an MPNN in the (φ, q)-domain is analogous to the equations describing KCNN in the (v, i)-domain (Section V-A).
2) To show that the state space in the (v, i)-domain of an MPNN can be foliated in invariant manifolds and that addressing the previous point 1) boils down to choosing via the ICs the suitable invariant manifold where an MPNN should evolve (Section V-B).
3) To conduct a thorough analysis of the optimization capabilities of an MPNN directly in the (φ, q)-domain (Section VI) using the analogy established in point 1).
4) To prove some fundamental peculiar properties of MPNNs in the (v, i)-domain, such as the property that capacitor voltages and hence power vanish when the transient is over (Section VI-E).

A. Analogy Between MPNN and KCNN

Consider again (4) satisfied by an MPNN in the (φ, q)-domain. The equations contain constant terms v_{i0} and $q^{\gamma}_{m_{i0}}$ corresponding to the ICs for the state variables in the voltage–current domain. The next result shows that by suitably choosing such ICs, we obtain an MPNN satisfying a system of differential equations analogous to that of the KCNN.

Proposition 1: Suppose to choose the ICs for the state variables in the voltage–current domain as follows:

$$v_{i0} = -\frac{1}{C}\left[a_i + \sum_{\ell=1}^{p} b_{\ell i} f(-c_i)\right],\ i = 1, 2, \ldots, q$$
$$q^{\gamma}_{m_{i0}} = -c_i, i = 1, 2, \ldots, p. \tag{7}$$

Then, (4) boils down to

$$C\dot{\varphi}_{m_i}(t) = -a_i - \sum_{j=1}^{q} g_{ij}\varphi_{m_j}(t) - \sum_{\ell=1}^{p} b_{\ell i} f\left(\sum_{k=1}^{q} b_{ik}\varphi_{m_k}(t) - c_i\right) \tag{8}$$

for $i = 1, 2, \ldots, q$. With this choice, an MPNN satisfies in the flux–charge domain a system analogous to that satisfied by KCNN in the voltage–current domain, that is, system (3), provided the vector of capacitor voltages v is replaced by that of neuron memristor fluxes φ_m and nonlinearity $s(\cdot)$ is replaced by the constraint neuron nonlinearity $f(\cdot)$.

Proof: Directly follows substituting (7) into (4). ∎

Let us define $\varphi(t) = (\varphi_{m_1}(t), \varphi_{m_2}(t), \ldots, \varphi_{m_q}(t))^{\mathrm{T}} \in \mathbb{R}^q$ and $F(\rho) = (f(\rho), f(\rho), \ldots, f(\rho))^{\mathrm{T}}$: $\mathbb{R}^q \to \mathbb{R}^p$ and let $B^{\mathrm{T}} = (B_1, B_2, \ldots, B_p) \in \mathbb{R}^{q\times p}$. Then, omitting dependence on t, and choosing $t_0 = 0$, in a matrix–vector form for QP problems (8) can be written as

$$C\dot{\varphi} = -a - G\varphi - B^{\mathrm{T}}F(B\varphi - c) \tag{9}$$

with ICs $\varphi(0) = 0$.

In the voltage–current domain, in a matrix–vector form

$$C\dot{v} = -Gv - \frac{d}{dt}\left[B^{\mathrm{T}}F\left(q^{\gamma}\right)\right] = -Gv - B^{\mathrm{T}}J_F\left(q^{\gamma}\right)Bv \tag{10}$$
$$\dot{q}^{\gamma} = Bv \tag{11}$$

where J_F is the Jacobian of F, we let $q^{\gamma} = (q^{\gamma}_{m_1}, q^{\gamma}_{m_2}, \ldots, q^{\gamma}_{m_p})^{\mathrm{T}} \in \mathbb{R}^p$ and considered that $q^{\gamma} = B\varphi - c$. The ICs are $v_0 = v(0)$ and $q^{\gamma}_0 = q^{\gamma}(0)$.

For LP problems, letting $G = 0$, we have

$$C\dot{\varphi} = -a - B^{\mathrm{T}}F(B\varphi - c) \tag{12}$$

with ICs $\varphi(0) = 0$, while in the voltage–current domain

$$C\dot{v} = -\frac{d}{dt}\left[B^{\mathrm{T}}F\left(q^{\gamma}\right)\right] = -B^{\mathrm{T}}J_F\left(q^{\gamma}\right)Bv \tag{13}$$
$$\dot{q}^{\gamma} = Bv \tag{14}$$

with ICs $v_0 = v(0)$ and $q^{\gamma}_0 = q^{\gamma}(0)$.

Remark 2: From a circuit viewpoint, the interconnecting structure of an MPNN is analogous to that of a Hopfield NN and a KCNN for LP and QP problems. The fundamental difference is the use of memristors in the neurons and constraint neurons of an MPNN. Memristors enable to process signals in the (φ, q)-domain, instead of the traditional (v, i)-domain, as it happens for Hopfield and KCNNs. As we shall see, this enables to implement an effective computation scheme according to the principle of in-memory computing, where memristors process and also store the computation output. In addition, computing in the (φ, q)-domain permits to lower the power requirements.

B. First Integrals, Invariant Manifolds, and Reduced-Order Dynamics

At first glance, it seems surprising that the order $q + p$ of the equations describing the dynamics of an MPNN in the voltage–current domain is different from the order q of the equations in the flux–charge domain. As shown next, such an apparent inconsistency can be explained via FCAM and the principle of the foliation of the state space of memristor NNs in the voltage–current domain [49].

1) QP Problems: Consider for QP problems function $Q_{\mathrm{QP}} : \mathbb{R}^{q+p} \to \mathbb{R}^q$ of the state variables in the voltage–current domain

$$Q_{\mathrm{QP}}\left(v(t), q^{\gamma}(t)\right) = BG^{-1}\left(Cv(t) + B^{\mathrm{T}}F\left(q^{\gamma}(t)\right)\right) + q^{\gamma}(t).$$

Theorem 1: The time derivative of Q_{QP} along the solutions of (10) and (11) identically vanishes, namely, $\dot{Q}_{\mathrm{QP}}(v(t), q^{\gamma}(t)) = 0$ for any $t \geq 0$.

Proof: From (10), we obtain $v = -G^{-1}(C\dot{v}) - G^{-1}(d[B^{\mathrm{T}}F(q^{\gamma})]/dt)$. Substituting in (11), $\dot{q}_{\gamma} + BG^{-1}(C\dot{v}) + BG^{-1}(d[B^{\mathrm{T}}F(q^{\gamma})]/dt) = 0$ and hence

$$\frac{d}{dt}\left[q^{\gamma} + BG^{-1}(Cv) + BG^{-1}B^{\mathrm{T}}F\left(q^{\gamma}\right)\right] = 0$$

for any $t \geq 0$. ∎

Theorem 1 implies that the p scalar functions $[Q_{\mathrm{QP}}(v, q^{\gamma})]_i : \mathbb{R}^{q+p} \to \mathbb{R}$, $i = 1, 2, \ldots, p$, are a set of p independent invariants of motion for the dynamics of an MPNN for QP problems in the voltage–current domain.

Let

$$\mathcal{M}(Q_0) = \left\{\left(v, q^{\gamma}\right) \in \mathbb{R}^{q+p} : Q_{\mathrm{QP}}\left(v, q^{\gamma}\right) = Q_0\right\} \subset \mathbb{R}^{q+p}$$

where $Q_0 \in \mathbb{R}^p$ is a constant vector. Note that for a fixed $Q_0 \in \mathbb{R}^q$, $\mathcal{M}(Q_0)$ is a q-dimensional manifold in $\mathbb{R}^{q+p}$ and that, by varying Q_0, we obtain ∞^p nonintersecting manifolds into which $\mathbb{R}^{q+p}$ is decomposed. Due to Theorem 1, each manifold $\mathcal{M}(Q_0)$ is positively invariant for the dynamics of an MPNN. Namely, if $(v(t), q^{\gamma}(t))$ is a solution of (10) and (11) with ICs $(v_0, q^{\gamma}_0) \in \mathcal{M}(Q_0)$, then we have $(v(t), q^{\gamma}(t)) \in \mathcal{M}(Q_0)$ for any $t \geq 0$. On each manifold, the dynamics has a reduced-order q and is described in the flux–charge domain by (4).[2]

2) LP Problems: Analogous results hold for LP problems, for which we define the function $Q_{\mathrm{LP}} : \mathbb{R}^{q+p} \to \mathbb{R}^p$ as follows:

$$Q_{\mathrm{LP}}(v(t), q^{\gamma}(t)) = Cv(t) + B^{\mathrm{T}}F\left(q^{\gamma}(t)\right).$$

[2]The relationship between the solutions of MPNN in the voltage–current domain and those of the reduced-order system on each manifold can be found via a technique analogous to that discussed in [56, Sec. III].

