A Primal-Dual Neural Network for Joint Torque Optimization of Redundant Manipulators Subject to Torque Limit Constraints

Wai Sum Tang and Jun Wang
Department of Mechanical and Automation Engineering
The Chinese University of Hong Kong
Shatin, New Territories, Hong Kong, China
E-mail: wstang@mae.cuhk.edu.hk, jwang@mae.cuhk.edu.hk

ABSTRACT

In this paper, a primal-dual neural network is proposed for the joint torque optimization of redundant manipulators subject to torque limit constraints. The neural network generates the minimum driving joint torques which never exceed the hardware limits and keep the end-effector to track a desired trajectory. The consideration of physical limits prevents the manipulator from torque saturation and hence ensuring a good tracking accuracy. The neural network is proven to be globally convergent to the optimal solution. The simulation results show that the neural network is capable of effectively computing the optimal redundancy resolution.

1 INTRODUCTION

Redundant manipulators are those having more degrees of freedom than required to perform a given task. The extra degrees of freedom enable a redundant manipulator to avoid singularity, obstacles, joint limits, and to optimize various performance criteria [1]-[5]. One example of the performance criteria to be minimized is joint torque. Since optimizing the joint torque makes an effective utilization of the actuator input power, torque optimization is an appealing area in redundant manipulator research.

The initial study for local torque minimization was conducted by Hollerbach and Suh [2] who presented the Null-Space (NS) algorithm which instantaneously optimizes joint torques using the null space of Jacobian such that all the joint torques are placed closest to the midpoint between the upper and lower joint torque limits. However, the NS algorithm exhibits an unstable movement for a long-range motion. In order to eliminate the stability problem, Suh and Hollerbach [3] developed a global torque optimization technique using the calculus of variation, which results in stable optimal solutions. The global torque minimization technique, however, requires intensive computation. It is therefore unsuitable for real-time motion control. Since then, many researchers tried to formulate other local torque optimization schemes for solving the instability problem. For example, Nedungadi and Kazemian [4] presented an approach that minimizes the local inertia inverse weighted joint torque which corresponds to global kinetic energy minimization and leads to global stable joint torque. Kang and Freeman [5] proposed the Null Space Damped Joint Torque Minimization (NDJTM) method in which damping forces are generated by using appropriate null space to stabilize the local joint torque.

In recent years, neural networks have been developed for the kinematic and dynamic control of redundant robots; e.g., [6]-[11]. Specifically, in [7] Ding and Chan presented an approach that incorporates the Tank-Hopfield (TH) network [12] into the NS algorithm for the redundancy resolution that locally minimizes the joint torques. In [10, 11] Tang and Wang proposed a two-layer recurrent neural network for local torque optimization of redundant manipulators. These research works showed neural network is efficient for real-time torque optimization of redundant manipulators. However, the joint torque limits were not included in the formulations. The driving torques generated by these approaches may therefore exceed the actuator limits and these large demand joint torques can cause torque saturation, which in turn reduces tracking accuracy. Hence, the joint torque limits should be taken into ac-
In this paper, a primal-dual neural network is presented for torque optimization of redundant robots with joint torque limits being considered. The proposed neural network explicitly minimizes the joint torque weighted by local inertia inverse and the computed driving joint torques are guaranteed never to exceed the upper and lower joint torque limits. The neural network is proven to converge asymptotically to the optimal solution. The effectiveness of the proposed neural network approach is demonstrated by use of computer simulation.

2 PROBLEM FORMULATION

The torque optimization of robot manipulators is based on the well-known arm dynamics [2]:

\[ \tau = H(\theta)\ddot{\theta} + c(\theta, \dot{\theta}) + g(\theta), \]  

where \( \tau \in \mathbb{R}^n \) denotes the joint torques, \( \theta, \dot{\theta}, \ddot{\theta} \in \mathbb{R}^n \) are the joint variables, joint velocities and joint accelerations respectively; \( H(\theta) \in \mathbb{R}^{n \times n} \) is the positive-definite symmetric inertia matrix, and \( c(\theta, \dot{\theta}) \in \mathbb{R}^n \) and \( g(\theta) \in \mathbb{R}^n \) are components of the torque depending on Coriolis, centrifugal and gravity forces respectively.

