Proceedings of the 12th KACC, October 1997

Global Torque Optimization of Redundant Manipulator Using Dynamic Programming

°Ick-Chan Shim and Yong-San Yoon

Dept. of Mechanical Eng., KAIST, 373-1 Kusong-dong, Yusong-gu, Taejon 305-701, KOREA
Tel : +82-42-869-3062; E-mail : sic@hms.kaist.ac.kr

Abstracts In this paper, the torque optimization of a kinematically redundant manipulator for minimizing the torque demands is discussed. The minimum torque solution based on a local optimization has been known to encounter the instability problem and then the global torque optimization was suggested as one of the alternatives. Herein, by adopting the infinity-norm rather than the 2-norm for the magnitude of torques, we are to propose a new cost function more advantageous to the avoidance of torque limits. By the way, a solution to the global torque optimization formulated with the new cost function can not be obtained by the previous methods due to their difficulties such as inability to treat discontinuous cost functions and various constraints on the joint variables. Thus, to overcome those deficiencies, we are developing a new approach using the dynamic programming. The effectiveness of the proposed method is shown through simulation examples for a 3-link planar redundant manipulator.

Keywords Redundant Manipulator, Global Torque Optimization, Dynamic Programming, Infinity-Norm

1. Introduction

To make a robot more versatile and dexterous, the redundancy is often required. Recently, the research on a manipulator having such redundant degrees of freedom has received many interests. So far, the singularity avoidance, the obstacle avoidance, and the joint limit avoidance, etc. were proposed for the utilization of redundancy.[1] Most of those studies are based on the local optimization of certain objective functions. However, the methods may sometimes face to unexpected problems because the local optimization does not consider the resulting future states. Particularly, the local torque optimization severely suffers from the instability problem known as the unrealistic whipping-like motion.[2,3]

The method based on the global optimization can be applied to the redundancy resolution in order to remedy the instability problem. Since it has a great advantage of guaranteeing the global optimality, the resulting motion is always stable. However, its real-time control is impossible due to a heavy amount of computation. Fortunately, for many industrial applications requiring repetitive motion tasks, the global optimization may be well suited for finding a solution by the off-line trajectory planning.[3]

In 1987, Nakamura and Hanafusa[4] have introduced a formalism for the global optimization through Pontryagin's minimum principle. They globally optimized a cost functional being a sum of the norm of joint velocities and the manipulability over the entire motion interval. Suh and Hollerbach[3] obtained a solution of the global torque optimization by using the calculus of variations. Also, using the variational method, Kazerounian and Nedungadi[5] derived the equation of redundancy resolution at the acceleration level for the global minimization of the kinetic energy. In fact, the variational method is theoretically equivalent with the method through Pontryagin's minimum principle. The results of both formulation are the two-point boundary value problems which are difficult to solve. Even for the shooting method commonly adopted to solve a boundary value problem, it may often fail to converge in a case of multiple shooting parameters. In the area of the time-optimal control for a non-redundant manipulator, it has been reported that the numerical algorithms do not converge to a solution.[6] Moreover, those formulations cannot treat discontinuous cost functions and various constraints on the joint variables, velocities, and torques which are often encountered practically.

The study of this paper aims at overcoming the above deficiencies. We are to adopt the technique of dynamic programming which has few restrictions on the cost function and the constraints. The possibility of this approach was already addressed by Shin and Mckay[7], Singh and Leu[8]. They conducted the time-optimal control of a non-redundant manipulator for a given path of the end-effector using the dynamic programming.

In Section 2, the formulation of global torque optimization at the torque level is described together with the local method. In Section 3, we develop a new algorithm using the dynamic programming approach. Section 4 gives several numerical simulations and the results for a planar 3-link redundant manipulator to verify effectiveness of the algorithm. Finally, the conclusions are presented in Section 5.