Theorem 2: The time derivative of Q_{LP} along the solutions of (13) and (14) identically vanishes, that is, $\dot{Q}_{\mathrm{LP}}(v(t), q^{\gamma}(t)) = 0$ for any $t \geq 0$.

Proof: From (13), we obtain $d[Cv(t) + B^{\mathrm{T}}F(q^{\gamma}(t))]/dt = 0$ for any $t \geq 0$. ∎

Theorem 2 implies that the p scalar functions $[Q_{\mathrm{LP}}(v, q^{\gamma})]_i : \mathbb{R}^{q+p} \to \mathbb{R}$, $i = 1, 2, \ldots, p$, are a set of p independent invariants of motion for the dynamics of an MPNN for LP problems in the voltage–current domain.

Analogous properties as those previously discussed hold for the invariant manifolds of an MPNN for LP problems

$$\mathcal{M}(Q_0) = \left\{\left(v, q^{\gamma}\right) \in \mathbb{R}^{q+p} : Q_{\mathrm{LP}}\left(v, q^{\gamma}\right) = Q_0\right\} \subset \mathbb{R}^{q+p}$$

where $Q_0 \in \mathbb{R}^p$ is a constant vector.

VI. Dynamic Analysis

A. Nonsmooth MPNN Model

Consider (9) and (12) describing the dynamics of an MPNN for QP and LP problems, respectively. Due to the presence of constraint neurons implemented with filamentary-type memristors, such equations contain stiff nonlinearities $f(\rho)$ with a very low slope for positive ρ and a very high slope for negative ρ. As discussed in [57], in such a case, it is convenient to analyze the limiting case where f is approximated by a nonsmooth discontinuous nonlinearity. The analysis of the discontinuous case is in fact able to highlight salient features of motion as the presence of sliding modes along discontinuity surfaces and the intriguing phenomenon of convergence in finite time. Such properties would be instead difficult to study via a traditional analysis using smooth nonlinearities.

More precisely, we replace f with a discontinuous nonlinearity

$$d(\rho) = \begin{cases} 0, & \rho > 0 \\ [-\sigma, 0], & \rho = 0 \end{cases} \tag{15}$$

where $\sigma > 0$. Moreover, $d(\cdot)$ is a continuous, single-valued, nondecreasing function when $\rho < 0$.

The assumption d is nondecreasing is justified by the fact that, as already mentioned, a passive memristor has a monotone characteristic relating charge and flux [7]. Actually, in the analysis in this article, we do not need to know the exact behavior of $d(\cdot)$ for $\rho < 0$. The only property that matters is that $d(\cdot)$ is nondecreasing and so $d(\rho) \leq -\sigma$ when $\rho < 0$. Again, this is consistent with the fact that actual memristors have boundary effects and saturations such that the characteristic behavior is not exactly known beyond saturation [58]. A possible function $d(\cdot)$ is depicted in Fig. 4(b).

Then, we obtain for an MPNN the nonsmooth model

$$\dot{x} \in -a - Gx - B^{\mathrm{T}}D(Bx - c) \doteq \mathfrak{F}(x) \tag{16}$$

where we let for simplicity, $x = \varphi$ and $D(x) = (d(x_1), d(x_2), \ldots, d(x_q))^{\mathrm{T}}$. We also consider a normalized value $C = 1$ for the capacitance. Note that the vector field $\mathfrak{F}$ defining (16) is multivalued at points where some components of $Bx - c$ vanish, hence we are actually dealing with a differential inclusion rather than an ordinary differential [59].

By a solution of (16) in $[0, T]$ we mean an absolutely continuous function $x(\cdot)$ in $[0, T]$ such that we have $\dot{x}(t) \in \mathfrak{F}(x(t))$ for almost all (a.a.) $t \in [0, T]$. An EP is a stationary solution, that is, $\xi \in \mathbb{R}^q$ is an EP if and only if we have $0 \in \mathfrak{F}(\xi)$. This is equivalent to

$$\eta(\xi) \doteq -G\xi - a \in N_{\mathcal{P}}(\xi) \tag{17}$$

where $N_{\mathcal{P}}(\xi)$ is the normal cone to the convex set $\mathcal{P}$ at ξ [60] (see also [61, Sec. I-a)]). Let $\mathcal{E}$ be the set of EPs of (16).

B. Gradient Differential Inclusion

Let

$$d_{\mathrm{ns}}(\rho) = \begin{cases} 0, & \rho > 0 \\ [-\sigma, 0], & \rho = 0 \\ -\sigma, & \rho < 0 \end{cases} \tag{18}$$

and also consider the single-valued function $d_{\mathrm{s}}(\rho) = d(\rho) - d_{\mathrm{ns}}(\rho)$, which is continuous and nondecreasing in $\mathbb{R}$.

On this basis, define the energy function

$$E(x) = E_0(x) + E_{\mathrm{ns}}(x) + E_{\mathrm{s}}(x) : \mathbb{R}^q \to \mathbb{R}$$

where

$$E_0(x) = \frac{1}{2}x^{\mathrm{T}}Gx + x^{\mathrm{T}}a$$

$$E_{\mathrm{ns}}(x) = \sigma \sum_{j=1}^{p} \max\left\{0, -\gamma_j(x)\right\}$$

$$E_{\mathrm{s}}(x) = \sum_{j=1}^{p} \int_0^{\gamma_j(x)} d_{\mathrm{s}}(\rho)d\rho$$

and $\gamma_j(x) = \langle B_j, x\rangle - c_j = B_j^{\mathrm{T}}x - c_j$ is as in (2).

Function E is convex, (hence) continuous, but not differentiable (Appendix A). Moreover, it is seen that (16) corresponds to the gradient differential inclusion

$$\dot{x} \in \mathfrak{F}(x) = -\partial E(x)$$

where ∂E is the generalized (Clarke's) gradient of E (Appendix B).[3]

The next result is a direct consequence of the fact that (16) is the gradient of a convex function.

Proposition 2:

1) The vector field $\mathfrak{F}$ defining (16) is maximal monotone, that is, we have $\langle x_a - x_b, \zeta_a - \zeta_b\rangle \geq 0$ for any $x_a, x_b \in \mathbb{R}^q$ and any $\zeta_a \in \mathfrak{F}(x_a)$ and $\zeta_b \in \mathfrak{F}(x_b)$.
2) For any $x_0 \in \mathbb{R}^q$, there is a unique solution $x(t)$ of (16) with IC $x(0) = x_0$, which is defined for $t \geq 0$. Furthermore, $t \to E(x(t))$ is a monotone nonincreasing, convex function such that for a.a. $t \geq 0$
$$\dot{E}(x(t)) = -\|\dot{x}(t)\|^2$$
3) The function $t \to \|\dot{x}(t)\|$ is monotone nonincreasing.

Proof: Since E is convex in $\mathbb{R}^q$, the result in point 1) follows from [59, Proposition 1, p. 159]. Points 2) and 3) follow from the fact that $\mathfrak{F}$ is maximal monotone [59, Th. 1, p. 147 and Th. 1, p. 159]. ∎

[3]We refer the reader to [62] and [57, Sec. I(a)] for the definition and basic properties of Clarke's gradient. We recall that since E is convex, ∂E coincides with the classical subdifferential of convex analysis. We also note that $\partial W = \nabla W$, where ∇ denotes the gradient, when W is smooth (C^1).

C. Convergence to the Feasibility Region

The next result addresses the convergence of MPNN solutions to the feasibility region.

Proposition 3: Choose $R > 0$ such that $\mathcal{P} \subset B_R = \{x \in \mathbb{R}^q : \|x\| < R\}$ and let

$$\sigma_{\text{th}} = \frac{2\lambda_M R^2 + \|a\|R}{\chi_{\min}} \tag{19}$$

where $\tilde{x} \in \text{int}\mathcal{P}$, hence $\gamma_i(\tilde{x}) > 0$, $i = 1, 2, \ldots, p$, and

$$\chi_{\min} = \min_{i=1,2,\ldots,p} \{\gamma_i(\tilde{x})\} > 0.$$

Moreover, $\lambda_M = \max_{i=1,2,\ldots,q}\{\lambda_i\}$ and λ_i are the eigenvalues of G.

