

Correspondence

A Dual Neural Network for Constrained Joint Torque Optimization of Kinematically Redundant Manipulators

Yunong Zhang and Jun Wang

Abstract—A dual neural network is presented for the real-time joint torque optimization of kinematically redundant manipulators, which corresponds to global kinetic energy minimization of robot mechanisms. Compared to other computational strategies on inverse kinematics, the dual network is developed at the acceleration level to resolve redundancy of limited-joint-range manipulators. The dual network has a simple architecture with only one layer of neurons and is proved to be globally exponentially convergent to optimal solutions. The dual neural network is simulated with the PUMA 560 robot arm to demonstrate effectiveness.

Index Terms—Dual neural network, joint limits, joint torque optimization, redundant manipulators.

I. INTRODUCTION

Kinematically redundant manipulators are those having more degrees of freedom (DOF) than required to perform a given task [1]. The redundancy of such manipulators including intrinsic redundancy and functional redundancy can be utilized to avoid obstacles, joint limits, and singularities, and to optimize various performance criteria [2]–[5], while conducting the end-effector motion task. One example of the performance criteria to be optimized is a norm of joint torques [6], [9], [10], which is aimed at making a more effective utilization of input power from actuators by exploiting the extra DOF in redundant manipulators.

Many efforts have been made in the redundancy resolution of manipulators through joint torque optimization. An initial study was conducted by Hollerbach and Suh [5] who presented the null-space (NS) algorithm that instantaneously minimizes the joint torque using the NS of Jacobian such that all the joint torques are placed closest to the mid-points between the upper and lower joint torque limits. Kazerounian and Nedungadi [11] derived a local joint torque minimizing pseudoinverse instead of using the NS of the Jacobian. Both approaches, however, were found to exhibit instabilities for long-range motions. Since then, many researchers attempted to formulate other joint torque optimization schemes to eliminate the instability problem. For example, Suh and Hollerbach [12] developed a global torque optimization technique using the calculus of variations, which results in globally stable optimal solutions. However, since the global optimization technique requires complete trajectory information in advance and is very computationally intensive, it is not well suited for real-time motion control usually requiring continuous trajectory modification based on sensory feedback. Kang and Freeman [7] presented the NS damped joint torque minimization method in which the local joint torques are stabilized by using the damping forces generated from appropriate NS. Ma [8] proposed a balancing scheme in which the solution of local joint

torque minimization is balanced against the solution of joint velocity minimization through a weighting factor. Unfortunately, a systematic and efficient procedure to select the damping gain or the weighting factor was not found and such damping/balancing schemes (via trial and error) entails extra computation. Nedungadi and Kazerounian [6] presented an approach that locally minimizes joint torque weighted by the inverse of inertia matrix, which corresponds to global kinetic energy minimization, and the local solutions are thus optimal and internally stable.

In recent years, neural network approaches have been developed for the redundancy resolution of robot manipulators [2], [13]–[16]. In particular, Ding and Tso [14] incorporated the Tank–Hopfield neural network into the NS algorithm for a redundancy solution which locally minimizes joint torques in the Chebyshev sense. Tang and Wang [10], [15] proposed two neural networks, namely, the Lagrangian network and the primal-dual network, to explicitly minimize the joint torque weighted by inertia inverse via torque-based formulation.

Among the aforementioned torque-minimization computational strategies (including the modified NS, pseudoinverse numerical algorithms, and the neural network-based schemes), it is always assumed implicitly that there exists no joint limit when solving such inverse kinematic problems. However, joint limits are physical constraints of the work space of a robot and they do exist for all kinds of robot. If a solution exceeds the mechanical joint rotation limits, the desired path becomes impossible to follow and the solution is then inapplicable.

To reduce scheme complexity and increase computational efficiency, Xia and Wang [16] proposed a dual neural network capable of solving equality-constrained optimization problems and then applied to the inverse kinematics problem at the velocity level. In this correspondence, as an extension of [16], our aim is to extend the dual neural network model to solve equality and bound-constrained optimization problems and then apply it to minimize the joint torque instantaneously at the acceleration level with joint limits considered. The proposed dual neural network model is simulated on the limited-joint-range 6-DOF PUMA 560 robot arm with effectiveness shown.

II. PROBLEM FORMULATION

Consider a forward kinematics equation

$$r = f(\theta) \quad (1)$$

where

- $r \in R^m$ position and orientation vector of the end-effector in the Cartesian space;
- $\theta \in R^n$ joint variable vector ($m < n$ in redundant manipulators);
- $f(\cdot)$ continuous nonlinear mapping function with a known structure and parameters for a given manipulator.

Differentiating (1) with respect to time gives the linear relation between the Cartesian velocity \dot{r} and joint velocity $\dot{\theta}$

$$\dot{r} = J(\theta)\dot{\theta} \quad (2)$$

where $J(\theta) \in R^{m \times n}$ is the Jacobian matrix defined as $J(\theta) = \partial f(\theta)/\partial \theta$. Differentiating (2) with respect to time yields the relation between the joint acceleration $\ddot{\theta}$ and Cartesian acceleration \ddot{r}

$$J(\theta)\ddot{\theta} = \ddot{r} - \dot{J}(\theta)\dot{\theta} \quad (3)$$

Manuscript received July 10, 2001; revised March 14, 2002. This study was supported by the Hong Kong Research Grants Council under Grant CUHK4165/98E. This paper was recommended by Associate Editor M. S. de Queiroz.

The authors are with the Department of Automation and Computer-Aided Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong (e-mail: ynzhang@acae.cuhk.edu.hk, jwang@acae.cuhk.edu.hk).

Publisher Item Identifier S 1083-4419(02)04835-5.

where $\dot{J}(\theta) \in R^{m \times n}$ is the time derivative of the Jacobian matrix. In a redundant manipulator, (2) and (3) are underdetermined since $m < n$ and hence they may admit an infinite number of solutions.

It is well known [5] that the joint torque τ can be expressed as

$$\tau = H(\theta)\ddot{\theta} + c(\theta, \dot{\theta}) + g(\theta) \quad (4)$$

where

- $H(\theta) \in R^{n \times n}$ symmetric positive definite inertia matrix;
- $c(\theta, \dot{\theta}) \in R^n$ component of the torque depending on Coriolis centrifugal forces;
- $g(\theta) \in R^n$ component depending on gravity forces.

As shown in [6] by using calculus of variations, the local optimization of joint torque weighted by inertia inverse results in resolutions with global characteristics; that is, the solution to the local torque minimization problem

$$\begin{aligned} \min \quad & \frac{1}{2} \tau^T H^{-1} \tau \\ \text{s.t.} \quad & J(\theta)\ddot{\theta} + \dot{J}(\theta)\dot{\theta} - \ddot{r} = 0 \end{aligned} \quad (5)$$

also expresses the governing solution equation to the ensuing global optimization problem of kinetic energy

$$\begin{aligned} \min \quad & \frac{1}{2} \int_{t_0}^{t_f} \dot{\theta}^T H \dot{\theta} \\ \text{s.t.} \quad & G_k(\theta, t) = 0, \quad k = 1, \dots, m \end{aligned}$$

where

- t_0 beginning time (usually zero);
- t_f final time of the given end-effector task;
- G_k kinematic constraint on the end-effector, e.g., (2).

Substituting (4) into (5) results in the local objective function at acceleration level

$$\frac{1}{2} \tau^T H^{-1} \tau = \frac{1}{2} \ddot{\theta}^T H \ddot{\theta} + b^T(\theta, \dot{\theta}) \ddot{\theta} + p(\theta, \dot{\theta}) \quad (6)$$

where $b(\theta, \dot{\theta}) := c(\theta, \dot{\theta}) + g(\theta)$ and $p(\theta, \dot{\theta}) := b^T(\theta, \dot{\theta}) H^{-1} b(\theta, \dot{\theta}) / 2$. Moreover, in robotics, compared to the physical limits on joint velocity or acceleration, joint limits are more crucial and significant, since velocity and acceleration limits can be generally escaped by increasing the task time (i.e., $t_f - t_0$), but joint limits cannot be avoided by such a simple scheme. Thus, with joint limits $[\theta^-, \theta^+]$ considered and defining $\ddot{r}_d := \ddot{r} - \dot{J}\dot{\theta}$, the torque optimization problem (5) can be rewritten as

$$\begin{aligned} \min \quad & \frac{1}{2} \ddot{\theta}^T H \ddot{\theta} + b^T \ddot{\theta} \\ \text{s.t.} \quad & J(\theta)\ddot{\theta} = \ddot{r}_d, \\ & \theta^- \leq \theta(t) \leq \theta^+. \end{aligned} \quad (7)$$

Different from other reports on torque optimization, (7) resolves redundancy at acceleration level and simultaneously takes into account the feature of joint limit avoidance. In the next section, our attention will focus on the conversion of (7) to a time-varying quadratic programming problem in terms of $\ddot{\theta}$ and then the development of a general dual neural network model for solving the problem.

