2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

On The LVI-based Primal-Dual Neural Network for Solving Online Linear and Quadratic Programming Problems

Yunong Zhang

***Abstract*— Motivated by real-time solution to robotic problems, researchers have to consider the general unified formulation of linear and quadratic programs subject to equality, inequality and bound constraints simultaneously. A primal-dual neural network is presented in this paper for the online solution based on linear variational inequalities (LVI). The neural network is of simple piecewise-linear dynamics, globally convergent to optimal solutions, and able to handle linear and quadratic problems in the same manner. Other robotics-related properties of the LVI-based primal-dual network are also investigated, like, the convergence starting within feasible regions, and the case of no solutions.**

***Index Terms*— Linear programming (LP), quadratic programming (QP), primal-dual neural network (PDNN), linear variational inequalities, global convergence.**

I. Introduction

In view of its fundamental role arising in numerous fields of science and engineering, the problem of solving linear and quadratic programs has been investigated extensively for the past decades. For example, about the recent research based on recurrent neural networks (specifically, the Hopfield-type neural networks), see [1]-[4] and the references therein. The neural network (NN) approach is now thought to be a powerful tool for online computation, in view of its parallel distributed computing nature and hardware-implementation availability [5]-[10].

In the literature, researches are usually solving linear programming and quadratic programming problems separately. In addition, they handle optimization problems only subject to one or two kinds of constraints [3]. Motivated by engineering applications of linear/quadratic programming in robotics [4][11]-[16], however, the following general problem formulation is preferred as the basis of discussion:

$$\text{minimize} \quad x^T W x/2 + q^T x, \tag{1}$$

$$\text{subject to} \quad Jx = d, \tag{2}$$

$$Ax \leqslant b, \tag{3}$$

$$\xi^- \leqslant x \leqslant \xi^+, \tag{4}$$

where W is assumed only positive semi-definite such that quadratic programming and linear programming are both to be handled in this formulation.

Before proceeding, we briefly review the following existing neural networks. It is known that the early neural model like [1][17] contains finite penalty parameters and generates approximate solutions only. The Lagrange neural network may have premature defect when applied to inequalities-constrained QP problems [18]. To always obtain optimal/exact solutions, traditional primal-dual neural networks were proposed based on the Karush-Kuhn-Tacker condition and projection operator [19]. However, due to minimizing the duality gap by gradient descent methods, the dynamic equations of such primal-dual networks are usually complicated, even containing high-order nonlinear terms [3][15]. To reduce implementation and computation complexities, a dual neural network has been finally developed for handling general QP (1)-(4) with simple piecewise linearity and global convergence to optimal solutions [13]. The disadvantage of dual networks, however, is that they require the inversion of coefficient matrix W and thus only able to handle strictly convex quadratic programs preferably with fixed W [3].

As the research evolves spirally, a primal-dual neural network model has recently been "discovered" with simple piecewise linear dynamics, global convergence to optimal solutions, and capability of handling QP and LP problems in the same/unified manner [4][16][20]. Because the primal-dual neural network model is designed based on linear variational inequalities, it is termed the LVI-based primal-dual neural network. In view of only partial results existing (e.g., in [4][16][20], $q \equiv 0$ or no inequality constraint $Ax \leqslant b$), this paper presents the whole picture and further study of the LVI-based primal-dual network for solving QP/LP (1)-(4), including rigorous derivation, convergence properties, and solution characteristics.

The remainder of this paper is organized in four sections. The problem formulation, its utility and online primal-dual neural solver are discussed in Section II. Detailed derivation and proofs of the LVI-based primal-dual neural network are presented in Section III. The network convergence behavior is further investigated via numerical experiments in Section IV. Lastly, Section V concludes this paper with final remarks.

II. Problem Formulation & Solution

Redundant manipulators are robots having more degrees-of-freedom (e.g., n DOF) than required to perform a given end-effector task (e.g., m dimensional). One fundamental issue in robotics is the inverse kinematics problem [4][11]-[16][19]-[22]. By resolving the redundancy (i.e., $n-m$), the robots can avoid obstacles, joint physical limits and configuration singularities, and as well as optimize various performance criteria for particular purposes.