Assume that $\sigma > \sigma_{\text{th}}$ and let $x(t)$, $t \geq 0$, be the solution of (16) with IC $x(0) = x_0 \in B_R$. Then:

1) the set B_R is positively invariant for the dynamics of (16), that is, $x_0 \in B_R$ implies $x(t) \in B_R$ for any $t \geq 0$;
2) the feasibility region $\mathcal{P}$ is positively invariant for the dynamics of (16);
3) the solution $x(\cdot)$ reaches $\mathcal{P}$ in finite time $\bar{t}$ and stays in $\mathcal{P}$ thereafter, that is, $x(t) \in \mathcal{P}$ for any $t \geq \bar{t}$;
4) within $\mathcal{P}$, that is, for $t \geq \bar{t}$, the solution $x(\cdot)$ satisfies the differential variational inequality (DVI)

$$\dot{x} \in -Gx - a - N_{\mathcal{P}}(x).$$

Proof: If we let $y = x - \tilde{x}$, then (16) becomes

$$\dot{y} \in -a - G\tilde{x} - Gy - B^{\text{T}}D(By + B\tilde{x} - c) \tag{20}$$

that has the form of the nonsmooth gradient system (3) studied in [61]. Moreover, $y = 0 \in \text{int}\mathcal{P}$ and $(B\tilde{x} - c)_i = \gamma_i(\tilde{x}) > 0$, $i = 1, 2, \ldots, p$. Due to [61, Properties 1 and 2], the results in the proposition hold whenever $\sigma > M_a M_b$, where

$$M_a = \max_{y \in B_R} \|Gy + a + G\tilde{x}\|, M_b = \frac{R}{\chi_{\min}}.$$

To complete the proof, note that $\|Gy + a + G\tilde{x}\| \leq \|Gy\| + \|a\| + \|G\tilde{x}\| \leq \lambda_M R + \|a\| + \lambda_M R = 2R\lambda_M + \|a\|$.[4] ■

Proposition 4: If $\sigma > \sigma_{\text{th}}$, we have $\mathcal{E} \cap B_R = \mathcal{M}$.

Proof: Follows by applying [61, Property 3] to (20). ■

Proposition 4 means that for sufficiently large σ, MPNN implements an exact penalty function method, where the set $\mathcal{E}$ of EPs of (16) in B_R coincides with the set of global minimizers of the QP or LP problem.

D. Convergence to Constrained Minima

Here, we give the main results on the convergence of solutions of the MPNN and global optimization capabilities.

Theorem 3: Assume that $\sigma > \sigma_{\text{th}}$ and let $x(t)$, $t \geq 0$, be the solution of (16) with IC $x(0) = x_0 \in B_R$. Then, $x(\cdot)$ has a finite length on $[0, +\infty)$, that is

$$\int_0^{+\infty} \|\dot{x}(t)\| dt < +\infty$$

and converges to an EP in $\mathcal{M}$ as $t \to +\infty$.

[4]Nonlinearity $d(\rho)$ in model (16), which is nondecreasing for $\rho < 0$, is slightly more general than the nonlinearity $d_{\text{ns}}(\rho)$ used in [61], which has 0 slope for $\rho < 0$. However, the proof of [61, Properties 1 and 2] can be straightforwardly adapted to be applied to d. We omit the details for brevity.

Proof: Once more, consider the system (20). The result follows from [61, Th. 3], where a nonsmooth version of the Łojasiewicz inequality is used to prove the finiteness of the trajectory length and convergence of each solution toward an EP in $\mathcal{M}$. ■

The next results state that for QP problems, there is either global exponential convergence with a known rate, or global convergence in finite time, to a global optimum. Moreover, for LP problems, global convergence is always in finite time. These convergence properties make it clear that MPNNs are effective for solving LP and QP problems in real time.

Theorem 4: Consider the MPNN (16) for QP problems and let $\sigma > \sigma_{\text{th}}$. Let $\mathcal{M} = \{x^*\}$, where x^* is the global minimizer of the QP problem and let $x(\cdot)$ be the solution of (16) with IC $x(0) = x_0 \in B_R$. The following hold.

1) If $\eta(x^*) = -Gx^* - a \in \text{int}N_{\mathcal{P}}(x^*)$, then $x(\cdot)$ converges in finite time to x^*, that is, there exists t_{f} such that $x(t) = x^*$ for any $t \geq t_{\text{f}}$.
2) If $\eta(x^*) = -Gx^* - a \in \text{bd}N_{\mathcal{P}}(x^*)$, where bd denotes the boundary, then $x(\cdot)$ converges exponentially to x^*, that is, there exist $\bar{t} > 0$ and $\kappa', \kappa'' > 0$ such that

$$\left\|x(t) - x^*\right\| \leq \kappa' \exp\left\{-\kappa''\left(t - \bar{t}\right)\right\}, \ t \geq \bar{t}.$$

Proof: The result is a consequence of a nonsmooth version of the Łojasiewicz inequality for QP problems [61, Th. 1] and Theorem 3 in the quoted paper. ■

Theorem 5: Consider the MPNN (16) for LP problems and let $\sigma > \sigma_{\text{th}}$. Let $x(\cdot)$ be the solution of (16) with IC $x(0) = x_0 \in B_R$. Then, $x(\cdot)$ converges in finite time to a singleton in $\mathcal{M}$, that is, there exist $\bar{x}^* \in \mathcal{M}$ (depending upon x_0) and t_{f} such that $x(t) = \bar{x}^*$ for any $t \geq t_{\text{f}}$.

Proof: The result follows from a nonsmooth version of the Łojasiewicz inequality for LP problems [61, Th. 2] and Theorem 4 in the quoted paper. ■

It is worth to stress that due to Theorem 5, for LP problems convergence in finite time to a global minimizer holds even if (16) has a *continuum* of (nonisolated) EPs.

E. Convergence to 0 of Capacitor Voltages

In view of the practical applications of MPNNs, we are interested not only in the convergence of $x(\cdot)$ toward a global minimizer but also in convergence to 0 of the capacitor voltages v. In fact, if $v \to 0$, it is easy to see that all voltages and currents drop off in an MPNN. Namely, power drops off after a quick transient, with advantages in terms of energy consumption with respect to traditional NNs operating in the voltage–current domain.

Since a solution $x(\cdot)$ of an MPNN is absolutely continuous, $x(\cdot)$ is differentiable for a.a. $t \geq 0$ and $\dot{x}(\cdot)$ is a measurable function. Recalling that $C = 1$, the vector of capacitor voltages is simply given by $v(t) = C\dot{x}(t) = \dot{x}(t)$ for a.a. $t \geq 0$.

It is worth to remark that since we are dealing with a differential equation with a discontinuous right-hand side, the convergence of $x(\cdot)$ does not, in general, ensure convergence to 0 of $\dot{x}$. The reader is referred to [57, Example 2] for an example of this kind. In what follows, we show however that for an MPNN we can rule out such an unwanted phenomenon.

We start for simplicity with LP problems.

Corollary 1: Consider the MPNN (16) for LP problems and let $\sigma > \sigma_{\text{th}}$. Let $x(\cdot)$ be the solution of (16) with IC $x(0) = x_0 \in B_R$. Then, there exists t_{f} such that $v(t) = 0$, $t > t_{\text{f}}$.

Proof: Simply follows from Theorem 5. ∎

Consider now an MPNN for QP problems. In this case, in general, we do not have convergence in finite time of a solution $x(\cdot)$. Moreover, there may exist instants accumulating at $t = +\infty$ for which $\dot{x}(\cdot)$ is not defined. In such a case, there are difficulties in using the standard concept of the limit as $t \to +\infty$. Following the discussion in [59, p. 311], we can however use the concept of almost limit, as in the next definition, which is in practice as useful as that of the limit when dealing with functions that are only differentiable for a.a. t.

Definition 1 [59]: Let $\rho(t) : [0, +\infty) \to \mathbb{R}^q$ be a measurable function. We say that $\bar{\rho} \in \mathbb{R}^q$ is the almost limit of ρ as $t \to +\infty$, and we write $\bar{\rho} = a\lim_{t\to+\infty}\rho(t)$, if for any $\epsilon > 0$ there exists $t_\epsilon > 0$ such that $\mu\{t \in [t_\epsilon, +\infty) : \|\rho(t) - \bar{\rho}\| > \epsilon\} = 0$, where μ denotes the measure.

Theorem 6: Consider the MPNN (9) for QP problems and let $\sigma > \sigma_{\text{th}}$. Let $x(\cdot)$ be the solution of (9) with IC $x(0) = x_0 \in B_R$. The following hold.

1) If $\eta(x^*) = -Gx^* - a \in \text{int}N_{\mathcal{P}}(x^*)$, then there exists t_{f} such that we have $v(t) = 0$ for any $t \geq t_{\text{f}}$.
2) If $\eta(x^*) = -Gx^* - a \in \text{bd}N_{\mathcal{P}}(x^*)$, then we have $a\lim_{t\to+\infty} v(t) = 0$.