III. A DUAL NEURAL NETWORK

Before developing a dual neural network, the limited-joint-range $[\theta^-, \theta^+]$ in (7) has to be converted to some kind of dynamically-updated bound constraints on joint acceleration variables. The proposed effective conversion scheme is to replace $\theta^- \leq \theta(t) \leq \theta^+$ with

$$\xi^-(t) \leq \ddot{\theta}(t) \leq \xi^+(t) \quad (8)$$

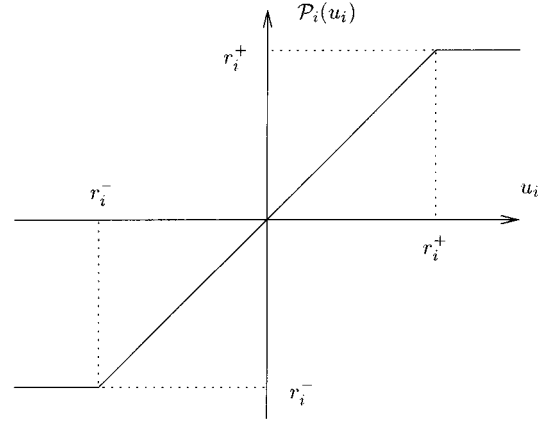


Fig. 1. Projection operator $\mathcal{P}_i(u_i)$ onto Ω_i .

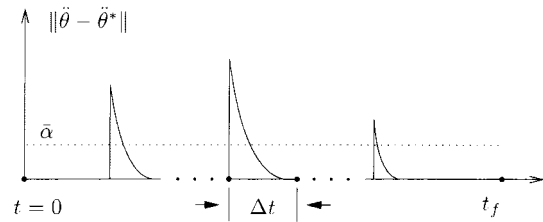


Fig. 2. Average of $\|\ddot{\theta}(t) - \ddot{\theta}^*(t)\|$ can be approximated by $\bar{\alpha}$.

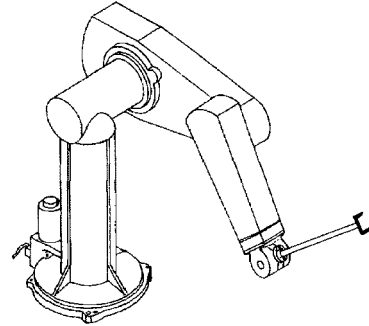


Fig. 3. PUMA 560 robot arm with a long tool.

TABLE I
PUMA 560 DH PARAMETERS AND JOINT LIMITS

#	a	d	α	θ^-	θ^+
1	0	0	$\pi/2$	-2.775	2.775
2	0.4318	0	0	-3.892	0.750
3	0.0203	0.15005	$-\pi/2$	-0.905	4.049
4	0	0.4318	$\pi/2$	-1.919	2.967
5	0	0	$-\pi/2$	-1.745	1.745
6	0	0.25625	0	-4.625	4.625

where $\xi^-(t) := \mu(\eta\theta^- - \theta(t))$, $\xi^+(t) := \mu(\eta\theta^+ - \theta(t))$, the critical coefficient $0 < \eta < 1$ is selected such that there appears a deceleration when the robot arm enters the critical areas $[\theta^-, \eta\theta^-]$ or $[\eta\theta^+, \theta^+]$, and the intensity coefficient $\mu > 0$ determines the deceleration magnitude when the robot is entering such critical areas. In the ensuing PUMA 560 simulations of Section V, η is usually selected to be 0.9 and μ is limited by bounds on joint accelerations or actuator torques and usually selected to be 20. The repelling mechanism at the joint limits is based on the existence of critical areas and the dynamical updating of joint acceleration bound constraint. For any $i \in \{1, \dots, n\}$, if the i th joint variable $\theta_i(t)$ is within its feasible range $(\eta\theta_i^-, \eta\theta_i^+)$, then

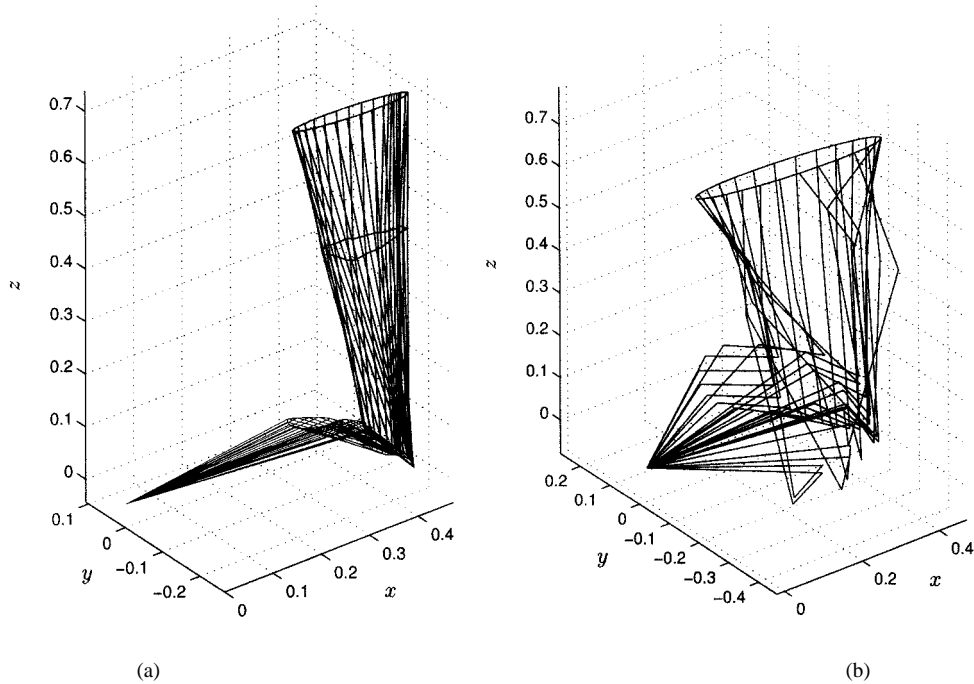


Fig. 4. Motion trajectories of the PUMA 560 manipulator tracking circles: (a) 10-cm circle. (b) 20-cm circle.

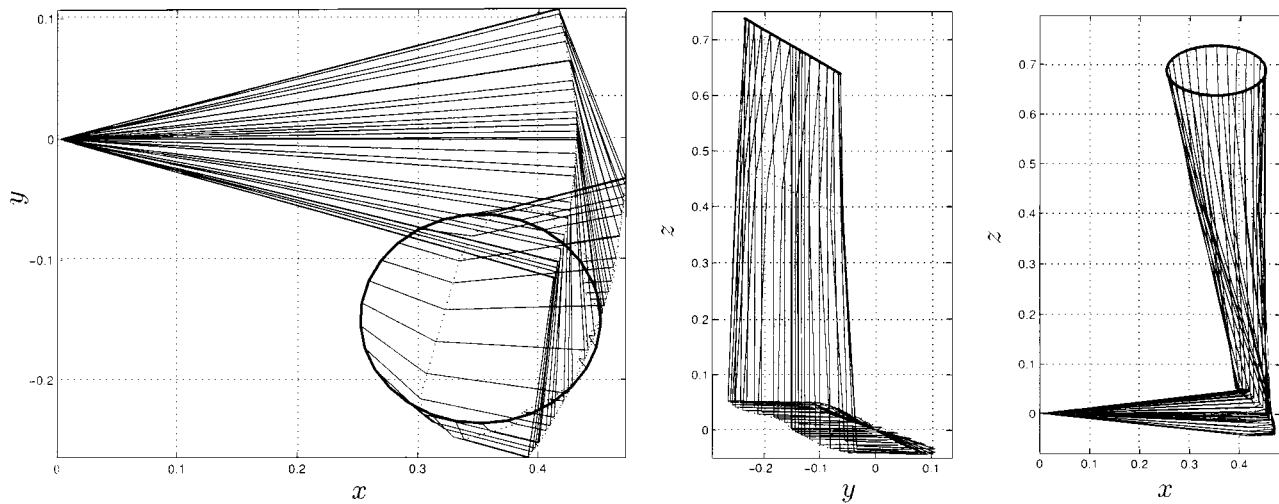


Fig. 5. Projection trajectories of the PUMA 560 manipulator tracking a 10-cm circle.

