'89 KACC 1989. 10. 27～28

Near-Optimum Trajectory Planning for Robot Manipulators

Motoji YAMAMOTO, Shinya MARUSHIMA and Akira MOHRI

Department of Mechanical Engineering for Production,
Faculty of Engineering, Kyushu University 36,
6-10-1 Hakozaki, Higashi-ku, Fukuoka 812, JAPAN

Abstract - An efficient algorithm for planning near-optimum trajectory of manipulators is proposed. The algorithm is divided into two stages. The first one is the optimization of time trajectory with given spatial path. And the second one is the optimization of the spatial path itself. To consider the second problem, the manipulator dynamics is represented using the path parameter "s", then a differential equation corresponding to the dynamics is solved as two point boundary value problem. In this procedure, the gradient method is used to calculate improved input torques.

1. Introduction

Optimum trajectory planning for robot manipulators is one of the most essential problem to realize the off-line teaching systems for industrial robots. Although many algorithms of optimum trajectory planning for manipulators are proposed, the efficient ones for general cost have not necessarily been obtained yet. This paper treats this general trajectory planning problem which includes the optimization of a spatial path.

The trajectory planning problems, in which a limitation for a range of the joint angle is not considered and a work space is free from obstacles, are broadly divided into two categories. One is the case that a spatial path of a manipulator is pre-planned and another is the case that the spatial path is not given, so the optimization of the spatial path is required. We will treat the latter case.

As for the former case that the spatial path is given, Shin et al.[1], Vukobratović et al.[2] and Ozaki et al.[3] have proposed the methods which use Dynamic Programming. These methods can generate the optimum trajectory under the general cost function which includes the time optimal problem.

On the other hand, the latter general trajectory planning problem which includes the optimization of the spatial path is studied by some researchers. Geering et al.[4] and Yamamoto et al.[5] have proposed methods based on the maximum principle. Rajan [6] has proposed an approximate method in which the joint trajectory is approximated by spline function. However these methods are restricted to time optimal problem in which the cost function is the traveling time. The studies for general cost functions are little. The algorithm presented here is an approximate method for this problem. In this algorithm, the manipulator dynamics is represented by a path parameter "s" and a differential equation corresponding to the dynamics is solved as two point boundary value problem. To calculate the improved input torque, the gradient method is used in the algorithm.

2. Optimum Trajectory Planning Problem

In this section, we will formulate the general trajectory planning problem. The dynamics of the manipulator system is expressed by

$$H(\theta)\ddot{\theta} + h(\theta,\dot{\theta}) + g(\theta) + D\dot{\theta} = u \qquad (1)$$

Where $\theta \in \Re^n$ is a joint variable vector, $H(\theta) \in \Re^{n\times n}$ is an inertia moment matrix, $h(\theta,\dot{\theta}) \in \Re^n$ is a Coriolis and centrifugal forces vector, $D \in \Re^{n\times n}$ represents viscous friction coefficients, $g(\theta) \in \Re^n$ is a gravitation vector and $u \in \Re^n$ is an actuator driving torque/force vector. Usually the output of actual actuator is limited, so in this formulation, we may restrict the actuator driving torque/force as follows.

$$u_{i\ min} \leq u_i(t) \leq u_{i\ max} \qquad (2)$$
$$(i = 1, \cdots, n)$$

Where $u_{i\ min}$ and $u_{i\ max}$ are i-th lower and upper bounds of actuator driving torque/force. At the general trajectory planning problem in which obstacles in a work space are not considered, only initial and terminal conditions are given as follows.

$$\theta(0) = \theta_0 \quad \theta(t_e) = \theta_e$$
$$\dot{\theta}(0) = 0 \quad \dot{\theta}(t_e) = 0 \qquad (3)$$

Performance index for this problem is

$$J = \int_0^{t_e} f_0(\theta(t), \dot{\theta}(t), u(t))dt \qquad (4)$$

Here t_e is terminal time and it is not given.

Then our problem becomes to how to obtain the joint trajectory which minimizes the cost eq.(4) under the constraint eq.(2) and boundary conditions, eq.(3).