Proof: 1) Follows from point 1) of Theorem 4. 2) According to point 3) of Proposition 2, function $t \to \|\dot{x}(t)\|$ is monotone nonincreasing for $t \geq 0$. Let us verify that $\|\dot{x}(t)\|$ decreases to 0 as $t \to +\infty$. We argue by a contradiction assuming $\|\dot{x}(\cdot)\|$ decreases to $\bar{\varepsilon} > 0$ as $t \to +\infty$. In this case, we have $\|\dot{x}(t)\| \geq \bar{\epsilon}$ for a.a. $t \geq 0$ and so $\lim_{\tau\to+\infty} \int_0^\tau \|\dot{x}(t)\|dt = +\infty$, contradicting the fact that any solution has finite length on $[0, +\infty)$ (Theorem 3). Since $\|\dot{x}(t)\|$ decreases to 0 as $t \to +\infty$, then for any $\epsilon > 0$, there is $t_\epsilon > 0$ such that $\|\dot{x}(t)\| < \epsilon$ for a.a. $t \geq t_\epsilon$, which implies $a\lim_{t\to+\infty} \|\dot{x}(t)\| = 0$. ∎

VII. Robustness of Global Convergence

In Section VI, we have given a number of fundamental results concerning global convergence of solutions for an MPNN. Such results are heavily dependent on the fact that a nominal MPNN is described by a gradient system of differential inclusions. Since even arbitrarily small perturbations of a gradient system might result in a system that is no longer described by the gradient of some function, robustness issues concerning convergence become important.

Such robustness problems are well known in the context of the Hopfield-type NNs and CNNs. Recall that a symmetric CNN is a gradient system and enjoys convergence toward EPs when the interconnections are symmetric. However, as shown in [63], there are competitive CNNs that implement a globally inconsistent decision scheme and display large-size nonvanishing oscillations even arbitrarily close to symmetry. The reader is referred to [64] and [65] for other types of oscillatory or complex dynamics that can be observed for small perturbations of symmetric matrices in NNs.

The goal of this section is to show that an MPNN enjoys some basic robustness properties of convergence. In general, we may study the robustness of convergence with respect to perturbations of the interconnection matrices G or B, vector a, or the threshold σ of nonlinearity d. It is easily seen that the convergence results in Section VI are robust with respect to small variations of a or σ since under such variations, an MPNN is still described by a gradient system. We thus consider a perturbed MPNN for solving QP problems described by

$$\dot{x} \in -a - Hx - B^{\text{T}}D(Bx - c) \doteq \mathfrak{H}(x) \tag{21}$$

where $H = G + \Delta G$ and ΔG is a nonsymmetric perturbation of G, that is, $\Delta G^{\text{T}} \neq \Delta G$ and so $H^{\text{T}} \neq H$. Note that matrix B is assumed to be unperturbed. The case where B is perturbed can be dealt with analogous techniques but a detailed treatment is omitted here to avoid repetitions and an excessive length.

The loss of symmetry of H implies that (21) is no longer a gradient system. Nonetheless, a result analogous to that of Proposition 3 holds with σ_{th} replaced by

$$\tilde{\sigma}_{\text{th}} = \frac{2\sqrt{\xi_M}R^2 + \|a\|R}{\chi_{\min}}$$

where $\xi_M = \max_{i=1,2,\ldots,q}\{\xi_i\} > 0$ and ξ_i are the eigenvalues of $H^{\text{T}}H$ (Appendix C). The only difference is that, since $\mathfrak{H}$ is no longer maximal monotone, we are not able to prove the uniqueness of the solution in $B_R \backslash \mathcal{P}$. The uniqueness of the solution holds instead within set $\mathcal{P}$, where the perturbed system still satisfies a DVI, as stated in the next result.

Proposition 5: Assume that $\sigma > \tilde{\sigma}_{\text{th}}$. Then, within $\mathcal{P}$, system (21) is equivalent to the DVI

$$\dot{x} \in -Hx - a - N_{\mathcal{P}}(x). \tag{22}$$

Such a DVI enjoys the property of uniqueness of the solution with respect to the IC in $\mathcal{P}$.

Proof: Let $x(\cdot)$ be a solution of (21) such that $x(t) \in \mathcal{P}$ for $t \geq \bar{t}$. We have $-B^{\text{T}}D(Bx - c) = -\partial(E_{\text{ns}}(x) + E_{\text{ns}}(x)) \in N_P(x)$ [61, p. 1476], hence there exists $\eta(t) \in N_{\mathcal{P}}(x(t))$ such that $\dot{x}(t) \in -Hx(t) - a - \eta(t)$ for a.a. $t \geq 0$. This means that $x(\cdot)$ is also a solution of the DVI (22). The uniqueness of the solution in $\mathcal{P}$ holds due to [66, Property 3]. The uniqueness implies that (21) and (22) are equivalent within $\mathcal{P}$. ∎

Let us now study the existence, uniqueness, and global stability of the EP of (21).

Proposition 6: If the symmetric part of H, that is, $H^S \doteq (1/2)(H + H^{\text{T}})$, is positive definite, then (21) has a unique EP $\xi \in \mathcal{P}$.

Proof: By following a procedure as that to prove [61, Property 1, point 2)], it can be verified that in $B_R \backslash \mathcal{P}$, there is no EP of (21). Within $\mathcal{P}$, there exists instead at least an EP of (21) due to the viability theorem for DVIs (see [59, Th. 1, p. 267]). Finally, let us prove that the EP is unique. We argue by a contradiction assuming there exist two different EPs $\xi_1 \neq \xi_2$ of (21) in $\mathcal{P}$. Then, there exist $\eta_1 \in N_{\mathcal{P}}(\xi_1)$ and $\eta_2 \in N_{\mathcal{P}}(\xi_2)$ such that $-H\xi_1 - a = \eta_1$ and $-H\xi_2 - a = \eta_2$. Due to the monotonicity of the normal cone [59], it is known

that $\langle \xi_1 - \xi_2, \eta_1 - \eta_2 \rangle \geq 0$. However, we also obtain

$$\begin{aligned}\langle \xi_1 - \xi_2, \eta_1 - \eta_2 \rangle &= \langle \xi_1 - \xi_2, -H\xi_1 - a + H\xi_2 + a \rangle \\ &= \langle \xi_1 - \xi_2, -H(\xi_1 - \xi_2) \rangle \\ &= -(\xi_1 - \xi_2)^{\mathrm{T}} H (\xi_1 - \xi_2) \\ &= -(\xi_1 - \xi_2)^{\mathrm{T}} H^S (\xi_1 - \xi_2) < 0\end{aligned}$$

since H^S is positive definite and $\xi_1 \neq \xi_2$. Hence, we have reached a contradiction. ∎

Theorem 7: Assume that H^S is positive definite and let $x(\cdot)$ be a solution of (21) with IC $x(0) = x_0 \in B_R$. Then, $x(\cdot)$ converges exponentially to the unique EP ξ, namely, there exists $\bar{t} \geq 0$ such that

$$\|x(t) - \xi\| \leq \|x(\bar{t}) - \xi\| \exp\left\{-\tilde{\lambda}_m (t - \bar{t})/2\right\}, \quad t \geq \bar{t}$$

where $\tilde{\lambda}_m = \min_{i=1,2,\ldots,q}\{\tilde{\lambda}_i\} > 0$ and $\tilde{\lambda}_i$ are the eigenvalues of H^S.