Y. Zhang was with the Department of Electronic & Electrical Engineering, University of Strathclyde, Glasgow G1 1QE, U.K., and is now with the Hamilton Institute, National University of Ireland, Maynooth, Co. Kildare, Ireland ynzhang@ieee.org

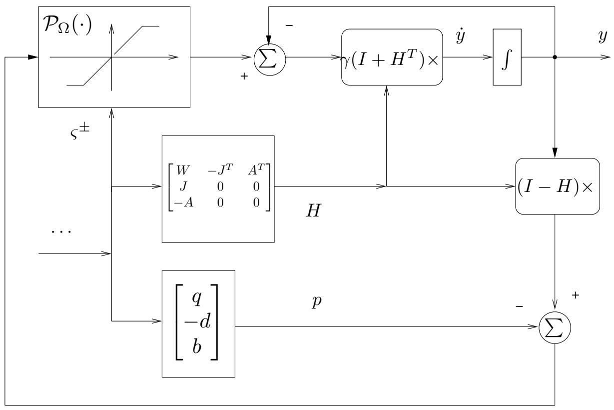


Fig. 1. Block diagram of LVI-based PDNN for solving (1)-(4).

Motivated by the real-time robotic applications, the general QP and LP problem formulation is required as in (1)-(4), which is simultaneously subject to equality, inequality and bound constraints. As for its usage in robotics, the performance index $x^TWx/2+q^Tx$ in (1) can be used for

- minimum-effort motion planning as in [12][16][19],
- minimum-energy / torque-optimization as in [4][11],
- drift-free motion planning like [13][21].

The equality constraint $Jx=d$ in (2) usually expresses a strong and main relation between some robotic variables; for example, a linear relation via Jacobian matrix J between the desired Cartesian velocity $d\in R^m$ and the joint velocity $x\in R^n$ to be resolved. On the other hand, the inequality constraint $Ax\leqslant b\in R^{\dim(b)}$ in (3) is entailed for robot obstacle avoidance as shown in [14] and [22], respectively, for point-obstacles and window-shaped obstacle avoidance. The bound constraint (4) is usually used to handle the avoidance of joint limits, joint velocity limits and joint acceleration/torque limits, like [4][11][13].

Based on the equivalence of QP/LP, LVI and a system of piecewise linear equations, a new type of primal-dual neural network solver can be developed to solve online (1)-(4). Simply saying, we can first convert QP/LP (1)-(4) to the following linear variational inequalities: to find a vector $y^*\in\Omega$ such that $\forall y\in\Omega:=\{y|\varsigma^-\leqslant y\leqslant\varsigma^+\}\subset R^{n+m+\dim(b)}$,

$$(y-y^*)^T(Hy^*+p)\geqslant 0, \tag{5}$$

where the primal-dual decision vector y and its lower/upper bounds are defined respectively as

$$y=\begin{bmatrix}x\\u\\v\end{bmatrix},\ \varsigma^-=\begin{bmatrix}\xi^-\\-\infty\\0\end{bmatrix},\ \varsigma^+=\begin{bmatrix}\xi^+\\+\infty\\+\infty\end{bmatrix}, \tag{6}$$

with ∞ here representing a sufficient large positive constant (or, vector of appropriate dimensions). The coefficients in (5) are defined as

$$H=\begin{bmatrix}W & -J^T & A^T\\ J & 0 & 0\\ -A & 0 & 0\end{bmatrix},\ p=\begin{bmatrix}q\\-d\\b\end{bmatrix}. \tag{7}$$

Moreover, in light of the equivalence of LVI (5) and the ensuing system of piecewise linear equation (also called the linear projection equation),

$$\mathcal{P}_\Omega(y-(Hy+p))-y=0, \tag{8}$$

it follows naturally from our neural-network design experience [3][4][11]-[16][22]-[26] that the LVI-based primal-dual neural network solver can be of the following dynamic equation:

$$\dot{y}=\gamma(I+H^T)\{\mathcal{P}_\Omega(y-(Hy+p))-y\} \tag{9}$$

where $\gamma>0$ is the design parameter used to scale the network convergence.

The block diagram on realizing (9) is presented in Fig. 1, where the piecewise-linear activation function $\mathcal{P}_\Omega(\cdot)$ can be implemented by using operational amplifiers known as limiter [2][3][6]. As bound constraint (4) is neatly cast into projection set Ω, the size of the LVI-based primal-dual network (9) is only the dimension sum of equality constraint (2), inequality constraint (3) and primal decision vector x, smaller than the existing neural networks' [2][15][18][19]. In addition, the LVI-based primal-dual network does not involve matrix inversion, matrix-matrix multiplication or high-order nonlinear computation, thus reducing the implementation and computation complexity, as compared to other recurrent neural models [2][3][15][19].

III. Theoretical Results

For clarity and better readability, the problem formulation (1)-(4) and its LVI-based primal-dual neural solver (9) are given in the previous section, while the detailed derivation and theoretical analysis are separated from them and presented as follows. Note that as a basis of theoretical analysis, the existence of at least one optimal solution x^* to the optimization problem (1)-(4) is always assumed throughout the paper unless stated otherwise.

