International Journal of Control, Automation, and Systems, vol. 4, no. 1, pp. 10-16, February 2006

Time-optimal Trajectory Planning for a Robot System under Torque and Impulse Constraints

Bang Hyun Cho, Byoung Suk Choi, and Jang Myung Lee*

Abstract: In this paper, moving a fragile object from an initial point to a specific location in the minimum time without damage is studied. In order to achieve this goal, initially, the maximum acceleration and velocity ranges are specified. These ranges can be dynamically generate on the planned path by the manipulator. The path can be altered by considering the geometrical constraints. Later, considering the impulsive force constraint on the object, the range of maximum acceleration and velocity are obtained to preserve object safety while the manipulator is carrying it along the curved path. Finally, a time-optimal trajectory is planned within the maximum allowable range of acceleration and velocity. This time-optimal trajectory planning can be applied to real applications and is suitable for both continuous and discrete paths.

Keywords: Time optimal, optimal trajectory, trajectory planning.

1. INTRODUCTION

In order to produce higher productivity and profit, industrial automations requires faster and safer manipulators. For this purpose, the manipulators have to follow a designated trajectory in the shortest time interval taking the minimum time and safety requirements into consideration [1]. The first requirement for optimal trajectory planning follows from the torque limits of the manipulator, which specify the ranges of acceleration and velocity along the designated path. Therefore, when the manipulator moves along this path with maximum acceleration and velocity, it takes the least amount of time.

However, solving the inverse dynamics of the manipulator is not easy, because the exact dynamics are not known a priori. This is especially difficult for redundant manipulators, because of the numerous paths possible [2]. As for the second requirement, the impulsive force must be considered in order not to damage the object. That is, while the manipulator is following the trajectory, the impulsive force should be kept within the durable range that is obtained from the curvature of the path as well as the acceleration and velocity of the manipulator.

There are several cases in which impulsive force constraints are required: 1. The speed of an automobile must be reduced before entering a rapid curvature to prevent it from overturning. 2. When a robot carries a delicate object such as a cup of water, a large-sized glass, or an explosive, it must be very carefully controlled to prevent accidents [3,4]. The control systems used in real applications require simpler algorithms. Practically, this can be achieved by representing the path as a certain number of points, N, to be fit to the memory limit and processor capability instead of being represented by a functional description. Therefore optimal trajectory planning should be used for the discontinuous path [5,6].

In this paper, the cost function to achieve the minimum time trajectory is defined in Section 2.1; the ranges of acceleration and velocity for the torque limits are discussed in Section 2.2; the safe ranges of acceleration and velocity in which to move an object without violation of the impulsive force limits can be found in Section 2.3; an optimal trajectory planning algorithm is implemented within the two allowable ranges in Section 3.1; the algorithm is applied to a real-time situation in Section 3.2 and finally, the optimal trajectory planning algorithm is verified through real experiments, as described in Section 4. Concluding remarks are given in Section 5, in expectation that the result – safe and fast carrying algorithms – will be helpful for improving productivity as well as the control of manipulators.

Manuscript received September 25, 2004; revised March 29, 2005 and September 21 2005; accepted November 15, 2005. Recommended by Editorial Board member Wankyun Chung under the direction of past Editor-in-Chief Myung Jin Chung. This work was supported by "Research Center for Logistics Information Technology (LIT)" hosted by the Ministry of Education & Human Resources Development in Korea.

Bang Hyun Cho is with the Daewoo Shipbuilding & Marine Engineering Co., Ltd., Aju-dong, Geoje-si, Gyeongnam 656-714, Korea (e-mail: runfortop@pusan.ac.kr).

Byoung Suk Choi and Jang Myung Lee are with the School of Electronics Engineering, Pusan National University, 30 Jangjeon-dong, Geumjeong-gu, Busan 609-735, Korea (e-mails: {woodc, jmlee}@pusan.ac.kr).

* Corresponding author.

