A Non-feasible Gradient Projection Recurrent Neural Network for Equality Constrained Optimization Problems

Maria P. Barbarosou, and Nicholas G. Maratos, *Member, IEEE*

***Abstract*—A recurrent neural network for both convex and nonconvex equality constrained optimization problems is proposed, which makes use of a cost gradient projection onto the tangent space of the constraints. The proposed neural network constructs a generically non-feasible trajectory, satisfying the constraints only as $t \to \infty$. Local convergence results are given which do not assume convexity of the optimization problem to be solved. Global convergence results are established for convex optimization problems. An exponential convergence rate is shown to hold both for the convex and the nonconvex case. Numerical results indicate that the proposed method is efficient and accurate.**

***Index Terms*—Recurrent neural networks, constrained optimization, convergence, convex and non-convex problems.**

I. Introduction

A large number of recurrent neural networks have been proposed in the literature for solving constrained optimization problems. Historically, the first approach towards designing such networks employed penalty functions which were used to convert the original constrained optimization problem to an (approximately or exactly) equivalent unconstrained optimization problem; the latter was usually solved by gradient descent, thus yielding the neural network. Although this is not immediately apparent, both the Tank and Hopfield network [1] and the Chua and Lin nonlinear programming circuit [2], [3] can be demonstrated to be, [4], gradient dynamical systems based on the L_2 penalty function. Other gradient-based architectures related to penalty functions include the switched capacitor neural networks proposed in [5], the neural network proposed in [6] which is based on the L_1 exact penalty function, and a multitude of network architectures given in [7] for solving constrained optimization problems. Also, in [8], various combinations of the L_1, L_2 and L_∞ penalty functions are used to obtain a class of neural networks which are rigorously analyzed in [9]. More recently, the nonlinear programming circuit [3] has been generalized for solving nonsmooth optimization problems, [10], and applied to quadratic and linear programming problems, [11], with strong convergence results. Apart from penalty functions, the logarithmic barrier function has also been used, together with Newton-type descent, to produce an interior point recurrent neural network, [12].

A second class of recurrent neural networks, proposed in [13], [14], makes direct use of the Lagrangian function and of Lagrangian optimality conditions in order to solve equality constrained optimization problems. These Lagrangian neural networks are capable of solving general non-convex problems; local convergence results are given for such problems in [13], [14]. For convex problems, global convergence is obtained in [15] for the neural network of [13].

More recently, a general methodology has been developed, [18] – [31], for solving constrained optimization problems over convex feasible sets defined by simple bounds on the variables and, in some cases, (linear or nonlinear) convex inequality constraints and/or linear equality constraints. According to this approach, optimality conditions for such optimization problems are written in the form of a variational inequality which is then equivalently transformed into a projection equation. Neural networks are designed to solve the projection equation and thus provide a solution to the associated optimization problem. The work described in [16] (and in the related paper [17]) for solving positive definite quadratic programs with bound constraints makes use of piecewise linear neuron outputs in order to satisfy exactly Kuhn–Tucker conditions, and thus it may be considered as an early form of this approach, though it does not use explicitly a variational inequality or projection equation. The same can be said about the convex quadratic programming neural network proposed in [18], for which global convergence is proven. In [19], a recurrent neural network for solving piecewise linear projection equations with asymmetric connection matrices is proposed and applied to semidefinite quadratic programming problems; global convergence is proven and an exponential convergence rate is obtained for the positive definite case.

Manuscript received August 9, 2005; revised April 4, 2006, November 30, 2006, June 3, 2007 and November 2, 2007; accepted December 27, 2007. This work was supported financially by the project “HRAKLEITOS”. The project “HRAKLEITOS” is co-funded by the European Social Fund (75%) and Greek National Resources (25%).

M. P. Barbarosou was with the School of Electrical and Computer Engineering, National Technical University of Athens, Athens 15773, Greece. She is now with the Department of Electronics, Technological Educational Institute of Piraeus, 250 Thivon St. and P. Ralli St., Aigaleo, Athens 12244, Greece (e-mail: mbarbar@ central.ntua.gr).

N. G. Maratos is with the School of Electrical and Computer Engineering, National Technical University of Athens, 9 Iroon Polytechniou St., Athens 15773, Greece (phone: +30 210 7723693; fax: +30 210 772 2281; e-mail: maratos@ece.ntua.gr).

Globally exponentially convergent neural networks for positive definite quadratic programming are proposed in [20] and [21] based on piecewise linear projection equations; the proposed dynamical equations make use of matrix inversion. Also based on a piecewise linear projection equation, the neural network of [22] solves a minimax problem with a convex–concave quadratic cost function and bound constraints.

More general problems over convex feasible sets can be tackled within this approach, by employing nonlinear projection equations. Thus, in [23], convex and non-convex optimization problems with convex constraints are considered, a projection equation and/or penalty functions are used to define a number of existing neural networks and LaSalle's theorem is employed to prove global convergence for convex problems and local convergence in the absence of convexity. Also a new neural network is proposed which is proven to be locally convergent, for convex problems. A neural network is proposed in [24] for minimizing a not necessarily convex function $f(x)$ with bound constraints, based on a nonlinear projection equation. For feasible starting points, global convergence is proven if $f(x)$ is convex. In [25], a single-layer neural network for solving a nonlinear projection equation is proposed and applied to minimization problems with a not necessarily convex cost function and simple bound constraints or a single hypersphere constraint on the variables. Neural networks for minimizing convex functions subject to convex inequality and linear equality constraints are proposed in [26] – [28], based on nonlinear projection equations. Global convergence to the set of optimal solutions is proven in [26], [27] and global convergence to the unique optimal solution is proven in [28] under a strict convexity assumption. In [29] – [33] neural networks for solving variational inequalities are analysed; when applied to convex optimization problems, these networks are shown to be globally convergent.

Other neural networks which are not based on a projection equation or on a variational inequality include [34] and [35] for solving linear and quadratic programming problems and [36] for solving convex nonlinear programming problems. In [34], global convergence is proven for positive definite cost functions, and [36] is a globally convergent feedback network which sequentially updates a lower bound for the optimal solution.

Apart from recurrent neural networks, several other methods have been proposed, mainly in the optimization literature, which make use of dynamical systems for solving optimization (see e.g. [37], [38]) and other related problems, [39]. Following [40], we consider such methods as constituting the "dynamical systems" approach. Although both approaches make use of differential equations to construct trajectories convergent to the desired solutions, they do so from quite different perspectives. The "dynamical systems" approach, being more mathematically oriented, relies on numerical integration of the differential equations on a digital computer; indeed, within this approach, well known optimization algorithms are often obtained as discrete integration schemes of the underlying differential equations. The neural network approach, on the other hand, gives emphasis on hardware implementation of the underlying dynamical system, either as an electrical circuit or as an interconnection of hardware integrators and function blocks; thus hardware integration of the underlying differential equations is obtained in real time. Surprisingly, cross-referencing between the two approaches seems rather scarce; however see [40] for a review of both. Within the "dynamical systems" framework, the main contribution of this paper is a practical one: we propose hardware implementations of the dynamical systems involved. In Section II both an electrical circuit and a function block diagram implementing the proposed neural network are given. A detailed comparison of our results to the dynamical systems of [37], [38] is also given in Section II.

Within the recurrent neural network framework, a novel neural network is proposed in this paper for the solution of the following, not necessarily convex equality constrained optimization problem (P),

$$\text{(P)} \qquad \min_{x\in R^n}\left\{ f(x):\ h_i(x)=0,\quad i=1,\ldots,m \right\} \qquad (1)$$

where the functions $f:R^n\to R$, and $h_i:R^n\to R$, $i=1,\ldots,m$ are assumed to be continuously differentiable. Let F denote the set of feasible points for problem (P), i.e.

$$\mathrm{F}=\left\{ x\in R^n:\quad h_i(x)=0,\quad i=1,\ldots,m \right\} \qquad (2)$$

The proposed neural network, first presented in [41], is capable of minimizing nonconvex cost functions over nonconvex feasible sets. It does not make use of a penalty function or of a projection equation; instead, it solves problem (P) directly, based on the well known gradient projection method of nonlinear optimization [42], in the sense that it makes use of the orthogonal projection of the cost gradient onto the nullspace of the constraint gradients. The proposed neural network does not require feasibility of the initial point x_o and, if $x_o\notin\mathrm{F}$ it defines a non-feasible solution $x(t)$ which, in normal operation, approaches the feasible set F only as $t\to\infty$. It may be considered as a continuous time version of a first order recursive quadratic programming method, [43], for nonlinear optimization. In case a feasible starting point $x_o\in\mathrm{F}$ is given, the proposed neural network defines a feasible solution $x(t)\in\mathrm{F},\ \forall t\geq 0$ which is a descent trajectory for the cost function f. In this case the network may be considered as a continuous time version of Rosen's gradient projection method, [42], for nonlinear optimization. From another viewpoint, since our network aims at satisfying Lagrangian conditions of optimality for problem (P), it may be considered as a Lagrangian network; differences from existing Lagrangian networks are detailed in the next section.

The following convergence results are established for the proposed neural network. First, local convergence results are given for the general case: without assuming convexity of either the cost function or of the constraints we prove that strict local minimizers of problem (P) are exponentially stable equilibrium points of the proposed neural network. Next, assuming convexity of the problem to be solved, we prove that solutions emanating from arbitrary initial points are bounded and converge globally to the set of points satisfying first order Lagrangian conditions for problem (P). Finally, an exponential convergence rate is established for both convex and non-convex problems.

The paper is organized as follows. In Section II the proposed neural network is derived and in Section III convergence results are given. Section IV contains numerical results and the paper is concluded in Section V.

II. Derivation of the Neural Network

In this section the proposed non-feasible gradient projection neural network (NFGPNN) is derived in implicit and explicit forms, block diagram and analog circuit realizations are given, its relationship to Lagrangian neural networks and to the dynamical systems of [37], [38] is explored, and its ability to tackle inequality constrained optimization problems is considered.

We define the set D of desirable points for problem (P) to be the set of all points in R^n which satisfy first order necessary conditions of optimality (Lagrangian conditions) for (P), i.e. $\mathrm{D} = \mathrm{D}' \cap \mathrm{F}$, where

$$\mathrm{D}' = \left\{ x \in R^n : \nabla f(x) + \nabla h(x)\lambda = 0, \text{ for some } \lambda \in R^m \right\}. \quad (3)$$

Here $\nabla h(x) = [\nabla h_1(x), \nabla h_2(x), \ldots, \nabla h_m(x)]^{\mathrm{T}}$, $\lambda \in R^m$ is a vector of Lagrange multipliers, and $h(x) = [h_1(x), h_2(x), \ldots, h_m(x)]^{\mathrm{T}}$. Apart from local minima, the set D also contains local maxima and saddle points of problem (P).