In addition, in order to optimize joint torques, the redundancy resolutions at acceleration level should be considered. At this level the direct kinematics can be expressed as [5]:

\[ J(\theta)\ddot{\theta} = \ddot{\bar{r}} - \ddot{J}(\theta)\dot{\theta}, \]  

where \( \ddot{\bar{r}} \in \mathbb{R}^m \) represents accelerations of the end-effector in Cartesian space derived from the given task, and \( J(\theta) \in \mathbb{R}^{m \times n} \) is the Jacobian matrix and \( \ddot{J}(\theta) \in \mathbb{R}^{m \times n} \) is its time derivative. For redundant manipulators, (2) is underdetermined since \( m < n \).

It is shown in [4] that the local optimization of joint torque weighted by inertia inverse results in solutions with global characteristics and it was formulated as the following constrained optimization problem subject to the redundancy resolutions at acceleration level:

\[ \begin{align*}
\text{minimize} & \quad \frac{1}{2} \tau^T H^{-1} \tau, \\
\text{subject to} & \quad J(\theta) \dot{\theta} = \ddot{\bar{r}} - \ddot{J}(\theta)\dot{\theta},
\end{align*} \]  

where the superscript \( T \) denotes the transpose operator. However, the calculation of joint accelerations is not required in the torque optimization of manipulator arms, where only the joint torques need to be calculated. Thus, it makes a needless computation. It is more desirable to have torque-based constraints.

In [13], Ma used (1) and (2) to derive the relation between the joint torque and the end-effector acceleration. Inverting (1), we have the joint accelerations for given joint torques:

\[ \ddot{\theta} = H^{-1}(\tau - c - g). \]  

Substituting (5) into (2), the joint torque can be related to the end-effector acceleration as:

\[ JH^{-1}\tau = JH^{-1}(c + g) + \ddot{\bar{r}} - J\ddot{\theta}. \]  

Equation (6) can be simplified by introducing the terms \( \ddot{\bar{r}} = JH^{-1}(c + g) + \ddot{\bar{r}} - J\ddot{\theta} \) and \( J_r = JH^{-1} \).

Hence, we have the linear torque-based constraint:

\[ J_r \tau = \ddot{\bar{r}}. \]  

By taking account the joint torque limits further, the local optimization of inertia inverse weighted joint torque problem can be reformulated to a quadratic programming problem subject to linear equality constraints as well as bound constraints:

\[ \begin{align*}
\text{minimize} & \quad \frac{1}{2} \tau^T H^{-1} \tau, \\
\text{subject to} & \quad J(\theta) \dot{\theta} = \ddot{\bar{r}} - \ddot{J}(\theta)\dot{\theta}, \\
& \quad \tau^- \leq \tau \leq \tau^+, \\
& \quad \tau^-, \tau^+ \in \mathbb{R}^n \text{ denote the lower and the upper joint torque limits, respectively.}
\end{align*} \]  

According to the dual theory [14], the dual problem for the quadratic program defined in (8) is

\[ \begin{align*}
\text{maximize} & \quad \ddot{\bar{r}}^T y - \frac{1}{2} \dddot{\bar{r}}^T H^{-1} \dddot{\bar{r}} - u^T \tau^- \\
& \quad - v^T \tau^+, \\
\text{subject to} & \quad H^{-1} \tau - J_r^T y + u - v = 0, \\
& \quad u, v \geq 0,
\end{align*} \]  

where \( y \in \mathbb{R}^m \), and \( u, v \in \mathbb{R}^n \) are dual decision variable vectors.