2. TIME-OPTIMAL TRAJECTORY PLANNING

2.1. Definition of a cost function

Time-optimal trajectory planning is necessary when a manipulator is carrying an object to a specific location, as shown in Fig. 1. The motion of the manipulator can be denoted as a position vector, $\vec{S}$, which begins from the starting point, l_0, and proceeds to the end point, l_f. Then, the arc length, $l(l=\int_{\vec{S}_0}^{\vec{S}}\|d\vec{S}\|,\ \vec{S}_0\le\vec{S}\le\vec{S}_f)$, which is scalar, can be defined as

$$l=\int_{t_0}^{t}\left\|\frac{d\vec{S}}{dt}\right\|dt,\quad t_0\le t\le t_f. \tag{1}$$

And then, in accordance with a cost function, the minimum time trajectory can be defined as follows:

$$t=\int_{t_0}^{t_f}dt=\int_{l_0}^{l_f}\frac{dt}{dl}dl=\int_{l_0}^{l_f}\frac{1}{v}dl,\quad l_0\le l\le l_f. \tag{2}$$

In Fig. 1, v, which is scalar, represents the magnitude of velocity. When the manipulator velocity, v, is maximized while it is kept under manipulator torque and impulsive force limits for maintaining object safety, the cost function is minimized.

Furthermore the inverse kinematics solution of the manipulator is necessary to control the manipulator. However, in this paper, we are not going to address this problem in detail.

2.2. Manipulator torque limit

The maximum acceleration and velocity range of the manipulator at a certain point on the path will be determined in this section. The dynamics of an n degrees-of-freedom robot can be represented as follows:

$$\tau=M(q)\ddot{q}+V(q,\dot{q})+F(\dot{q})+G(q), \tag{3}$$

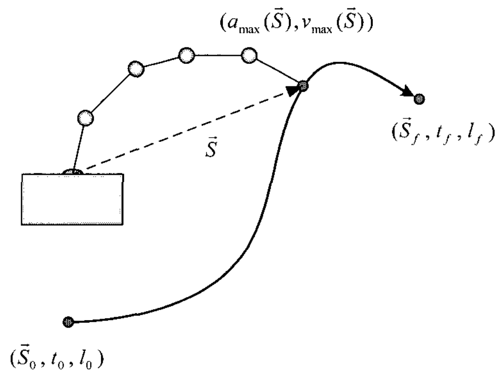


Fig. 1. Time optimal trajectory planning.

where q is a coordinate vector, τ is a torque vector, $M(q)$ is an inertia matrix, $V(q,\dot{q})$ is a Coriolis/centrifugal force vector, $F(\dot{q})$ is a viscous friction matrix, and $G(q)$ is a gravity torque vector.

Also q is assumed that it can be determined uniquely and can be represented as a function of arc length l as follow:

$$q=f(l)\in R^n,\quad l_0\le l\le l_f. \tag{4}$$

When we parameterize q and differentiate with respect to time t, we have

$$\dot{q}=\frac{df}{dl}\frac{dl}{dt}=\frac{df}{dl}v, \tag{5}$$

where $\frac{dl}{dt}=v$.

The dynamic equations along the path then becomes

$$\begin{aligned}\tau&=M(f(l))\frac{df}{dl}\dot{v}+M(f(l))\frac{d^2f}{dl^2}v^2\\&+V(f(l),\frac{df}{dl})v^2+F(\frac{df}{dv})v+G(f(l)),\end{aligned} \tag{6}$$

where l represents the length of the path, and v and $\dot{v}$ represent velocity and acceleration, respectively.

With the torque limits represented as

$$\tau_{i,\min}\le\tau_i\le\tau_{i,\max},\quad i=1,2,\cdots,n, \tag{7}$$

where τ_i is the i^{th} component of τ, and thus the possible range of $\dot{v}$ on a point can be obtained as $LB_i(l,v)\le\dot{v}\le UB_i(l,v)$ on the (l,v) phase plane from (6).

When all the limit values of max $[LB_i]$ and min $[UB_i]$ are given as $\dot{v}_{\max}$ and $\dot{v}_{\min}$, the range of $\dot{v}$ can be represented as

$$\begin{aligned}D_a&=\{\dot{v}\mid \max\ [LB_i]\le\dot{v}\le\min\ [UB_i]\}\\&=\{\dot{v}\mid\dot{v}_{\min}\le\dot{v}\le\dot{v}_{\max}\},\quad i=1,2,\cdots,n.\end{aligned} \tag{8}$$