In order to obtain our non-feasible gradient projection neural network (NFGPNN) for problem (P), we set out to define a dynamical system with solutions $x(t)$ which start at any initial point $x_o \in R^n$ and, if possible, converge to a local minimum of (P). Hence, the following two requirements on the solutions $x(t)$ of NFGPNN are made: in normal operation (i.e. when $x(t)$ extends to infinite time), (i) $x(t)$ should converge to the set D', and (ii) $x(t)$ should converge to the feasible set F.

In a first instance, the desired solution $x(t)$ is determined as part of the solution $(x(t), \lambda(t))$ of the following system of implicit ordinary differential equations (implicit ODE's):

$$\dot{x}(t) = -\mu\left[\nabla f(x(t)) + \nabla h(x(t))\lambda(t)\right], \qquad \forall t \in [0, \omega) \quad (4)$$

$$\nabla h(x(t))^{\mathrm{T}} \dot{x}(t) = -\rho\, h(x(t)), \qquad \forall t \in [0, \omega) \quad (5)$$

where μ and ρ are positive constants, $x_o \in R^n$ is an initial condition for $x(t)$, and $\lambda(t) \in R^m$ are additional variables (multipliers). Later in this section, the multipliers $\lambda(t)$ are eliminated from (4), (5) and a set of ordinary differential equations (eq. (13)) is obtained for $x(t)$.

Let $(x(t), \lambda(t))$ be a solution of (4), (5) with initial condition (x_o, λ_o) and let $[0, \omega)$ be its maximum interval of existence. Equation (4) aims at satisfying the first requirement on $x(t)$. Indeed, let (x^*, λ^*) be an equilibrium point of (4), (5). It then follows from (3) and (4) that $x^* \in \mathrm{D}'$, therefore in normal operation (i.e. when $\omega = \infty$, and $(x(t), \lambda(t)) \to (x^*, \lambda^*)$ as $t \to \infty$), the solution $x(t)$ will approach the set D'.

In order to satisfy the second requirement on $x(t)$, i.e. to ensure that $x(t)$ will eventually approach the feasible set F, we use the continuous Newton-Raphson method for solving the constraint equations $h(x) = 0$. This method, introduced by Branin [44], when applied to the equations $h(x) = 0$ yields immediately the differential equation (5). The relationship of (5) to the classical (iterative) Newton-Raphson method can be appreciated by applying an Euler numerical integration scheme to equations (5); then $\nabla h(x_k)^{\mathrm{T}}(x_{k+1} - x_k) = -\rho \Delta t\, h(x_k)$ is obtained. If $\rho \Delta t = 1$, this is indeed the Newton-Raphson iteration for solving the equations $h(x) = 0$.

Put together, equations (4) and (5) ensure that, in normal operation, any solution $x(t)$ will eventually approach the set D of desirable points. These equations define, in implicit form, the dynamics of the proposed neural network NFGPNN. The fact that the variables $\lambda(t)$ do not appear explicitly in equation (5) should not confuse the reader: equations (4) and (5) may be written as

$$\begin{bmatrix} \mathrm{I} & 0 \\ \nabla h(x(t))^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \dot{x}(t) \\ \dot{\lambda}(t) \end{bmatrix} = \begin{bmatrix} -\mu\left(\nabla f(x(t)) + \nabla h(x(t))\lambda(t)\right) \\ -\rho h(x(t)) \end{bmatrix}. \quad (6)$$

This is indeed a system of *implicit* ordinary differential equations (see e.g. [45]) with state vector $z(t) = [x(t)^{\mathrm{T}}, \lambda(t)^{\mathrm{T}}]^{\mathrm{T}}$.

Next, two equivalent versions of the dynamical system (4), (5) are obtained. Since $\nabla h(x(t))^{\mathrm{T}} \dot{x}(t) = \dfrac{dh(x(t))}{dt}$, integration of equation (5) gives

$$h(x(t)) = \exp(-\rho t) h(x_o), \qquad \forall t \in [0, \omega). \quad (7)$$

Also by replacing $\dot{x}(t)$ form (4) into (5) we get, $\forall t \in [0, \omega)$,

$$\nabla h(x(t))^{\mathrm{T}}\left[\nabla f(x(t)) + \nabla h(x(t))\lambda(t)\right] - \frac{\rho}{\mu} h(x(t)) = 0. \quad (8)$$

The dynamics of the proposed neural network are defined, in implicit form, by equations (4) and (5) or, equivalently, by equations (4) and (7) or, equivalently, by equations (4) and (8). Since (7) and (8) are algebraic equations, definition of NFGPNN by equations (4) and (7) or (4) and (8) leads to its description as a system of differential-algebraic equations (DAE's). If, on the other hand, equations (4) and (5) are used to define NFGPNN, then the description (6), i.e. a system of implicit ordinary differential equations (implicit ODE's), is obtained. Obviously, these definitions are equivalent. Such dynamical systems arise naturally as descriptions of analog electronic circuits; when writing circuit equations, based on Kirchoff's laws, one usually obtains a system of implicit ODE's or a system of DAE's. Recurrent neural networks described by implicit dynamical systems have been proposed in [46] and [47].

Two ways of realizing the proposed neural network are given next, based on equations (4) and (5). The first realization of NFGPNN, shown in Fig. 1, is a block diagram of the type introduced in [46] and [47]. The second realization, depicted in Fig. 2, is an ideal nonlinear analog circuit which makes use of ideal ap amps, nonlinear voltage controled current sources and nonlinear voltage controlled conductances. The values of conductances $G_i(x)$ and $\hat{G}_j(x)$ are taken to be:

$$G_i(x) = -\sum_{j=1}^{m} \frac{\partial h_j(x)}{\partial x_i}, \qquad i = 1, \ldots, n,$$

and

$$\hat{G}_j(x) = -\sum_{i=1}^{n} \frac{\partial h_j(x)}{\partial x_i}, \qquad j = 1, \ldots, m.$$

Then, writing nodal equations for the circuit of Fig. 2 we obtain,

$$\begin{bmatrix} -\mathrm{I} & -\nabla h(x(t)) \\ -\nabla h(x(t))^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\mu}\dot{x}(t) \\ \lambda(t) \end{bmatrix} = \begin{bmatrix} \nabla f(x(t)) \\ \frac{\rho}{\mu} h(x(t)) \end{bmatrix}.$$

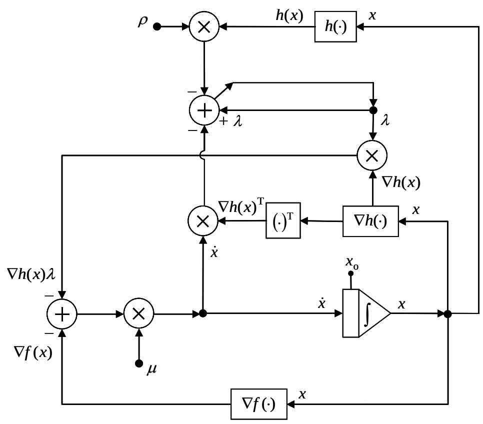

Fig. 1. Block diagram realization of NFGPNN.

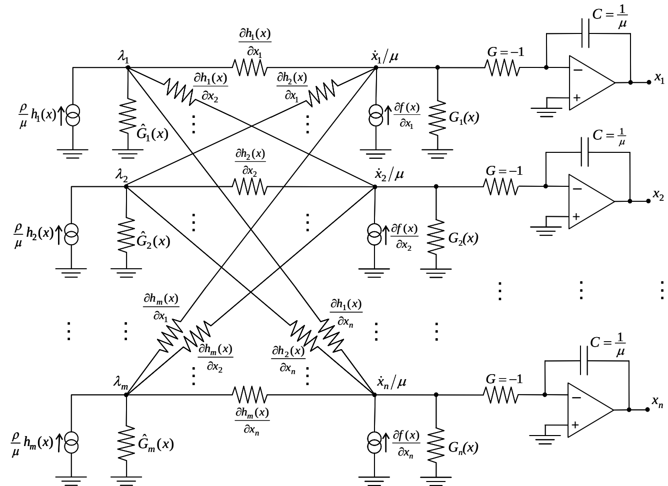

Fig. 2: Ideal nonlinear circuit realization of NFGPNN.

These are indeed equations (4) and (5) (or, equivalently, equations (6)) of NFGPNN. It should be stressed that the ideal circuit of Fig. 2 is given here simply in order to illustrate that, in principle, a circuit realization of the proposed neural network is feasible. Practical implementation of the circuit would require investigations beyond the scope of this paper. Note however that, for the special case of linearly constrained optimization problems, ordinary linear conductors are only required in Fig. 2.

Next, we compare NFGPNN to existing Lagrangian neural networks and to other dynamical systems proposed in the literature for solving problem (P). The Lagrangian networks of [13] and [14] are defined by the following differential equations

$$\dot{x}(t) = -\left[\nabla f\left(x(t)\right) + \nabla h\left(x(t)\right)\lambda(t)\right] \tag{9.1}$$

$$\dot{\lambda}(t) = h(x(t)) \tag{9.2}$$

and

$$\frac{\partial^2 L(x(t),\lambda(t))}{\partial x^2}\dot{x}(t) + \nabla h(x(t))\dot{\lambda}(t) = = -\left[\nabla f\left(x(t)\right) + \nabla h\left(x(t)\right)\lambda(t)\right] \tag{10.1}$$

$$\nabla h\left(x(t)\right)^{\mathrm{T}}\dot{x}(t) = -h(x(t)) \tag{10.2}$$

respectively. Partial similarities may be observed among equations (4), (5), (9) and (10). More specifically, for $\mu = \rho = 1$, equation (4) reduces to equation (9.1) of [13], and equation (5) reduces to equation (10.2) of [14]. However, in contrast to (9) and (10), the variables $\lambda(t)$ of the proposed dynamical system (4), (5) are non-dynamic variables. Indeed, (4), (5) is a dynamical system described by implicit ordinary differential equations (ODE's), therefore it is substantially different to both (9) and (10) which are dynamical systems described by explicit ODE's. In essence, equations (9.1), (9.2) are gradient flows aiming to approach the sets D' and F respectively, and equations (10.1), (10.2) are Newton flows with the same respective aims. On the other hand, NFGPNN makes use of a gradient flow (equation (4)) in order to approach the set D' and of a Newton flow (equation (5)) in order to satisfy the constraints. Thus the use of second derivatives is avoided and fast convergence to the feasible set is obtained. Therefore, if viewed as a Lagrangian system, the proposed neural network differs substantially compared to existing Lagrangian networks.