Theorem 1 (LP/QP-LVI equivalence) Optimization problem (1)-(4) can be reformulated as the LVI problem (5).

Proof. It follows from [27] that the Lagrangian dual problem of (1)-(4) can be derived as

$$\text{max.}\ -\frac{1}{2}x^TWx+d^Tu-b^Tv+\xi^{-T}\nu^--\xi^{+T}\nu^+ \tag{10}$$

$$\text{s.t.}\ \ Wx+q-J^Tu+A^Tv-\nu^-+\nu^+=0, \tag{11}$$

$$\text{with}\ \ u\ \text{unrestricted},\ v\geqslant 0,\ \nu^-\geqslant 0,\ \nu^+\geqslant 0. \tag{12}$$

where $u\in R^m$, $v\in R^{\dim(b)}$, $\nu^-\in R^n$ and $\nu^+\in R^n$ are the corresponding dual decision variables. Then, a necessary and sufficient condition for optimum $(x^*,y^*,v^*,\nu^{\pm *})$ of primal problem (1)-(4) and its dual problem (10)-(12), is

[27]

Primal feasibility:

$$Jx^* - d = 0, \quad (13)$$

$$-Ax^* + b \geqslant 0, \quad (14)$$

$$\xi^- \leqslant x^* \leqslant \xi^+;$$

Dual feasibility:

$$Wx + q - J^T u + A^T v - \nu^- + \nu^+ = 0, \quad (15)$$

$$u \text{ unrestricted},\ v \geqslant 0,\ \nu^- \geqslant 0,\ \nu^+ \geqslant 0;$$

Complementarity:

$$v^{*T}(-Ax^* + b) = 0, \quad (16)$$

$$\nu^{-*T}(-x^* + \xi^-) = 0, \quad (17)$$

$$\nu^{+*T}(-\xi^+ + x^*) = 0. \quad (18)$$

To simplify the above necessary and sufficient formulation, we further study dual variable vectors ν^{-*} and ν^{+*} in eqns. (15), (17) and (18), which correspond to bound constraint (4). It follows from (17) and (18) that [11][13][28][29]

$$\begin{cases} x_i^* = \xi_i^+ & \text{iff } \nu_i^{+*} > 0, \nu_i^{-*} = 0, \\ \xi_i^- < x_i^* < \xi_i^+ & \text{iff } \nu_i^{+*} = 0, \nu_i^{-*} = 0, \\ x_i^* = \xi_i^- & \text{iff } \nu_i^{+*} = 0, \nu_i^{-*} > 0. \end{cases}$$

By defining $\nu^* = \nu^{-*} - \nu^{+*}$, dual feasibility constraint (15) becomes

$$Wx^* + q - J^T u^* + A^T v^* = \nu^* \begin{cases} \leqslant 0, & x_i^* = \xi_i^+ \\ = 0, & x_i^* \in (\xi_i^-, \xi_i^+) \\ \geqslant 0, & x_i^* = \xi_i^- \end{cases}$$

which equals the following linear variational inequality [23][30]-[32]: to find an $x^* \in \Omega_1$ such that $\forall x \in \Omega_1$,

$$(x - x^*)^T (Wx^* + q - J^T u^* + A^T v^*) \geqslant 0, \quad (19)$$

where $\Omega_1 := \{x | \xi^- \leqslant x^* \leqslant \xi^+\}$. Similarly, defining $\Omega_2 := \{v | v \geqslant 0\}$, we have the following LVI for (14) and (16): to find a $v^* \in \Omega_2$ such that

$$(v - v^*)^T (-Ax^* + b) \geqslant 0,\ \forall v \in \Omega_2; \quad (20)$$

and equality constraint (13) is equivalent to the following LVI: to find a $u^* \in \Omega_3 := \{u | u \in R^m\}$ such that

$$(u - u^*)^T (Jx^* - d) \geqslant 0,\ \forall u \in \Omega_3. \quad (21)$$

Define $\Omega = \Omega_1 \times \Omega_2 \times \Omega_3 = \{y := (x^T, u^T, v^T)^T \in R^{n+m+\dim(b)} | \xi^- \leqslant x^* \leqslant \xi^+, u \text{ unrestricted}, v \geqslant 0,\}$ [30][33]. Linear variational inequalities (19)-(21) can be combined into one LVI problem formulation; i.e., to find $u^* \in \Omega$ such that $\forall y := [x^T, u^T, v^T]^T \in \Omega$,

$$\left(\begin{bmatrix} x \\ u \\ v \end{bmatrix} - \begin{bmatrix} x^* \\ u^* \\ v^* \end{bmatrix} \right)^T \left(\begin{bmatrix} W & -J^T & A^T \\ J & 0 & 0 \\ -A & 0 & 0 \end{bmatrix} \begin{bmatrix} x^* \\ u^* \\ v^* \end{bmatrix} + \begin{bmatrix} q \\ -d \\ b \end{bmatrix} \right) \geqslant 0.$$