$\xi_i^- < 0$, $\xi_i^+ > 0$, and with an appropriate value of μ , it follows from (8) that $\ddot{\theta}_i$ might be a positive or negative solution to (5) if existent. However, if $\theta_i(t)$ enters into its critical area $[\eta\theta_i^+, \theta_i^+]$, then $\xi_i^- < 0$ and $\xi_i^+ \leq 0$; that is, the bound constraint (8) becomes $\ddot{\theta} \leq 0$, which implies an acceleration of opposite direction to the original joint movement and drives the i th joint away from its physical limit. The same is true for the other critical area $[\theta_i^-, \eta\theta_i^-]$.

Now, the real-time torque minimization problem (7) for joint-constrained redundant manipulators can be reformulated as the following time-varying quadratic program:

$$\min \frac{1}{2} \ddot{\theta}^T H \ddot{\theta} + b^T \ddot{\theta} \text{ s.t. } J_E \ddot{\theta} \in \Omega \quad (9)$$

where defining $r^- = [\ddot{r}_d^T, (\xi^-)^T]^T$ and $r^+ = [\ddot{r}_d^T, (\xi^+)^T]^T$, $\Omega := \{u \in R^{n+m} | r^- \leq u \leq r^+\}$, and $J_E := [J^T, I_n^T]^T$ with I_n denoting the $n \times n$ identity matrix.

At any time instant, the above problems may be viewed as parametric optimization problems. Now we consider the equivalent formulation for the optimal solution to (9). By the Karush–Kuhn–Tucker condition [17], we know that $\ddot{\theta}$ is a solution to (9) if and only if there exists the dual decision vector $u \in R^{m+n}$ such that $H\ddot{\theta} - J_E^T u + b = 0$ and

$$\begin{cases} [J_E \ddot{\theta}]_i = r_i^-, & \text{if } u_i > 0 \\ [J_E \ddot{\theta}]_i = r_i^+, & \text{if } u_i < 0 \\ r_i^- \leq [J_E \ddot{\theta}]_i \leq r_i^+, & \text{if } u_i = 0 \end{cases}$$

which is equivalent to the system of piecewise equations $J_E \ddot{\theta} = \mathcal{P}(J_E \ddot{\theta} - u)$ [18]–[20], where the projection operator $\mathcal{P}(u) = [\mathcal{P}_1(u_1), \dots, \mathcal{P}_{m+n}(u_{m+n})]^T$ is illustrated in Fig. 1 and defined as

$$\mathcal{P}_i(u_i) = \begin{cases} r_i^-, & \text{if } u_i < r_i^- \\ u_i, & \text{if } r_i^- \leq u_i \leq r_i^+, i = 1, 2, \dots, m+n \\ r_i^+, & \text{if } u_i > r_i^+ \end{cases} \quad (10)$$

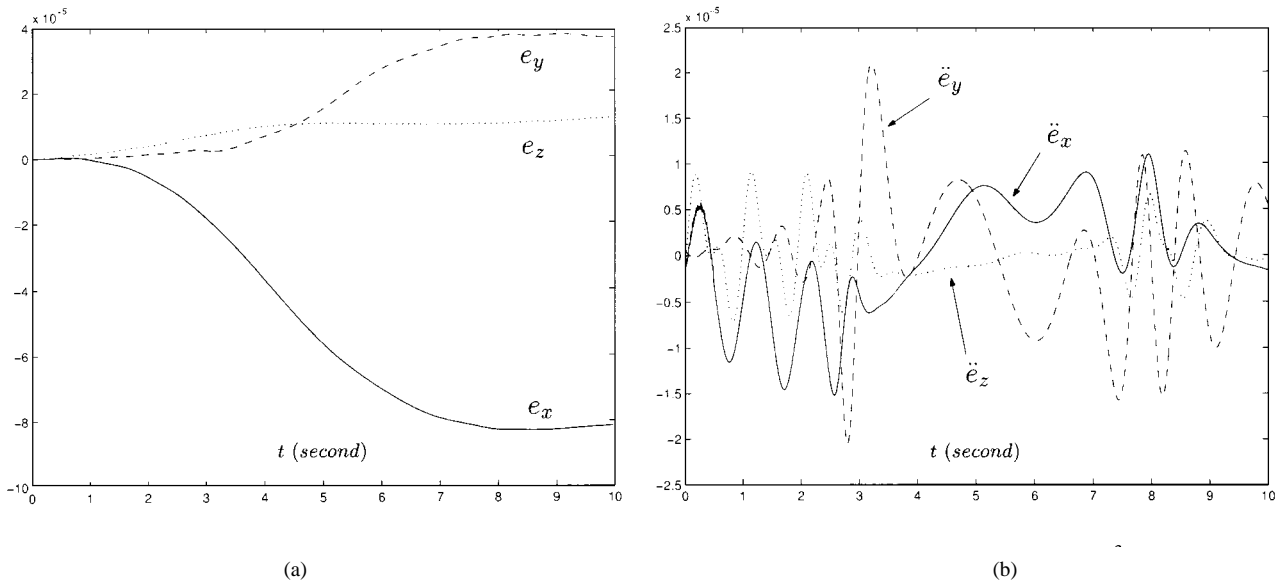


Fig. 6. Cartesian tracking errors of the PUMA 560 manipulator tracking a 10-cm circle. (a) Position error in meters. (b) Acceleration error in m/s^2 .

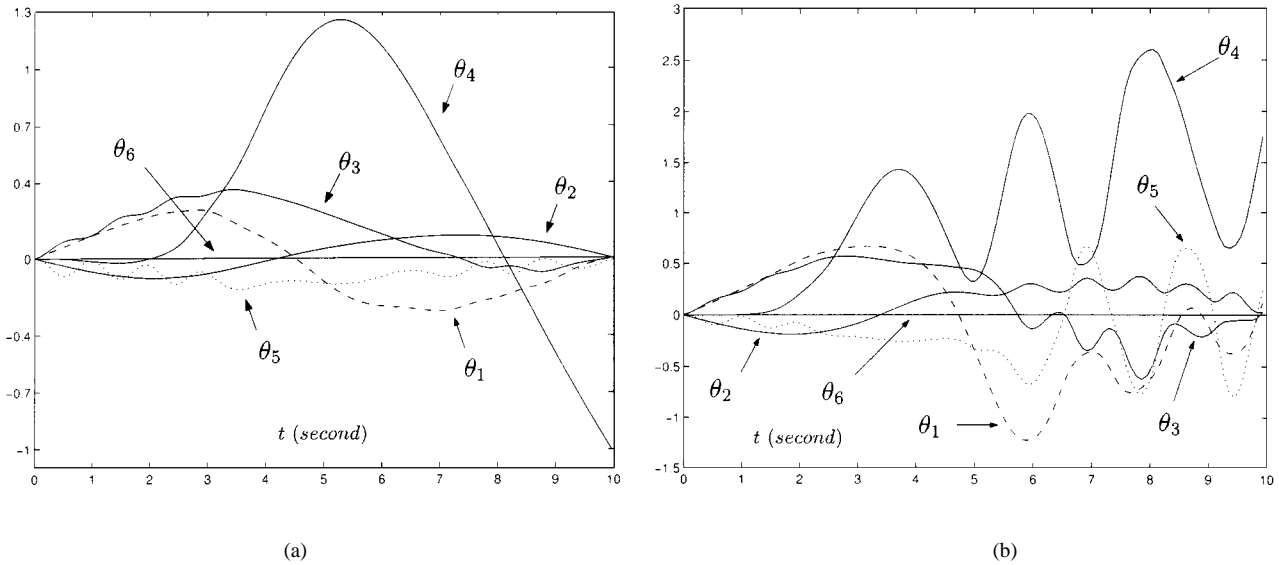


Fig. 7. Joint rotation variables of the PUMA 560 manipulator tracking circles: (a) 10-cm circle. (b) 20-cm circle.