NFGPNN is more similar to the dynamical systems of [37], [38]. The dynamical system analysed in [37] is the same as equations (4), (5) with $\mu = \rho = 1$. The dynamical system of [38] also makes use of equation (5) with $\rho = 1$, however it employs curvature terms of $f(.)$ and $h(.)$ in equation (4), by multiplying its left hand side by a positive definite matrix. Local convergence results are proven in [37] and [38] for the corresponding dynamical systems. Global convergence results are obtained in [37] by assuming boundedness of the solutions. We provide better local convergence results for NFGPNN, by proving exponential stability of strict local minima of (P); in comparison only asymptotic stability is proven in [37] and [38]. In addition, we obtain strong global convergence results for the case of convex problems in Section III-B, where both boundedness and global convergence of the solutions is proven; no such results are given in [37] or [38]. Finally, NFGPNN is hardware impementable both as a circuit and as an interconnection of function blocks, i.e. it is a neural network and not a dynamical system integrated on a digital computer.

In order to facilitate analysis of the proposed neural network, we obtain next an explicit expression for the differential-algebraic dynamical system which defines NFGPNN. Let $x_{\mathrm{o}} \in R^n$ be an arbitrary initial point and let $\lambda_{\mathrm{o}} \in R^m$ be such that $(x_{\mathrm{o}}, \lambda_{\mathrm{o}}) \in R^{n+m}$ satisfies equation (8). We define the set

$$\begin{aligned}\Upsilon &= \bigcup_{\alpha\in[0,1]}\left\{x \in R^n : h(x) = ah(x_{\mathrm{o}})\right\} \\ &= \left\{x \in R^n : h(x) = ah(x_{\mathrm{o}}), \quad \text{for any } a \in [0,1]\right\}\end{aligned} \tag{11}$$

and we make the following assumption.

Assumption 1: (a) The functions $f: R^n \to R$, $h_i: R^n \to R$, $i = 1,\ldots,m$ are continuously differentiable in R^n. (b) For any $x \in \Upsilon$ the matrix $\nabla h(x) = [\nabla h_1(x), \nabla h_2(x), \ldots, \nabla h_m(x)]^{\mathrm{T}}$ has full rank.

Remark 1: Part (b) of Assumption 1 may appear to be too strong. In fact this is a standard regularity assumption, which corresponds to causality of the system, made in order to obtain dynamical systems described by explicit ODE's from DAE descriptions (see e.g. [45]). Moreover, in the literature of optimization algorithms, this is a standard assumption associated with gradient projection methods (see e.g. [42, p. 331], [48, p. 190]).

If $(x(t),\lambda(t))$ is a solution of the dynamical system (4), (8) with initial point $(x(0),\lambda(0))=(x_o,\lambda_o)$ and if $[0,\omega)$ is its maximal interval of existence, then equations (5) and (7) will hold for every $t\in[0,\omega)$ (see also proof of Theorem 1). Hence from (7), $x(t)\in Y$ will hold $\forall t\in[0,\omega)$. It then follows from Assumption 1 that equation (8) can be solved for $\lambda(t)$ to give,

$$\lambda(t)=-\left[\nabla h(x(t))^{\mathrm{T}}\nabla h(x(t))\right]^{-1}\left(\nabla h(x(t))^{\mathrm{T}}\nabla f(x(t))-\frac{\rho}{\mu}h(x(t))\right) \tag{12}$$

Replacement of $\lambda(t)$ from (12) into (4) yields the dynamical system which determines NFGPNN in explicit form:

$$\begin{aligned}\dot{x}(t)&=-\mu\, P(x(t))\nabla f(x(t))\\ &\quad-\rho\nabla h(x(t))\left[\nabla h(x(t))^{\mathrm{T}}\nabla h(x(t))\right]^{-1}h(x(t))\end{aligned} \tag{13}$$

with initial condition

$$x(0)=x_o\in R^n \tag{14}$$

In equation (13), $P(x(t))$ is the well known projection matrix, [42], which orthogonally projects an R^n vector onto the nullspace of $\nabla h(x(t))$, i.e.

$$P(x)=I-\nabla h(x)\left[\nabla h(x)^{\mathrm{T}}\nabla h(x)\right]^{-1}\nabla h(x)^{\mathrm{T}} \tag{15}$$

Thus the right hand side of (13) consists of two components: the first is an orthogonal projection of the cost gradient onto the tangent plane of the constraints, and the second is a Newton step aiming to establish feasibility of the constraints. It is also noted that if the initial point x_o is feasible, i.e. if $h(x_o)=0$, then it follows from (7) that $h(x(t))=0,\ \forall t\in[0,\omega)$ hence (13) and (14) reduce to $\dot{x}(t)=-\mu P(x(t))\nabla f(x(t))$, $x(0)=x_o\in \mathrm{F}$. It can be seen that these equations define a continuous time version of Rosen's gradient projection algorithm, [42]. If, on the other hand, the initial point x_o is not feasible, then (13) can be considered as a continuous time version of the algorithm proposed in [43].

We close this section by considering inequality constrained optimization problems (PI) of the following general form,

$$\text{(PI)}\quad \min_{x\in R^n}\left\{f(x):\ h_i(x)=0,\ \ i=1,\ldots,m,\quad g_j(x)\le 0,\quad j=1,\ldots,l\right\}$$

where f, h_i, $i=1,\ldots,m$ and g_j, $j=1,\ldots,l$ are continuously differentiable (not necessarily convex) functions. A simple way to extend applicability of the proposed neural network in order to solve problem (PI) is to transform the latter into the following equality constrained optimization problem (PE),

$$\text{(PE)}\qquad \min_{x\in R^n,\,y\in R^l}\left\{\begin{aligned}&f(x):\ \ h_i(x)=0,\ \ i=1,\ldots,m,\\ &\qquad\quad g_j(x)+(y_j)^2=0,\quad j=1,\ldots,l\end{aligned}\right\}$$

where $y=[y_1,y_2,\ldots,y_l]^{\mathrm{T}}$ is a vector of additional variables. It is well known, [49, p. 286], that problems (PI) and (PE) are equivalent in the following sense: x^* is a local minimum of (PI) if and only if (x^*,y^*) is a local minimum of (PE), where $y_j{}^*=(-g_j(x^*))^{1/2}$, $j=1,\ldots,l$. Thus local minima of (PI) may be obtained by straightforward application of NFGPNN to (PE) (see also Example 1 in Section V).

III. Convergence Results

This section contains convergence results for the proposed non-feasible gradient projection neural network (NFGPNN) defined, in explicit form, by equations (13). First, local convergence results are given in Section III-A where exponential stability of strict local minimizers of problem (P) is proven. In Section III-B global convergence results are given for the case of convex optimization problems. Finally, in Section III-C, the rate of convergence of NFGPNN is examined.

A. Local Convergence Results

In this subsection we do not assume convexity of problem (P). Our main result shows that strict local minimizers of problem (P) are exponentially stable equilibrium points of NFGPNN.

Theorem 1: Let the functions $f:R^n\to R$, $h_i:R^n\to R$, $i=1,\ldots,m$ be twice continuously differentiable in R^n. Let $x^*\in R^n$ be a local minimizer of problem (P) which satisfies the sufficient conditions of optimality for (P) and assume that x^* is a regular point of the constraints (i.e. assume that $\nabla h_1(x),\nabla h_2(x),\ldots,\nabla h_m(x)$ are linearly independent). Then x^* is an exponentially stable equilibrium point of NFGPNN described by equations (13) with initial condition (14).

Proof: The proof is given in the Appendix. □

B. Global Convergence Results for the Convex Case

We turn now to the convex version of problem (P), i.e. we consider the following problem, denoted in the sequel as problem (CP),

$$\text{(CP)}\qquad \min_{x\in R^n}\left\{f(x):A^{\mathrm{T}}x-b=0\right\} \tag{16}$$

where $f:R^n\to R$ is assumed to be a convex continuously differentiable function, A an $n\times m$ full rank matrix with $m<n$ and b an m vector. For such problems it is well known, [42] that the set of desirable points D, the set of local minimizers LM and the set of global minimizers G are identical, i.e. D = LM = G. The following assumption is made for the convex problem (CP).

Assumption 2: (a) The function $f:R^n\to R$ is continuously differentiable and convex in R^n. (b) The $n\times m$ matrix A (with $m<n$) has full rank. (c) The set D of minimizers of problem (CP) is nonempty.

Remark 2: This is a mild assumption. Part (b) is simply a constraint qualification which excludes redundant constraints from being present in the description of the feasible set F. Furthermore, for a large part of the neural network literature (see e.g. [14], [27], [31], [34], [36]), assumption 2(b) is a standard assumption made when solving problems with linear equality constraints.

When applied to problem (CP), the proposed dynamical system (13) becomes:

$$\dot{x}(t)=-\mu\, P\nabla f(x(t))-\rho\, A(A^{\mathrm{T}}A)^{-1}(A^{\mathrm{T}}x(t)-b) \tag{17}$$

where the projection matrix $P=I-A(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}$ is now independent of t.

The main objective of this section is to show that, under mild assumptions, every solution of (17) with initial conditions (14) is bounded and converges to a global minimum of (CP).

Theorem 2: Let Assumption 2 hold. Then every solution of (17) with initial conditions (14) is bounded, extends to infinite time and has limit points, each of which is a global minimizer of problem (CP).

Proof : The proof is given in the Appendix. □

If f is assumed to be strictly convex and if D is nonempty, then (CP) has a unique desirable point which is also a strict local and global minimizer. Under these circumstances, the following, stronger result is obtained as a direct consequence of Theorem 2.

Corollary 2.1: If f is continuously differentiable and strictly convex, if $\text{rank}\{A\} = m$ and if D is nonempty, then every solution $x(t)$ of (22) with initial conditions (14) is bounded, extends to infinite time and converges to the unique global minimizer x^* of problem (CP), i.e. $\lim_{t\to\infty}\{x(t)\} = x^*$.