3 NETWORK DESCRIPTIONS

The minimum driving joint torques subject to physical limit constraints for redundant manipulators are obtained by solving the quadratic program (8) in real-time. However, solving time-varying optimization
problems by using numerical iteration algorithms presented in literatures is generally computationally intensive. Neural networks with inherent massively parallel distribution nature, however, are a good alternative for solving optimization problems in real-time. Since Hopfield and Tank's seminal work [12], a number of neural networks have been developed for solving linear and quadratic programming problems in recent years; e.g., [15]-[17]. In particular, Xia and Wang [17] presented a recurrent neural network for solving linear programs subject to bound constraints based on the duality theory. Although this neural network is developed for solving linear programs, it can be extended to solve quadratic programs subject to linear equality and bound constraints like (8). In this section, we follow their approach to develop a primal-dual neural network which solves the pair of primal and dual quadratic programs (8) and (9) simultaneously to obtain the bounded minimum driving joint torques of redundant manipulators. From the optimality conditions for (8) and (9), we map the optimal solutions of (8) and (9) to the equilibrium points of a set of asymptotically stable differential equations.

By the complementary slackness theorem [14], \( \tau^* \) and \((y^*, u^*, v^*)\) are the optimal solutions to the pair of primal (8) and dual (9) programs, respectively, if and only if \((\tau^*, y^*, u^*, v^*)\) satisfies

\[
J_r \tau^* = \bar{r}_r, \quad (10)
\]

\[
H^{-1} \tau^* - J^T_r y^* + u^* - v^* = 0, \quad (11)
\]

\[
\tau^- \leq \tau^* \leq \tau^+, \quad (12)
\]

\[
u^*, v^* \geq 0, \quad (13)
\]

and the following complementary conditions:

\[
u^*T(\tau^* - \tau^-) = 0, \quad (14)
\]

\[
u^*T(\tau^+ - \tau^*) = 0. \quad (15)
\]

The complementary slackness theorem indicates that at optimal solution the restricted primal variable and its dual constraint cannot be satisfied simultaneously and one of these constraints must be zero, or vice versa for the restricted dual slack variable and its primal constraint as summarized in (14) and (15). Equations (14) and (15) thus imply that if \( \tau^* = \tau^- \), then \( u^* \geq 0 \), \( v^* = 0 \) and hence \( H^{-1} \tau^- - J^T_r y^- \leq 0 \). Similarly, if \( \tau^* = \tau^+ \) then \( u^* = 0 \), \( v^* \geq 0 \) and hence \( H^{-1} \tau^+ - J^T_r y^+ \geq 0 \). Also, when \( \tau^- \leq \tau^* \leq \tau^+ \), \( u^* = v^* = 0 \) and hence \( H^{-1} \tau^* - J^T_r y^* = 0 \). Therefore equations (11)-(15) are equivalent to the following projection equation:

\[
P_{\Omega}(\tau^* + H^{-1} \tau^- - J^T_r y^-) = \tau^*, \quad (16)
\]

Hence, by solving the following optimality conditions, we have the optimal solutions of the pair of primal and dual quadratic programs defined in, respectively, (8) and (9):

\[
J_r \tau - \bar{r}_r = 0, \quad (17)
\]

\[
\tau - P_{\Omega}(\tau + H^{-1} \tau - J^T_r y) = 0. \quad (18)
\]

Extending the structure of the neural network presented in [17], we proposed the following primal-dual neural network for solving the pair of primal (8) and dual (9) quadratic programs:

\[
C_1 \frac{dx}{dt} = -[J^T_r(J_r \tau - \bar{r}_r) + \beta z(\tau, y)], \quad (19)
\]

\[
C_2 \frac{dy}{dt} = -\beta [J_r q(\tau, y) - \bar{r}_r], \quad (20)
\]

where \( z(\tau, y) = 2H^{-1} \tau - J^T_r \tau - J^T_r y \), \( q(\tau, y) = P_{\Omega}(\tau + H^{-1} \tau - J^T_r y) \), \( \beta = \|z(\tau, y)\|_2^2 \), and \( C_1 \in \mathbb{R}^{n \times n} \), \( C_2 \in \mathbb{R}^{n \times n} \) are positive diagonal capacitive matrices which are used to scale the convergence rate of the recurrent neural network. The values of the capacitive matrices \( C_1 \) and \( C_2 \) should be selected as small as possible so long as the hardware is allowed. Obviously, the equilibrium points of the system (19) and (20) equal the optimal solutions of (8) and (9).