Therefore, $\ddot{\theta}$ is a solution to (9) if and only if there exists $u \in R^{m+n}$ such that $H\ddot{\theta} - J_E^T u + b = 0$ and $J_E \ddot{\theta} = \mathcal{P}(J_E \ddot{\theta} - u)$, that is

$$\begin{cases} \ddot{\theta} = H^{-1} J_E^T u - H^{-1} b \\ \mathcal{P}(J_E H^{-1} J_E^T u - J_E H^{-1} b - u) = J_E H^{-1} J_E^T u - J_E H^{-1} b \end{cases} \quad (11)$$

which gives rise to the proposed dual neural network for solving (9) with the following dynamics:

$$c \dot{u} = \mathcal{P}(J_H J_E^T u - J_H b - u) - (J_H J_E^T u - J_H b) \quad (12)$$

where $J_H := J_E H^{-1}$ and $c \in R$ is a strictly positive capacitive parameter to scale the convergence rate of the proposed network. The dynamic equation described in (12) shows that the dual neural network, even with the extra n joint bound constrained torque minimization problem, is composed of only one layer of $m + n$ neurons and without any multiplier, penalty parameter, or high-order terms. Compared to the dual neural network approach for solving (7), the Lagrange neural network or the prime-dual neural network [10], [13]–[15] has more complicated dynamics and structure in terms of high-order terms and the

number of neurons/layers. In comparison, the dual neural network proposed in [16] handles equality constraint only and solve inverse kinematics problem at velocity level without considering any joint limits.

IV. THEORETICAL RESULTS

In this section, we prove the global convergence of the proposed dual neural network (12) for torque minimization of redundant manipulators.

Some related definitions and a lemma are first presented. A neural network is said to be globally convergent if, starting from any initial point taken in a given domain of the whole associated Euclidean space, every trajectory of the corresponding dynamic system converges to an equilibrium point that depends on the initial state of the trajectory. Furthermore, the system is said to be globally exponentially convergent if every trajectory starting from any initial point $x(t_0)$ in a given domain satisfies $\|x(t) - x^*\| \leq \alpha \|x(t_0) - x^*\| \exp(-\beta(t - t_0))$, $\forall t \geq t_0 \geq 0$ where α and β are strictly positive constant; x^* is an equilibrium point depending on initial states; and $\|\cdot\|$ denotes hereafter the Euclidean

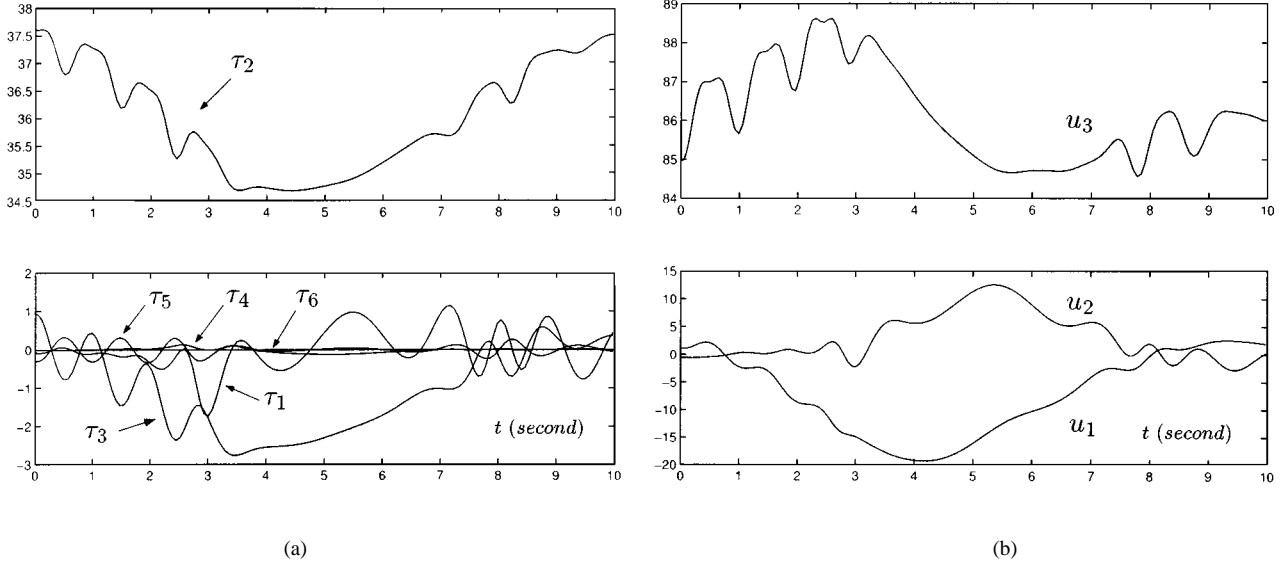


Fig. 8. Joint torques and dual decision variables of the PUMA 560 manipulator tracking a 10-cm circle. (a) Joint torques. (b) Dual variables.

norm of a vector or the Frobenius norm of a matrix. The following projection property is often used in optimization literature [16], [18], [21].

Lemma 1: Assume that the set $\Omega \subset R^m$ is a closed convex set, then the following two inequalities hold:

$$\begin{aligned} (v - P_\Omega(v))^T (P_\Omega(v) - u) &\geq 0, \forall v \in R^m, u \in \Omega \\ \|P_\Omega(u) - P_\Omega(v)\| &\leq \|u - v\|, \forall u, v \in R^m \end{aligned}$$

where $P_\Omega : R^m \rightarrow \Omega$ is a projection operator defined as $P_\Omega(x) = \arg \min_{v \in \Omega} \|x - v\|$.

It is clear that Ω and $\mathcal{P}_i(u_i)$, defined in (9) and (10), respectively, satisfy the above projection property. The convergence results of the proposed dual network model (12) are then obtained as follows.

Theorem 1: Starting from any initial state, the dual neural network (12) is exponentially convergent to an equilibrium point u^* and, together with $\dot{\theta}^* = H^{-1}J_E^T u^* - H^{-1}b$, is an optimal solution ($u^*, \dot{\theta}^*$) of the original minimization problem (9).

Proof: At any equilibrium point u^* of the dual neural network (12), it follows from

$$\mathcal{P} \left(J_H J_E^T u^* - J_H b - u^* \right) - \left(J_H J_E^T u^* - J_H b \right) = 0 \quad (13)$$

and $\ddot{\theta}^* = H^{-1}J_E^T u^* - H^{-1}b$ that $(u^*, \ddot{\theta}^*)$ satisfies the Lagrangian optimality condition (including the Karush–Kuhn–Tucker complementary condition) and thus is an optimal solution to the original minimization problem (9).

At u^* , we have the following inequality property [16], [20]:

$$\left(v - \left(J_H J_E^T u^* - J_H b \right) \right)^T u^* \geq 0, \forall v \in \Omega \quad (14)$$

which can be obtained by discussing the following three cases:

Case 1) If for some $i \in \{1, 2, \dots, m+n\}$, $u_i^* = 0$, $r_j^- \leq [J_E \ddot{\theta}^*]_i \leq r_j^+$, then $(v_i - [J_E \ddot{\theta}^*]_i)u_i^* = 0$.

Case 2) If for some $j \in \{1, 2, \dots, m+n\}$, $u_j^* > 0$, $[J_E \ddot{\theta}^*]_j = r_j^-$ and $r_j^- \leq v_j \leq r_j^+$, then $v_j - [J_E \ddot{\theta}^*]_j \geq 0$ and thus $(v_j - [J_E \ddot{\theta}^*]_j)u_j^* \geq 0$.

Case 3) If for some $k \in \{1, 2, \dots, m+n\}$, $u_k^* < 0$, $[J_E \ddot{\theta}^*]_k = r_k^+$ and $r_k^- \leq v_k \leq r_k^+$, then $v_k - [J_E \ddot{\theta}^*]_k \leq 0$ and thus $(v_k - [J_E \ddot{\theta}^*]_k)u_k^* \geq 0$.