C. Rate of Convergence

In this subsection we show that, under mild assumptions, the proposed neural network exhibits an exponential convergence rate when applied to both the general problem (P) and to the convex problem (CP).

In order to examine the rate of convergence of NFGPNN for the general (not necessarily convex) problem (P), it is assumed that NFGPNN generates a solution $x(t)$ which extends to infinite time and which converges to a strict local minimizer x^* of problem (P). In practice, these assumptions are almost always satisfied when NFGPNN is applied to specific examples of problem (P); in this sense they express the usual operating conditions of the neural network. It is noted that similar assumptions are made when examining the convergence rate of discrete optimization algorithms (see [42], [48] – [49]). Under these circumstances, Theorem 1 provides a local exponential convergence rate. In the next Theorem we extend this result to the entire solution $x(t)$.

Theorem 3: Let Assumption 1 hold and let the functions $f: R^n \to R,\ h_i: R^n \to R,\quad i = 1,\ldots,m$ be twice continuously differentiable in R^n. Let $x^* \in R^n$ be a local minimizer of problem (P) which satisfies the sufficient conditions of optimality for (P). Let $x(t)$ be a solution of (13) with initial conditions (14) and assume that $x(t)$ extends to infinite time and converges to x^*. Then the rate of convergence of $x(t)$ is exponential, i.e. there exist real numbers $\psi > 0$ and $\xi > 0$ such that:

$$\|x(t) - x^*\| \le \psi \exp(-\xi t), \qquad \forall t \ge 0 . \qquad (18)$$

Proof: Clearly the assumptions of Theorem 1 are satisfied, therefore there exist $T > 0$, $\psi_1 > 0$ and $\xi > 0$ such that $\|x(t) - x^*\| \le \psi_1 \exp(-\xi t)$, $\forall t \ge T$. For the finite time interval $[0, T]$ we have: $\max_{t\in R}\{\|x(t) - x^*\| \exp(\xi t) : t \in [0, T]\} = \psi_2 > 0$ where ψ_2 exists and is finite from Weierstrass' theorem. Hence $\|x(t) - x^*\| \le \psi_2 \exp(-\xi t)$ holds $\forall t \in [0, T]$. Now (18) follows immediately by taking $\psi = \max\{\psi_1, \psi_2\}$. □

Similar results are obtained when NFGPNN is applied to the convex problem (CP). However, in this case some of the assumptions on the solution $x(t)$ are redundant. Indeed, if Assumption 2 holds, it follows from Theorem 2 that $x(t)$ will extend to infinite time. Furthermore, if $f(x)$ is strictly convex then any minimizer x^* of (CP) is isolated, therefore the set D contains a single point, $D = \{x^*\}$, hence Theorem 2 yields that $x(t)$ will converge to x^*. Thus we have proven the following:

Corollary 3.1: Let Assumption 2 hold, let the function $f: R^n \to R$ be twice continuously differentiable in R^n, and let $x(t)$ be a solution of (17), (14). Assume either that $f(x)$ is strictly convex, or that $x(t)$ converges to a minimizer x^* of (CP). Then the rate of convergence of $x(t)$ is exponential, i.e. there exist real numbers $\psi > 0$ and $\xi > 0$ such that $\|x(t) - x^*\| \le \psi \exp(-\xi t)$, $\forall t \ge 0$. □

Numerical results given in the next section confirm the exponential convergence rate of NFGPNN on both convex and non-convex problems.

IV. Simulation Results

Performance of the proposed neural network is evaluated by using MATLAB to simulate its response for several test problems. The simulation is based on the block diagram of Fig. 1, i.e. on equations (4) and (5).

Example 1: This was originally, [24], a non-convex two dimensional bound-constrained optimization problem, which we converted to an equality constrained problem by adding the variables x_3 and x_4. Thus a problem with $n = 4$ variables and $m = 2$ constraints was obtained.

$$\min_{x\in R^4}\left\{ \left(x_1 \cos(x_2) - x_2 \sin(x_1)\right)^2 \ :\ \begin{array}{l} x_1^{\,2} + x_3^{\,2} - 1 = 0, \\ x_2^{\,2} + x_4^{\,2} - 1 = 0 \end{array} \right\} \qquad (19)$$

This problem has an infinite number of global minimizers including the set $X^* = \{x \in R^4 : x_1 = 0, x_3^{\,2} = 1, x_2^{\,2} + x_4^{\,2} - 1 = 0\}$.

In Fig.3a the solution $x(t)$ obtained from the non-feasible initial point $x_o = [-2\ \ -2\ \ \ 1\ \ \ 1]^T$ is shown. The final point obtained at the end of the simulation time was $[0\ \ -0.99666138\ \ 1.00008951\ \ \ 0.08273509]^T$ which is close to the global minimizer $x^* = [0\ \ \ -0.99657218\ \ \ 1\ \ \ 0.08271768]^T$. In this example, the solution $x(t)$ converges to a particular global minimizer $x^* \in X^*$, although the set X^* is a continuum of global minimizers. Figure 3b shows cost function convergence to zero; since the initial point x_o is not feasible, $f(x(t))$ is non-monotonic. In Fig. 3c, $\log_{10}(\text{dist}(x(t), X^*))$ is plotted as a function of time, where $\text{dist}(x(t), X^*) = \min_{y\in X^*}\|x(t) - y\| =$

$$= \left[x_1(t)^2 + \left(\sqrt{x_2(t)^2 + x_4(t)^2} - 1\right)^2 + \left(|x_3(t)| - 1\right)^2 \right]^{1/2}$$

is the distance of $x(t)$ from the set X^*. An exponential decrease is clearly observable, although one of the assumptions of Theorem 3 is not satisfied (x^* does not satisfy sufficient conditions of optimality). The values $\mu = \rho = 10$ were used for compatibility with [24].

Trajectories obtained by NFGPNN from 1000 random initial points with elements $x_{oi} \in [-1, 1]$, $i = 1,\ldots,4$ are plotted (in the (x_1, x_2) plane) in Fig. 3d. As in [24], convergence is observed to various minimizers which belong to the continuum of global minimizers of the problem. An additional random experiment was conducted with 10000 random initial points in

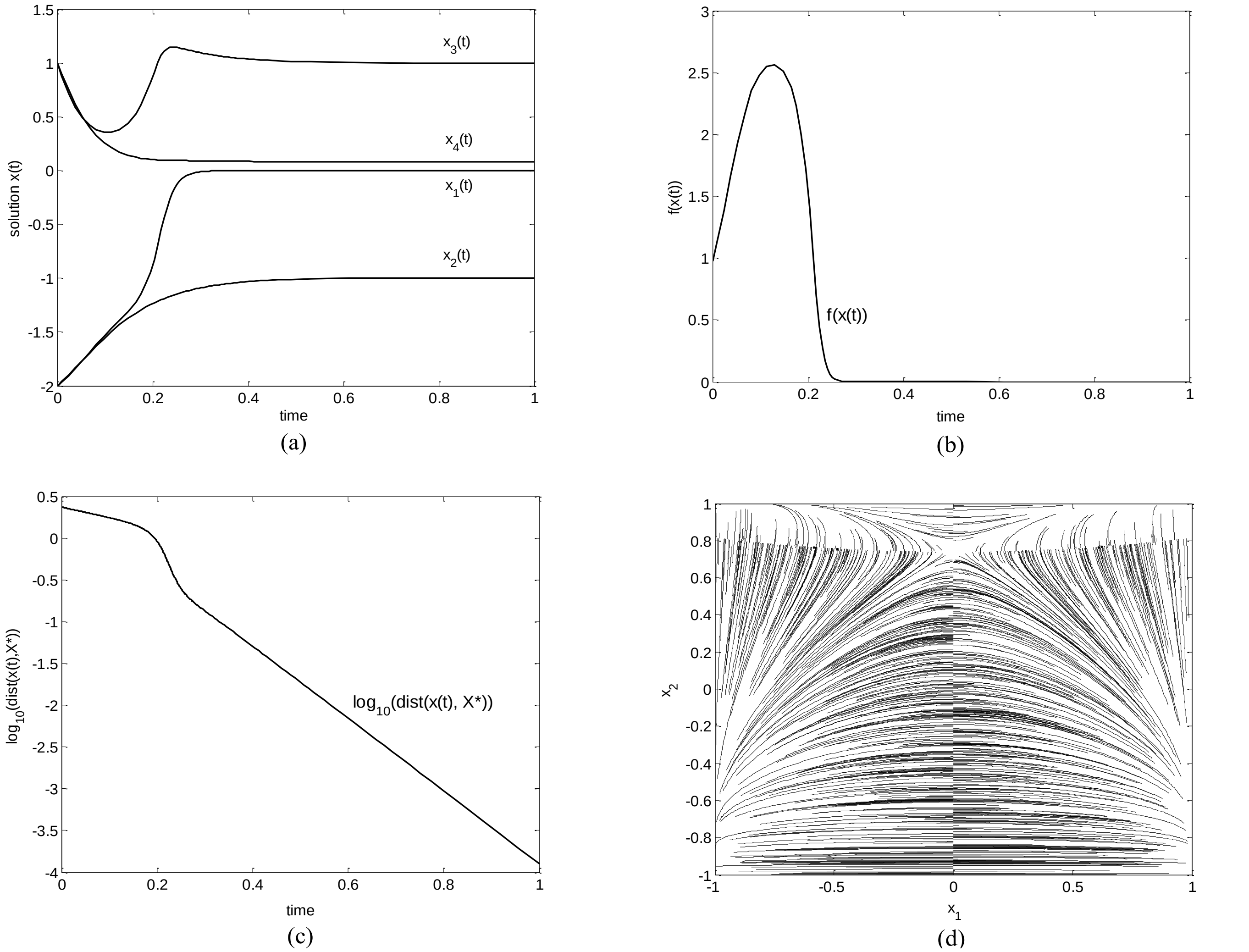

Fig. 3. Simulation results for Example 1: (a) Solution $x(t)$, (b) Convergence of the cost function, (c) $\log_{10}(\mathrm{dist}(x(t), X^*))$ as a function of time, (d) Trajectories obtained from 1000 random initial points.

the same intervals as above. As in [24], for each trajectory, a final cost value not greater than 10^{-6} was considered to be a criterion of successful solution of the problem. Only 14 failures were recorded with NFGPNN. This corresponds to a success rate of 99.86% and compares favorably with the published results for the neural network of [24] (65 failures, i.e. a success rate of 99.35%).