3. Path Parameter "s" and Spatial Path

In this section, we will make clear the difference of the spatial path and time trajectory. Then we will represent the manipulator dynamics by a path parameter.

Here we introduce a scalar parameter "s" which is a general length along the path for the tip of manipulator's end-effecter (See Fig.1). We call this parameter "s" as path parameter. $\theta(s)$ is called as spatial path when the path is represented by the path parameter "s". And $\theta(t)$ is called as time trajectory when it is represented by time "t". $\lambda(s)$ is defined to connect the path parameter "s" with time "t" directly, then dt is described as

$$dt = \frac{1}{\lambda(s)}ds \qquad (5)$$

Obviously the condition $\lambda(s) > 0$ is necessary. Time t can be described with $\lambda(s)$ as

$$t(s) = \int_0^s \frac{1}{\lambda(s)}ds \qquad (6)$$

There exist the following relations between the spatial path $\theta(s)$ and time trajectory $\theta(t)$.

$$\theta(t) = \theta(s) \qquad (7)$$

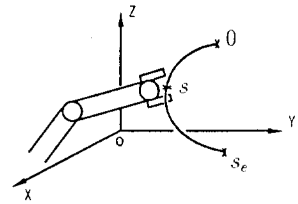

Fig.1 Path parameter "s" and spatial path

$$\dot{\theta}(t) = \lambda(s)\frac{d\theta(s)}{ds} \qquad (8)$$

$$\ddot{\theta}(t) = \lambda^2(s)\frac{d^2\theta(s)}{ds^2} + \lambda(s)\frac{d\lambda(s)}{ds}\frac{d\theta(s)}{ds} \qquad (9)$$

Substituting eqs.(7)~(9) into eq.(1) yield

$$u(s) = \lambda^2(s)[H(\theta(s))\frac{d^2\theta(s)}{ds^2} + h(\theta(s), \frac{d\theta(s)}{ds})]$$
$$+\lambda(s)[\frac{d\lambda(s)}{ds}H(\theta(s))\frac{d\theta(s)}{ds} + D\frac{d\theta(s)}{ds}] + g(\theta(s)) \qquad (10)$$

Actuator driving torque/force is formally described as $u(s)$ instead of $u(t)$ because "s" is directly related with "t" by eq.(6). Substituting eqs.(7)~(10) into eq.(4) yield

$$J = \int_0^{s_e} \frac{1}{\lambda(s)} f_0(\theta(s), \frac{d\theta(s)}{ds}, \lambda(s), u(s))ds \qquad (11)$$

From this equation, we find that if a spatial path is given, the value of cost function could be calculated by determining $\lambda(s)$. Therefore the trajectory planning problem of this type is to generate the optimal trajectory of $\lambda(s)$. The algorithms to generate the optimal $\lambda(s)$ have been proposed by some researchers. Those algorithms are called "MCTP (Minimum Cost Trajectory Planning) algorithm". Most of the researchers use one of the optimization techniques, Dynamic Programming in their methods. However, their methods are not necessarily effective because they are very time consuming. Another approximate method is also proposed by Ozaki et al. in which the trajectory of $\lambda(s)$ is described by B–spline function. In our method, we use the approximate method in view of accuracy of solution and calculation time. This algorithm is described in the following section.

4. MCTP Algorithm

In this section, the MCTP algorithm is presented briefly. First, we represent the trajectory $\lambda(s)$ by B-spline function. Dividing $0 \leq s \leq s_e$ into m sections $(s_0(=0), \cdots, s_m(=s_e))$, $\lambda(s)$ in $s_k \leq s \leq s_{k+1}$ is expressed by B–spline function (order four) as follows.

$$s(l) = s_k + l\Delta s_k \qquad 0 \leq l \leq 1 \qquad (12)$$

$$\lambda(s) = \lambda(l) = \sum_{j=-1}^{2} \hat{\lambda}_{k+j} N_{k+j,4}(l) \qquad (13)$$

$$\frac{d\lambda(s)}{ds} = \frac{1}{\Delta s_k} \sum_{j=-1}^{2} \hat{\lambda}_{k+j} \frac{dN_{k+j,4}(l)}{dl} \qquad (14)$$