Proof: We already noted that any solution of (21) with IC in B_R reaches $\mathcal{P}$ in finite time and stays in $\mathcal{P}$ thereafter. Then, it suffices to consider a solution $x(\cdot)$ such that $x(\bar{t}) \in \mathcal{P}$ and show that it converges exponentially to ξ for $t \geq \bar{t}$. To this end, consider the candidate Lyapunov function

$$V(x) = \frac{1}{2}\|x - \xi\|^2 = \frac{1}{2}(x - \xi)^{\mathrm{T}}(x - \xi).$$

The function $V(x(t))$ is differentiable for a.a. $t \geq 0$ and we have $\dot{V}(x(t)) = \langle \nabla V(x(t)), \dot{x}(t) \rangle = \langle x(t) - \xi, \dot{x}(t) \rangle$. From (22), for a.a. $t \geq \bar{t}$, there exists $\eta(t) \in N_{\mathcal{P}}(x(t))$ such that $\dot{x}(t) = -Hx(t) - a - \eta(t)$. Furthermore, since ξ is an EP of (21), $-H\xi - a = \eta_\xi$, for some $\eta_\xi \in N_{\mathcal{P}}(\xi)$. Therefore, we obtain

$$\begin{aligned}\dot{V}(x(t)) &= \langle x(t) - \xi, \dot{x}(t) \rangle \\ &= \left\langle x(t) - \xi, -Hx(t) - a - \eta(t) + H\xi + a + \eta_\xi \right\rangle \\ &= \left\langle x(t) - \xi, -H(x(t) - \xi) - \eta(t) + \eta_\xi \right\rangle \\ &= -\langle x(t) - \xi, H(x(t) - \xi) \rangle \\ &\quad - \left\langle x(t) - \xi, \eta(t) - \eta_\xi \right\rangle \\ &\leq -\langle x(t) - \xi, H(x(t) - \xi) \rangle \\ &\leq -\left\langle x(t) - \xi, H^S(x(t) - \xi) \right\rangle \\ &\leq -\tilde{\lambda}_m \|x(t) - \xi\|^2 = -\tilde{\lambda}_m V(x(t))\end{aligned}$$

where we have taken into account that the normal cone $N_{\mathcal{P}}(\cdot)$ is a maximal monotone operator and so $\langle x(t) - \xi, \eta(t) - \eta_\xi \rangle \geq 0$. Now, [66, Th. 6] yields the result. ∎

It is worth remarking that standard techniques based on LMIs [67] can be effectively used to find the allowed perturbations ΔG for which H^S is positive definite and the result on global convergence in Theorem 7 holds.

VIII. Application Examples

Here, we present some toy examples of QP and LP problems to verify the theoretical results and illustrate the optimization capabilities of MPNNs.

Example 1. Consider the following QP problem. Minimize $\psi(x) = (1/2)x^{\mathrm{T}}Gx + ax$, where

$$G = \frac{1}{4}\begin{pmatrix} 1.5 & -1 \\ -1 & 2 \end{pmatrix}, \quad a = (-1.2, -3.5)^{\mathrm{T}} \tag{23}$$

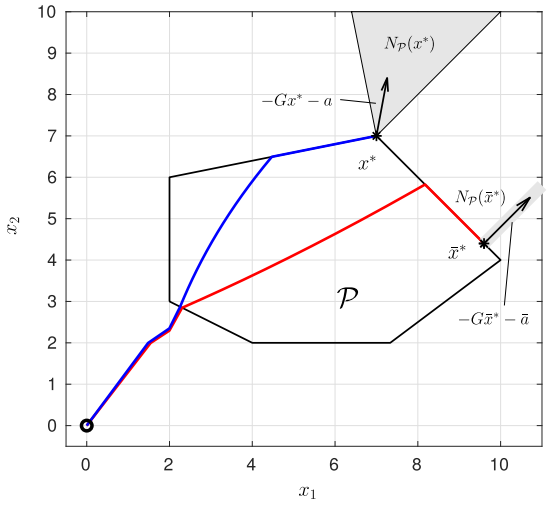

Fig. 5. Trajectory of MPNN for solving the QP problem in Example 1 (blue) and that in Example 2 (red).

and $x = (x_1, x_2)^{\mathrm{T}} \in \mathbb{R}^2$ $(q = 2)$, subject to the affine constraints $B^{\mathrm{T}}x - c \geq 0$, where $(p = 6)$

$$B = \begin{pmatrix} 1 & 1.5 & -1 & -4/3 & 1/2 & 0 \\ 0 & -1 & -1 & 1 & 1 & 1 \end{pmatrix} \tag{24}$$

and

$$c = (2, -28/5, -14, -7/2, 4, 2)^{\mathrm{T}} \tag{25}$$

defining the polyhedron in Fig. 5. In this case, $\mathcal{M} = \{x^*\} = \{(7, 7)^{\mathrm{T}}\}$ is a vertex of $\mathcal{P}$ and $-Gx^* - a \in \mathrm{int}\mathcal{N}_{\mathcal{P}}(x^*)$ (Fig. 5). If we pick $R = 11$ and $\tilde{x} = (7, 4.5)^{\mathrm{T}}$, we obtain from (19) that $\sigma_{\mathrm{th}} = 76.8$.

We used MPNN (16) for solving this QP problem. The matrices G and B and vector c in the MPNN model (16) are chosen as in (23)–(25), respectively. The discontinuous nonlinearity $d(\cdot)$ is as in (15), where we have chosen $\sigma = 80 > \sigma_{\mathrm{th}}$. Next, we report the numerical simulations of the MPNN (16) for this QP problem. Recall that according to the design procedure of an MPNN in Section V, we are interested in the evolution of the solution of (16) starting at $x(0) = \varphi(0) = 0$. Fig. 5 shows the trajectory starting at the origin and converging to the global minimizer x^*, while Fig. 6 shows the time evolution of the memristor fluxes $x_1(\cdot) = \varphi_1(\cdot)$ and $x_2(\cdot) = \varphi_2(\cdot)$ showing convergence to the exact solution x^* in finite time. These results are in accordance with Theorem 4. Fig. 6 also shows the evolution of the capacitor voltages $v_1(\cdot) = \dot{x}_1(\cdot)$ and $v_2(\cdot) = \dot{x}_2(\cdot)$, confirming that they tend to 0 in finite time (Theorem 6). Hence, as predicted, power turns off in the MPNN after the transient is over, with advantages in terms of power consumption. The memristor in the neurons are able to hold in memory the result of the computation, that is, the memristor fluxes, which can be indirectly measured via the memristance. This behavior is in accordance with the in-memory computing principle.

For comparison, we also simulated KCNN (3) for solving the same QP problem. In particular, the nonlinearity $s(\cdot)$ in (3) has been chosen according to [28] as $s(\rho) = 0$ for $\rho \geq 0$, while $s(\rho) = \kappa\rho$ for $\rho < 0$, where $\kappa > 0$. Fig. 7 shows the corresponding time-domain behavior of the capacitor voltages $v_1(\cdot)$

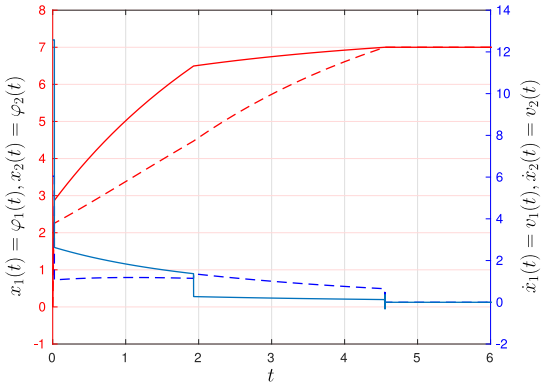

Fig. 6. Evolution of $x_1(\cdot) = \varphi_1(\cdot)$ (red solid), $x_2(\cdot) = \varphi_2(\cdot)$ (red dashed) and $\dot{x}_1(\cdot) = v_1(\cdot)$ (blue solid), $\dot{x}_2(\cdot) = v_2(\cdot)$ (blue dashed) for the MPNN in Example 1.

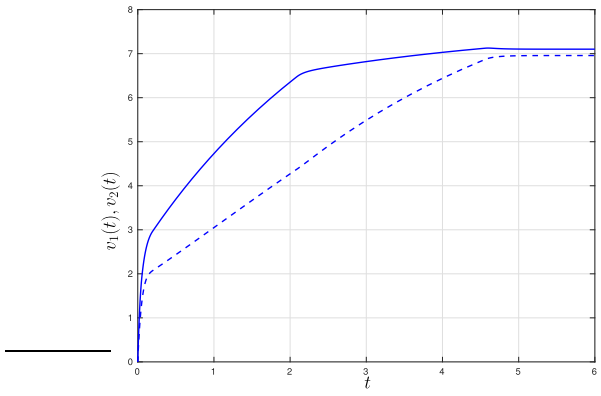

Fig. 7. Evolution of the capacitor voltages $v_1(\cdot)$ (blue solid) and $v_2(\cdot)$ (blue dashed) for KCNN (3) used for solving the QP problem in Example 1.

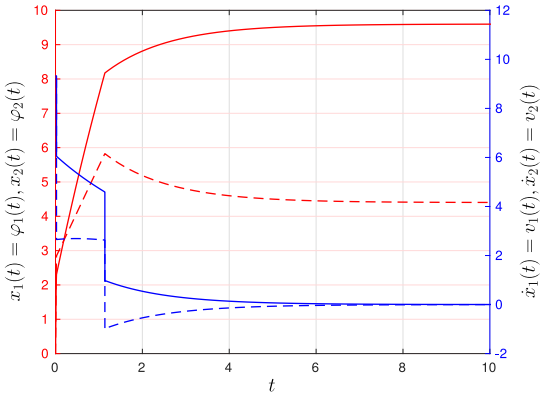

Fig. 8. Evolution of $x_1(\cdot) = \varphi_1(\cdot)$ (red solid), $x_2(\cdot) = \varphi_2(\cdot)$ (red dashed) and $\dot{x}_1(\cdot) = v_1(\cdot)$ (blue solid), $\dot{x}_2(\cdot) = v_2(\cdot)$ (blue dashed) for the MPNN in Example 2.

