# A Balancing Technique to Stabilize Local Torque Optimization Solution of Redundant Manipulators

**Shugen Ma**
*Ibaraki University, Faculty of Engineering*
*Department of Systems Engineering*
*4-12-1 Naka-Narusawa-Cho, Hitachi-Shi 316, Japan*

Received January 12, 1995; revised October 13, 1995; accepted October 26, 1995

A technique that stabilizes the existing local torque optimization solutions for redundant manipulators is proposed in this article. The technique is based on a balancing scheme, which balances a solution of joint torque-minimization against a solution of joint velocity-minimization. Introducing the solution of joint velocity-minimization in the approach prevents occurrence of high joint velocities, and thus results in stable optimal arm motions and guarantees the joint velocities at end of motion to be near zero. Computer simulations were executed on a three-link planar rotary manipulator to verify the performance of the proposed local torque optimization technique and to compare its performance with existing ones for various straight line trajectories. © 1996 *John Wiley & Sons, Inc.*

この発表では、冗長性のあるマニピュレータに対して、存在するローカル・トルク最適化の解を安定させる方法を提案する。この技法は平衡法を基本とし、ジョイント速度最小化の解に対するジョイント・トルク最小化の解の平衡を取る。接近時にジョイント速度最小化の解を導入することで、ジョイント速度が上がるのを防止して安定で最適なアーム動作を実現するとともに、動作の最終段階におけるジョイント速度をほぼゼロにすることを保証している。3リンク・プレイナー・ロータリー・マニピュレータのコンピュータ・シミュレーションを実行し、提案したローカル・トルク最適化法の性能を確認するとともに、様々な直線軌道において既存の方法と性能を比較する。

Journal of Robotic Systems 13(3), 177–185 (1996)

CCC 0741-2223/96/030177-09

## 1. INTRODUCTION

Kinematically redundant manipulators have been receiving increasing attention in recent years because of their ability to perform tasks of high complexity. Besides the end-effector task, one or more additional tasks can be accomplished. One example of the additional tasks is to minimize joint torques. Since minimizing the joint torques is equivalent to minimizing the input power of the manipulator, thus minimizing joint torques makes the manipulator execute heavy work with small power input.

To minimize the joint torques of the redundant manipulator, the inverse kinematics must be solved at the acceleration level to resolve the redundancy. At this level, direct kinematics of the redundant manipulator is expressed as

$$\mathbf{J}\ddot{\mathbf{q}} = \ddot{\mathbf{x}} - \dot{\mathbf{J}}\dot{\mathbf{q}}, \tag{1}$$

where $\ddot{\mathbf{x}} \in R^m$ represents the end-effector accelerations derived from the given task, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}} \in R^n$ are joint velocities and joint accelerations respectively, $\mathbf{J} \in R^{m\times n}$ is the Jacobian matrix and $\dot{\mathbf{J}} \in R^{m\times n}$ is its time-derivative. For redundant manipulators, equation (1) will be underdetermined since $m < n$.

If a desired task acceleration vector $\ddot{\mathbf{x}}$ of the end-effector is given, and current joint positions and velocities are known, equation (1) can be solved for joint accelerations by means of some generalized inverse $\mathbf{J}^-$ of the non-square Jacobian

$$\ddot{\mathbf{q}} = \mathbf{J}^-(\ddot{\mathbf{x}} - \dot{\mathbf{J}}\dot{\mathbf{q}}) + (\mathbf{I} - \mathbf{J}^-\mathbf{J})\ddot{\phi}. \tag{2}$$

Equation (2) comprises two components for joint accelerations; the first term on the right-hand side stands for a particular solution of the underdetermined system (1), and the second term represents the general solution of the homogeneous system $\mathbf{J}\ddot{\mathbf{q}} = 0$ (*the homogeneous solution*). Nontrivial homogeneous solutions can be obtained by varying the $n$ vector $\ddot{\phi} \neq 0$ arbitrarily and projecting it onto the nullspace of the Jacobian matrix by means of the projecting operator $(\mathbf{I} - \mathbf{J}^-\mathbf{J})$. Thus, different joint accelerations can be generated for the same $\ddot{\mathbf{x}}$.

On the other hand, it is well known that the joint torques are expressed as

$$\tau = \mathbf{H}\ddot{\mathbf{q}} + \mathbf{c}_{(\mathbf{q},\dot{\mathbf{q}})} + \mathbf{g}_{(\mathbf{q})}, \tag{3}$$

where $\mathbf{H}_{(\mathbf{q})} \in R^{n\times n}$ is the inertia matrix, and $\mathbf{c}_{(\mathbf{q},\dot{\mathbf{q}})} \in R^n$ and $\mathbf{g}_{(\mathbf{q})} \in R^n$ are components of the torques depending upon Coriolis, centrifugal, and gravity forces, respectively. Substituting (2) into (3), it can be seen that due to the redundancy, a joint torque component $\mathbf{H}(\mathbf{I} - \mathbf{J}^-\mathbf{J})\ddot{\phi}$ is added. This component can be extracted from the nullspace of the Jacobian matrix and varied arbitrarily through $\ddot{\phi}$ to minimize joint torques for the given $\ddot{\mathbf{x}}$. Another alternative for varying joint torques is to utilize different generalized inverse $\mathbf{J}^-$ in the particular solution of equation (2).

Hollerbach et al.[1,2] proposed the utilization of equations (2) and (3) for local optimization of joint torques by the Moor-Penrose generalized inverse (*pseudoinverse*) $\mathbf{J}^+$. Another technique for torque minimization was proposed by Kazerounian et al.[3,4] Lagrangian multipliers were utilized for the minimization of joint torques under consideration of equations (1) and (3). These local solutions are much more modest in complexity and computational requirements, and have therefore proved more practical for on-line motion control. However, these approaches revealed a major obstacle: the local algorithm resulted in instability during tracking long trajectories, leading to a whipping-like motion with increasingly high joint velocities. In addition, no one had taken the joint velocities at end of motion into consideration in control of redundant manipulators, even though this is important in the practical use of them.