Example 2: This example is a nonconvex problem with $n = 3$ variables and $m = 1$ nonlinear constraint:

$$\min_{y\in R^3}\left\{f(x):\; x_1^2 + x_2^2 - x_3 = 0\right\},$$

where

$$f(x)=\begin{cases}\left(\sqrt{x_3}-1\right)^4\left(1-\dfrac{x_2}{2\sqrt{x_3}}\cos\left(\dfrac{1}{\sqrt{x_3}-1}\right)-\dfrac{x_1}{2\sqrt{x_3}}\sin\left(\dfrac{1}{\sqrt{x_3}-1}\right)\right), & \\ & if\;\; x_3\neq 1 \;\; and \;\; x_3>0\\ 0, & if\;\; x_3=1\end{cases}$$

The origin of this problem can be traced in a two dimensional unconstrained example from [49]. The cost function $f:\;\Omega\to R$ is continuously differentiable in its domain $\Omega=\{x\in R^3:\; x_3>0\}$. Minimizers of this problem contain the set $X^*=\{x\in R^3:\;\; x_3=1,\;\; x_1^2+x_2^2-1=0\}$.

Starting from the non-feasible initial point $x_o=[-2\;\;2\;\;2\,]^T$, the solution $x(t)$ depicted in Fig. 4a is obtained. Figure 4b shows the corresponding trajectory, in the (x_1,x_2) plane. It can be observed both from Fig. 4a and from Fig. 4b that the solution $x(t)$ does not appear to converge to any particular point in X^*; instead $x(t)$ seems to contain subsequences which converge to every point in X^*. Thus it appears that, for this example, the set Λ^+ of limit points of $x(t)$ satisfies $\Lambda^+ = X^*$, i.e. the solution $x(t)$ seems to converge to the entire set X^*. In Fig. 4c, the distance of $x(t)$ from the set X^*: $\log_{10}(\mathrm{dist}(x(t),X^*))=$

$$=\log_{10}\left\{\min_{y\in X^*}\|x(t)-y\|\right\}=\log_{10}\left[\left(\sqrt{x_1(t)^2+x_2(t)^2}-1\right)^2+\left(x_3(t)-1\right)^2\right]^{1/2}$$

is plotted as a function of time; the rate of decrease does not appear to be exponential. However this does not contradict the results of Section IV since in this example, at least one of the assumptions (i.e. that x^* satisfies sufficient conditions of

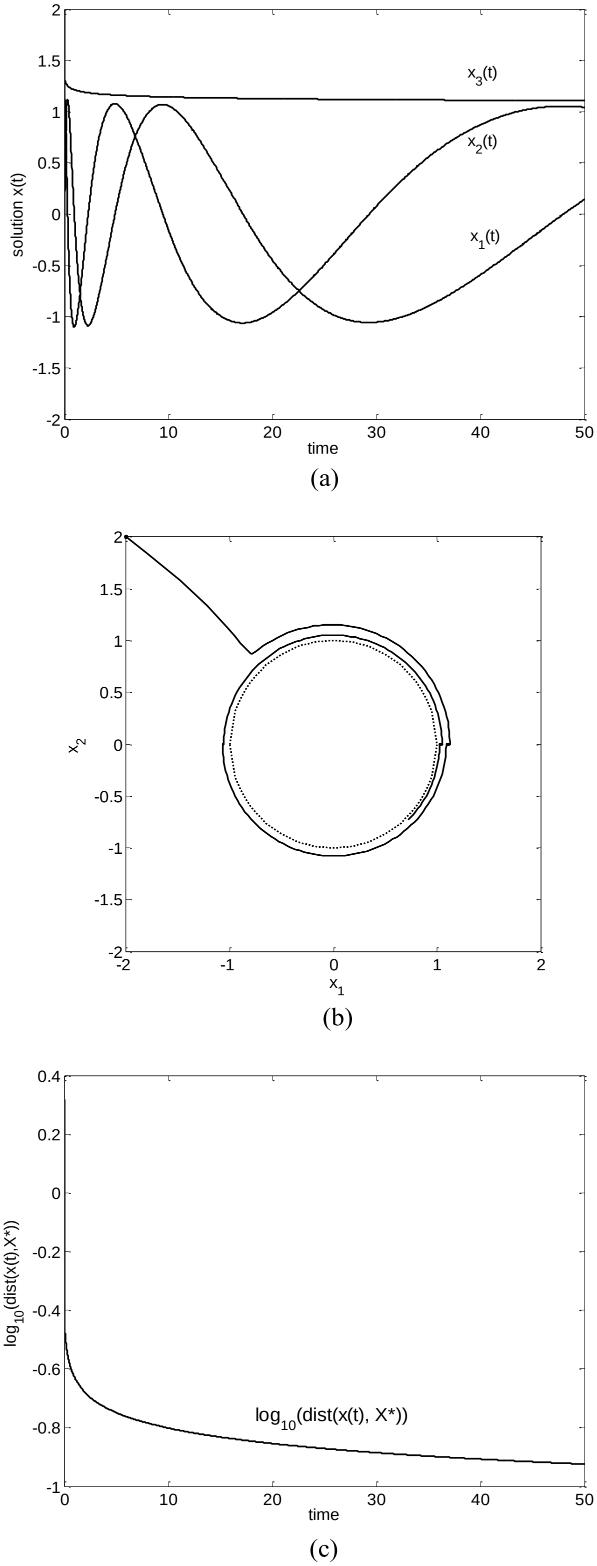

Fig. 4. Simulation results for Example 2: (a) Solution $x(t)$, (b) Trajectory of $x(t)$ in the (x_1,x_2) plane, (c) $\log_{10}(\text{dist}(x(t), X^*))$ as a function of time.

optimality) of Theorem 3 is not satisfied. The values $\mu = \rho = 10\,000$ were used.

I. Conclusions

In this paper a recurrent neural network is proposed for equality constrained nonlinear optimization problems. When started from a feasible initial point, the proposed neural network (NFGPNN) constructs a feasible trajectory, satisfying the constraints at all times. In the usual case however, when the initial point is non-feasible, NFGPNN constructs a non-feasible trajectory, which satisfies the constraints only in the limit as $t \to \infty$. To reduce the cost function, a projection of the cost gradient onto the tangent space of the constraints is used.

Local convergence results are given for the general case (i.e. without assuming convexity of the problem to be solved) which show that strict local minimizers of the optimization problem (P) are exponentially stable equilibrium points of NFGPNN. Global convergence results are given for convex optimization problems: solutions of NFGPNN emanating from arbitrary initial points are shown to convergence to the set of global minimizers of the optimization problem. The rate of convergence of the proposed neural network is shown to be exponential, both for convex and non-convex problems.

Numerical results confirm these findings and indicate that NFGPNN is both efficient and accurate.

Appendix

Proof of Theorem 1: We first note that since x^* is a regular point and $\nabla h_i(x)$, $i=1,\ldots,m$ are continuous, there exists a neighborhood $N(x^*,\varepsilon)=\{x\in R^n : \|x-x^*\| \le \varepsilon\}$ of x^*, such that the gradients $\nabla h_i(x)$, $i=1,\ldots,m$ are linearly independent $\forall x \in N(x^*,\varepsilon)$. Thus Assumption 1 holds for every $x\in N(x^*,\varepsilon)$ therefore, within $N(x^*,\varepsilon)$, NFGPNN is well defined (in explicit form) by equations (13). In this proof, $x\in N(x^*,\varepsilon)$ will be assumed, unless otherwise stated. Also, the Euclidean norm of a vector y is denoted simply as $\|y\|$.

We shall obtain some useful bounds and, based on these, we shall define a Lyapunov function which satisfies the assumptions of a well known theorem, [50, Theorem 4.10, p. 154], on exponential stability.

Let $L(x,\lambda)=f(x)+\lambda^{\mathrm{T}}h(x)$ be the Lagrangian function for problem (P) and let $\lambda^* \in R^m$ be the Lagrange multipliers corresponding to x^*, so that we have

$$\nabla L(x^*,\lambda^*) = \nabla f(x^*) + \nabla h(x^*)\lambda^* = 0 \tag{20}$$

Since $P(x)\nabla h(x)=0$ we obtain

$$P(x)\nabla f(x) = P(x)\left(\nabla f(x) + \nabla h(x)\lambda^*\right) = P(x)\nabla L(x,\lambda^*) \tag{21}$$

Taylor expansion of $P(x)\nabla L(x,\lambda^*)$ around x^* yields

$$\begin{aligned} P(x)\nabla L(x,\lambda^*) &= P(x^*)\nabla L(x^*,\lambda^*) + \\ &+ \nabla\{P(x)\nabla L(x,\lambda^*)\}\Big|_{x=x^*}^{\mathrm{T}}(x-x^*) + o(\|x-x^*\|) \end{aligned} \tag{22}$$

Let $p_i(x)^{\mathrm{T}} \in R^n$, $i=1,\ldots,n$ denote the i$^{\text{th}}$ row of the $n\times n$ matrix $P(x)$. Differentiation of $p_i(x)^{\mathrm{T}}\nabla L(x,\lambda^*)$ with respect to x gives, in view of (20),

$$\begin{aligned} \nabla\{p_i(x)^{\mathrm{T}}\nabla L(x,\lambda^*)\}\Big|_{x=x^*} &= \left[\nabla p_i(x)\nabla L(x,\lambda^*) + \frac{\partial^2 L(x,\lambda^*)}{\partial x^2}p_i(x)\right]_{x=x^*} \\ &= \frac{\partial^2 L(x^*,\lambda^*)}{\partial x^2}p_i(x^*) \end{aligned}$$

Hence

$$\nabla\{P(x)\nabla L(x,\lambda^*)\}\big|_{x=x^*} = \nabla\begin{bmatrix} p_1(x)^{\mathrm T}\nabla L(x,\lambda^*) \\ \vdots \\ p_n(x)^{\mathrm T}\nabla L(x,\lambda^*)\end{bmatrix}_{x=x^*}$$
$$= \left[\nabla\{p_1(x)^{\mathrm T}\nabla L(x,\lambda^*)\},\cdots,\nabla\{p_n(x)^{\mathrm T}\nabla L(x,\lambda^*)\}\right]_{x=x^*} \quad (23)$$
$$= \frac{\partial^2 L(x^*,\lambda^*)}{\partial x^2}P^{\mathrm T}(x^*)$$

From (20) – (23) it follows that

$$P(x)\nabla f(x) = P(x^*)\frac{\partial^2 L(x^*,\lambda^*)}{\partial x^2}(x-x^*) + o(\|x-x^*\|) \qquad (24)$$