2. Formulation of Global Torque Optimization

Generally, the kinematic relations between the joint variables and the task variables of a robot can be given as follows[10]:

$$x = f(\theta) \tag{1}$$

$$\dot{x} = J(\theta)\dot{\theta} \tag{2}$$

$$\ddot{x} = J(\theta)\ddot{\theta} + \dot{J}(\theta,\dot{\theta})\dot{\theta} \tag{3}$$

where x is the m-dimensional vector specifying the end-effector variables, θ is the n-dimensional vector specifying the joint variables, and $J \in \Re^{m\times n}$ represents the Jacobian matrix. Also, it is well known that the dynamics of a manipulator can be written as

$$\tau = M(\theta)\ddot{\theta} + h(\theta,\dot{\theta}) \tag{4}$$

where $\tau \in \Re^{n}$ specifies the joint torque vector, $M \in \Re^{n\times n}$ is the inertia matrix, and $h \in \Re^{n}$ is the nonlinear force vector containing Coriolis, centrifugal, and gravity forces. Usually, the individual components of the joint torque vector are restricted to be available only within ranges between its upper and lower bounds given by the hardware limits for actuators as follows:

$$-\tau_i^{ub} \le \tau_i \le \tau_i^{ub} \qquad i = 1,\cdots,n \tag{5}$$

where the upper and lower bounds are assumed to have the same absolute value for convenience.[10]

For a redundant manipulator, Eqs. (1), (2), and (3) are underdetermined forms since $n > m$ with the desired end-effector path $x_d(t)$ given. Then, the inverse kinematics of the redundant manipulator has an infinite number of solutions due to its redundancy. Choosing one of such an infinite number of solutions under certain objective is called the resolution of redundancy. One important example of the redundancy resolution considering the manipulator's dynamics is the torque optimization. Actually, in controlling the manipulator's motion, the saturation of the actuator's force or torque often occurs due to a requirement of excessive control force or torque. Since the torque saturation degrades remarkably the control performance, its avoidance is very crucial. Also, through the torque optimization, the redundant manipulator can achieve larger acceleration capability for the end-effector.

Hollerbach and Suh[2] first suggested the local torque optimization for reducing the torque demands in the least squares sense. With a consideration of the weighted magnitude of joint torques, its formulation can be expressed as follows at the torque level:

$$\begin{aligned} \min \quad & \Psi_{L2}(\tau) = \tfrac{1}{2}\tau^T W^T W \tau \\ \text{s. t.} \quad & JM^{-1}\tau = \ddot{x}_d - \dot{J}\dot{\theta} + JM^{-1}h \end{aligned} \tag{6}$$

where the joint variables and velocities are known at any instant of time, and the constraint equation can be derived by substituting the joint accelerations from Eq. (4) into Eq. (3). Since the constraints are linear with respect to the joint torques, the optimal solution to Eq. (6) can be easily determined as the least-squares solution. Also, since the major objective of the torque optimization is to avoid exceeding the torque limits, the weighting matrix should be set as $W = diag\{(\tau_i^{ub})^{-1}\}$ in order to normalize the magnitude of the torque vector.

By the way, the normalized 2-norm of torques is not completely consistent with the torque limits given as Eq. (5). Thus, it is more reasonable to minimize the ∞-norm of the joint torque vector than the 2-norm for the avoidance of torque limits[9]:

$$\begin{aligned} \min \quad & \Psi_{L\infty}(\tau) = \|W\tau\|_\infty \\ \text{s. t.} \quad & JM^{-1}\tau = \ddot{x}_d - \dot{J}\dot{\theta} + JM^{-1}h \end{aligned} \tag{7}$$

where the ∞-norm of a n-dimensional vector x is defined as $\|x\|_\infty = \max_{1\le i\le n}|x_i|$. For convenience, Eqs. (6) and (7) will be called as LTO-2 and LTO-∞ problems hereafter.

As known previously, the local minimization of the joint torques has the instability problem for the end-effector's long trajectories. Many researchers have developed methods to prevent the instability.[9] Although those improved methods based on the local optimization are effective in many cases, only the global torque optimization provides the stable motions in any cases and outperforms the local methods. From the previous studies[3,4], the global torque optimization can be given as follows:

$$\begin{aligned} \min \quad & \Psi_{G2}[\theta,\dot{\theta},\tau] = \int_0^{t_f} \tfrac{1}{2}\tau^T W^T W \tau \, dt \\ \text{s. t.} \quad & \frac{d}{dt}\begin{bmatrix}\theta \\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\dot{\theta} \\ M^{-1}(\tau - h)\end{bmatrix} \\ & JM^{-1}\tau = \ddot{x}_d - \dot{J}\dot{\theta} + JM^{-1}h \end{aligned} \tag{8}$$