The instabilities caused by local optimization methods gave impetus to developing global techniques for torque optimization.[5,6] Although the global techniques result in stable optimal solutions, and guarantee the zero joint velocities at end of motion, the main drawbacks are the amount of computation required, especially for a larger number of degrees of freedom, and the fact that global solutions are unsuitable for on-line motion control.

In this article, we introduce a local technique for minimization of joint torques. A balancing technique is utilized in the approach, where a solution of joint torque-minimization is balanced against a solution of joint velocity-minimization. Minimization of joint velocities in the approach can prevent occurrence of high joint velocities and guarantee the joint velocities at end of motion to be near zero, thus resulting in stable arm motions. Computer simulations were executed to verify the performance of the proposed local torque optimization technique.

## 2. FORMULATION OF THE TECHNIQUE

As previously mentioned, results obtained by conventional approaches agree that local torque-minimization is inconsistent with global stability. It has been pointed out that instabilities are due to high joint velocities, which obviously are caused by the torque-minimization performance.[1–4] Hence, if a means can be found to balance properly between the torque-minimization and the velocity-minimization, the performance of the technique could be increased in the sense of global stability. A balancing scheme of joint torque-minimization against joint acceleration-minimization was developed,[7] wherein joint acceleration-minimization prevents a big change in joint velocities. The approach has improved the conventional approaches somewhat, but never guarantees the joint velocities at end of motion to be zero. Directly balancing joint torque-minimization against velocity-minimization prevents the high norm of joint velocities, and thus should give better performance.

Torque-based formulation (*the relation between the joint torques and the end-effector accelerations*) can be derived from equations (1) and (3),[8] which is expressed as

$$\ddot{\mathbf{x}}_0 + \mathbf{J}_\tau(\mathbf{c} + \mathbf{g}) = \mathbf{J}_\tau \tau, \tag{4}$$

where $\mathbf{J}_\tau$ is called the inertia-based Jacobian matrix, given by $\mathbf{J}_\tau = \mathbf{J}\mathbf{H}^{-1}$, and $\ddot{\mathbf{x}}_0 = \ddot{\mathbf{x}} - \dot{\mathbf{J}}\dot{\mathbf{q}}$.

The solution for local torque minimization can be directly derived from equation (4), on the basis of pseudoinverse. This is given by

$$\tau = \mathbf{J}_\tau^+[\ddot{\mathbf{x}}_0 + \mathbf{J}_\tau(\mathbf{c} + \mathbf{g})] + (\mathbf{I} - \mathbf{J}_\tau^+\mathbf{J}_\tau)\psi, \tag{5}$$

where $\mathbf{J}_\tau^+$ is pseudoinverse of the inertia-based Jacobian $\mathbf{J}_\tau$, and $\psi$ is an arbitrary vector that is used for optimizing other criteria in torque-based formulation.

It has been pointed out that the unstable arm motion comes from high joint velocities. The buildup of high joint velocities is due to the lack of a kinematic criterion in local optimization techniques to reflect the change of joint velocities. Thus, to generate stable arm motion, the velocities should be directly managed in the local optimization approach. Based on the author's former results,[9] we know that the solution to minimize the norm of joint velocities in acceleration level is given by

$$\ddot{\mathbf{q}} = \mathbf{J}^+\ddot{\mathbf{x}}_0 + (\mathbf{I} - \mathbf{J}^+\mathbf{J})\dot{\mathbf{J}}^+\dot{\mathbf{x}}, \tag{6}$$

where $\dot{\mathbf{J}}^+ = (\mathbf{I} - \mathbf{J}^+\mathbf{J})\dot{\mathbf{J}}^T(\mathbf{J}\mathbf{J}^T)^{-1} - \mathbf{J}^+\dot{\mathbf{J}}\mathbf{J}^+$.

Substituting equation (6) for equation (5), we can obtain the nullspace vector $\psi$ by

$$\begin{aligned}\psi = {} & (\mathbf{I} - \mathbf{J}_\tau^+\mathbf{J}_\tau)\mathbf{H}\mathbf{J}^+\ddot{\mathbf{x}}_0 \\ & + (\mathbf{I} - \mathbf{J}_\tau^+\mathbf{J}_\tau)\mathbf{H}(\mathbf{I} - \mathbf{J}^+\mathbf{J})\dot{\mathbf{J}}^+\dot{\mathbf{x}} \\ & + (\mathbf{I} - \mathbf{J}_\tau^+\mathbf{J}_\tau)(\mathbf{c} + \mathbf{g}),\end{aligned} \tag{7}$$

wherein $(\mathbf{I} - \mathbf{M}^+\mathbf{M})^+ = \mathbf{I} - \mathbf{M}^+\mathbf{M}$, $(\mathbf{I} - \mathbf{M}^+\mathbf{M})(\mathbf{I} - \mathbf{M}^+\mathbf{M}) = \mathbf{I} - \mathbf{M}^+\mathbf{M}$ are used.