Let F be a $n\times(n-m)$ full rank matrix such that $\nabla h(x^*)^{\mathrm T}F = 0$ and $F^{\mathrm T}F = I_{n-m}$. Such a matrix F can be easily obtained as part of the QR decomposition of the matrix $\nabla h(x^*)^{\mathrm T}$. It is easy to show that:

$$P(x^*) = FF^{\mathrm T} \qquad (25)$$

Let the $n\times n$ symmetric matrix $B(x)$ be defined as follows,

$$B(x) = \nabla h(x)\left(\nabla h(x)^{\mathrm T}\nabla h(x)\right)^{-1}\nabla h(x)^{\mathrm T} \qquad (26)$$

From the definition (15) of $P(x)$, (25) and (26) we obtain,

$$\begin{aligned} I_n &= FF^{\mathrm T} + \nabla h(x^*)\left(\nabla h(x^*)^{\mathrm T}\nabla h(x^*)\right)^{-1}\nabla h(x^*)^{\mathrm T} \\ &= FF^{\mathrm T} + B(x^*) \end{aligned} \qquad (27)$$

Then (27) yields

$$\left\|F^{\mathrm T}(x-x^*)\right\|^2 = \|x-x^*\|^2 - (x-x^*)^{\mathrm T}B(x^*)(x-x^*) \qquad (28)$$

Now, from (13), (24) and (25) we get (where x stands for $x(t)$ in the right hand sides of the equations that follow),

$$\begin{aligned} \frac{d(x(t)-x^*)}{dt} &= -\mu FF^{\mathrm T}\frac{\partial^2 L(x^*,\lambda^*)}{\partial x^2}(x-x^*) \\ &\quad -\rho\nabla h(x)\left(\nabla h(x)^{\mathrm T}\nabla h(x)\right)^{-1}h(x) + o(\|x-x^*\|) \end{aligned} \qquad (29)$$

Premultiplication by $(x-x^*)^{\mathrm T}$ and replacement of I_n from (27) into the right-hand-side yields

$$\begin{aligned} \frac{1}{2}\frac{d\|x(t)-x^*\|^2}{dt} &= (x(t)-x^*)^{\mathrm T}\frac{d(x(t)-x^*)}{dt} = \\ &= -\mu(x-x^*)^{\mathrm T}FF^{\mathrm T}\frac{\partial^2 L(x^*,\lambda^*)}{\partial x^2}FF^{\mathrm T}(x-x^*) \\ &\quad -\rho(x-x^*)^{\mathrm T}\nabla h(x)\left(\nabla h(x)^{\mathrm T}\nabla h(x)\right)^{-1}h(x) \\ &\quad -\mu(x-x^*)^{\mathrm T}FF^{\mathrm T}\frac{\partial^2 L(x^*,\lambda^*)}{\partial x^2}B(x^*)(x-x^*) + o(\|x-x^*\|) \end{aligned} \qquad (30)$$

Let α be the minimum eigenvalue of the matrix $F^{\mathrm T}\frac{\partial^2 L(x^*,\lambda^*)}{\partial x^2}F$. Since x^* satisfies the sufficient conditions of optimality for problem (P), it follows that $\alpha > 0$ therefore we have

$$(x-x^*)^{\mathrm T}FF^{\mathrm T}\frac{\partial^2 L(x^*,\lambda^*)}{\partial x^2}FF^{\mathrm T}(x-x^*) \ge \alpha\left\|F^{\mathrm T}(x-x^*)\right\|^2 \qquad (31)$$

Making use of (30), (31) and (28) we obtain, after simple algebra,

$$\begin{aligned} \frac{1}{2}\frac{d\|x(t)-x^*\|^2}{dt} &\le -\mu\alpha\|x-x^*\|^2 + (x-x^*)^{\mathrm T}W\nabla h(x^*)^{\mathrm T}(x-x^*) \\ &\quad -\rho(x-x^*)^{\mathrm T}\nabla h(x)\left(\nabla h(x)^{\mathrm T}\nabla h(x)\right)^{-1}h(x) + o(\|x-x^*\|^2) \end{aligned} \qquad (32)$$

where,

$$W = \mu\left(\alpha I_n - FF^{\mathrm T}\frac{\partial^2 L(x^*,\lambda^*)}{\partial x^2}\right)\nabla h(x^*)\left(\nabla h(x^*)^{\mathrm T}\nabla h(x^*)\right)^{-1}.$$

Next, we seek bounds for the terms on the right hand side of (32). Taylor expansion of $h(x)$ around x gives $h(x^*) = h(x) + \nabla h(x)^{\mathrm T}(x^*-x) + o(\|x-x^*\|)$, and, since $h(x^*) = 0$,

$$\nabla h(x)^{\mathrm T}(x-x^*) = h(x) + o(\|x-x^*\|) \qquad (33)$$

Similarly, Taylor expansion of $h(x)$ around x^* gives $h(x) = h(x^*) + \nabla h(x^*)^{\mathrm T}(x-x^*) + o(\|x-x^*\|)$, hence, since $h(x^*) = 0$,

$$\nabla h(x^*)^{\mathrm T}(x-x^*) = h(x) + o(\|x-x^*\|) \qquad (34)$$

Since the matrix $B(x)$ is positive semidefinite $\forall x\in N(x^*,\varepsilon)$, it follows from (33) that,

$$\begin{aligned} -\rho(x-x^*)^{\mathrm T}\nabla h(x)\left(\nabla h(x)^{\mathrm T}\nabla h(x)\right)^{-1}h(x) &= \\ = -\rho(x-x^*)^{\mathrm T}B(x)(x-x^*) + o(\|x-x^*\|^2) & \\ \le o(\|x-x^*\|^2) & \end{aligned} \qquad (35)$$

If $\sigma = \|W\|_{\mathrm F}$, it follows from (34) that,

$$\begin{aligned} (x-x^*)^{\mathrm T}W\nabla h(x^*)^{\mathrm T}(x-x^*) &= (x-x^*)^{\mathrm T}W h(x) + o(\|x-x^*\|^2) \\ &\le \sigma\|x-x^*\|\|h(x)\| + o(\|x-x^*\|^2) \end{aligned} \qquad (36)$$

Then, from (32), (35) and (36) we obtain,

$$\frac{1}{2}\frac{d\|x(t)-x^*\|^2}{dt} \le -\mu\alpha\|x-x^*\|^2 + \sigma\|x-x^*\|\|h(x)\| + o(\|x-x^*\|^2) \qquad (37)$$

Also, from the definition of $o(.)$, it follows that given any $\delta\in(0,\mu\alpha)$, there exists $\hat\varepsilon\in(0,\varepsilon)$ such that $o(\|x-x^*\|^2) \le \delta\|x-x^*\|^2$ holds $\forall x\in N(x^*,\hat\varepsilon)$. Hence we obtain from (37),

$$\frac{1}{2}\frac{d\|x(t)-x^*\|^2}{dt} \le -(\mu\alpha-\delta)\|x-x^*\|^2 + \sigma\|x-x^*\|\|h(x)\|, \quad \forall x\in N(x^*,\hat\varepsilon) \qquad (38)$$

We shall now define a suitable Lyapunov function $V: R^n \to R$ as follows

$$V(x) = \frac{1}{2}\|x-x^*\|^2 + \frac{c}{2}\|h(x)\|^2$$

where the parameter $c > 0$ will be determined in the sequel. It follows from (34) and the definition of $o(.)$ that there exist $k_1 > 0$ and $\tilde\varepsilon\in(0,\hat\varepsilon)$ such that

$$\frac{1}{2}\|x-x^*\|^2 \le V(x) \le k_1\|x-x^*\|^2, \qquad \forall x\in N(x^*,\tilde\varepsilon) \qquad (39)$$

It follows from (5) that $\dot h(x(t)) = -\rho h(x(t))$, $\forall t\ge 0$; hence (38) yields,

$$\frac{dV(x(t))}{dt} \le -(\mu a-\delta)\|x-x^*\|^2 + \sigma\|x-x^*\|\|h(x)\| - c\rho\|h(x)\|^2, \quad \forall x\in N(x^*,\tilde\varepsilon) \qquad (40)$$

Then, by choosing $\zeta\in(0,\mu\alpha-\delta)$ and $c = \frac{\sigma^2}{4\rho\zeta} > 0$, we obtain from (40),

$$\begin{aligned} \frac{dV(x(t))}{dt} &\le -\left(\sqrt{\zeta}\|x-x^*\| - \frac{\sigma}{2\sqrt{\zeta}}\|h(x)\|\right)^2 - (\mu a-\delta-\zeta)\|x-x^*\|^2 \\ &\le -(\mu a-\delta-\zeta)\|x-x^*\|^2, \qquad \forall x\in N(x^*,\tilde\varepsilon) \end{aligned} \qquad (41)$$

Conditions (39) and (41) ensure that the function $V(x)$ defined above satisfies the assumptions of [50, Theorem 4.10, p. 154].