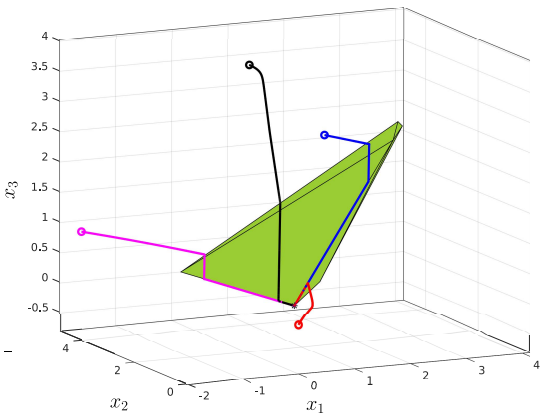

Fig. 9. Polyhedron $\mathcal{P}$ (green) and trajectories of MPNN for solving the LP problem in Example 3.

and $v_2(\cdot)$, when the ICs are $v_1(0) = v_2(0) = 0$ and $\kappa = 10$. It is seen that v_1 tends to 7.1, while v_2 tends to 6.96, which approximates the exact solution $(7, 7)$ of the problem. This is due to the fact that KCNN is a smooth NN that implements an inexact penalty method. The most important difference is that the capacitor voltages do not tend to 0 for KCNN, hence power does not turn off in the steady state. As already discussed, in practice, the capacitors in a KCNN are unable to hold in memory the result of computation so that an extra memory would be needed with potential problems related to the Von Neumann bottleneck.

Example 2. Consider the same QP problem as in Example 1 with a replaced by $\bar{a} = (-6.2, -3.5)^{\mathrm{T}}$. Now, $\mathcal{M} = \{\bar{x}^*\} = \{(9.6, 4.4)^{\mathrm{T}}\}$ is a point on a face of $\mathcal{P}$ and $-G\bar{x}^* - \bar{a} \in \mathrm{bd}\mathcal{N}_{\mathcal{P}}(\bar{x}^*)$ (Fig. 5). Fig. 8 shows the simulation results obtained by applying MPNN (16) to this QP problem. Note that we have exponential convergence of $x(\cdot)$ to $\bar{x}^*$ and exponential convergence to 0 of $\dot{x}(\cdot)$ in accordance with Theorems 4 and 6.

Example 3. Consider the following LP problem. Minimize $\psi(x) = a'x$ with $a = (6, 3, 4)^{\mathrm{T}}$ and $x = (x_1, x_2, x_3)^{\mathrm{T}} \in \mathbb{R}^3$ $(q = 3)$ subject to the constraints $B^{\mathrm{T}}x - c \geq 0$, where

$$B^{\mathrm{T}} = \begin{pmatrix} -4 & 7 & 1 & 1 & 0 & -1 \\ 3 & 1 & 0 & 1 & 0 & -1 \\ 3 & -2 & 0 & 0 & 1 & -1 \end{pmatrix}$$

and $c = (2, 6, 0, 0, 0, -5)^{\mathrm{T}}$ $(p = 6)$ defining the polyhedron in Fig. 9. The theoretic optimal is $\mathcal{M} = \{x^*\} = \{(16/25, 38/25, 0)^{\mathrm{T}}\}$ on a vertex of $\mathcal{P}$. By choosing $R = 5$ and $\tilde{x} = (0.5, 0.3, 1)^{\mathrm{T}}$, we obtain $\sigma_{\mathrm{th}} = 12.2$. Let $\sigma = 15$. Next, we show the simulation results obtained by applying MPNN (16) to this LP problem. Fig. 9 shows in red the trajectory in the state space $\mathbb{R}^3$ starting at the origin and converging to $(0.6398, 1.5202, -0.0005)^{\mathrm{T}}$, in close agreement with the theoretic results x^*. From the time evolution, we can also observe convergence in finite time of $x(\cdot)$ in accordance with Theorem 5 and convergence to 0 in finite time of $\dot{x}(\cdot)$ as predicted by Corollary 1. As shown in the figure, convergence to x^* holds also for other trajectories starting at different ICs.

IX. Conclusion

To the best of our knowledge, the MPNNs introduced in this article are the first NNs using the unconventional features of memristors to obtain advantages for solving QP and LP problems with respect to traditional NNs. One main feature is that an MPNN operates and computes in the flux–charge domain, rather than in the traditional voltage–current domain. This yields advantages as reduced power consumption and the possibility of implementing a computation scheme according to the principle of in-memory computing. The approach

presented in this article is based on constructing an NN whose behavior in the flux–charge domain is analogous to that displayed by a KCNN in the voltage–current domain. Such an approach can, in principle, be extended also to other NNs in the literature for solving QP and LP problems, as those cited in Section I. The key point is to construct via the use of memristors an NN that is analogous to the original one but works in the flux–charge domain. In doing so, memristors should be carefully selected so that their flux–charge (or charge–flux) characteristics correspond to the voltage–current nonlinearities used in the original NN models. Addressing these problems is in our opinion a good topic to be further investigated in future works. Further work will be also devoted to address the case of more general functions to optimize or include more general constraints, as equality constraints, in the optimization problem.

APPENDIX A

The function E_0 is convex since G is symmetric and positive definite. It is known that $E_{\rm ns}$ is nonsmooth but convex [61]. Let us now verify that $E_{\rm s}$ is a (smooth) convex function. Note that $E_{\rm s}$ is convex if and only if, for any $j = 1, 2, \ldots, p$, function $\int_0^{B_j^{\rm T}x-c_j} d_{\rm s}(\rho)d\rho : \mathbb{R}^q \to \mathbb{R}$ is convex, that is, the following inequality holds for any $x_1, x_2 \in \mathbb{R}^q$ and $\lambda \in [0, 1]$:

$$\begin{aligned}&\int_0^{B_j^{\rm T}[\lambda x_1+(1-\lambda)x_2]-c_j} d_s(\rho)d\rho\\ &= \int_0^{\lambda\left(B_j^{\rm T}x_1-c_j\right)+(1-\lambda)\left(B_j^{\rm T}x_2-c_j\right)} d_s(\rho)d\rho\\ &\le \lambda\int_0^{B_j^{\rm T}x_1-c_j} d_s(\rho)d\rho + (1-\lambda)\int_0^{B_j^{\rm T}x_2-c_j} d_s(\rho)d\rho.\end{aligned}$$

This in turn is true if function $\nu(y) = \int_0^y d_s(\rho)d\rho : \mathbb{R} \to \mathbb{R}$ is convex in $\mathbb{R}$. Since $\nu'(y) = d_s(y)$ and $d_s(y)$ is a monotone nondecreasing function, ν is indeed convex. Then, E is convex being the sum of three convex functions.

APPENDIX B

Let us rewrite system (16) as

$$\dot{x} = -a - Gx - B^{\rm T}D_{\rm ns}(Bx-c) - B^{\rm T}D_{\rm s}(Bx-c) \qquad (26)$$

where $D_{\rm ns}(x) = (d_{\rm ns}(x_1), d_{\rm ns}(x_2), \ldots, d_{\rm ns}(x_q))^{\rm T}$ and $D_{\rm s}(x) = (d_{\rm s}(x_1), d_{\rm s}(x_2), \ldots, d_{\rm s}(x_q))^{\rm T}$. Due to the symmetry of G, $-Gx - a = -\nabla E_0(x)$. Moreover, it has been shown in [61, Sec. II] that $-B^{\rm T}D_{\rm ns}(Bx-c) = -\partial E_{\rm ns}(x)$. Finally, let us verify that $-B^{\rm T}D_{\rm s}(Bx-c) = -\nabla E_{\rm s}(x)$. We have

$$\begin{aligned}\nabla E_{\rm s}(x) &= \sum_{j=1}^{q}\nabla\left(\int_0^{B_j^{\rm T}x-c_j} d_{\rm s}(\rho)d\rho\right)\\ &= \sum_{j=1}^{q} B_j d_{\rm s}\left(B_j^{\rm T}x-c_j\right) = B^{\rm T}D_{\rm s}(Bx-c).\end{aligned}$$

The result is proved considering that since E_0, $E_{\rm ns}$, and $E_{\rm s}$ are convex, $\partial\sum = \sum\partial$ [29, Sec. I-a)].