Equation (7) is the nullspace vector in the torque-based formulation for minimization of the norm of joint velocities. If we introduce a scalar factor $\alpha$ ($0 \le \alpha \le 1$) between the solution of torque-minimization given by the pseudoinverse of the torque-based formulation and the solution of velocity-minimization given by the homogeneous solution of the torque-based formulation, we can obtain the joint torques that are given by

$$\begin{aligned}\tau &= \mathbf{J}_\tau^+[\ddot{\mathbf{x}}_0 + \mathbf{J}_\tau(\mathbf{c} + \mathbf{g})] + \alpha(\mathbf{I} - \mathbf{J}_\tau^+\mathbf{J})\psi \\ &= \{\mathbf{J}_\tau^+ + \alpha(\mathbf{I} - \mathbf{J}_\tau^+\mathbf{J}_\tau)\mathbf{H}\mathbf{J}^+\}\ddot{\mathbf{x}}_0 \\ &\quad + \alpha(\mathbf{I} - \mathbf{J}_\tau^+\mathbf{J}_\tau)\mathbf{H}(\mathbf{I} - \mathbf{J}^+\mathbf{J})\dot{\mathbf{J}}^+\dot{\mathbf{x}} \\ &\quad + \{\alpha\mathbf{I} + (1 - \alpha)\mathbf{J}_\tau^+\mathbf{J}_\tau\}(\mathbf{c} + \mathbf{g}).\end{aligned} \tag{8}$$

Simplifying equation (8) by $\mathbf{J}_\tau = \mathbf{J}\mathbf{H}^{-1}$ and $\mathbf{J}\mathbf{J}^+ = \mathbf{I}$, we obtain the balanced solution between the torque-minimization and the velocity-minimization, which is given by

$$\begin{aligned}\tau = {} & \alpha\{\mathbf{H}\mathbf{J}^+\ddot{\mathbf{x}}_0 + \mathbf{H}(\mathbf{I} - \mathbf{J}^+\mathbf{J})\dot{\mathbf{J}}^+\dot{\mathbf{x}} + (\mathbf{c} + \mathbf{g})\} \\ & + (1 - \alpha)\{\mathbf{J}_\tau^+\ddot{\mathbf{x}}_0 + \mathbf{J}_\tau^+\mathbf{J}_\tau(\mathbf{c} + \mathbf{g})\}.\end{aligned} \tag{9}$$

From equation (9), we know that the solution of torque-minimization is balanced by the solution of velocity-minimization through the weighting factor $\alpha$. Therefore, when $\alpha = 0$ only the torque-minimization is performed, on the other hand, when $\alpha = 1$ only the velocity-minimization is performed.

## 3. COMPUTER SIMULATIONS

To guarantee comparability with the previous results, we used a planar RRR mechanism geometry. Link lengths ($\ell_1 = \ell_2 = \ell_3 = 1$ *m*), link masses ($m_1 = m_2 = m_3 = 10$ kg), inertia parameter (*derived as a uniform beam*) and parameters of the trajectory (*straight Cartesian paths, starting and ending with zero velocity, constant* **bang-bang** *type acceleration of* 1 $m/s^2$ *and zero joint velocities at the beginning*) were chosen. The manipulator dynamics were integrated at a time interval of 5 ms, where gravity is neglected.

The control system was simulated utilizing the Resolved Acceleration Control method[10] with position and velocity error feedback-gains of 200 rad/m · $s^2$ and 30 rad/m · s, respectively. The full-rank pseudoinverse matrix $\mathbf{M}^+$ is calculated from the equation

$$\mathbf{M}^+ = \mathbf{M}^T(\mathbf{M}\mathbf{M}^T)^{-1}. \tag{10}$$

Programs were written in C and were run on a Gateway2000 (4DX2-66V) microcomputer.

First of all, the weighting factor $\alpha$ has to be defined. The first thought was to utilize a constant weighting factor or the switching technique. However, the constant weighting factor is difficult to determine, since it should be changed depending upon the task to generate better performance. The switching technique generates discontinuous joint torque. In this article, we define the weight factor $\alpha$ by analyzing the Jacobian matrix. In the example of a 3-DOF planar RRR manipulator, there exist three virtual 2-DOF manipulators. The determinates of each virtual 2-DOF manipulator Jacobian matrix are given by

$$d_{12} = \ell_1\ell_2(s_{12}c_1 - c_{12}s_1),$$
$$d_{23} = \ell_2\ell_3(s_{123}c_{12} - c_{123}s_{12}),$$
$$d_{13} = \ell_1\ell_3(s_{123}c_1 - c_{123}s_1),$$

where

$$s_1 = \sin(q_1),\ s_{12} = \sin(q_1 + q_2),\ s_{123} = \sin(q_1 + q_2 + q_3),$$
$$c_1 = \cos(q_1),\ c_{12} = \cos(q_1 + q_2),\ c_{123} = \cos(q_1 + q_2 + q_3).$$

For 3-DOF manipulator, where the manipulator end-effector position is given, 1-DOF of redundancy will change arm configuration through self-motion (*homogeneous part of solution*), just as with a 1-DOF 4-link planar mechanism. For a 4-link planar mechanism there exists a singular posture depending upon each link length. The singular posture, called *Dead point*, occurs at the arm configuration shown in Figure 1. In the case of *dead point* of the arm configuration, one of the $d_{12}$, $d_{23}$, $d_{13}$ must be zero. The 1-DOF of redundancy in the 3-DOF manipulator is lost in this case, thus the minimization of joint torques cannot be performed through self-motion, and only the manipulator end-effector demand will be guaranteed. Velocity-minimization can be executed even in this singular posture to guarantee the manipulator end-effector demand. Thus, in the case of no extra DOF in the redundant manipulator, the weighting