Correspondingly to Eq. (7), we are also going to minimize the maximum value of $\|W\tau\|_\infty$ over the entire motion from the concept of L_∞-function norm:

$$\begin{aligned} \min \quad & \Psi_{G\infty}[\theta,\dot{\theta},\tau] = \max_{0\le t\le t_f} \|W\tau\|_\infty \\ \text{s. t.} \quad & \frac{d}{dt}\begin{bmatrix}\theta \\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\dot{\theta} \\ M^{-1}(\tau - h)\end{bmatrix} \\ & JM^{-1}\tau = \ddot{x}_d - \dot{J}\dot{\theta} + JM^{-1}h \end{aligned} \tag{9}$$

where the cost functional to be minimized is defined as the maximum value of the weighted ∞-norm of torques over the time interval $[0, t_f]$. As similar to the local methods, let's call Eqs. (8) and (9) as GTO-2 and GTO-∞ problems respectively. It is noteworthy that the cost functional of the GTO-∞ has a discontinuity. That is, the ∞-norm of torques is not differentiable with respect to time.

3. Dynamic Programming Method for Global Torque Optimization

In this section, a new method for finding solutions to the problems GTO-2 and GTO-∞ using the dynamic programming approach is developed. Its motivation is to develop a general algorithm capable of taking into account various types of cost functions and constraints.

The global optimization problems can be represented as the following general form in optimal control theory[11]:

$$\begin{aligned} \min \quad & \int \psi(x,u,t)\,dt \\ \text{s. t.} \quad & \dot{x} = f(x,u,t) \\ & x^- \le x \le x^+ \\ & u^- \le u \le u^+ \end{aligned} \tag{10}$$

To make up a state equation, we parameterize the redundancy of a manipulator by specifying $(n-m)$ independent joint variables. For the parameterization, the joint variables can be partitioned as

$$\theta = \begin{bmatrix}\tilde{\theta} \\ \bar{\theta}\end{bmatrix} \tag{11}$$

where the variables $\bar{\theta} \in \Re^{(n-m)}$ and $\tilde{\theta} \in \Re^m$ denote the independent and dependent joint variables respectively. The dependent joint variables $\tilde{\theta}$ can be given as $\tilde{\theta}(\bar{\theta}, x_d, t)$, a nonlinear algebraic function of the independent joint variables $\bar{\theta}$ and the desired path x_d, through the kinematic relation of Eq. (1). In other words, $\bar{\theta}$ together with x_d yield the unique joint configuration. Then, provided that $J = \begin{bmatrix}\tilde{J}_{m\times m} & \bar{J}_{m\times(n-m)}\end{bmatrix}$, the dependent joint velocities $\dot{\tilde{\theta}}$ are determined from Eq. (2) as

$$\dot{\tilde{\theta}} = \tilde{J}^{-1}(\dot{x}_d - \bar{J}\dot{\bar{\theta}}) \tag{12}$$

Defining $x \equiv \begin{bmatrix}\bar{\theta}^T & \dot{\bar{\theta}}^T\end{bmatrix}^T \in \Re^{2(n-m)}$ as a state vector, the state equation is as follows:

$$\frac{d}{dt}\begin{bmatrix}\bar{\theta} \\ \dot{\bar{\theta}}\end{bmatrix} = \begin{bmatrix}\dot{\bar{\theta}} \\ \begin{bmatrix}0_{(n-m)\times m} & I_{(n-m)}\end{bmatrix} M^{-1}(\tau - h)\end{bmatrix} \tag{13}$$