This implies exponential stability of x^* and completes our proof. □

Proof of Theorem 2: (a) Boundedness of the solutions. We shall first show that any solution $x(t)$ of (22) is bounded. Let $x^* \in R^n$ be any minimizer of problem (CP). Then we have $A^{\mathrm T}x^* - b = 0$ and (17) may be written as,

$$\dot x(t) = -\mu P\nabla f\big(x(t)\big) - \rho A\big(A^{\mathrm T}A\big)^{-1}A^{\mathrm T}\big(x(t)-x^*\big) \qquad (42)$$

Let the function $V: R^n \to R$ (with $c>0$) be defined as,

$$V(x) = f(x) + \frac{\rho}{2\mu}\|x-x^*\|_2^2 + c\big\|A^{\mathrm T}(x-x^*)\big\|_1 \qquad (43)$$

First we prove that $x(t)$ is a descent solution for the function $V(x)$, i.e. that $\dot V(x(t)) \le 0$, $\forall t \ge 0$. Premultiplication of (17) by $A^{\mathrm T}$ gives $A^{\mathrm T}\dot x(t) = -\rho\,(A^{\mathrm T}x(t) - b)$ which upon integration yields (7) with $h(x) = A^{\mathrm T}x(t) - b = A^{\mathrm T}(x - x^*)$, i.e.

$$A^{\mathrm T}\big(x(t)-x^*\big) = \exp(-\rho t)A^{\mathrm T}\big(x_{\mathrm o}-x^*\big) \qquad (44)$$

It then follows from (43) and (44),

$$V\big(x(t)\big) = f\big(x(t)\big) + \frac{\rho}{2\mu}\|x(t)-x^*\|_2^2 + c\exp(-\rho t)\big\|A^{\mathrm T}\big(x_{\mathrm o}-x^*\big)\big\|_1$$

hence

$$\dot V\big(x(t)\big) = \nabla f\big(x(t)\big)^{\mathrm T}\dot x(t) + \frac{\rho}{\mu}\big(x(t)-x^*\big)^{\mathrm T}\dot x(t) - c\rho\exp(-\rho t)\big\|A^{\mathrm T}\big(x_{\mathrm o}-x^*\big)\big\|_1 \qquad (45)$$

Let $B = A(A^{\mathrm T}A)^{-1}A^{\mathrm T}$ and let x stand for $x(t)$. From (45), (42) and (44) we obtain,

$$\begin{aligned}\dot V\big(x(t)\big) = &-\mu\nabla f(x)^{\mathrm T}P\nabla f(x) - \rho\nabla f(x)^{\mathrm T}B(x-x^*)\\ &-\rho(x-x^*)^{\mathrm T}P\nabla f(x) - \frac{\rho^2}{\mu}(x-x^*)^{\mathrm T}B(x-x^*)\\ &-c\rho\big\|A^{\mathrm T}(x-x^*)\big\|_1\end{aligned}$$

and, since $P+B=I$, $P=P^{\mathrm T}P$, $B=B^{\mathrm T}B$, we get

$$\begin{aligned}\dot V\big(x(t)\big) = &-\mu\|P\nabla f(x)\|_2^2 - \rho\nabla f(x)^{\mathrm T}(x-x^*)\\ &-\frac{\rho^2}{\mu}\|B(x-x^*)\|_2^2 - c\rho\big\|A^{\mathrm T}(x-x^*)\big\|_1\end{aligned} \qquad (46)$$

As $f(x)$ is convex it follows that $(x-x^*)^{\mathrm T}(\nabla f(x) - \nabla f(x^*)) \ge 0$, $\forall x \in R^n$ (see e.g. [49]). Hence, since $P\nabla f(x^*) = 0$, we have

$$\begin{aligned}-(x-x^*)^{\mathrm T}\nabla f(x) &\le -(x-x^*)^{\mathrm T}(P+B)\nabla f(x^*)\\ &= -(x-x^*)^{\mathrm T}B\nabla f(x^*)\end{aligned} \qquad (47)$$

An upper bound is obtained for the expressions in (47) by using norm properties:

$$-(x-x^*)^{\mathrm T}B\nabla f(x^*) \le \|B(x-x^*)\|_2\|\nabla f(x^*)\|_2 \le \big\|A(A^{\mathrm T}A)^{-1}\big\|_{\mathrm F} \times \times\big\|A^{\mathrm T}(x-x^*)\big\|_2\|\nabla f(x^*)\|_2$$

therefore there exists $\xi > 0$ such that

$$-(x-x^*)^{\mathrm T}B\nabla f(x^*) \le \xi\big\|A^{\mathrm T}(x-x^*)\big\|_1\|\nabla f(x^*)\|_2 \qquad (48)$$

It now follows from (46), (47), (48) that, for $c \ge \xi\|\nabla f(x^*)\|_2$,
$\dot V\big(x(t)\big) \le -\mu\|P\nabla f(x)\|_2^2 + \rho\big\|A^{\mathrm T}(x-x^*)\big\|_1\big(\xi\|\nabla f(x^*)\|_2 - c\big) \le 0$.
Thus it holds

$$V\big(x(t)\big) \le V(x_{\mathrm o}), \qquad \forall t \ge 0 \qquad (49)$$

i.e. the solution $x(t)$, $\forall t \ge 0$ is contained in the $V(x_{\mathrm o})$ level set of the function $V(x)$. We next show that this level set is bounded. A lower bound for $f(x)$ is obtained by using a property of convex functions, [49], and the facts $P+B=I$ and $P\nabla f(x^*) = 0$:

$$\begin{aligned}f(x) &\ge f(x^*) + (x-x^*)^{\mathrm T}\nabla f(x^*)\\ &= f(x^*) + (x-x^*)^{\mathrm T}(P+B)\nabla f(x^*)\\ &= f(x^*) + (x-x^*)^{\mathrm T}B\nabla f(x^*)\end{aligned}$$

It then follows from the above bound and equations (49), (48), (43) that, for $c \ge \xi\|\nabla f(x^*)\|_2$, and $\forall t \ge 0$:

$$\begin{aligned}V(x_{\mathrm o}) \ge V\big(x(t)\big) \ge\ & f(x^*) + \big(x(t)-x^*\big)^{\mathrm T}B\nabla f(x^*) +\\ &+\frac{\rho}{2\mu}\|x(t)-x^*\|_2^2 + c\big\|A^{\mathrm T}(x(t)-x^*)\big\|_1\\ \ge\ & f(x^*) + \big(c - \xi\|\nabla f(x^*)\|_2\big)\big\|A^{\mathrm T}(x(t)-x^*)\big\|_1 +\\ &+\frac{\rho}{2\mu}\|x(t)-x^*\|_2^2\end{aligned} \qquad (50)$$

Let the function $g: R^n \to R$ be defined as $g(x) = f(x^*) + \tilde\psi\big\|A^{\mathrm T}(x-x^*)\big\|_1 + \frac{\rho}{2\mu}\|x-x^*\|_2^2$, where $\tilde\psi = c - \xi\|\nabla f(x^*)\|_2 > 0$. Then, by (50), the solution $x(t)$ is contained ($\forall t \ge 0$) in the $V(x_{\mathrm o})$ level set of the function $g(x)$. Moreover, $g(x)$ is a strictly convex function with a unique minimum at x^*, hence every level set of $g(x)$ is bounded, [49, Prop. B.9, p. 569]. This proves boundedness of the solution $x(t)$.

(b) Convergence. To prove the remaining conclusions of the Theorem, let Λ^+ be the positive limit set of $x(t)$. Since $x(t)$ is bounded, Λ^+ is nonempty, [51, Theorem 4, p. 364], and the maximal interval of existence of $x(t)$ is $[0,\infty)$, [51, ex. 6, p. 365]. It then follows from (44) that $\lim_{t\to\infty}\big\{A^{\mathrm T}\big(x(t)-x^*\big)\big\} = \lim_{t\to\infty}\big\{A^{\mathrm T}x(t) - b\big\} = 0$, i.e.

$$\Lambda^+ \subset \mathrm F \qquad (51)$$

where F is the set of feasible points for problem (P) defined in (2).

Since, by (3), $\mathrm D = \mathrm D' \cap \mathrm F$, it remains to show that $\Lambda^+ \subset \mathrm D'$. We shall do so by applying LaSalle's theorem [51, Theorem 3.2, p. 243] to the dynamical system (17). Let the function $W(x)$ be defined as $W(x) = f(x) + c\big\|A^{\mathrm T}x - b\big\|_1$, where $c > 0$ will be determined in the sequel. It follows from (7) (see also (44)) that $\big\|A^{\mathrm T}x(t) - b\big\|_1 = \exp(-\rho t)\big\|A^{\mathrm T}x_{\mathrm o} - b\big\|_1$, hence the function

$$\begin{aligned}W\big(x(t)\big) &= f\big(x(t)\big) + c\big\|A^{\mathrm T}x(t) - b\big\|_1\\ &= f\big(x(t)\big) + c\exp(-\rho t)\big\|A^{\mathrm T}x_{\mathrm o} - b\big\|_1\end{aligned}$$

is differentiable with respect to t. Then

$$\begin{aligned}\dot W\big(x(t)\big) &= \frac{d}{dt}f\big(x(t)\big) - c\rho\exp(-\rho t)\big\|A^{\mathrm T}x_{\mathrm o} - b\big\|_1\\ &= \nabla f\big(x(t)\big)^{\mathrm T}\dot x(t) - c\rho\big\|A^{\mathrm T}x(t) - b\big\|_1\end{aligned}$$

Replacing $\dot x$ from (17) into the above we obtain, $\forall x \in R^m$,

$$\begin{aligned}\dot W(x) = &-\mu\nabla f(x)^{\mathrm T}P\nabla f(x) - \rho\nabla f(x)^{\mathrm T}A\big(A^{\mathrm T}A\big)^{-1}(A^{\mathrm T}x - b)\\ &-c\rho\sum_{i=1}^{m}\big|e_i^{\mathrm T}(A^{\mathrm T}x - b)\big|\end{aligned}$$

i.e.,

$$\dot{W}(x)=-\mu\|P\nabla f(x)\|_2^2-\rho\sum_{i=1}^{m}\left|e_i^{\mathrm{T}}(A^{\mathrm{T}}x-b)\right|\times \times\left[c+\mathrm{sgn}\left(e_i^{\mathrm{T}}(A^{\mathrm{T}}x-b)\right)\psi_i(x)\right] \quad (52)$$

where the fact $P=P^{\mathrm{T}}P$ has been used, and $\psi: R^n \to R^m$ is defined by $\psi(x)=(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}\nabla f(x)$. Let the set $\mathrm{X}\subset R^n$ be defined as $\mathrm{X}=\{x(t): t\in[0,\infty)\}$. It follows from [51, Theorem 5, p. 365], that the set $\overline{\mathrm{X}}=\mathrm{X}\cup\Lambda^+$ is compact. It also follows from Assumption 1 that $\psi(x)$ defined above is a continuous function, hence, Weierstrass' theorem yields that the quantities $\bar{c}_i=\max_y\{|\psi_i(y)|: y\in\overline{\mathrm{X}}\}$, $i=1,\ldots,m$ exist and are finite. Now, by choosing $c>\max\{\bar{c}_1,\bar{c}_2,\ldots,\bar{c}_m\}$ we obtain $c+\mathrm{sgn}\{e_i^{\mathrm{T}}(A^{\mathrm{T}}x-b)\}\psi_i(x)>0$, $\forall x\in\overline{\mathrm{X}}$, hence (52) yields $\dot{W}(x)\le-\mu\|P\nabla f(x)\|_2^2\le 0$, $\forall x\in\overline{\mathrm{X}}$. Since $x\in\mathrm{D}'$ iff $P\nabla f(x)=0$, it follows from LaSalle's Theorem and the definitions of D' and P that

$$\Lambda^+\subset\left\{x\in R^n : P\nabla f(x)=0\right\}=\mathrm{D}' \quad (53)$$

Now (51), (53) and (3) yield the desired result. □

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers for their comments which have greatly improved the manuscript. In particular the authors are grateful to reviewer #4 for bringing some important references to their attention.