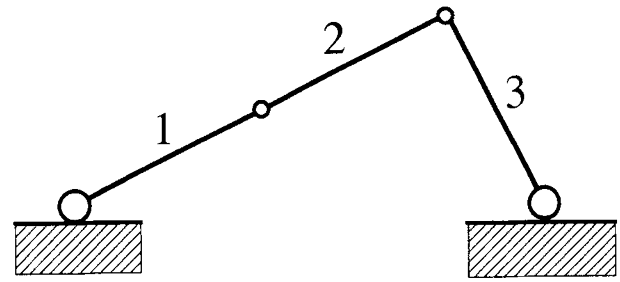


**Figure 1.** Configuration of a 4-link planar mechanism in singular posture (dead point), in this case $d_{12} = 0$.

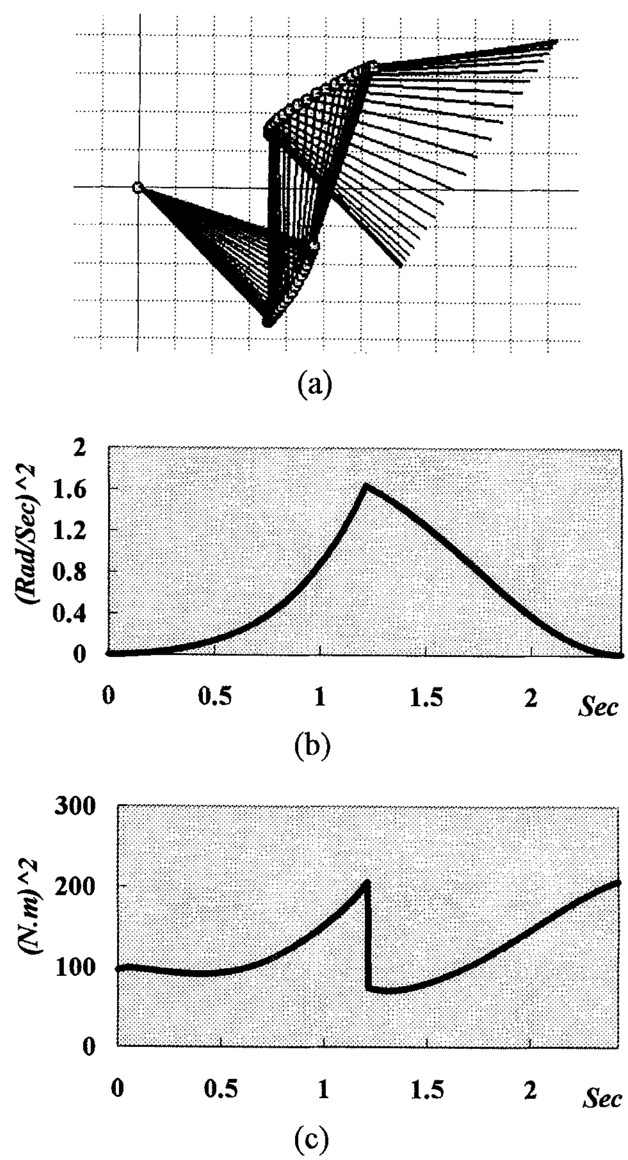


**Figure 2.** Simulation results by the proposed technique where the solution of joint torque-minimization is balanced against the solution of joint velocity-minimization. (a) Arm motion. (b) Profile of the square sum of joint velocities. (c) Profile of the square sum of joint torques.

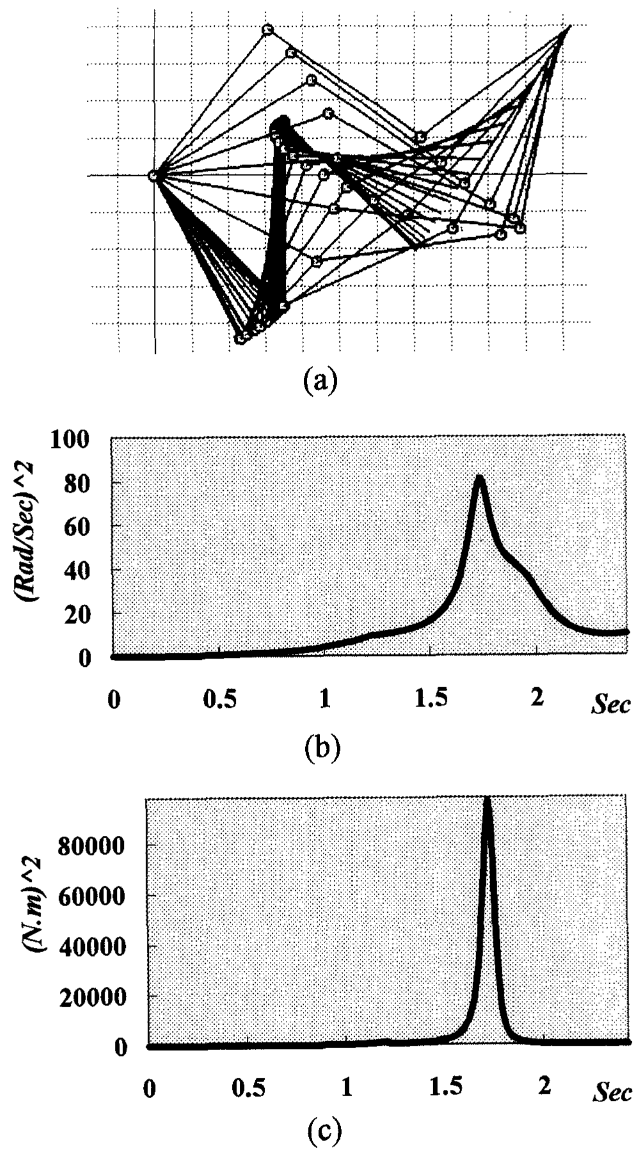


**Figure 3.** Simulation results by the nullspace approach where only the joint torques are minimized.[1] (a) Arm motion. (b) Profile of the square sum of joint velocities. (c) Profile of the square sum of joint torques.

factor $\alpha$ will be set to 1 because there is no possibility of minimizing joint torques further.