Therefore, it follows from (14) that

$$\left(\mathcal{P} \left(J_H J_E^T u - J_H b - u \right) - \left(J_H J_E^T u^* - J_H b \right) \right)^T u^* \geq 0. \quad (15)$$

On the other hand, from Lemma 1 we have $\forall u \in R^m$

$$\begin{aligned} &\left[\mathcal{P} \left(J_H J_E^T u - J_H b - u \right) - \left(J_H J_E^T u^* - J_H b \right) \right]^T \\ &\cdot \left[\left(J_H J_E^T u - J_H b - u \right) - \mathcal{P} \left(J_H J_E^T u - J_H b - u \right) \right] \geq 0. \quad (16) \end{aligned}$$

Then, adding (15) and (16) yields

$$\begin{aligned} &\left[\mathcal{P} \left(J_H J_E^T u - J_H b - u \right) - \left(J_H J_E^T u^* - J_H b \right) \right]^T \\ &\cdot \left[u^* + \left(J_H J_E^T u - J_H b - u \right) - \mathcal{P} \left(J_H J_E^T u - J_H b - u \right) \right] \geq 0. \quad (17) \end{aligned}$$

Defining $\tilde{\mathcal{P}} := \mathcal{P}(J_H J_E^T u - J_H b - u) - (J_H J_E^T u - J_H b)$, it follows from (17) that

$$\left[\tilde{\mathcal{P}} + J_H J_E^T (u - u^*) \right]^T \left[(u - u^*) + \tilde{\mathcal{P}} \right] \leq 0$$

and thus

$$\begin{aligned} (u - u^*)^T \tilde{\mathcal{P}} + \tilde{\mathcal{P}}^T J_H J_E^T (u - u^*) &\leq -\|\tilde{\mathcal{P}}\|^2 \\ - (u - u^*)^T J_E H^{-1} J_E^T (u - u^*) &\leq 0. \quad (18) \end{aligned}$$

Now we choose a Lyapunov function candidate as

$$V(u(t)) = \frac{1}{2} \|Q(u(t) - u^*)\|^2 \quad (19)$$

where Q is symmetric positive definite and $Q^2 = c(I + J_E H^{-1} J_E^T) > 0$. Clearly, $V(u)$ is positive definite, that is, $V > 0$ if $u \neq u^*$ and $V = 0$ iff $u = u^*$ for u taken in any domain $\Gamma(u^*) \in R^{m+n}$ (i.e., the attraction region of u^*). \dot{V} is negative definite in $\Gamma(u^*)$ since, in view of (18), we have

$$\begin{aligned} \dot{V} &= (u - u^*)^T Q^2 \dot{u} \\ &= (u - u^*) \left(I + J_E H^{-1} J_E^T \right) \tilde{\mathcal{P}} \\ &\leq -\|\mathcal{P} \left(J_H J_E^T u - J_H b - u \right) - \left(J_H J_E^T u - J_H b \right)\|^2 \\ &\quad - (u - u^*)^T J_E H^{-1} J_E^T (u - u^*) \quad (20) \end{aligned}$$

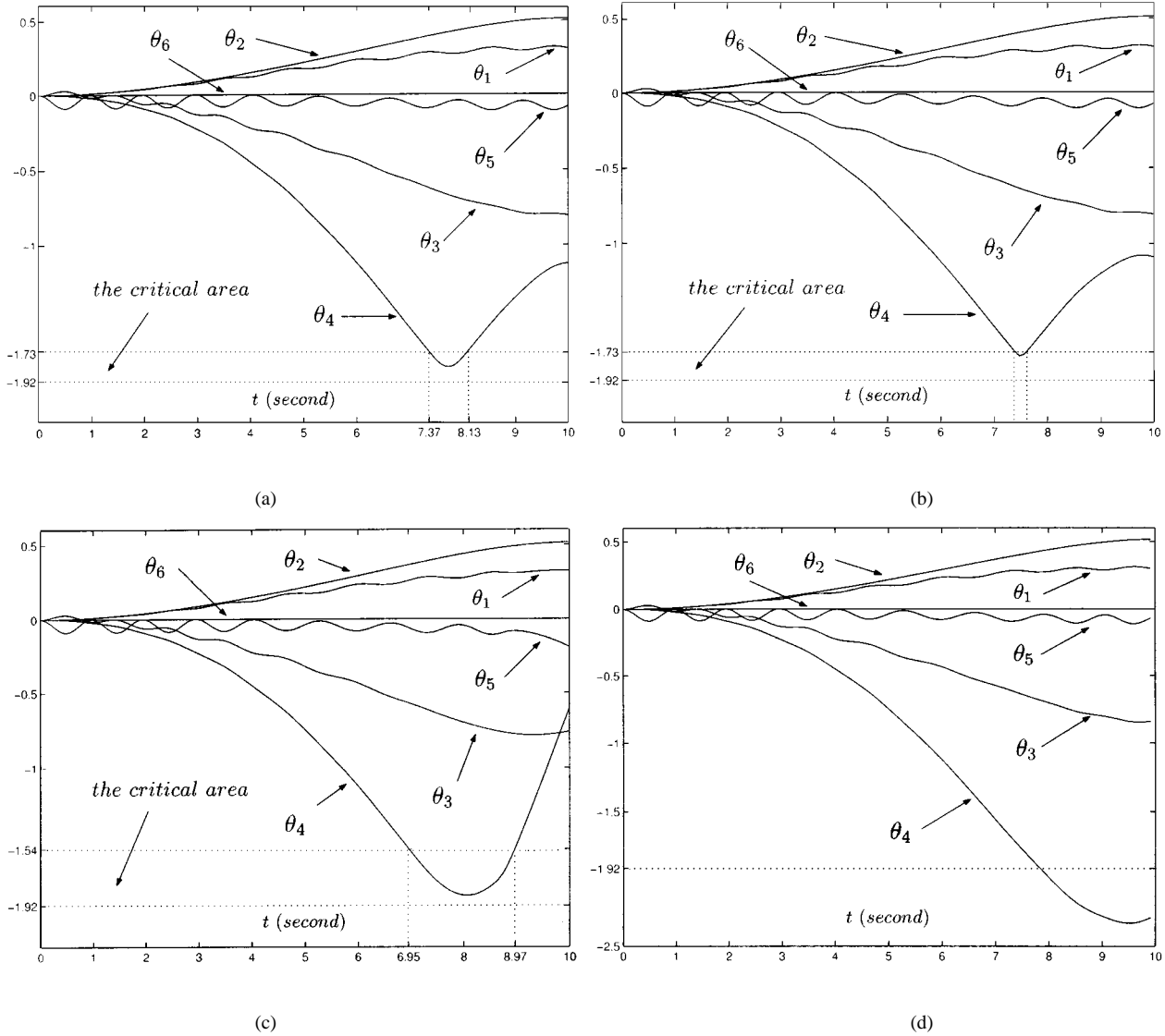


Fig. 9. Transients of the joint rotation variables synthesized by dual neural network (12) with different design parameters to avoid joint limits while the robot end-effector tracking straight line, where (b), (c), and (d) are presented for comparison purposes. (a) $\mu = 20$, $\eta = 0.9$. (b) $\mu = 200$, $\eta = 0.9$. (c) $\mu = 2$, $\eta = 0.8$. (d) No joint limits.

and $\dot{V} < 0$ if $u \neq u^*$, $\dot{V} = 0$ iff $u = u^*$ in $\Gamma(u^*)$. Thus, it follows that the dual neural network (12) is convergent to u^* .