REFERENCES

[1] 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 and Systems*, vol. 33, pp. 533-541, 1986.

[2] L.O. Chua and G.-N. Lin, "Nonlinear programming without computation", *IEEE Trans. Circuits and Systems*, vol. 31, pp. 182-188, 1984.

[3] M.P. Kennedy and L.O. Chua, "Unifying the Tank and Hopfield linear programming circuit and the canonical nonlinear programming circuit of Chua and Lin", *IEEE Trans. Circuits and Systems*, vol. 34, pp. 210-214, 1986.

[4] W.E. Lillo, M.H. Loh, S. Hui, and S.H. Zak, "On solving constrained optimization problems with neural networks: a penalty method approach", *IEEE Trans. Neural Networks*, vol. 4, pp. 931-940, 1993.

[5] 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 and Systems*, vol. 37, pp. 384-398, 1990.

[6] W.E. Lillo, S. Hui, and S.H. Zak, "Neural networks for constrained optimization problems", *Int. J. of Circuit Th. and Appl*., vol. 21, pp. 385-399, 1993.

[7] A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, John Wiley & Sons, 1993.

[8] S.H. Zak, V. Upatising, and S. Hui, "Solving linear programming problems with neural networks: a comparative study", *IEEE Trans. Neural Networks*, vol. 6, pp. 94-103, 1995.

[9] E. Chong, S. Hui, and S.H. Zak, "An analysis of a class of neural networks for solving linear programming problems", *IEEE Trans. Automatic Contr.*, vol. 44, pp. 1995-2006, 1999.

[10] 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, 2004.

[11] M. Forti, P. Nistri, and M. Quincampoix, "Convergence of Neural Networks for programming problems via a nonsmooth Lojasiewicz inequality", *IEEE Trans. Neural Networks*, vol. 17, no. 6, pp. 1471-1486, 2006.

[12] P.T. Krasopoulos and N.G. Maratos, "A neural network for convex optimization", in *Proc. ISCAS 2006*, pp.747-750, 2006.

[13] S. Zhang and A.G. Constandinides, "Lagrange programming neural networks", *IEEE Trans. Circuits Syst. II,* vol. 39, pp. 441-452, July 1992.

[14] S. Zhang, X. Zhu and L.-H. Zou, "Second-order neural nets for constrained optimization", *IEEE Trans. Neural Networks*, vol. 3, pp. 1021-1024, Nov. 1992.

[15] Y. Xia, "Global convergence analysis of Lagrangian networks", *IEEE Trans. Circuits Syst. I*, vol. 50, pp. 818-822, June 2003.

[16] A. Bouzerdoum and T. Pattison, "Neural network for quadratic optimization with bound constraints", *IEEE Trans. Neural Networks*, vol. 4, pp. 293-304, 1993.

[17] S. Sudharsanan and M. Sundareshan, "Exponential stability and a systematic synthesis of a neural network for quadratic minimization", *Neural Networks*, vol. 4, pp. 599-613, 1991.

[18] Y. Xia, "A new neural network for solving linear and quadratic programming problems", *IEEE Trans. Neural Networks*, vol. 7, pp. 1544-1547, Nov. 1996.

[19] Y. Xia and J. Wang, "A recurrent neural network for solving linear projection equations", *Neural Networks*, vol. 13, pp. 337-350, 2000.

[20] Y. Zhang and J. Wang, "A dual neural network for convex quadratic programming subject to linear equality and inequality constraints", *Physics Letters A*, vol. 298, pp. 271-278, 2002.

[21] Y. Xia, G. Feng, and J. Wang, "A recurrent neural network with exponential convergence for solving convex quadratic program and related linear piecewise equations", *Neural Networks*, vol.17, pp. 1003-1015, 2004.

[22] X.-B. Gao, L.-Z. Liao, and W. Xue, "A neural network for a class of convex quadratic minimax problems with constraints", *IEEE Trans. Neural Networks*, vol. 15, no. 3, pp. 622-628, 2004.

[23] Y. Xia and J. Wang, "A general methodology for designing globally convergent optimization neural networks", *IEEE Trans. Neural Networks*, vol. 9, no. 6, pp. 1331-1343, 1998.

[24] X.-B. Liang and J.Wang, "A recurrent neural network for nonlinear optimization with a continuously differentiable objective function and bound constraints", *IEEE Trans. Neural Networks*, vol. 11, no.6, pp. 1251-1262, 2000.

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

[26] Q. Tao, J. Cao, M. Xue, and H. Qiao, "A high performance neural network for solving nonlinear programming problems with hybrid constraints", *Physics Letters A*, vol. 288, no. 2, pp. 88-94, 2001.

[27] X.-B. Gao, "A novel neural network for nonlinear convex programming", *IEEE Trans. Neural Networks*, vol. 15, no.3, pp. 613-621, 2004.

[28] Y. Xia and J. Wang, "A recurrent neural network for solving nonlinear convex programs subject to linear constraints", *IEEE Trans. Neural Networks*, vol. 16, no. 2, pp. 379-386, 2005.

[29] X.-B. Gao, "Exponential stability of globally projected dynamic systems", *IEEE Trans. Neural Networks*, vol. 14, no.2, pp. 426-431, 2003.

[30] Y. Xia and J. Wang, "A general projection neural network for solving monotone variational inequality and related optimization problems", *IEEE Trans. Neural Networks*, vol. 15, no. 2, pp. 318-328, 2004.

[31] X.-B. Gao, L.-Z. Liao and L. Qi, "A novel neural network for variational inequalities with linear and nonlinear constraints", *IEEE Trans. Neural Networks*, vol. 16, no. 6, pp. 1305-1317, 2005.

[32] X. Hu and J. Wang, "Solving pseudomonotone variational inequalities and pseudoconvex optimization problems using the projection neural network", *IEEE Trans. Neural Networks*, vol. 17, no. 6, pp. 1487-1499, 2006.

[33] X. Hu and J. Wang, "Solving generally constrained generalized linear variational inequalities using the general projection neural networks", *IEEE Trans. Neural Networks*, vol. 18, no. 6, pp. 1697-1708, 2007.

[34] 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 Networks*, vol. 12, no. 5, pp. 1074-1083, 2001.

[35] S. Liu and J. Wang, "A simplified dual neural network for quadratic programming with its KWTA application", *IEEE Trans. Neural Networks*, vol. 17, no. 6, pp. 1500-1510, 2006.

[36] Y. Leung, K.-Z. Chen, and X.-B. Gao, "A high-performance feedback neural network for solving convex nonlinear programming problems", *IEEE Trans. Neural Networks*, vol. 14, no. 6, pp. 1469-1477, 2003.

[37] K. Tanabe, "A geometric method in nonlinear programming", *J. of Optimization Th. and Appl.*, vol. 30, no.2, pp. 181-210, 1980.

[38] H. Yamashita, "A differential equation approach to nonlinear programming", *Mathematical Programming*, vol.18, pp. 155-168, 1980.

[39] V. Hemke and J.B. Moore, *Optimization and Dynamical Systems*, Springer-Verlang, London, 1994.

[40] L.-Z. Liao, H. Qi and L. Qi, "Neurodynamical optimization", *J. of Global Optimization*, vol. 28, pp. 175-195, 2004.

[41] M. Barbarosou and N.G. Maratos, "A non-feasible gradient projection recurrent neural network for equality constrained optimization", in *Proc. IJCNN 2004*, pp.2251-2256, 2004.

[42] D.G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, 1984.

[43] D. Q. Mayne and N. Maratos, "A first order, exact penalty function algorithm for equality constrained optimization problems", *Mathematical Programming*, vol. 16, pp. 303-324, 1979.

[44] F.H. Branin, "Widely convergent method for finding multiple solutions of simultaneous nonlinear equations", *IBM Journal of Research and Development,* vol. 16, pp.504-521, 1972.

[45] H.G. Kwatny, R.F. Fischl and C.O. Nwankpa, "Local bifurcation in power systems: theory, computation, and application", *Proceedings of the IEEE*, vol. 83, no. 11, pp. 1456-1483, 1995.

[46] Y. Zhang, D. Jiang and J. Wang, "A recurrent neural network for solving Sylvester equation with time – varying coefficients", *IEEE Trans. Neural Networks*, vol. 13, no. 5, pp. 1053-1063, 2002.

[47] Y. Zhang and S.S. Ge, "Design and analysis of a general recurrent neural network model for time – varying matrix inversion", *IEEE Trans. Neural Networks*, vol. 16, no. 6, pp. 1477-1490, 2005.

[48] E. Polak, *Optimization Algorithms and Consistent Approximations*, Springer-Verlag New York, 1997.

[49] D. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1995.

[50] H.K. Khalil, *Nonlinear Systems,* Prentice Hall, New Jersey, 2002.

[51] N. Rouche, P. Hebets, and M. Laloy, *Stability Theory by Liapunov's Direct Method*, Springer-Verlag, New York, 1977.

Maria P. Barbarosou recieved the B.Sc. degree in Physics from the University of Athens, Creece, in 1996, the M.Sc. degree in Applied Mathematics and the Ph.D. degree in Neural Networks, both from the National Technical University of Athens, Greece, in 2004 and 2008 respectively.

From 1997 to 1999 she was with the Institute of Accelerating Systems and Applications (IASA), in Athens, working on the implementation of the control system for the IASA microtron. Since 1998, she is working in the Department of Electronics, Technological Educational Institute of Piraeus, Creece, as a teaching and research cooperator. Her research interests include neural networks, systems theory and state estimation.

Nicholas G. Maratos (M'06) received the Diploma in Electrical and Mechanical Engineering from the National Technical University of Athens, Greece in 1973, the M.Sc. degree in Control Systems and the Ph.D. degree in Optimization, from Imperial College of Science and Technology, London, UK, in 1974 and 1978 respectively.

From 1980 to 1983 he was a Postdoctoral Research Assistant in the CAD Group, Dept. of Electrical Engineering, Imperial College of Science and Technology, London. In 1985 he joined the School of Electrical and Computer Engineering, National Technical University of Athens, Greece, where he is currently a Professor. His research interests include nonlinear optimization (theory, algorithms, and applications), computer-aided design of electrical circuits, and neural networks.