Where $\Delta s_k = s_{k+1} - s_k$. Base functions of B–spline are

$$\begin{aligned} N_{k-1,4}(l) &= (-l^3 + 3l^2 - 3l + 1)/6 \\ N_{k,4}(l) &= (3l^3 - 6l^2 + 4)/6 \\ N_{k+1,4}(l) &= (-3l^3 + 3l^2 + 3l + 1)/6 \\ N_{k+2,4}(l) &= l^3/6 \\ &(k = 0, \cdots, m) \end{aligned} \qquad (15)$$

Control points $\hat{\lambda}_k(k = -1, 0, 1, m-1, m, m+1)$ of $\lambda(s)$ are chosen so as to satisfy the boundary conditions, eq.(3). This means $\lambda(0) = \lambda(s_e) = 0$ from eq.(8). It is obvious that $\lambda(s)$ and $\frac{d\lambda(s)}{ds}$ depend on only the control points $\hat{\lambda}_k(k = 2, \cdots, m-2)$. Consequently if in the case that the spatial path $\theta(s)$ and $\frac{d\theta(s)}{ds}$ are given, the cost function can be calculated only by defining the control points $\hat{\lambda}_k$. With above discussions, the approximate algorithm, in the case that the spatial path is given, is described as follows.

1. Give the spatial joint path $\theta(s)$.
2. Give appropriately large values for $\hat{\lambda}_k(k = 2, \cdots, m-2)$ and generate the initial path of $\lambda(s)$.
3. Search the control points $\hat{\lambda}_k(k = 2, \cdots, m-2)$ so as to make the cost J minimum under the constraints.
4. Repeat searching $\hat{\lambda}_k$. When the decrease of cost J equals 0 or nearly equals 0, the algorithm is stopped.

5. Optimization of Spatial Path

In this section, the efficient method which improves the spatial path using the dynamics of manipulator represented by the path parameter "s" is described.

Firstly, because the inertia matrix $H(\theta)$ is nonsingular, we may rewrite the eq.(10) as

$$\frac{d^2\theta(s)}{ds^2} = H(\theta(s))^{-1}\{\frac{u(s) - g(s)}{\lambda^2(s)} - \frac{1}{\lambda(s)} \cdot [\frac{d\lambda(s)}{ds} H(\theta(s)) \frac{d\theta(s)}{ds} + D\frac{d\theta(s)}{ds}] - h(\theta(s), \frac{d\theta(s)}{ds})\} \qquad (16)$$

Here we define the state variable vector $x \in \Re^{2n}$ as follows.

$$\begin{cases} x_i = \theta_i \\ x_{n+i} = \frac{d\theta_i}{ds} \end{cases} \qquad (17)$$
$$(i = 1, \cdots, n)$$

Then, eq.(16) is expressed as

$$\frac{dx(s)}{ds} = f(x(s), \lambda(s), \frac{d\lambda(s)}{ds}, u(s)) \qquad (18)$$

This non-linear differential equation corresponds to the manipulator dynamics represented by the path parameter "s".

Now we consider how to solve the differential equation with given $\lambda(s)$, $\frac{d\lambda(s)}{ds}$ and $u(s)$. As for the boundary conditions, two terminal joint positions $\theta(0) = \theta_0$ and $\theta(s_e) = \theta_e$ are specified and two terminal values of $\frac{d\theta(s)}{ds}$ are not specified. That means there is two-point boundary value problem with given $x_i(0)$, $x_i(s_e)$ $(i = 1, \cdots, n)$ and with free $x_i(0)$, $x_i(s_e)$ $(i = n+1, \cdots, 2n)$. In the eq.(18), $\lambda(s)$, $\frac{d\lambda(s)}{ds}$ and $u(s)$ can not be calculated until $\theta(s)$ is defined, so we can not solve the eq.(18) with this form. Therefore, based on the eq.(18), the updating formula to get the spatial path $x(s)$ is

$$\frac{dx^{j+1}(s)}{ds} = f(x^{j+1}(s), \lambda^j(s), \frac{d\lambda^j(s)}{ds}, u^{*j}(s)) \qquad (19)$$