As explained in Section 2, the joint torques should satisfy the relation with the end-effector accelerations rewritten as

$$A(\theta)\tau = b(\theta,\dot{\theta},\ddot{x}_d) \tag{14}$$

where $A(\theta) \equiv JM^{-1}$ and $b(\theta,\dot{\theta},\ddot{x}_d) \equiv \ddot{x}_d - \dot{J}\dot{\theta} + JM^{-1}h$. If the joint torques is parameterized as $\tau = \begin{bmatrix}\tilde{\tau}^T & \bar{\tau}^T\end{bmatrix}^T$ consistently with Eq. (11), then the dependent joint torques $\tilde{\tau} \in \Re^m$ can be obtained in terms of the independent joint torques $\bar{\tau} \in \Re^{(n-m)}$ as

$$\tilde{\tau} = \tilde{A}^{-1}(b - \bar{A}\bar{\tau}) \tag{15}$$

where A is assumed to be partitioned as $A = \begin{bmatrix}\tilde{A}_{m\times m} & \bar{A}_{m\times(n-m)}\end{bmatrix}$. Finally, by introducing a vector of control variables $u \equiv \bar{\tau} \in \Re^{(n-m)}$, the derivation of the state equation $\dot{x} = f(x,u,t)$ is completed. Note that the constraint equation (14) on the control variables does not explicitly appear in the optimal control problem (10).

Now, the dynamic programming technique can be easily applied to the optimal control problem [11]. For the stage index k of the discrete times over the time interval $[0, T_f]$, let $\Delta\Psi(x(k),k)$ denote the incremental cost between stages k and $k+1$, and $\Psi^*(x(k),k)$ denote the minimum cost functional to reach the final stage from the state at stage k. Applying the optimality principle yields the recurrence equation,

$$\Psi^*(x(k),k) = \min_{u(k)} \{\Delta\Psi(x(k),k) + \Psi^*(x(k+1),k+1)\} \quad (16)$$

where $x(k+1)$ results from the state equation for given $x(k)$ and $u(k)$. The solution of this recurrence equation is an optimal control policy $u^*(x(k),k)$, which is obtained by trying all admissible control values at each admissible state value. To perform the computational procedure, it is necessary to discretize the admissible state and control values specified by the bounds on x and u into a finite number of levels.

In general, the dynamic programming method may be severely restricted by the *curse of dimensionality*. Hence we paid much attention to reduction of the number of state and control variables. By such parameterization of redundancy as above, there are only $2r$ state and r control variables for the degree of redundancy $r = n - m$. Although the proposed method may become impractical due to the curse of dimensionality as the degree of redundancy increases, for one or two degrees of redundancy, it must be still applicable.

4. Simulations and Discussions

In this section, we are presenting some simulations for the global torque optimization using the dynamic programming method. For the simulations, a three-link planar manipulator shown in Fig. 1 is used which is encountered frequently in literature. All links of the manipulator are modeled by thin uniform rods with length of 1 m and mass of 10 kg. The upper and lower torque limits are set equal in magnitude as 54, 24, and 6 $N\cdot m$ for joints 1, 2, and 3. Then, the weighting matrix W is given as $diag\{1/54, 1/24, 1/6\}$ by considering these torque limits.

For the verification of the dynamic programming method, we first examine one of the examples treated by Suh and Hollerbach[3]. The desired end-effector's path is a straight-line bang-bang motion with constant acceleration and deceleration of $[1 \;\; 1]^T \, m/s^2$ over the first and last half of the movement for the duration time $t_f = 2$ seconds, and the initial configuration of the manipulator is $\theta_0 = [-45^\circ \; 135^\circ \; -45^\circ]^T$ with $\dot{\theta}_0 = 0$. In the previous work, this trajectory was applied with some modifications: $\ddot{x}_d(t)$ was patched in the middle for one tenth of the total duration with a third order polynomial to make $\ddot{x}_d(t)$ continuous. However, as mentioned before, the proposed method using dynamic programming can treat easily such discontinuity. Fig. 2 and 4 show the link motion and the torque curves obtained by the proposed method. As expected, except small difference in the middle, these results are almost identical with those of the Suh and Hollerbach. Also, as a by-product of the proposed method, the cost map for initial conditions can be obtained as shown in Fig. 3. From this cost map, the best initial condition can be determined.

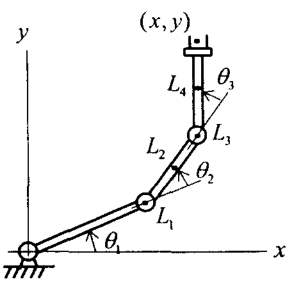


Fig. 1 Three-link planar manipulator

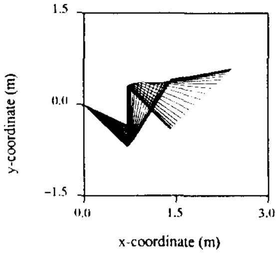


Fig. 2 Link motion of the GTO-2 solution for the straight path.