After defining $\varsigma^{\pm}$, H and p respectively as in (6) and (7) for notation and implementation simplicity, the above LVI is exactly in the same compact matrix form as in (5), being the equivalence of QP (1)-(4). □

Theorem 2 (PDNN convergence) Starting from any initial state, the state vector $y(t)$ of the primal-dual neural network (9) is convergent to an equilibrium point y^*, of which the first n elements constitute the optimal solution x^* to the QP problem (1)-(4). Moreover, the exponential convergence can be achieved, provided that there exists a constant $\varrho > 0$ such that

$$\|y - \mathcal{P}_\Omega(y - (Hy + p))\|_2^2 \geqslant \varrho \|y - y^*\|_2^2$$

Proof. The proof can be generalized from [12][13][23]. To show the convergence, the finally useful inequality (25) needs to be obtained through (22) and (23).

First, it follows from the projection inequality $(\mathcal{P}_\Omega(\omega) - \varpi)^T (\omega - \mathcal{P}_\Omega(\omega)) \geqslant 0$, $\forall \omega \in R^{n+m+dim(b)}, \varpi \in \Omega$ [12][23][29] that

$$(\mathcal{P}_\Omega(y - (Hy + p)) - y^*)^T \times \\ (y - (Hy + p) - \mathcal{P}_\Omega(y - (Hy + p))) \geqslant 0$$

or written as follows for consistency,

$$(y^* - \mathcal{P}_\Omega(y - (Hy + p)))^T \times \\ (Hy + p - y + \mathcal{P}_\Omega(y - (Hy + p))) \geqslant 0. \quad (22)$$

Second, it follows from the projection-equation formulation of linear variational inequalities, (8), that

$$(y^* - \mathcal{P}_\Omega(y - (Hy + p)))^T (-Hy^* + p) \geqslant 0. \quad (23)$$

Summing up (22) and (23) yields

$$(y^* - \mathcal{P}_\Omega(y - (Hy + p)))^T \times \\ (H(y - y^*) - y + \mathcal{P}_\Omega(y - (Hy + p))) \geqslant 0;$$

and

$$(y^* - y + y - \mathcal{P}_\Omega(y - (Hy + p)))^T \times \\ (H(y - y^*) - y + \mathcal{P}_\Omega(y - (Hy + p))) \geqslant 0. \quad (24)$$

Then extending (24) gives

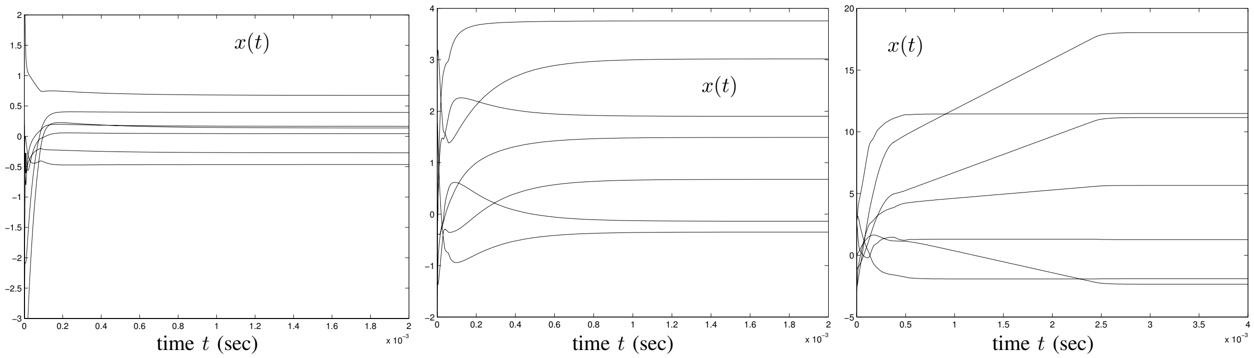
$$(y - y^*)^T (I + H^T)(y - \mathcal{P}_\Omega(y - (Hy + p))) \\ \geqslant \|y - \mathcal{P}_\Omega(y - (Hy + p))\|_2^2 + (y - y^*)^T H (y - y^*).$$