Where j represents the iteration and $\lambda^j(s)$ is the optimum trajectory of $\lambda(s)$ for the joint path $(x^j(s)\ (= \{[\theta^j(s)]^{\mathrm{T}}, [\frac{d\theta^j(s)}{ds}]^{\mathrm{T}}\}^{\mathrm{T}})$ at the former iteration. $\lambda^j(s)$ is obtained for the spatial path $x^j(s)$ by using the MCTP algorithm. $u^{*j}(s)$ must be improved input torque, so we calculate this $u^{*j}(s)$ using the gradient method. To obtain the gradient function we may represent eq.(19) with the expression of time "t" as

$$\frac{dx^{j+1}(t)}{dt} = f(x^{j+1}(t), u^{*j}(t)) \qquad (20)$$

The cost function, eq.(4) is modified to obtain the gradient function as follows.

$$J = F(x(t_e)) + \int_0^{t_e} f_0(x(t), u(t))dt \quad (21)$$

First term of the right hand side is added to satisfy the boundary condition of terminal point x_e.

Now let $\eta^j(t)$ denote the gradient function for the j-th iteration. By using the gradient function $\eta^j(t)$, the improved input torque $u^{*j}(t)$ is given as

$$u^{*j}(t) = u^j(t) - \alpha\eta^j(t) \quad (22)$$

Where α is an appropriate positive constant. The gradient function $\eta^j(t)$ is calculated as

$$\eta^j(t)^{\mathrm{T}} = (\frac{\partial F(x(t_e))}{\partial x(t_e)})^{\mathrm{T}}\Phi^j(t_e, t)(\frac{\partial f}{\partial u})^{\mathrm{T}}_{x^j,u^j}$$

$$+(\frac{\partial f_0}{\partial u})^{\mathrm{T}}_{x^j,u^j} + \int_t^{t_e}(\frac{\partial f_0}{\partial x})^{\mathrm{T}}_{x^j,u^j}\Phi^j(\tau, t)(\frac{\partial f}{\partial u})^{\mathrm{T}}_{x^j,u^j}d\tau$$

$$0 \le t \le t_e \quad (23)$$

Where Φ is the state transition matrix for the following linear time varying system.

$$\frac{d(\delta x)}{dt} = (\frac{\partial f}{\partial x})_{x^j,u^j}\delta x + (\frac{\partial f}{\partial u})_{x^j,u^j}\delta u$$

$$\begin{cases} \delta x = x - x^j \\ \delta u = u - u^j \end{cases} \quad (24)$$

$\Phi^j(t_e, t)$ can be obtained by solving the following matrix differential equation.

$$\Phi^j(t_e, t_e) = E$$

$$\frac{d}{dt}[\Phi^j(t_e, t)] = -\Phi^j(t_e, t)A^j(t)$$

$$0 \le t \le t_e \quad (25)$$

Where

$$A^j(t) = (\frac{\partial f}{\partial x})_{x^j,u^j} \quad (26)$$

and E represents a unit matrix. In our problem, terminal time t_e is not specified. So the gradient function, eq.(23) is not appropriate if it is used just this form. Therefore we modify the gradient function as follows.

$$\eta^j(t)^{\mathrm{T}}_{\pm\Delta t} = (\frac{\partial F(x(t_e \pm \Delta t))}{\partial x(t_e \pm \Delta t)})^{\mathrm{T}}\Phi^j(t_e \pm \Delta t, t)$$

$$\cdot(\frac{\partial f}{\partial u})^{\mathrm{T}}_{x^j,u^j} + (\frac{\partial f_0}{\partial u})^{\mathrm{T}}_{x^j,u^j}$$

$$+\int_t^{t_e \pm \Delta t}(\frac{\partial f_0}{\partial x})^{\mathrm{T}}_{x^j,u^j}\Phi^j(\tau, t)(\frac{\partial f}{\partial u})^{\mathrm{T}}_{x^j,u^j}d\tau$$