In this context, the desired accelerations of the end-effector for a given task \( \bar{r} \) is incorporated into \( \bar{r}_r \) which
is fed into the neural network, and the neural network generates the minimum computed joint torques to drive the robot arm at the same time. Fig. 1 delineates the torque control process of redundant robot manipulators based on the proposed neural network. Fig. 2 illustrates the configuration of the proposed primal-dual neural network.

4 STABILITY ANALYSIS

In this section, we prove the convergence of the proposed neural network defined in equations (19) and (20). First, we introduce the two lemmas.

Lemma 1. Let \( r, y \in \mathbb{R}^m \), then
\[
[q(r, y) - r]^T[H^{-1}r - J_T^T y] \geq \|r - q(r, y)\|_2^2.
\]

Proof. Since \( \Omega \) is a convex set, the angle between vectors \( q(r, y) - (r + H^{-1}r - J_T^T y) \) and \( r - q(r, y) \) are acute, then the inner product of these two vectors gives
\[
[q(r, y) - (r + H^{-1}r - J_T^T y)]^T[r - q(r, y)] \geq 0. \tag{21}
\]
Rearranging (21), we have
\[
[q(r, y) - r]^T[H^{-1}r - J_T^T y] \geq \|r - q(r, y)\|_2^2.
\]

Lemma 2. Let \( \phi(r, y) = \|r_T - \tilde{r}_T\|^2 + \beta[q(r, y) - r]^T[H^{-1}r - J_T^T y] \) for \( r \in \Omega, y \in \mathbb{R}^n \), then \( \phi(r, y) \geq 0 \), and \( \phi(r, y) = 0 \) if and only if \( (r, y) \) is an optimal solution to the pair of primal and dual problems defined in (8) and (9). Hence \( (r, y) \) is an optimal solution to (8) and (9) if and only if \( \phi(r, y) = 0 \). When \( \phi(r, y) = 0 \), it is obvious from (19) and (20) that \( (r, y) \) makes \( dr/dt = 0 \) and \( dy/dt = 0 \), and thus \( (r, y) \) is an equilibrium point of the system defined in (19) and (20).

Theorem 2. The primal-dual neural network defined in (19) and (20) is asymptotically stable and convergent to the optimal solutions of the primal problem and its dual problem defined in (8) and (9), respectively.

Proof. From Lemma 1, we have \( [q(\tau, y) - \tau]^T[H^{-1}\tau - J_T^T y] \geq 0 \), then \( \phi(\tau, y) \geq 0 \). Also, \( \phi(\tau, y) = 0 \) if and only if \( J_T^T \gamma = \tilde{r}_T \) and \( \gamma = q(\tau, y) \) which are the optimality conditions (17) and (18) for the pair of primal and dual problems defined in (6) and (9). Hence \( (\tau, y) \) is an optimal solution to (8) and (9) if and only if \( \phi(\tau, y) = 0 \). When \( \phi(\tau, y) = 0 \), it is obvious from (19) and (20) that \( (\tau, y) \) makes \( dr/dt = 0 \) and \( dy/dt = 0 \), and thus \( (\tau, y) \) is an equilibrium point of the system defined in (19) and (20).