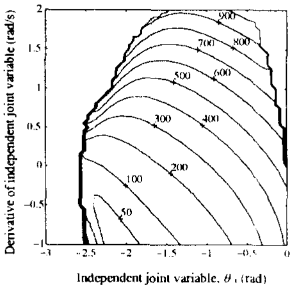


Fig. 3 Cost map for initial states

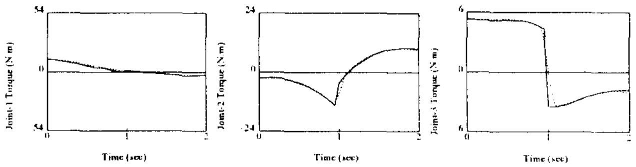
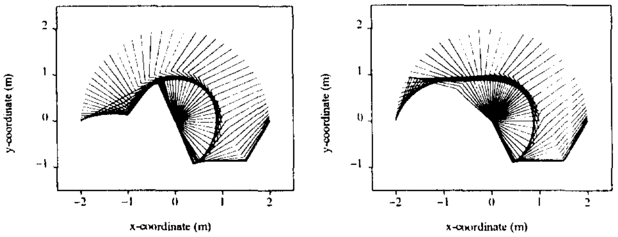


Fig. 4 Joint torques of the GTO-2 solution for the straight path. (—— proposed method, ····· Suh and Hollerbach method)

Next, to conduct a comparison in performance between the solutions of GTO-2 and GTO-∞, we consider a half-circular path with center at origin and radius of 2 m. The end-effector moves on the circumference with constant angular acceleration and deceleration about origin of $4\pi/t_f^2 \; rad/s^2$ over the first and last half of the rotation for the duration time $t_f = 5$ seconds. At starting, the manipulator has the state given as $\theta_0 = [-60^\circ \; 60^\circ \; 60^\circ]^T$ and $\dot{\theta}_0 = 0$. By applying the dynamic programming method, we could obtain the link motions and the torque curves for GTO-2 and GTO-∞ shown in Fig. 5 and 6. For the comparison of two solutions, the 2-norm and the ∞-norm of the torque vector for them are also plotted in Fig. 7a and 7b respectively. From Fig. 7a, regarding to the cost $\sum\sqrt{\tau^T W \tau}$, the solution of GTO-∞ has larger value 5.91 than 4.72 for that of GTO-2. However, for the peak value of the normalized ∞-norm of torques in Fig. 7b, the solution of GTO-∞ and GTO-2 has 0.87 and 1.21 respectively. Since these peak values represent the cost $\max\|W\tau\|_\infty$ for the formulation of GTO-∞, they should be lower than one in order to satisfy the torque limits given. Therefore, we can say that the solution of GTO-∞ is more advantageous than that of GTO-2 for the avoidance of torque limits.



(a) GTO-2 solution (b) GTO-∞ solution

Fig. 5 Link motions for the half-circular path.

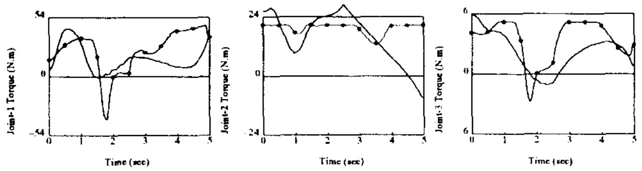
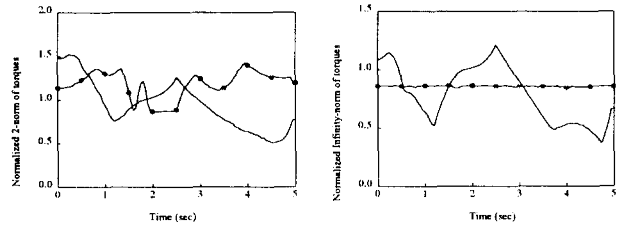


Fig. 6 Joint torques for the half-circular path
(— GTO-2 solution, -•- GTO-∞ solution)



(a) Normalized 2-norm (b) Normalized ∞-norm

Fig. 7 Normalized norms of the joint torques for the GTO-2 and GTO-∞ solutions in Fig. 6.