APPENDIX C

Let $y = x - \tilde{x}$ so that (21) becomes

$$\dot{y} \in -a - H\tilde{x} - Hy - B^{\rm T}D(By + B\tilde{x} - c)$$

that has the form of the nonsmooth system (3) studied in [61]. We also have $y = 0 \in {\rm int}\mathcal{P}$ and $(B\tilde{x}-c)_i = \gamma_i(\tilde{x}) > 0$, $i = 1, 2, \ldots, p$. We can use [61, Properties 1 and 2] to show that the claimed results hold whenever $\sigma \ge M_a M_b$, where

$$M_a = \max_{x\in B_R}\|Hx + a + H\tilde{x}\|,\ M_b = \frac{R}{\chi_{\min}}.$$

Now, it is enough to note that $\|Hx + a + H\tilde{x}\| \le \|Hx\| + \|a\| + \|H\tilde{x}\|$, moreover, for any $y \in \mathbb{R}^q$, we have $\|Hy\|^2 = \langle Hy, Hy\rangle = y^{\rm T}H^{\rm T}Hy \le \xi_M\|y\|^2$, hence $\|Hx + a + H\tilde{x}\| \le 2\sqrt{\xi_M}R + \|a\|$.

REFERENCES

[1] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017.

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. ACM 1st Ed. MCC Workshop Mobile Cloud Comput.*, 2012, pp. 13–16.

[3] O. Krestinskaya, A. P. James, and L. O. Chua, "Neuromemristive circuits for edge computing: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 1, pp. 4–23, Jan. 2020, doi: 10.1109/TNNLS.2019.2899262.

[4] M. M. Waldrop, "The chips are down for Moore's law," *Nat. News*, vol. 530, no. 7589, p. 144, 2016.

[5] R. S. Williams, "What's next? [The end of Moore's law]," *Comput. Sci. Eng.*, vol. 19, no. 2, pp. 7–13, 2017.

[6] M. A. Zidan, J. P. Strachan, and W. D. Lu, "The future of electronics based on memristive systems," *Nat. Electron.*, vol. 1, no. 1, p. 22, 2018.

[7] L. O. Chua, "Memristor-the missing circuit element," *IEEE Trans. Circuit Theory*, vol. CT-18, no. 5, pp. 507–519, Sep. 1971.

[8] R. Tetzlaff, Ed., *Memristors and Memristive Systems*. New York, NY, USA: Springer, 2014.

[9] L. Chua, G. Sirakoulis, and A. Adamatzky, Eds., *Handbook of Memristor Networks*, vols. 1–2. New York, NY, USA: Springer, 2019.

[10] D. Ielmini and H.-S. P. Wong, "In-memory computing with resistive switching devices," *Nat. Electron.*, vol. 1, no. 6, p. 333, 2018.

[11] F. Corinto, M. Di Marco, M. Forti, and L. Chua, "Nonlinear networks with MEM-elements: Complex dynamics via flux-charge analysis method," *IEEE Trans. Cybern.*, early access, Apr. 3, 2019, doi: 10.1109/TCYB.2019.2904903.

[12] F. L. Traversa and M. Di Ventra, "Universal memcomputing machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2702–2715, Nov. 2015.

[13] S. R. Nandakumar, S. R. Kulkarni, A. V. Babu, and B. Rajendran, "Building brain-inspired computing systems: Examining the role of nanoscale devices," *IEEE Nanotechnol. Mag.*, vol. 12, no. 3, pp. 19–35, Nov. 2018.

[14] C. Li *et al.*, "Long short-term memory networks in memristor crossbar arrays," *Nat. Mach. Intell.*, vol. 1, no. 1, p. 49, 2019.

[15] H. Kim, M. Sah, C. Yang, T. Roska, and L. O. Chua, "Memristor bridge synapses," *Proc. IEEE*, vol. 100, no. 6, pp. 2061–2070, Jun. 2012.

[16] E. Covi, S. Brivio, A. Serb, T. Prodromakis, M. Fanciulli, and S. Spiga, "Analog memristive synapse in spiking networks implementing unsupervised learning," *Front. Neurosci.*, vol. 10, p. 482, Oct. 2016.

[17] S. P. Adhikari, C. Yang, H. Kim, and L. O. Chua, "Memristor bridge synapse-based neural network and its learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 9, pp. 1426–1435, Sep. 2012.

[18] D. Ielmini, "Brain-inspired computing with resistive switching memory (RRAM): Devices, synapses and neural networks," *Microelectron. Eng.*, vol. 190, pp. 44–53, Apr. 2018.

[19] J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, vol. 79, pp. 2554–2558, Apr. 1982.

[20] L. O. Chua and L. Yang, "Cellular neural networks: Theory," *IEEE Trans. Circuits Syst.*, vol. 35, no. 10, pp. 1257–1272, Oct. 1988.

[21] Q. Xiao and Z. Zeng, "Scale-limited Lagrange stability and finite-time synchronization for memristive recurrent neural networks on time scales," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 2984–2994, Oct. 2017.

[22] P. Liu, Z. Zeng, and J. Wang, "Multistability of recurrent neural networks with nonmonotonic activation functions and mixed time delays," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 4, pp. 512–523, Jul. 2016.

[23] A. Wu, S. Wen, and Z. Zeng, "Synchronization control of a class of memristor-based recurrent neural networks," *Inf. Sci.*, vol. 183, no. 1, pp. 106–116, 2012.

[24] Z. Guo, J. Wang, and Z. Yan, "Global exponential dissipativity and stabilization of memristor-based recurrent neural networks with time-varying delays," *Neural Netw.*, vol. 48, pp. 158–172, Dec. 2013.

[25] G. Bao and Z. Zeng, "Multistability of periodic delayed recurrent neural network with memristors," *Neural Comput. Appl.*, vol. 23, nos. 7–8, pp. 1963–1967, 2013.

[26] I. B. Pyne, "Linear programming on an electronic analogue computer," *Trans. Amer. Inst. Elect. Eng. I Commun. Electron.*, vol. 75, no. 2, pp. 139–143, 1956.

[27] D. W. Tank and J. J. Hopfield, "Simple 'neural' optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit," *IEEE Trans. Circuits Syst.*, vol. CAS-33, no. 5, pp. 533–541, May 1986.

[28] M. P. Kennedy and L. O. Chua, "Neural networks for nonlinear programming," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 35, no. 5, pp. 554–562, May 1988.

[29] M. Forti, P. Nistri, and M. Quincampoix, "Generalized neural network for nonsmooth nonlinear programming problems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 9, pp. 1741–1754, Sep. 2004.

[30] Q. Liu and J. Wang, "A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming," *IEEE Trans. Neural Netw.*, vol. 19, no. 4, pp. 558–570, Apr. 2008.

[31] A. Rodriguez-Vazquez, R. Dominguez-Castro, A. Rueda, J. L. Huertas, and E. Sanchez-Sinencio, "Nonlinear switched capacitor 'neural' networks for optimization problems," *IEEE Trans. Circuits Syst.*, vol. 37, no. 3, pp. 384–398, Mar. 1990.

[32] Q. Liu and J. Wang, "A one-layer projection neural network for nonsmooth optimization subject to linear equalities and bound constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 5, pp. 812–824, May 2013.

[33] X. Gao and L.-Z. Liao, "A new one-layer neural network for linear and quadratic programming," *IEEE Trans. Neural Netw.*, vol. 21, no. 6, pp. 918–929, Jun. 2010.

[34] Y. Xia, "A new neural network for solving linear and quadratic programming problems," *IEEE Trans. Neural Netw.*, vol. 7, no. 6, pp. 1544–1548, Nov. 1996.

[35] Y. Leung, K.-Z. Chen, Y.-C. Jiao, X.-B. Gao, and K. S. Leung, "A new gradient-based neural network for solving linear and quadratic programming problems," *IEEE Trans. Neural Netw.*, vol. 12, no. 5, pp. 1074–1083, Sep. 2001.

[36] Y. Xia and J. Wang, "A dual neural network for kinematic control of redundant robot manipulators," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 1, pp. 147–154, Feb. 2001.

[37] Y. Xia, H. Leung, and J. Wang, "A projection neural network and its application to constrained optimization problems," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 49, no. 4, pp. 447–458, Apr. 2002.

[38] J. Wang, "Analysis and design of a recurrent neural network for linear programming," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 40, no. 9, pp. 613–618, Sep. 1993.

[39] X.-Y. Wu, Y.-S. Xia, J. Li, and W.-K. Chen, "A high-performance neural network for solving linear and quadratic programming problems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 7, no. 3, pp. 643–651, May 1996.

[40] Y. Xia and J. Wang, "Neural network for solving linear programming problems with bounded variables," *IEEE Trans. Neural Netw.*, vol. 6, no. 2, pp. 515–519, Mar. 1995.