Furthermore, to show the global exponential convergence [22], [23], review $V(u)$ and \dot{V} . It follows from (19) that $c_1 \|u - u^*\|^2 \leq V(u) \leq c_2 \|u - u^*\|^2$, $\exists c_2 \geq c_1 > 0$ where c_1 and c_2 are, respectively, the minimal and maximal eigenvalues of $c(I + J_E H^{-1} J_E^T)$. Moreover, for any $u(t) \neq u^*$ in $\Gamma(u^*)$, it is reasonably claimed that $\exists 1 \geq \gamma > 0$

$$\|\mathcal{P}(J_H J_E^T u - J_H b - u) - (J_H J_E^T u - J_H b)\|^2 \geq \gamma \|u - u^*\|^2$$

since $\|\mathcal{P}(J_H J_E^T u - J_H b - u) - (J_H J_E^T u - J_H b)\| > 0$ for any $u(t) \neq u^*$ in R^{n+m} and $\|\mathcal{P}(J_H J_E^T u - J_H b - u) - (J_H J_E^T u - J_H b)\| = 0$ amounts to $u = u^*$ only. By analyzing the linear/saturation cases of $\mathcal{P}_j([J_H J_E^T u - J_H b - u]_j)$, the range of ρ is $(0, 1]$. Therefore

$$\begin{aligned} \frac{dV(u)}{dt} &\leq -\gamma \|u - u^*\|^2 - (u - u^*)^T J_E H^{-1} J_E^T (u - u^*) \\ &= - (u - u^*)^T (\gamma I + J_E H^{-1} J_E^T) (u - u^*) \\ &\leq -\beta V(u(t)) \end{aligned}$$

where $\beta > 0$ and is proportional to the reciprocal of capacitive parameter c . Thus, we have $V(u(t)) = O(\exp(-\beta(t - t_0)))$, $\forall t \geq t_0$ and hence $\|u(t) - u^*\| = O(\exp(-\beta(t - t_0)/2))$, $\forall t \geq t_0$, which completes the exponential convergence of the proposed dual neural network (12). \square

Since the exponential convergence rate $\beta/2$ is proportional to $1/c$, we can expedite the convergence of dual neural network (12) sufficiently quickly by decreasing c and assume that the associated manipulator matrices $J(\theta)$, $H(\theta)$, and r_d are time-varying in a time scale sufficiently slower than that of (12). Specifically, starting from any $t_1 \in [0, t_f]$ and within some small time interval Δt , the maximal variation of (J, H, r_d) is sufficiently small, while the proposed dual neural network (12) has been asymptotically convergent to the corresponding theoretical solution u^* with a sufficiently small relative error; that is, for $t \in [t_1, t_1 + \Delta t]$, $\exists \alpha > 0$, $\|u(t) - u^*\| \leq \alpha \exp(-\beta t/2)$ and thus $\exists \tilde{\alpha} > 0$

$$\|\ddot{\theta}(t) - \ddot{\theta}^*\| \leq \|H^{-1} J_E^T\| \cdot \|u - u^*\| \leq \tilde{\alpha} \exp\left(-\beta \frac{t}{2}\right).$$

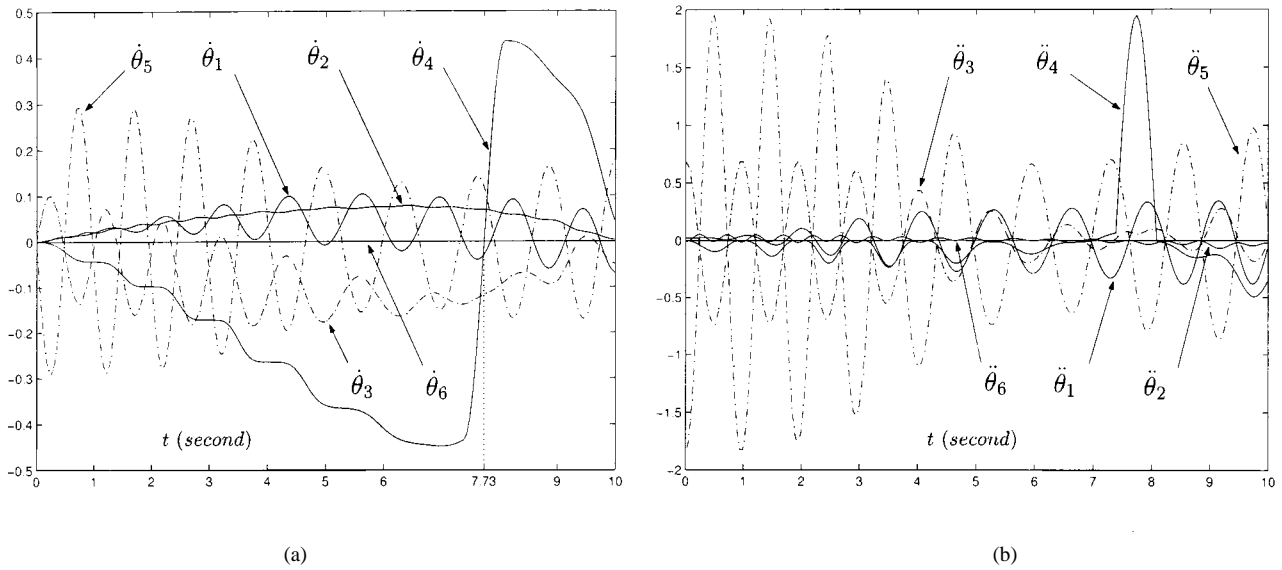


Fig. 10. Joint velocity and acceleration variables of the PUMA 560 manipulator tracking straight line. (a) Joint velocities in rad/s. (b) Joint accelerations in rad/s^2 .

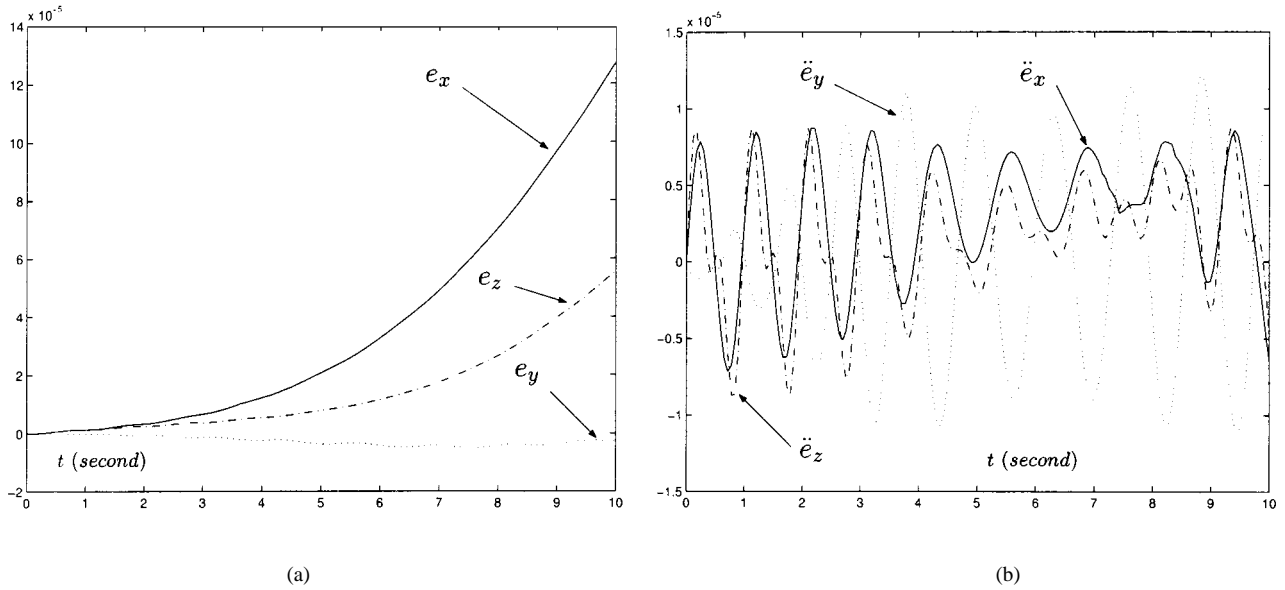


Fig. 11. Cartesian tracking errors of the PUMA 560 manipulator tracking straight line. (a) Position error in meters. (b) Acceleration error in m/s^2 .