Noting that H is positive semidefinite (not necessarily symmetric) [23][33] in terms of

$$y^T H y = y^T \frac{H + H^T}{2} y = y^T \begin{bmatrix} W & 0 \\ 0 & 0 \end{bmatrix} y \geqslant 0,$$

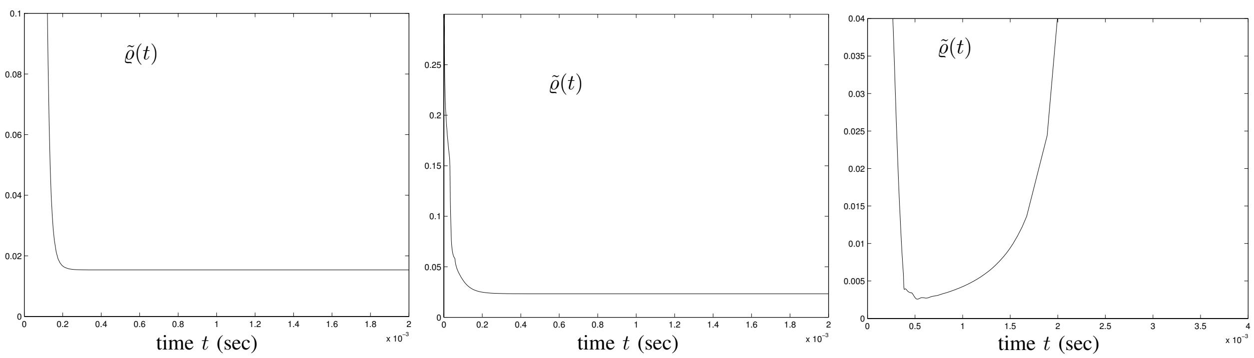
we have

$$(y - y^*)^T (I + H^T)(y - \mathcal{P}_\Omega(y - (Hy + p))) \\ \geqslant \|y - \mathcal{P}_\Omega(y - (Hy + p))\|_2^2 + \|y - y^*\|_H^2 \geqslant 0. \quad (25)$$



(a) QP, positive definite (b) QP, semi-definite (c) LP

Fig. 2. Global and exponential convergence of PDNN (9) to x^* starting from random initial states



(a) QP, positive definite (b) QP, semi-definite (c) LP

Fig. 3. Exponential-convergence factor $\tilde{\varrho}(t)$ of PDNN (9) starting from random initial states

Define the Lyapunov function $V(y) = \|y - y^*\|_2^2 \geqslant 0$. Its time derivative along the primal-dual neural network trajectory (9) is

$$
\begin{aligned}
&\frac{dV(y)}{dt} = \left(\frac{\partial V(y)}{\partial y}\right)^T \frac{dy}{dt} \qquad (26)\\
&= (y-y^*)^T \gamma (I + H^T)(\mathcal{P}_\Omega(y - (Hy+p)) - y)\\
&= -\gamma (y-y^*)^T (I+H^T)(y - \mathcal{P}_\Omega(y-(Hy+p)))\\
&\leqslant -\gamma \|y - \mathcal{P}_\Omega(y-(Hy+p))\|_2^2 - \gamma \|y-y^*\|_H^2\\
&\leqslant 0
\end{aligned}
$$

By Lyapunov theory, the network state $y(t)$ is stable and globally convergent to an equilibrium y^* in view of $\dot{V} = 0$ being $\dot{y} = 0$ and $y = y^*$. It follows from Theorem 1 and Eq. (8) that y^* is the solution to the linear variational inequality problem (5) and the first n elements of y^* constitute the optimal solution x^* to quadratic programming (1)-(4).

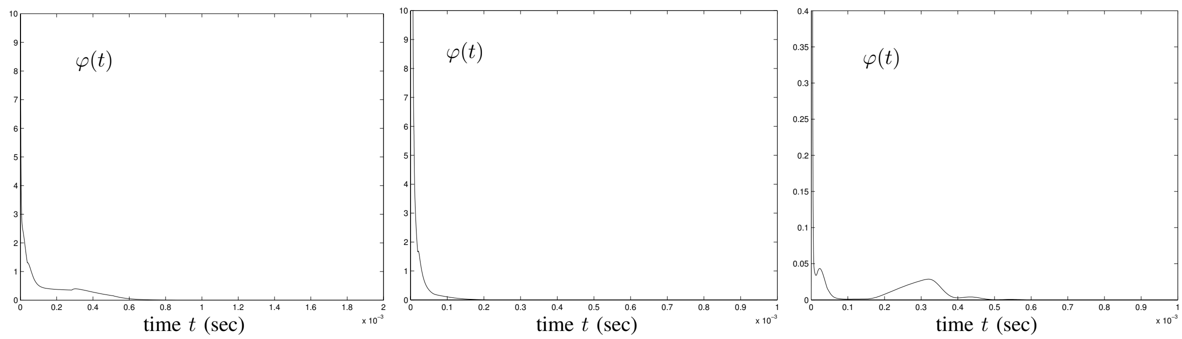
As for the exponential convergence, review $V(y)$ and $\dot{V}(y)$. From (26) and the extra condition called exponential-convergence condition (i.e., if there exists a $\varrho > 0$ such that $\|y - \mathcal{P}_\Omega(y-(Hy+p))\|_2^2 \geqslant \varrho \|y-y^*\|_2^2$), we have