Therefore, the weighting factor $\alpha$ can be defined as a continuous function based on the arm configuration. For example, we can select the function

$$\alpha = 1 - k \min(d_{12}, d_{23}, d_{13}) \qquad (11)$$

as weighting factor, where $k$ is an arbitrary scalar that guarantees the value of weighting factor $\alpha$ is larger than zero ($k = 0.002$ *was used in this paper*).

In Figure 2 simulation data is shown for a trajectory that is obtained by the proposed technique, where the solution of joint torque-minimization is balanced against the solution of joint velocity-minimization. For comparison, the results from the nullspace approach,[1] where only joint torque is minimized, and the balancing approach between the solution of joint torque-minimization and the solution of joint acceleration-minimization[7] are also provided in Figures 3 and 4. It can be seen that the proposed local torque optimization technique yields better performance from the stability point of view.

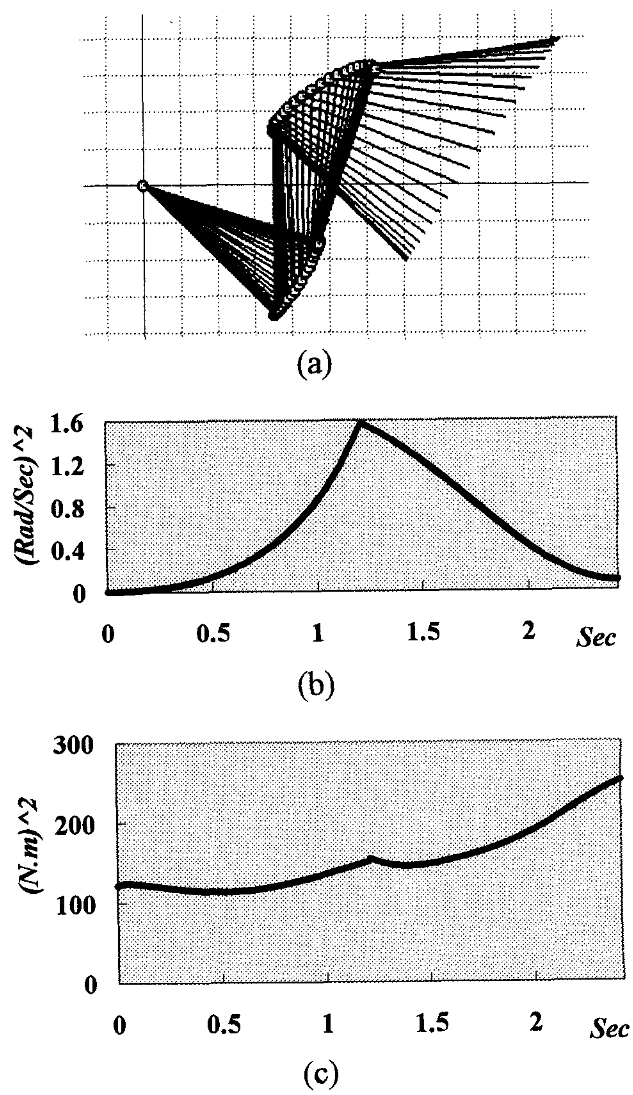


**Figure 4.** Simulation results by the balancing technique where the joint torque-minimization is balanced by the joint acceleration-minimization.[7] (a) Arm motion. (b) Profile of the square sum of joint velocities. (c) Profile of the square sum of joint torques.

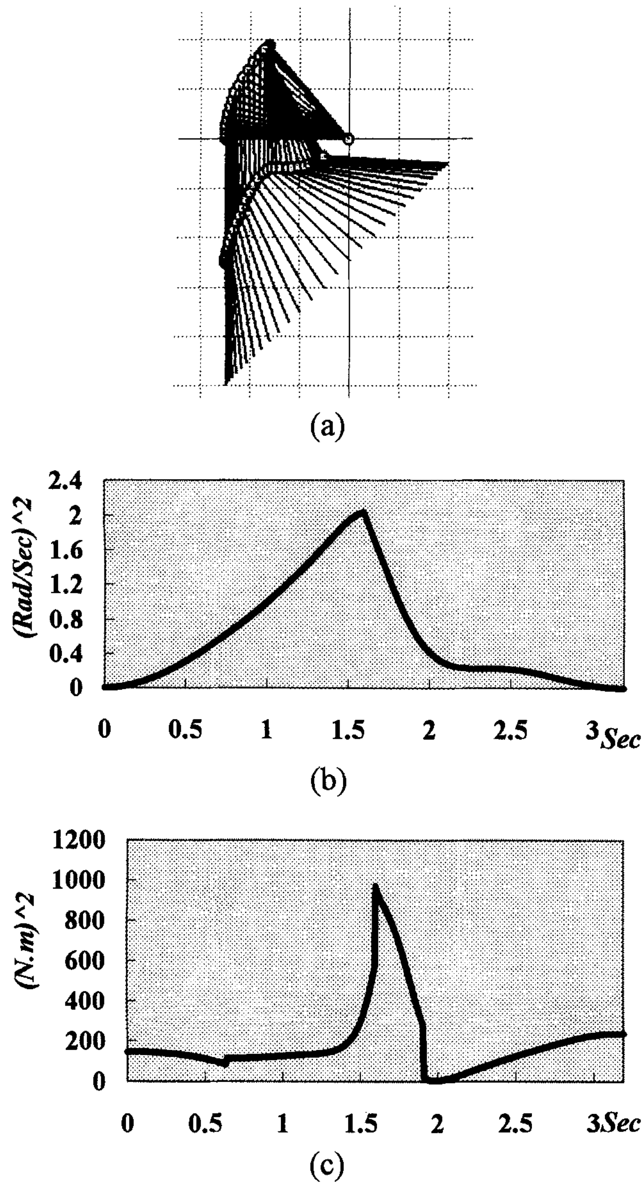


**Figure 5.** Simulation results by the proposed approach. (a) Arm motion. (b) Profile of the square sum of joint velocities. (c) Profile of the square sum of joint torques.