In view of $\theta^*(0) = \theta(0)$ and $\dot{\theta}^*(0) = \dot{\theta}(0)$, it follows that

$$\begin{aligned} \|\theta(t) - \theta^*(t)\| &= \left\| \int_0^t (\dot{\theta}(\omega_1) - \dot{\theta}^*(\omega_1)) d\omega_1 \right\| \\ &= \left\| \int_0^t \int_0^{\omega_1} (\ddot{\theta}(\omega_2) - \ddot{\theta}^*(\omega_2)) d\omega_2 d\omega_1 \right\| \\ &\leq \int_0^t \int_0^{\omega_1} \|\ddot{\theta}(\omega_2) - \ddot{\theta}^*(\omega_2)\| d\omega_2 d\omega_1. \end{aligned}$$

As shown in Fig. 2, for the finite-time path-following task, the worst case of $\|\ddot{\theta}(t) - \ddot{\theta}^*(t)\|$, $t \in [0, t_f]$ can be estimated on average as $\bar{\alpha}$ where $\bar{\alpha} > 0$ depends on the capacitive parameter c . Thus, we can estimate the inverse kinematics joint configuration deviation as

$$\|\theta(t) - \theta^*(t)\| \leq \int_0^t \int_0^{\omega_1} \bar{\alpha} d\omega_2 d\omega_1 \leq \int_0^t \bar{\alpha} \omega_1 d\omega_1 \leq \frac{1}{2} \bar{\alpha} t^2$$

where $\bar{\alpha}$ can be made arbitrarily small by decreasing c , namely, increasing the convergence rate of (12). Based on Taylor series expansion of (1), $\exists \hat{\alpha} > 0$, the position tracking error $r(t) - r^*(t)$ is roughly

bounded by the function $(1/2)\hat{\alpha}t^2$ with the final error $r(t_f) - r^*(t_f)$ less than $(1/2)\hat{\alpha}t_f^2$, where the coefficient $\hat{\alpha}$ can be made arbitrarily small by decreasing the capacitive parameter c .

From the previous results of position error estimation, we know that the error can be made smaller by decreasing the design parameter c to expedite the convergence of neural network (12); whereas, although a smaller value of t_f makes the position error decrease faster, it may cause large joint acceleration or velocity. The ensuing simulation results will verify the soundness of the proposed error estimation.

V. SIMULATION RESULTS BASED ON PUMA 560 ROBOT

The Unimation PUMA 560 robot arm has six joints [24]. When both the Cartesian position and orientation are considered, the PUMA 560 arm is not a redundant robot. However, if we consider only the positioning of the end-point of its attached tool, the PUMA 560 arm, as shown in Fig. 3, becomes a redundant manipulator and the associated Jacobian matrix $J(\theta) \in R^{3 \times 6}$. Table I shows the Denavit and Hartenberg parameters of the system with joint limits θ^\pm expressed in radians.

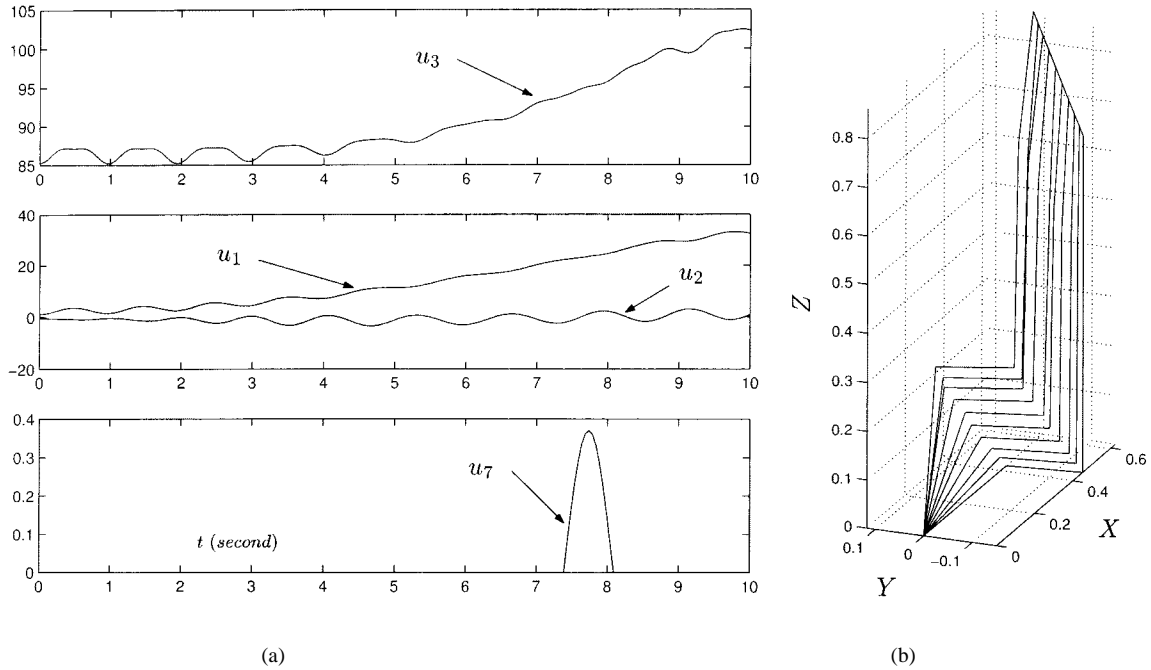


Fig. 12. Transients of the dual neural network (12) and the PUMA 560 manipulator tracking a straight line: (a) dual variables and (b) 3-D motion trajectory.

A. Circular Paths

In this subsection, by applying the proposed dual network model (12) to the PUMA 560 robot arm with the capacitive parameter $c = 10^{-6}$, we discuss the torque minimization problem of redundant manipulators when its end-effector tracks circular paths.

The first desired motion of the end-effector is a circle of radius $r = 10$ cm with the revolute angle about the x axis $\pi/6$. The task time of the motion is 10 s and the initial joint variables $\theta(0) = [0, 0, 0, 0, 0, 0]^T$. Figs. 4(a) and 5 illustrate the simulated motion of the PUMA 560 manipulator in the three-dimensional (3-D) workspace and its projection trajectories onto the xy , yz , and xz planes, which is sufficiently close to the desired one. As shown in Fig. 6, the maximal Cartesian position and acceleration tracking errors at the tool tip of PUMA 560 robot arm are less than 0.09 mm and 0.021 mm/s², respectively, where e_x, e_y , and e_z denote, respectively, the x -, y -, and z -axis components of position error with respect to the base frame and similarly \ddot{e}_x, \ddot{e}_y , and \ddot{e}_z denote acceleration error components. Figs. 7(a) and 8 show the transient behaviors of the dual neural network and the robot: joint rotations $\theta(t)$ in radians, joint torques $\tau(t)$ in Newton \cdot meters, dual decision variables $u(t)$ in Newton \cdot meters.

Another circular simulation, for comparison, has been conducted with the same specifications except for the desired radius $r = 20$ cm, which is shown in Figs. 4(b) and 7(b), and the maximal position error is less than 0.13 mm. Similar to the human arm motion trajectories, for a smaller radius circular motion [e.g., in Fig. 4(a)], the base and shoulder joints rotate with less amplitude to reduce power consumption, while the elbow and wrist joints rotate as far as needed to perform the tracking task. However, for a bigger radius circular motion [e.g., in Fig. 4(b)], the base and shoulder joints are also required to rotate with sufficiently large amplitude, since only the elbow and wrist rotation is not enough to perform such a desired path.

Incidentally, the solution vectors $\theta(t)$ of joint rotation variables depicted in Fig. 7 never exceed any joint boundary θ^\pm and thus are feasible, which demonstrates the capability of the proposed dual neural network model (12) for online optimization of joint torques by exploiting the extra DOF for redundant manipulators.

B. Straight Lines

Such a practical problem of avoiding joint limits arises when planning a straight-line Cartesian path starting/ending with zero velocity (the angles between the desired line and xy, yz, zx -planes are all $\pi/4$ and the end-point motion length is 30 cm). The first simulated results of using the proposed dual neural network (12) but ignoring joint limits are illustrated in Fig. 9(d). Although the maximal position and acceleration tracking errors are less than 0.13 mm and 0.013 mm/s², respectively, the joint variable θ_4 has exceeded its mechanical range $[-1.919, 2.967]$ rad, and thus the solution becomes inapplicable. If such a solution is directly applied to PUMA 560 with θ_4 finally locked at position -1.919 , the tracking error increases considerably, not to mention the physical damage possibly caused. Hence, the path-following task fails.