$$
\begin{aligned}
\frac{dV(y)}{dt} &\leqslant -\gamma \|y - \mathcal{P}_\Omega(y-(Hy+p))\|_2^2 - \gamma \|y-y^*\|_H^2\\
&\leqslant -\gamma\varrho \|y-y^*\|_2^2 - \gamma \|y-y^*\|_H^2\\
&= -\gamma (y-y^*)^T (\varrho I + H)(y-y^*)\\
&\leqslant -\lambda V(y)
\end{aligned}
$$

where $\lambda = \varrho\gamma > 0$ is the convergence rate. Note that the existence of the exponential-convergence condition can be justified in practice by considering the equivalence of $y - \mathcal{P}_\Omega(y-(Hy+p)) = 0$ and $y = y^*$, and that it is analyzed in [3][12][14][34] and the references therein. Thus we have $V(y) = O(e^{-\lambda(t-t_0)})$, $\forall t \geqslant t_0$, and hence $\|y-y^*\|_2 = O(e^{-\lambda(t-t_0)/2})$, $\forall t \geqslant t_0$, which completes the exponential convergence property of this primal-dual network. □

IV. Numerical Studies

Theoretical results about primal-dual neural network (9) depicted in the previous section are substantiated and supplemented by the following numerical observations. The first subsection is about the global convergence and exponential convergence. The second subsection is about the



(a) QP, positive definite (b) QP, semi-definite (c) LP

Fig. 4. The solution procedure may shift out of the feasible region, though started from it.

solution behavior starting from initial states within feasible region. The third subsection is about the convergence of the LVI-based PDNN under the circumstance of no x^* solutions. The last subsection is to show the incapability of a model simplified from this LVI-based primal-dual neural approach.

A. Global Convergence

The PDNN model (9) for solving QP/LP (1)-(4) is simulated with randomly generated coefficient matrices and vectors, randomly generated initial states $y(0)$, and without loss of generality, $n = 7$, $m = 3$, $\dim(b) = 1$, $\gamma = 10^4$. Assuming the existence of x^*, the global convergence of primal-dual network (9) is illustrated in Fig. 2 where the usual convergence time is less than 4×10^{-3} second. As shown in Fig. 3, the exponential-convergence factor $\tilde{\varrho}$ is estimated online as $\|y - \mathcal{P}_\Omega(y - (Hy + p))\|_2^2/\|y - y^*\|_2^2$, $\forall y \neq y^*$, and otherwise, $\lim_{y \to y^*} \|y - \mathcal{P}_\Omega(y - (Hy + p))\|_2^2/\|y - y^*\|_2^2$. The value of ϱ in the proof of Theorem 2 can thus be chosen as $\min_t(\tilde{\varrho}(t))$. This typical simulation result justifies the exponential convergence property of the LVI-based PDNN model. Furthermore, the x^* values are also compared with those by MATLAB QP routines: the difference is less than 10^{-7}.

B. Feasible-Region Solutions

As in the robotic applications, the initial state $x(0)$ (being the initial joint velocity, acceleration or torque variables) is usually within the feasible region constituted by constraints (1)-(4). Thus, we may be interested in the question whether the network output $x(t)$ starting from such an $x(0)$ will always be within the same feasible region. If not, what measurements could we adopt? Fig. 4 shows the typical situation where only $\varphi = 0$ corresponds to within the feasible region. That is, $x(t)$ sometimes does go out of the region. In light of the exponential convergence, this weakness can be remedied easily by increasing design parameter γ as large as hardware permits (like, $\gamma = 10^7$, such that the shifting period is only of level 10^{-7}sec), or using a limiter to force the network output $x(t)$ between $[\xi^-, \ \xi^+]$. This numerical result also applies to other neural networks design for robot manipulators and might be the source of the small end-effector positioning error of level $10^{-4} \sim 10^{-7}$m in [4][11]-[16][19]-[22][26][34].