Proof. Let \( x^T = (\tau^T, y^T) \) and without loss of generality the positive diagonal capacitive matrices in (19) and (20) be identity matrices. Then consider the Lyapunov function:
\[
V(x) = -\frac{1}{2}(x - x^*)^T(x - x^*), \tag{22}
\]
where \( x^* = (\tau^*, y^*) \), and \( \tau^*, y^* \) are the optimal solutions for the pair of primal and dual problems defined in (8) and (9), respectively. Differentiating (22) with respect to time gives
\[
\frac{dV(x)}{dt} = -[\tau - \tau^*]^T \begin{bmatrix} J_T^T (J_T^T \tau - \tilde{r}_T) + \beta z(\tau, y) \\
J_T^T (J_T^T \tau - \tilde{r}_T) + \beta z(\tau, y) \end{bmatrix}.
\]
Since \( J_T^T \tau^* = \tilde{r}_T \), and by the duality theorem [14], a pair of primal and dual problems has equal optimal objective function value; i.e., \( \tau^*^T y^* = -\frac{1}{2} \tau^*^T H^{-1} \tau^* - u^*^T T^* - v^*^T \tau^* = \frac{1}{2} \tau^*^T H^{-1} \tau^* \), we have
\[
-\frac{dV(x)}{dt} \leq -[\tau - \tau^*]^T \begin{bmatrix} J_T^T (J_T^T \tau - \tilde{r}_T) + \beta z(\tau, y) \\
J_T^T (J_T^T \tau - \tilde{r}_T) + \beta z(\tau, y) \end{bmatrix}.
\]
Using the identity \( J_T^T y^* - H^{-1} \tau^* = u^* - v^* \), we obtain
\[
\frac{dV(x)}{dt} \geq \beta \|r^+ - q(\tau, y)\|^2 + \beta [q(\tau, y) - r]^T u^* + \phi(\tau, y) + \beta (r - \tau^* )^T H^{-1} (r - \tau^* )^T u^* \geq 0.
\]
From Lemma 2 and because \( H^{-1} \) is positive definite, we have
\[
\frac{dV(x)}{dt} \leq -[\phi(\tau, y) + \beta (r - \tau^* )^T H^{-1} (r - \tau^* )] < 0.
\]
Hence, the primal-dual neural network defined in (19) and (20) is asymptotically stable. Furthermore, by Lemma 2, the neural network is therefore convergent to the optimal solutions of the pair of primal and dual problems defined in (8) and (9). The proof is thus complete.

\[ \text{Fig. 3: Three-link planar rotary manipulator.} \]

5 SIMULATION RESULTS

Computer simulation has been conducted to demonstrate the effectiveness of the proposed recurrent neural network for torque optimization of redundant manipulators. In order to be consistent with previous work, the simulation configuration is the same as the one in [11]. The simulation is based on a three-link unit mass and unit length planar rotary manipulator arm as shown in Fig. 3. The simulated movements of the end-effector are straight line Cartesian paths starting and ending with zero velocity and bang-bang typo acceleration, with equal and constant deceleration and acceleration in $x$ and $y$ directions with magnitude of $2 \text{ m/s}^2$. The duration taken for the simulation is 2 seconds. The initial states of the manipulator are $\theta(0) = [\pi, -\pi/2, -\pi/2]^T$ and $\dot{\theta}(0) = [0, 0, 0]^T$. The upper and lower joint torque limits for joints $1$–$3$ are set at $\pm 15$, $\pm 10$, and $\pm 5$ Nm, respectively.

In this simulation, the differential equations (19) and (20) are solved by the fourth-order Runge-Kutta method implemented by using MATLAB. The capacitive matrices in (19) and (20) are set to be $C_1 = 2 \times 10^{-6}I$ and $C_2 = 2 \times 10^{-6}I$.

Fig. 4 and Fig. 5 are the joint torque profiles obtained by using the two-layer recurrent neural network presented in [11] which does not consider the joint torque limits, and the proposed primal-dual neural network, respectively. It is clear that the computed driving joint torques for joints $1$ and $2$ are beyond the joint torque limits between $0.7$ and $1$ second by using the two-layer recurrent neural network. However, the computed driving joint torques are within the joint torque limits all over the motion by using the proposed primal-dual neural network. Fig. 6 shows the joint motion trajectories of the redundant arm driven by the torques computed by the proposed primal-dual neural network.

\[ \text{Fig. 4: Joint torque profiles obtained by using the previous neural network where joint torque limits are not considered.} \]

\[ \text{Fig. 5: Joint torque profiles obtained by using the proposed neural network where joint torque limits are considered.} \]

6 CONCLUDING REMARKS

The proposed primal-dual neural network provides a new parallel distributed model for computing the driving joint torques for redundant manipulators. Compared with previous models, the present neural network can explicitly take into account the joint torque
Fig. 6: Joint motion trajectories of the redundant arm driven by the torques computed by the proposed primal-dual neural network.

limits and generate optimized stable driving joint torques which guarantee for never exceeding the joint torque limits. Compared with the supervised learning neural networks, the present approach eliminates the need for training and guarantees to be asymptotically stable. Because of the inherently parallel distributed computation nature, the proposed neural network implemented in a dedicated ASIC can be used for real-time torque optimization of redundant manipulators.

REFERENCES