With Table I joint limits considered ($\mu = 20, \eta = 0.9$), dual network (12) is simulated to control PUMA 560 for the path-following torque minimization task. Fig. 9(a) and Figs. 10–12 illustrate the performance of the dual neural network-based inverse kinematic PUMA 560 control system. As seen in Figs. 9(a) and 10, the avoidance scheme of joint limits really works well. That is, when the fourth joint θ_4 enters the critical area $[-1.727, -1.919]$ at time $t = 7.37$ s, the lower bound ξ_4^- of its feasible acceleration range (8) becomes positive (namely, of the opposite direction) to decelerate the original fourth-joint motion and drive it away from the vicinity of joint limit $\theta_4^- = -1.919$ rad. The deceleration magnitude $|\xi_4^-|$ increases proportionally as $\theta_4(t)$ approaches its limit θ_4^- and the ratio μ can be tuned larger to expedite the escaping from such a critical area. Clearly, due to the inertia, the original fourth-joint movement continues until the velocity changes its sign at $t = 7.73$ s and then leaves the critical area at $t = 8.13$ s, which accomplishes the avoidance of joint limits effectively. Moreover, as seen from Fig. 11, the path-following task has been completed satisfactorily, with the maximal position and acceleration errors less than 0.127 mm and 0.124 mm/s², respectively.

It is worth mentioning that, as shown in the subplots of Fig. 9, when the intensity coefficient μ increases, the robot leaves the associated critical area (also known as the buffering area of a joint limit) within a shorter time. However, more actuator power might be consumed to provide the deceleration of larger magnitude. Thus, from the viewpoint

of torque minimization, a smaller value of μ is preferable, and as a tradeoff, a larger critical area has to be prescribed [e.g., $\eta = 0.8$ when selecting $\mu = 2.0$ as in Fig. 9(c)]. In summary, performance of the presented examples demonstrates the effectiveness of the proposed dual neural network model (12) on the path-following and torque minimization task of redundant manipulators with limited joint ranges.

VI. CONCLUDING REMARKS

The one-layer dual neural network model provides a new parallel distributed computational approach to real-time torque minimization of limited-joint-range redundant manipulators in real-time. Compared with other studies on torque optimization, the proposed formulation resolves redundancy at the acceleration level and simultaneously considers the joint limit avoidance. Different from other recurrent neural network approaches, the proposed dual neural network is developed for avoiding robot joint limits during the path-following torque optimization task. Future works include the neural network implementation on a dedicated hardware such as ASIC, and then experimental verifications on different manipulators.

REFERENCES

- [1] L. Sciacivco and B. Siciliano, *Modeling and Control of Robot Manipulators*. New York: Springer-Verlag, 2000.
- [2] Z. Mao and T. C. Hsia, "Obstacle avoidance inverse kinematics solution of redundant robots by neural networks," *Robotica*, vol. 15, pp. 3–10, 1997.
- [3] T. Yoshikawa, "Manipulability of robot mechanisms," *Int. J. Robot. Res.*, vol. 4, no. 2, pp. 3–9, 1985.
- [4] A. S. Deo and I. D. Walker, "Minimum effort inverse kinematics for redundant manipulators," *IEEE Trans. Robot. Automat.*, vol. 13, pp. 767–775, Oct. 1997.
- [5] J. M. Hollerbach and K. C. Suh, "Redundancy resolution of manipulators through torque optimization," *IEEE Trans. Robot. Automat.*, vol. RA-3, pp. 308–315, 1987.
- [6] A. Nedungadi and K. Kazerounian, "A local solution with global characteristics for joint torque optimization of a redundant manipulator," *J. Robot. Syst.*, vol. 6, no. 5, pp. 631–654, 1989.
- [7] H. J. Kang and R. A. Freeman, "Null space damping method for local joint torque optimization of redundant manipulators," *J. Robot. Syst.*, vol. 10, pp. 249–270, 1993.
- [8] S. Ma, "A balancing technique to stabilize local torque optimization solution of redundant manipulators," *J. Robot. Syst.*, vol. 13, no. 3, pp. 177–185, 1996.
- [9] —, "A new formulation technique for local torque optimization of redundant manipulators," *IEEE Trans. Ind. Electron.*, vol. 43, pp. 462–468, 1996.
- [10] W. S. Tang and J. Wang, "Two recurrent neural networks for local joint torque optimization of kinematically redundant manipulators," *IEEE Trans. Syst., Man, Cybern. B*, vol. 30, pp. 120–128, Jan. 2000.
- [11] K. Kazerounian and A. Nedungadi, "Redundancy resolution of robotic manipulators at the acceleration level," in *Proc. 7th World Congress Theory of Machines and Mechanisms (IFTOMM)*, Sevilla, Spain, 1987.
- [12] K. C. Suh and J. M. Hollerbach, "Local versus global torque optimization of redundant manipulators," in *Proc. IEEE Int. Conf. Robotics and Automation*, Raleigh, NC, Mar. 1987.
- [13] J. Wang, Q. Hu, and D. Jiang, "A Lagrangian network for kinematic control of redundant manipulators," *IEEE Trans. Neural Networks*, vol. 10, pp. 1123–1132, Sept. 1999.
- [14] H. Ding and S. K. Tso, "A fully neural-network-based planning scheme for torque minimization of redundant manipulators," *IEEE Trans. Ind. Electron.*, vol. 46, pp. 199–206, 1999.
- [15] 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 architectural complexity," *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, pp. 98–105, Jan. 2001.
- [16] Y. Xia and J. Wang, "A dual neural network for kinematic control of redundant robot manipulators," *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, pp. 147–154, Jan. 2001.

- [17] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming—Theory and Algorithms*. New York: Wiley, 1993.
- [18] D. P. Bertsekas, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [19] O. L. Mangasarian, "Solution of symmetric linear complementarity problems by iterative methods," *J. Optim. Theory Applicat.*, vol. 22, no. 2, pp. 465–485, 1979.
- [20] W. Li and J. Swetits, "A new algorithm for solving strictly convex quadratic programs," *SIAM J. Optim.*, vol. 7, no. 3, pp. 595–619, 1997.
- [21] J.-S. Pang and J.-C. Yao, "On a generalization of a normal map and equation," *SIAM J. Control Optim.*, vol. 33, pp. 168–184, 1995.
- [22] Y. Xia and J. Wang, "Global exponential stability of recurrent neural networks for solving optimization and related problems," *IEEE Trans. Neural Networks*, vol. 11, pp. 1017–1022, 2000.
- [23] —, "Global asymptotic and exponential stability of a dynamic neural system with asymmetric connection weights," *IEEE Trans. Automat. Contr.*, vol. 46, pp. 635–638, 2001.
- [24] P. I. Corke and B. Armstrong-Helouvy, "A search for consensus among model parameters reported for the PUMA 560 robot," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 2, 1994, pp. 1608–1613.

A Merge-Based Condensing Strategy for Multiple Prototype Classifiers

Ramón A. Mollineda, Francesc J. Ferri, and Enrique Vidal

Abstract—A class-conditional hierarchical clustering framework has been used to generalize and improve previously proposed condensing schemes to obtain multiple prototype classifiers. The proposed method conveniently uses geometric properties and clusters to efficiently obtain reduced sets of prototypes that accurately represent the data while significantly keeping its discriminating power. The benefits of the proposed approach are empirically assessed with regard to other previously proposed algorithms which are similar in their foundations. Other well-known multiple prototype classifiers have also been taken into account in the comparison.

Index Terms—Clustering, condensing, multiple prototypes, nearest neighbors (NNs).

I. INTRODUCTION

One of the best known and most extensively studied family of pattern classifiers is the k -nearest neighbor (NN) rule. Given a training set of previously labeled samples (or prototypes) and an unknown sample x , the k -NN rule assigns the most frequently represented class-label among the k closest prototypes to x . In spite of its conceptual simplicity, the rule is asymptotically optimal in the Bayes sense [1], provided there is an arbitrarily large number of prototypes n available so that k can be taken as large as necessary, while keeping the ratio k/n arbitrarily small which, in turn, implies that the neighbors are infinitely close to x . A trivial consequence of the large size of the sets of prototypes is the computational burden this searching problem implies. Another very important drawback comes from the erroneously labeled

Manuscript received March 21, 2002. This work was supported in part by Spanish project TIC2000-1703-C03-03. This paper was recommended by Associate Editor W. Pedrycz

R. A. Mollineda and E. Vidal are with the Institut Tecnològic d'Informàtica, Universitat Politècnica de València, 46071 València, Spain.

F. J. Ferri is with the Departamento d'Informàtica, Universitat de València, 46100 Burjassot, Spain.

Publisher Item Identifier S 1083-4419(02)06293-3.