C. No-Solution Case

Another interesting topic of using the PDNN (9) is about the case of no optimal/theoretical solutions x^* to the original QP/LP problem (1)-(4). If such, what convergence behavior will the network illustrate? Nonexistence of x^* actually means the feasible region being empty. In robotics, this is an extreme case, but does happen, like to command the robot arm to do an impossible task of positioning or lifting. Though theoretical result does not cover this case, a large number of random numerical tests shows that the network output $x(t)$ is always convergent (while the dual variables u and v are not). Due to space limitation, there is no figures to show. The typical $x(t)$ convergence is similar to Fig. 2. This numerical result is also applicable to other neural networks design for robot manipulators, and might explain why no divergence of joint variables was observed even in the extreme case (instead, a jump or loss of integration precision may occur) [4][11]-[16][19]-[22][26][34].

D. Incapacity of A Simplified Model

Review eq. (8). Based on the dual neural network design experience [3][11]-[15][22][34], we may hope to further simplify the PDNN dynamic equation (9) as the following by removing term $(I+H^T)$: $\dot{y} = \gamma\{\mathcal{P}_\Omega(y-(Hy+p))-y\}$. However, due to the asymmetry of H, the above simplified model may not work well. For example, for solving the LP problem with $W = 0$, the simplified model is typically oscillating and not convergent to any x^*, as shown in Fig. 5. Failure is also possible in using this simplified model to solve QP problems. That means the simplified model has lost the capability of handling QP and LP effectively and simultaneously.

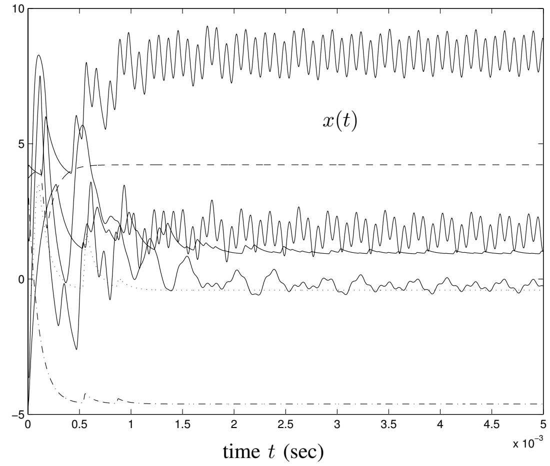


Fig. 5. The incapability of a simplified model for solving (1)-(4).

V. Conclusions

Robot motion planning and control have motivated the online solution to general QP/LP problems [4]. In this paper, the LVI-based primal-dual neural network has been investigated thoroughly for such an online solution. The network is with piecewise linear dynamics, global exponential convergence, and handling QP/LP at the same time. The numerical results and discussion have provided some insights and answers to related robotic problems [20][34].

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 Transaction on Circuits and Systems,* vol. 33, pp. 533-541, 1986.

[2] A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Precessing*, U. K.: Wiley, London, England, 1993.

[3] 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.

[4] Y. Zhang, S. S. Ge, and T. H. Lee, "A unified quadratic programming based dynamical system approach to joint torque optimization of physically constrained redundant manipulators," *IEEE Transactions on Systems, Man, and Cybernetics, Part B,* to appear, 2004.

[5] I. B. Pyne, "Linear programming on an electronic analogue computer," *Trans. Amer. Inst. Elec. Eng.,* vol. 75, pp. 139-143, 1956.

[6] C. Mead, *Analog VLSI and Neural Systems*, Reading, Mass.: Addison-Wesley, 1989.

[7] F. L. Luo and B. Zheng, "Neural network approach to computing matrix inversion," *Applied Mathematics and Computation*, vol. 47, no. 2-3, pp. 109-120, 1992.

[8] N. M. Botros and M. Abdul-Aziz, "Hardware implementation of an artificial neural network using field programmable gate arrays (FPGAs)," *IEEE Transactions on Industrial Electronics*, vol. 41, no. 6, pp. 665-667, 1994.

[9] J. Song and Y. Yam, "Complex recurrent neural network for computing the inverse and pseudo-inverse of the complex matrix," *Appiled Mathematics and Computation*, vol. 93, no. 2-3, pp. 195-205, 1998.

[10] C. Diorio and R. P. N. Rao, "Neural circuits in silicon," *Nature*, vol. 405, no. 6789, pp. 891-892, 2000.