$$0 \le t \le t_e \pm \Delta t \quad (27)$$

From this equation, we obtain two improved inputs $u^{*j}(t)$'s corresponding to $\eta^j(t)_{+\Delta t}$ and $\eta^j(t)_{-\Delta t}$. Then we get two spatial paths $x_+^{j+1}(s)$(the case for $+\Delta t$) and $x_-^{j+1}(s)$(the case for $-\Delta t$) by solving the differential equation (20) using two $u^{*j}(t)$'s. We chose the spatial path which gives smaller value of cost J as the improved spatial path at the next iteration.

6. Algorithm for Trajectory Planning

We will show an algorithm for general trajectory planning problem based on the method proposed in the previous section. To execute this algorithm using the updating equation (19), initial spatial path ($x^0(s)$ ($= \{[\theta^0(s)]^{\mathrm{T}}, [\frac{d\theta^0(s)}{ds}]^{\mathrm{T}}\}^{\mathrm{T}}$) is needed. In this algorithm, we use a straight line in the joint space as an initial spatial path for simplicity. This path is expressed as

$$\theta^0(s) = \frac{\theta_e - \theta_0}{s_e}s + \theta_0 \quad (28)$$

$$\frac{d\theta^0(s)}{ds} = \frac{\theta_e - \theta_0}{s_e} \quad (29)$$

Based on the result of previous section, the whole algorithm for general trajectory planning problem is given as follows.

1. Obtain the initial spatial path $\theta^0(s)$ from eqs. (28) and (29). Let J^0 be sufficiently large positive value. Set $j = -1$.
2. Set $j = j + 1$.
3. Obtain the optimum $\lambda(s)$, $\frac{d\lambda(s)}{ds}$ by applying the MCTP algorithm to the joint path $\theta^j(s)$. Calculate $\theta, \dot{\theta}, \ddot{\theta}, u$ from eqs. (7)~(10). Calculate costJ^j from eq.(4). If two J^j's (the case of $+\Delta t$ and $-\Delta t$) exist for $\theta^j(s)$, smaller one is used in the following procedures. Set $\Delta J = J^j - J^{j-1}$. If ΔJ is sufficiently small, the algorithm ends.
4. Set $\theta, \dot{\theta} \to x$, then calculate the state transition matrix $\Phi^j(t_e \pm \Delta t, t)$ from eq.(25). Calculate the gradient functions $\eta^j(t)_{\pm\Delta t}$ using eq.(27).
5. Set α in eq.(22) appropriately small positive value. Calculate two $u^{*j}(t)$'s in the case of $\pm\Delta t$ from eq.(22). If $u^{*j}(t)$ excesses the

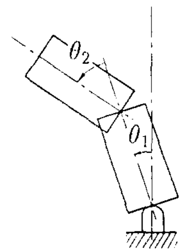

Fig.2 Structure of manipulator

limitation of the input considering the constraint eq.(2), $u^{*j}(t)$ is modified with the following manner.

$$\begin{cases} \text{if } u_i^{*j}(t) > u_{i\ max} & \text{then } u_i^{*j}(t) = u_{i\ max} \\ \text{if } u_i^{*j}(t) < u_{i\ min} & \text{then } u_i^{*j}(t) = u_{i\ min} \end{cases}$$

$$(i = 1, \cdots, n) \qquad (30)$$

6. Obtain an improved spatial path $x^{j+1}(s)$ solving the eq.(20) as two point boundary value problem.
7. Set $x^{j+1}(s) \rightarrow \theta^{j+1}(s)$. Go to 3.