Moreover, it is shown from the square sum of joint velocities that the proposed approach guarantees the joint velocities at end of motion to be near zero. This point is important in the practical use of redundant manipulators. The simulation data for other trajectories is shown in Figures 5–10. Similar results to those shown in Figures 2, 3, and 4 were obtained. However, it should be noted that the proposed technique in most cases, but not always, yields better performance in torque-optimization. There is a trade-off between the local optimization of joint torques and the zero end-motion velocities. If we need a solution minimizing the joint torques, the end-motion velocities must not be zero or near zero, and vice versa. To remove this trade-off, the global approaches should be applied in which the joint torques are minimized under the constraints of the zero end-motion velocities (*boundary condition*). However, as stated in the introduction, because we are concentrating on the point of view of real-time control, only the local optimization techniques were discussed in this article. The zero joint velocities at end of motion

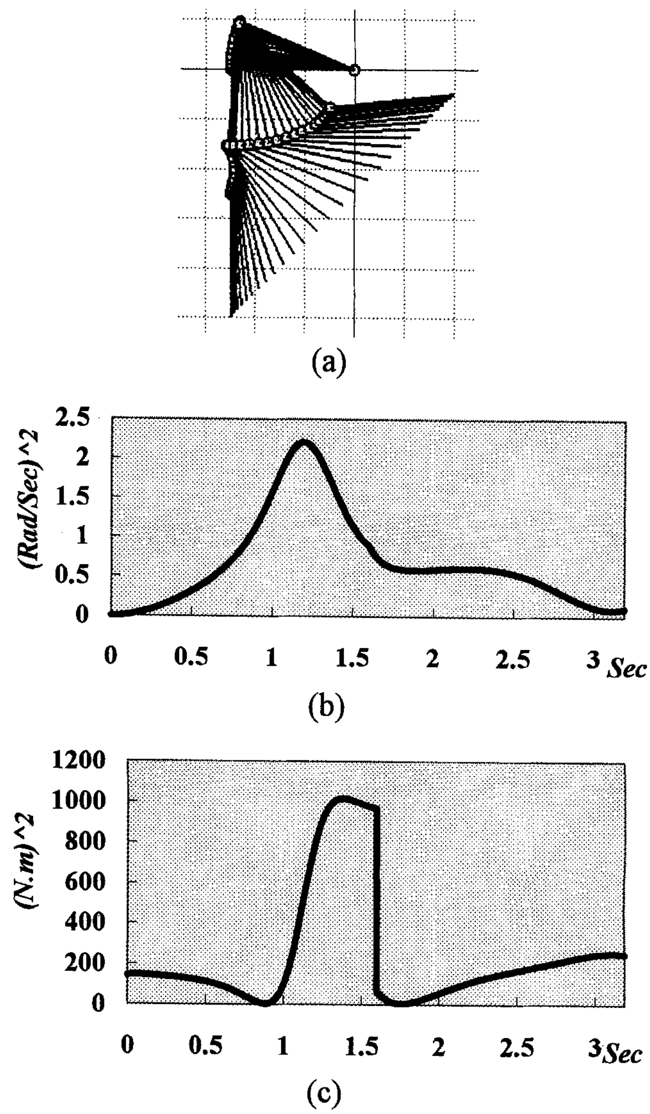


**Figure 6.** Simulation results by the nullspace approach.[1] (a) Arm motion. (b) Profile of the square sum of joint velocities. (c) Profile of the square sum of joint torques.

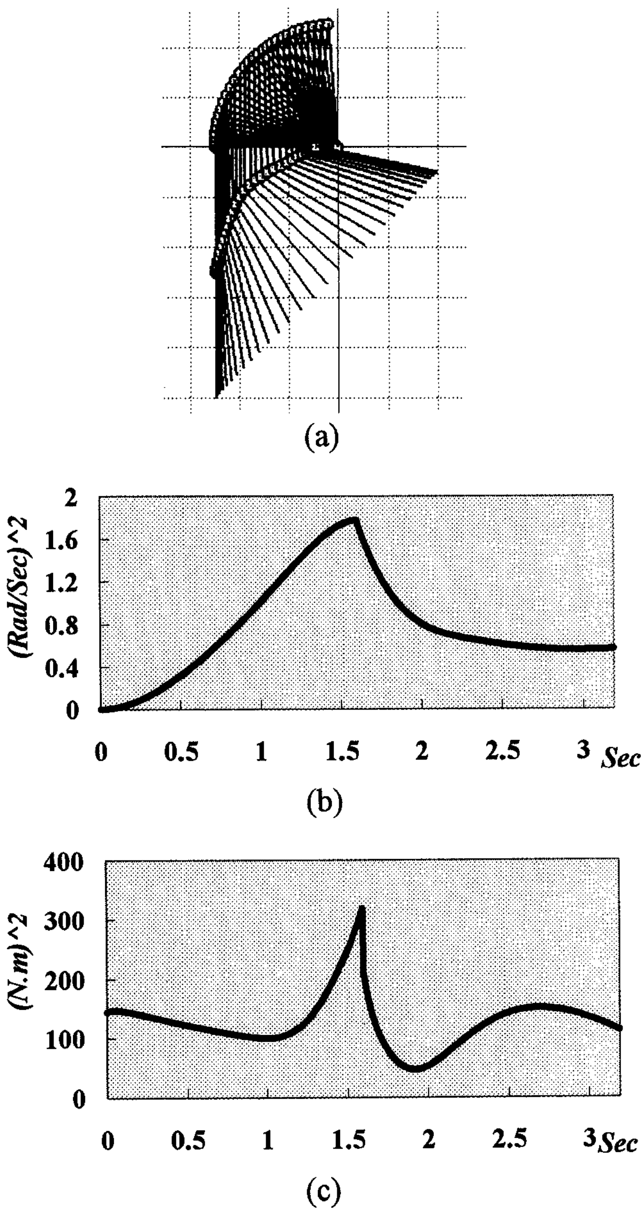


**Figure 7.** Simulation results by the balancing approach.[7] (a) Arm motion. (b) Profile of the square sum of joint velocities. (c) Profile of the square sum of joint torques.

and the stable arm motion have been emphasized on the control of a real redundant manipulator at the cost of torque optimization.