Lastly, to show the ability of the dynamic programming method capable of treating various constraints, GTO-∞ considering the obstacle avoidance is performed. As illustrated in Fig. 8, the manipulator is required to trace out the tip path specified by a curve a-b-c-d while avoiding three obstacles. At the same time, the torque loading at the joint actuators is being minimized. To prevent the manipulator itself from colliding with the obstacles, the constraints have been defined as follows:

$$d_{ij} = \overline{O_i L_j} > r_i \quad \text{for } i = 1,\cdots,3,\ j = 1,\cdots,4$$

where d_{ij} is a distance from the center of the i-th obstacle O_i to the point L_j specified on the links in Fig. 1 and r_i is a radius of the obstacle. Note that the obstacle-1 represents the base equipment of the manipulator and thus its collision with only link-1does not occur. From Fig. 8, we can know that the obtained solution well satisfies both the constraints for the obstacles and the torque limits.

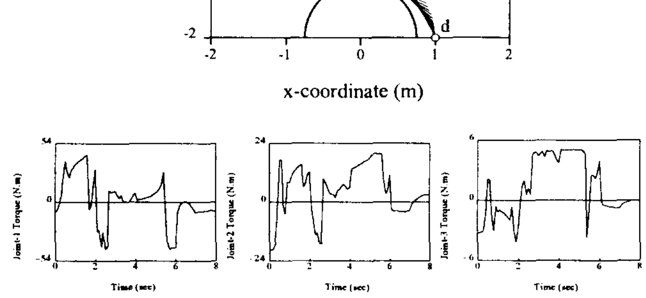


Fig. 8 Link motion and torque profiles for an example of the obstacle avoidance.

5. Conclusions

In this study, for the torque limit avoidance as the primary objective of the torque optimization, a new formulation GTO-∞ based on the concept of L_∞-function norm is suggested. Since GTO-∞ has discontinuous cost function, its solution cannot be produced by the previous methods based on Pontryagin's minimum principle or the calculus of variations. Thus, to solve GTO-∞ with various constraints, we have proposed a numerical method using the dynamic programming. The simulation results show that GTO-∞ is more advantageous than the previous formulation GTO-2, and that the dynamic programming method can treat various constraints such as obstacles and joint limits.

References

[1] D. N. Nenchev, "Redundancy resolution through local optimization: a review", *Journal of Robotic Systems*, vol. 6, no. 6, pp. 769-798, 1989.

[2] J. M. Hollerbach and K. C. Suh, "Redundancy resolution through torque optimization", *IEEE Journal of Robotics and Automation*, vol. 3, no. 4, pp. 308-316, 1987.

[3] K. C. Suh and J. M. Hollerbach, "Local versus global torque optimization of redundant manipulators", *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 619-624, 1987.

[4] Y. Nakamura and H. Hanafusa, "Optimal redundancy control of robot manipulators", *Int. Journal of Robotics Research*, vol. 6, no. 1, pp. 32-42, 1987.

[5] K. Kazerounian and Z. Wang, "Global versus local optimization in redundancy resolution of robotic manipulators", *Int. Journal of Robotics Research*, vol. 7, no. 5, pp. 3-12, 1988.

[6] J. E. Bobrow, S. Dubowsky and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths", *Int. Journal of Robotics Research*, vol. 4, no. 3, pp. 3-17, 1985.

[7] K. G. Shin and N. D. McKay, "A dynamic programming approach to trajectory planning of robotic manipulators", *IEEE Transaction on Automatic Control*, vol. 31, no. 6, 1986.

[8] S. Singh and M. C. Leu, "Optimal trajectory generation for robotic manipulators using dynamic programming", *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 109, pp. 88-96.

[9] I.-C. Shim and Y.-S. Yoon, "Stabilization constraint method for local torque optimization of redundant manipulator," *IEEE Int. Conference of Robotics and Automation*, pp. 2403-2408, 1997.

[10] T. Yoshikawa, *Foundations of Robotics: Analysis and Control*, The MIT Press, 1990.

[11] D. E. Kirk, *Optimal Control Theory: An Introduction*, Prentice-Hall Inc., 1970.