7. Example

In this section, the proposed algorithm for trajectory planning problem is applied to a two link manipulator which is illustrated in Fig.2. This manipulator moves in the vertical plane. The dynamics of this manipulator is

$$\{2I + 2ml^2(3 + 2\cos\theta_2)\}\ddot{\theta}_1 + \{I + ml^2(1 + 2\cos\theta_2)\}\ddot{\theta}_2 - 2ml^2\sin\theta_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) - mgl\{3\sin\theta_1 + \sin(\theta_1 + \theta_2)\} = u_1$$

$$\{I + ml^2(1 + 2\cos\theta_2)\}\ddot{\theta}_1 + (I + ml^2)\ddot{\theta}_2 + 2ml^2\sin\theta_2\dot{\theta}_1^2 - mgl\sin(\theta_1 + \theta_2) = u_2 \qquad (31)$$

Where I denotes the inertia moment ($I = 4.167 \times 10^{-3} kg \cdot m^2$) about the axis through the center of gravity for each link, m denotes the mass ($m = 1.0kg$) for each link and l denotes a half of each link ($l = 0.1m$). Let each element of the term D which represents viscous friction be 0 for simplicity. The cost (eq.(4)) is given as

$$f_0 = 1 + \beta(u_1^2 + u_2^2) \qquad (32)$$

In this example the value of β is 0.03. The cost function to evaluate the gradient (eq.(21)) is given as

Table 1 Constraints and boundary conditions

	Joint 1	Joint 2
$u_{i\ max}$ (N·m)	10.0	5.0
$u_{i\ min}$ (N·m)	-10.0	-5.0
$\theta_i(0)$ (deg)	45.0	-90.0
$\theta_i(t_e)$(deg)	-90.0	0.0

$$J = \gamma\sum_{i=1}^{4}(x_i(t_e) - x_{ei})^2 + \int_0^{t_e} f_0(t)dt \qquad (33)$$

Where the value of γ is 0.1 and x_{ei} denotes the given terminal condition. The limitations of driving torques and boundary conditions of joint trajectory are given in Table 1. Fig.3 shows the initial spatial path in work space. This is a straight line path in joint space. The dot-dash-lines represent the center lines of manipulator links. The cost for this path is $J = 0.95859$ applying the MCTP algorithm. Fig.4 and Fig.5 show the spatial path at 1-st iteration of algorithm and the spatial path obtained by proposed method (at 6-th iteration) respectively. Corresponding values of cost are J^1 =0.72659 and J^6=0.57766. Fig.6 ~ Fig.8 are optimum inputs for the spatial paths $\theta^0, \theta^1, \theta^6$ (Fig.3 ~ Fig.5) obtained by MCTP respectively. Table 2 shows the outline of the convergence for this example. In the table, index values and the values of α in eq.(22) are given at each iterations.

8. Conclusion

An approximate method has been proposed for the general trajectory planning problem of manipulator including the optimization of spatial path. In this method, the manipulator dynamics is represented by the path parameter "s" which does not depend on time. The algorithm is constructed based on the manipulator dynamics represented by "s", so the given boundary conditions at initial and terminal points can be easily satisfied. This is an advantage of this method. Furthermore, because the method takes the form of feasible method, the calculation with this algorithm may be stopped at any time if the desired accuracy of a solution is obtained. As an example, the proposed method is applied to