## 4. CONCLUSIONS

A local optimization technique that stabilizes the existing local torque-optimization solutions for redundant manipulators was proposed in this article. The technique is based on a balancing scheme, which balances a solution of joint torque-minimization against a solution of joint velocity-minimization. Computer simulations were performed on a 3-DOF planar RRR manipulator to compare the performance of the proposed local torque optimization technique with existing ones. For some long trajectories, the previous local torque optimization techniques showed instability due to the buildup of high joint velocities. Moreover, the joint velocities at end of

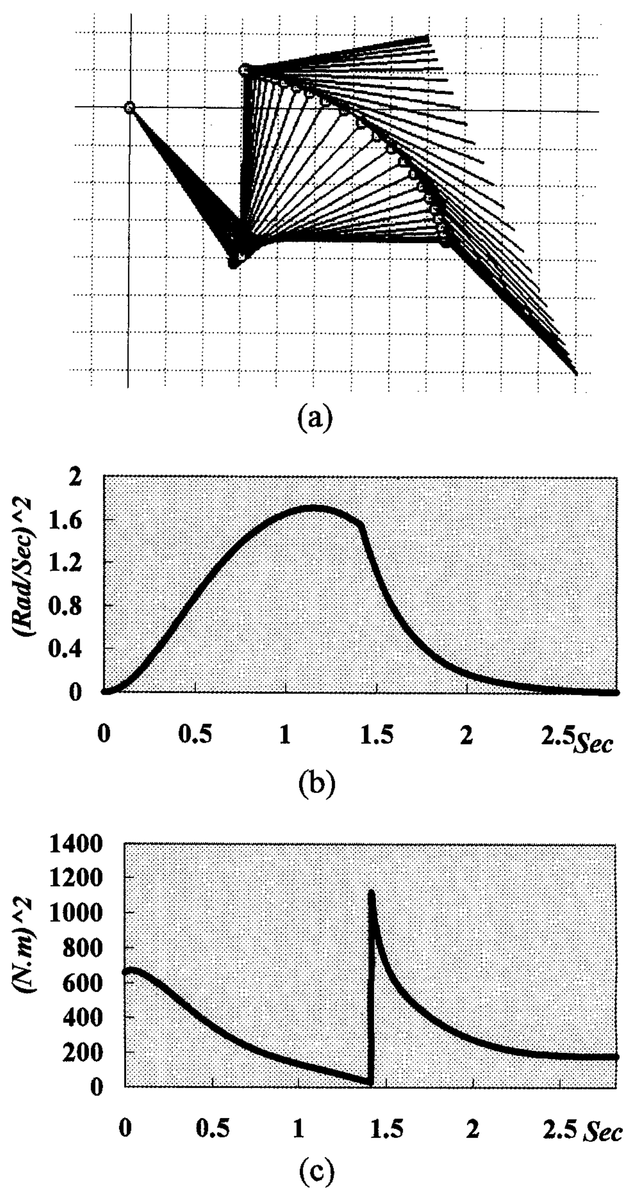


**Figure 8.** Simulation results by the proposed approach. (a) Arm motion. (b) Profile of the square sum of joint velocities. (c) Profile of the square sum of joint torques.

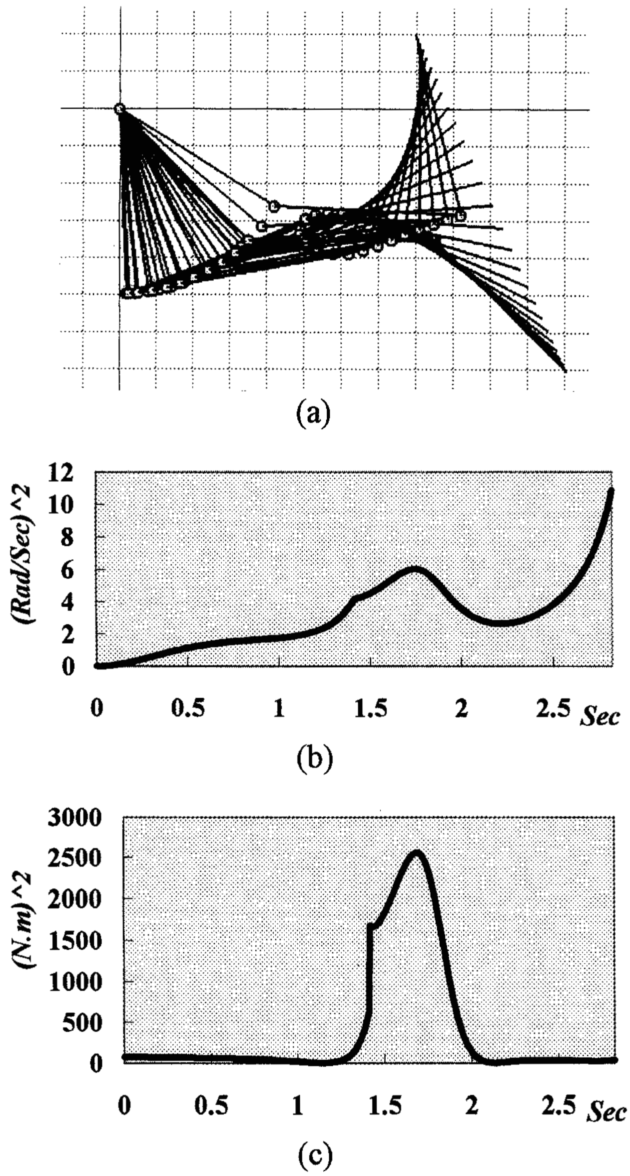


**Figure 9.** Simulation results by the nullspace approach.[1] (a) Arm motion. (b) Profile of the square sum of joint velocities. (c) Profile of the square sum of joint torques.