[11] Y. Zhang and J. Wang, "A dual neural network for constrainted torque optimization of kinematically redundant manipulators," *IEEE Transactions on Systems, Man and Cybernetics Part B,* vol. 32, no. 5, pp. 654-662, 2002.

[12] Y. Zhang, J. Wang, and Y. Xu, "A dual neural network for bi-criteria kinematic control of redundant manipulators," *IEEE Transactions on Robotics and Automation,* vol. 18, no. 6, pp. 923-931, 2002.

[13] Y. Zhang, J. Wang, and Y. Xia, "A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits," *IEEE Transactions on Neural Networks,* vol. 14, no. 3, pp. 658-667, 2003.

[14] Y. Zhang and J. Wang, "Obstacle avoidance of kinematically redundant manipulators using a dual neural network," *IEEE Transactions on Systems, Man, and Cybernetics, Part B,* vol. 34, no. 1, pp. 752-759, 2004.

[15] J. Wang and Y. Zhang, "Recurrent neural networks for real-time computation of inverse kinematics of redundant manipulators," *Machine Intelligence: Quo Vadis?* P. Sincak, J. Vascak, and K. Hirota (eds.), World Scientific: Singapore, pp. 299-319, 2004.

[16] S. S. Ge, Y. Zhang, and T. H. Lee, "Acceleration-based weighting scheme for minimum-effort manipulator inverse kinematics," *IEEE International Symposium on Intelligent Control,* Taiwan, September 2004.

[17] M. P. Kennedy and L. O. Chua, "Neural networks for nonlinear programming," *IEEE Transactions on Circuits and Systems,* vol. 35, no. 5, pp. 554-562, 1988.

[18] S. Zhang and A. G. Constantinides, "Lagrange Programming Neural Networks," *IEEE Transactions on Circuits and Systems,* vol. 39, no. 7, pp. 441-452, 1992.

[19] W. S. Tang and J. Wang, "A recurrent neural network for minimum infinity-norm kinematic control of redundant manipulators with an improved problem formulation and reduced architecture complexity," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 31, no. 1, pp. 98-105, 2001.

[20] S. S. Ge, Y. Zhang, and T. H. Lee, "On the weighting scheme for minimum-effort inverse kinematics of redundant manipulators," *IEEE Transactions on Robotics and Automation,* under review since May 2003.

[21] F.-T. Cheng, T.-H. Chen, and Y.-Y. Sun, "Resolving manipulator redundancy under inequality constraints," *IEEE Journal of Robotics and Automation,* vol. 10, no. 1, pp. 65-71, 1994.

[22] Y. Zhang and J. Wang, "Obstacle avoidance of redundant manipulators using a dual neural network," *Proceedings of the 2003 IEEE International Conference on Robotics and Automation,* vol. 2, pp. 2747-2752, 2003.

[23] Y. Xia and J. Wang, "A recurrent neural network for solving linear projection equations," *Neural Networks*, vol. 13, no. 3, pp. 337-350, 2000.

[24] Y. Zhang and J. Wang, "Global exponential stability of recurrent neural networks for synthesizing linear feedback control systems via pole assignment," *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 633-644, 2002.

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

[26] Y. Zhang and S. S. Ge, "A general recurrent neural network model for time-varying matrix inversion," *Proceedings of the IEEE 42nd Conference on Decision and Control,* pp. 6169-6174, 2003.

[27] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming – Theory and Algorithms.* New York: Wiley, 1993.

[28] W. Li and J. Swetits, "A new algorithm for solving strictly convex quadratic programs," *SIAM Journal of Optimization,* vol. 7, no. 3, pp. 595-619, 1997.

[29] D. P. Bertsekas, *Parallel and Distributed Computation: Numerical Methods.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[30] D. Kinderlehrer and G. Stampcchia, *An Introduction to Variational Inequalities and Their Applications.* New York: Academic Press, 1980.

[31] M. C. Ferris and J.-S. Pang, *Complementarity and Variational Problems: State of The Art.* Philadelphia: SIAM, 1997.

[32] B. He and H. Yang, "A neural network model for monotone linear asymmetric variational inequalities," *IEEE Trans. Neural Net.,* vol. 11, no. 1, pp. 3-16, 2000.

[33] B. He, "A projection and contraction method for a class of linear complementarity problems and its application in conves quadratic programming," Appl. Math. Optimiz., vol. 25, pp. 247-262, 1992.

[34] Y. Zhang, *Analysis and Design of Recurrent Neural Networks and Their Applications to Control and Robotic Systems*, Ph.D. Thesis, Department of Automation & Computer-Aided Enginering, Chinese University of Hong Kong, 2002.