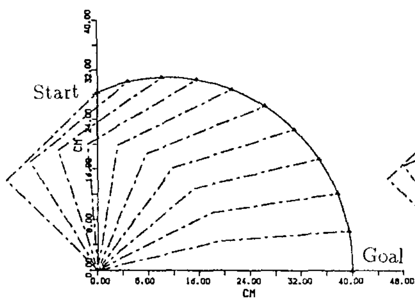

Fig.3 Initial spatial path $\theta^0(s)$

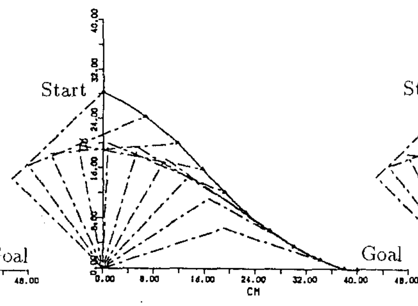

Fig.4 Spatial path at 1-st iteration $\theta^1(s)$

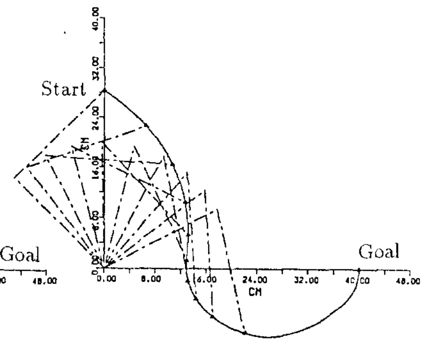

Fig.5 Improved spatial path $\theta^6(s)$

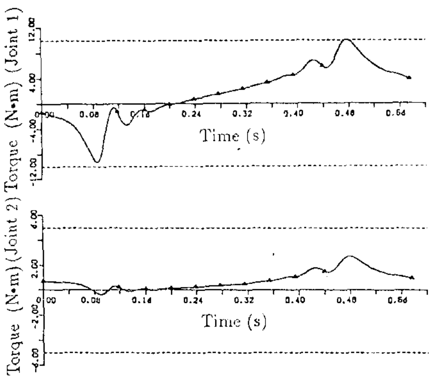

Fig.6 Optimum input torque for $\theta^0(s)$

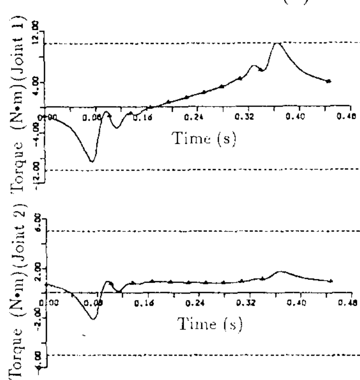

Fig.7 Optimum input torque for $\theta^1(s)$

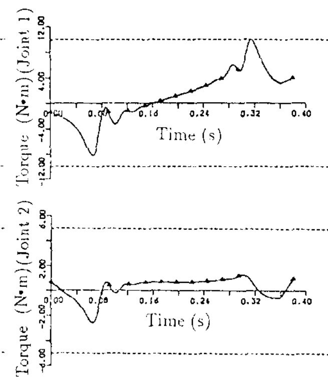

Fig.8 Optimum input torque for $\theta^6(s)$

a two link manipulator. Then the effectiveness of this method is shown by the result of this example.

Acknowledgment

We wish to thank the Ministry of Education of Japan for the financial support by a grant (No. 63790285) in Aid for Science Research.

References

[1] K.G. Shin and N.D. Mckay: A Dynamic Programming to Trajectory Planning of Robot Manipulators, IEEE Trans. Automatic Control, **AC-31**-6, 491/500 (1986).

[2] M. Vukobratović and M. Kirćanski: A Method for Optimal Synthesis of Manipulation Robot Trajectories, Trans. of ASME, **G-104**-6, 188/193 (1982).

[3] H. Ozaki, M. Yamamoto and A. Mohri: Optimal and Near-Optimal Manipulator Joint Trajectories with a Preplanned Path, Procs. of the 26th IEEE Conf. on Decision and Control, 1029/1034 (1987).

[4] H.P. Geering, L. Guzzela, S.A.R. Hepner and C.H. Onder: Time-Optimal Motions of Robots in Assembly Tasks, IEEE Trans. Automatic Control, **AC-31**-6, 512/518 (1986).

[5] M. Yamamoto and A. Mohri: Planning of quasi-minimum time trajectories for robot manipulators (Generation of a bang-bang control), Robotica, **7**-1, 43/47 (1989).

[6] V.T. Rajan: Minimum Time Trajectory Planning, Procs. of IEEE Intl. Conf. on Robotics and Automation, 759/764 (1985).

Table 2 Index value J and value α for each path

	Index value J	α	Corresponding figures
$\theta^0(s)$ (Initial path)	0.958 59	—	Fig.3, Fig.6
$\theta^1(s)$	0.726 59	0.30	Fig.4, Fig.7
$\theta^2(s)$	0.584 49	0.38	
$\theta^3(s)$	0.589 54	0.01	
$\theta^4(s)$	0.587 10	0.01	
$\theta^5(s)$	0.582 04	0.01	
$\theta^6(s)$	0.577 66	0.01	Fig.5, Fig.8