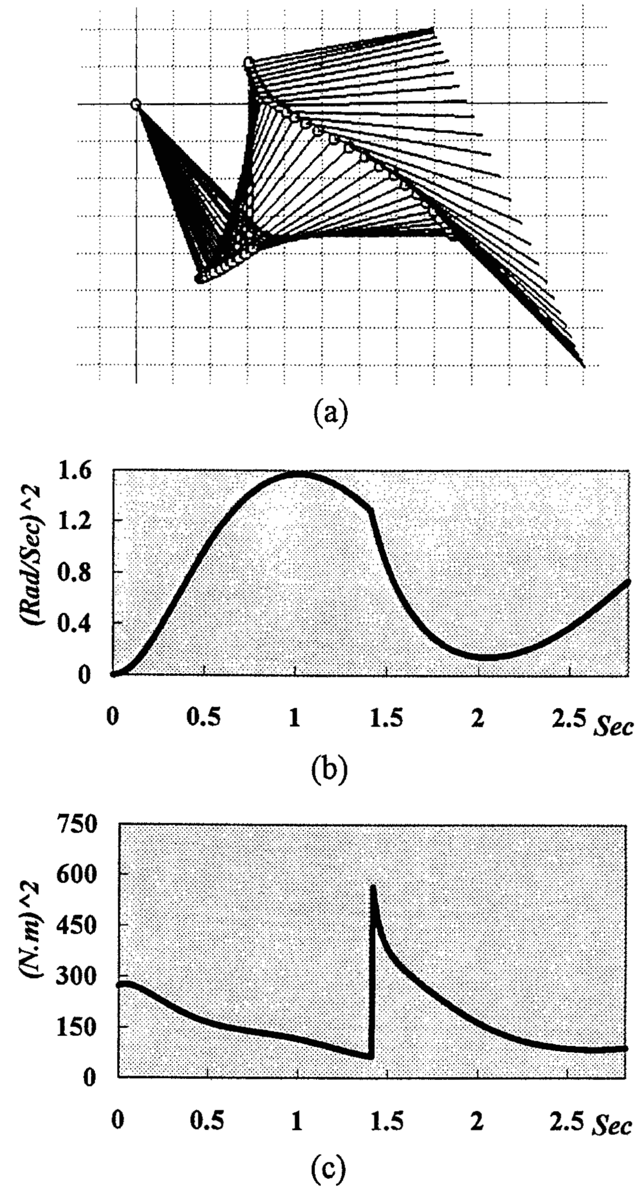


**Figure 10.** Simulation results by the balancing approach.[7] (a) Arm motion. (b) Profile of the square sum of joint velocities. (c) Profile of the square sum of joint torques.

motion were not guaranteed to be near zero in the previous approaches, thus there have been problems in practical uses. To avoid the buildup of high joint velocities and to guarantee near zero joint velocities at end of motion, the joint velocities are minimized in the proposed approach together with torque-minimization. This has been shown to be more stable than the previous techniques, and guarantees the joint velocities at end of motion to be near zero.

The author would like to thank the Editor and Referees for helpful and constructive comments on the previous version of this article.

## REFERENCES

1. J. M. Hollerbach and K. C. Suh, "Redundancy resolution of manipulators through torque optimization," *Proc. IEEE Int. Conf. Rob. Autom.*, St. Louis, MO, 1985, pp. 1016–1021.
2. J. M. Hollerbach and K. C. Suh, "Redundancy resolution of manipulators through torque optimization," *IEEE Trans. Rob. Autom.* **3**(4), 308–315, 1987.
3. K. Kazerounian and A. Nedungadi, "Redundancy resolution of robotic manipulators at the acceleration level," *Proc. IFToMM World Congress*, Sevilla, Spain, 1987, pp. 1207–1211.
4. A. Nedungadi, "Torque optimization of redundant

manipulators," *Proc. ISIR*, Chicago, IL, 1987, pp. 10(65)–10(83).
5. Y. Nakamura and H. Hanafusa, "Optimal redundancy control of robotic manipulators," *Int. J. Rob. Res.*, **6**(1), 32–42, 1987.
6. K. C. Suh and J. M. Hollerbach, "Local versus global torque optimization of redundant manipulators," *Proc. IEEE Int. Conf. Rob. Autom.*, Raleigh, NC, 1987, pp. 619–624.
7. S. Ma, D. Nenchev, and S. Hirose, "Improving local torque optimization techniques for redundant robotic mechanisms," *J. Rob. Syst.*, **8**(1), 75–91, 1991.
8. S. Ma, "Local torque optimization of redundant manipulators in torque-based formulation," *Proc. IEEE Int. Conf. Ind. Electron. Control Instrument.* (IECON'94), Bologna, Italy, 1994, Vol. 2, pp. 697–702.
9. S. Ma and S. Hirose, "Dynamic redundancy resolution of redundant manipulators with local optimization of a kinematic criterion," *Proc. Asian Conf. Rob. Appl.*, Beijing, China, 1994, pp. 236–243.
10. J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "Resolved acceleration control of mechanical manipulators," *IEEE Trans. Autom. Control*, **25**(3), 468–474, 1980.