[41] X. Hu and B. Zhang, "An alternative recurrent neural network for solving variational inequalities and related optimization problems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1640–1645, Dec. 2009.

[42] X. He, T. Huang, J. Yu, C. Li, and C. Li, "An inertial projection neural network for solving variational inequalities," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 809–814, Mar. 2017.

[43] X. Hu and J. Wang, "Design of general projection neural networks for solving monotone linear variational inequalities and linear and quadratic optimization problems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 5, pp. 1414–1421, Oct. 2007.

[44] Y. Xia and J. Wang, "A bi-projection neural network for solving constrained quadratic optimization problems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 2, pp. 214–224, Feb. 2015.

[45] Z. Peng and J. Wang, "Output-feedback path-following control of autonomous underwater vehicles based on an extended state observer and projection neural networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 4, pp. 535–544, Apr. 2018.

[46] W. Bian, L. Ma, S. Qin, and X. Xue, "Neural network for nonsmooth pseudoconvex optimization with general convex constraints," *Neural Netw.*, vol. 101, pp. 1–14, May 2018.

[47] K. M. Kim, D. S. Jeong, and C. S. Hwang, "Nanofilamentary resistive switching in binary oxide system; a review on the present status and outlook," *Nanotechnology*, vol. 22, no. 25, 2011, Art. no. 254002.

[48] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, May 2008.

[49] F. Corinto and M. Forti, "Memristor circuits: Flux–charge analysis method," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 11, pp. 1997–2009, Nov. 2016.

[50] F. Corinto and M. Forti, "Memristor circuits: Bifurcations without parameters," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 6, pp. 1540–1551, Jun. 2017.

[51] F. Corinto and M. Forti, "Memristor circuits: Pulse programming via invariant manifolds," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 4, pp. 1327–1339, Apr. 2018.

[52] M. Di Marco, M. Forti, and L. Pancioni, "Stability of memristor neural networks with delays operating in the flux–charge domain," *J. Franklin Inst.*, vol. 355, no. 12, pp. 5135–5162, 2018.

[53] N. Du *et al.*, "Practical guide for validated memristance measurements," *Rev. Sci. Instrum.*, vol. 84, no. 2, 2013, Art. no. 023903.

[54] L. Zhong, P. A. Reed, R. Huang, C. H. de Groot, and L. Jiang, "Resistive switching of Cu/SiC/Au memory devices with a high ON/OFF ratio," *Solid-State Electron.*, vol. 94, pp. 98–102, Apr. 2014.

[55] Y. Li, S. Long, Q. Liu, H. Lv, and M. Liu, "Resistive switching performance improvement via modulating nanoscale conductive filament, involving the application of two-dimensional layered materials," *Small*, vol. 13, no. 35, 2017, Art. no. 1604306.

[56] M. Di Marco, M. Forti, and L. Pancioni, "Convergence and multistability of nonsymmetric cellular neural networks with memristors," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 2970–2983, Oct. 2017.

[57] M. Forti and P. Nistri, "Global convergence of neural networks with discontinuous neuron activations," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 50, no. 11, pp. 1421–1435, Nov. 2003.

[58] A. Ascoli, F. Corinto, and R. Tetzlaff, "Generalized boundary condition memristor model," *Int. J. Circuit Theory Appl.*, vol. 44, no. 1, pp. 60–84, 2016.

[59] J. P. Aubin and A. Cellina, *Differential Inclusions. Set-Valued Maps and Viability Theory*. Berlin, Germany: Springer-Verlag, 1984.

[60] J. P. Aubin and H. Frankowska, *Set-Valued Analysis*. Boston, MA, USA: Birkäuser, 1990.

[61] M. Forti, P. Nistri, and M. Quincampoix, "Convergence of neural networks for programming problems via a nonsmooth Lojasiewicz inequality," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1471–1486, Nov. 2006.

[62] F. H. Clarke, *Optimization and Non-Smooth Analysis*. New York, NY, USA: Wiley, 1983.

[63] M. Di Marco, M. Forti, and A. Tesi, "Bifurcations and oscillatory behavior in a class of competitive cellular neural networks," *Int. J. Bifurcation Chaos*, vol. 10, no. 6, pp. 1267–1293, Jun. 2000.

[64] M. Di Marco, M. Forti, and A. Tesi, "Existence and characterization of limit cycles in nearly symmetric neural networks," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 49, no. 7, pp. 979–992, Jul. 2002.

[65] M. Di Marco, M. Forti, and A. Tesi, "Harmonic balance approach to predict period-doubling bifurcations in nearly symmetric neural networks," *J. Circuits Syst. Comput.*, vol. 12, no. 4, pp. 435–460, Jul. 2003.

[66] M. Di Marco, M. Forti, M. Grazzini, P. Nistri, and L. Pancioni, "Lyapunov method and convergence of the full-range model of CNNs," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 11, pp. 3528–3541, Dec. 2008.

[67] S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, vol. 15. Philadelphia, PA, USA: SIAM, 1994.

Mauro Di Marco was born in Firenze, Italy, in 1970. He received the Laurea degree in electronic engineering from the University of Firenze, Firenze, in 1997, and the Ph.D. degree from the University of Bologna, Bologna, Italy, in 2001.

From 1999 to 2000, he was a Visiting Researcher with LAAS, Toulouse, France. Since 2000, he has been with the University of Siena, Siena, Italy, where he is currently an Associate Professor of circuit theory. He has authored more than 80 technical publications. His current research interests are in analysis and modeling of nonlinear dynamics of complex systems and neural networks, in robust estimation and filtering.

Dr. Di Marco served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I: REGULAR PAPERS from 2007 to 2011.

Mauro Forti received the Laurea degree in electronics engineering from the University of Florence, Florence, Italy, in 1988.

From 1991 to 1998, he was an Assistant Professor of applied mathematics and network theory with the Electronic Engineering Department, University of Florence. In 1998, he joined the Department of Information Engineering and Mathematics, University of Siena, Siena, Italy, where he is currently a Professor of electrical engineering. His main research interests are in the field of nonlinear circuits and systems, with an emphasis on the qualitative analysis and stability of circuits modeling artificial neural networks. His research activity also includes aspects of electromagnetic compatibility.

Prof. Forti served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I: FUNDAMENTAL THEORY AND APPLICATIONS from 2001 to 2003 and the IEEE TRANSACTIONS ON NEURAL NETWORKS from 2001 to 2010. He is currently serving as an Associate Editor for the IEEE TRANSACTIONS ON CYBERNETICS and *Neural Networks*.

Luca Pancioni received the Laurea degree in telecommunication engineering and the Ph.D. degree in information engineering from the University of Siena, Siena, Italy, in 2001 and 2004, respectively.

He is currently an Assistant Professor of electrical engineering with the Department of Information Engineering and Mathematics, Siena University. His main research interests are in the field of analysis of nonlinear circuits modeling neural networks, mainly focused on stability and complex dynamics. Recent activities concern the study and the modeling of neural circuits including memristors. His research activity also includes modeling of source-coupled logic and electronic design of integrated analog and mixed signals.

Giacomo Innocenti received the master's degree in information engineering and the Ph.D. degree in nonlinear dynamics and complex systems from the University of Florence, Firenze, Italy, in 2004 and 2008, respectively.

He was a Postdoctoral Research Fellow, first with the University of Florence and then the University of Siena, Siena, Italy, from 2008 to 2010. Since 2012, he has been an Assistant Professor of automation with the University of Florence, where he is with the Department of Information Engineering. His main research is in the field of nonlinear dynamics with a particular interest in networks of interacting agents and neuron models.

Dr. Innocenti serves as an Associate Editor for scientific journals in the field of the analysis of nonlinear systems, and he was in the committees of workshops and scientific congresses on the same subject.

Alberto Tesi received the Laurea degree in electronics engineering from the University of Florence, Florence, Italy, in 1984, and the Ph.D. degree in systems engineering from the University of Bologna, Bologna, Italy, in 1989.

In 1990, he joined the Department of Systems and Computer Science, University of Florence, as a Research Assistant, where he is currently a Professor of control systems with the Department of Information Engineering. He has coauthored about 180 scientific publications. His research interests are in analysis of nonlinear dynamics of complex systems and robust control of linear systems and optimization.

Prof. Tesi was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS from 1994 to 1995, the IEEE TRANSACTIONS ON AUTOMATIC CONTROL from 1995 to 1998, and *Systems and Control Letters* from 1995 to 2010. He was a member of the conference editorial board of the Conference on Decision and Control from 1994 to 1999 and the American Control Conference from 1995 to 2000, and a member of the program committee of several international conferences.