For the existence of $\dot{v}$, min $[UB_i]-$ max $[LB_i]\ge 0$ and $v\ge 0$ should be satisfied. That is, $UB_i(l,v)-LB_i(l,v)\ge 0$ and $v\ge 0$ are the same conditions for all i and j, and the range of v on a certain l can be obtained using (6) and (7) as follow:

$$\begin{aligned}D_v=\{v\mid UB_i(l,v)-LB_i(l,v)\ge 0\ \ for\ \ all\\ i,\ j=1,2,\cdots,n\ \ and\ \ v\ge 0\}.\end{aligned} \tag{9}$$

When the common range of v is obtained as D_v under the joint limit along the path $l_0 \le l \le l_f$, the minimum and maximum values of $\dot{v}$ can be obtained on the (l, v) phase plane while $v \in D_v$ [7-9].

2.3. Impulsive force limit

To move the object as fast as possible without damage, the impulsive force limit should be kept with the assumption that the manipulator will grasp the object stably. This can be represented as the ranges of allowable acceleration and velocity on the path.

The impulsive force of an object with mass, m for Δt seconds can be represented as

$$F_{obj} = m\frac{\Delta v}{\Delta t} = ma_{obj}. \tag{10}$$

When the allowable impulsive force for the object is $|F_{obj\_max}| = ma_{obj\_max}$, the maximum acceleration for the object can be represented as

$$|a_{obj}| \le a_{obj\_max}. \tag{11}$$

Also, the acceleration on the curve consists of two components: the tangential component and the normal component.

Fig. 2 illustrates the two components of acceleration. Now the maximum acceleration of the object can be represented as

$$a_{obj} = \frac{dv}{dt}\vec{T} + cv^2\vec{N} = a_T\vec{T} + a_N\vec{N}, \tag{12}$$

where $a_T = \frac{dv}{dt}$ and $a_N = cv^2$ represent the tangential component, and normal component of a_{obj}, respectively, c represents the curvature of the path $\vec{S}$, and v is the magnitude of the velocity.

Now the impulse force limit can be represented as

$$|a_{obj}| = \sqrt{\left(\frac{dv}{dt}\right)^2 + (cv^2)^2} \le a_{obj\_max}, \quad v \ge 0. \tag{13}$$

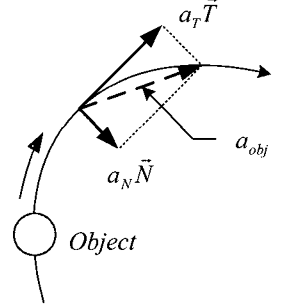


Fig. 2. Acceleration of an object.

From this inequality equation, if $c \ne 0$, the maximum velocity can be obtained as $v = \sqrt{\frac{a_{obj\_max}}{c}}$ with $a_T = 0$ and $a_N = cv^2 = a_{obj\_max}$, while the minimum velocity is $v = 0$. If $c = 0$, then $a_N = 0$. Therefore current velocity, v, does not have any effect on the acceleration and $|a_T| \le a_{obj\_max}$ can be obtained directly.

In summary, ranges for the velocity, v, and acceleration, a_{obj}, can be defined in terms of c as follows:

if $c \ne 0$,

$$I_v = \left\{ v \mid 0 \le v \le \sqrt{\frac{a_{obj\_max}}{c}} \right\} \tag{14a}$$

$$I_a = \left\{ a \mid \sqrt{a_T{}^2 + a_N{}^2} \le a_{obj\_max} \right\} \tag{14b}$$

otherwise,

$$I_v = \{v \mid all \quad v\} \tag{15a}$$

$$I_a = \{ a \mid \ |a_T| \le a_{obj\_max}, \ a_N = 0 \}. \tag{15b}$$

Notice that when c=0, there is no tangential acceleration component on the straight path $\vec{S}$. Also note that for trajectory planning, the constraints, I_v and I_a, should be satisfied along with D_v and D_a constraints.

3. TIME OPTIMAL TRAJECTORY

To this point, the ranges of acceleration and velocity for an object moving along a path have been delineated. In this section, the time-optimal trajectory is constructed within the specified ranges.

3.1. Construction of optimal trajectory

In Section 2.2, D_v and D_a were obtained; Section 2.3 provided the ranges of I_v and I_a in accordance with c. To minimize the execution time, the velocity, v, should be maintained at as large a value as possible while the constraints of D_v, D_a, I_v and I_a are simultaneously satisfied. The common region which satisfies all the constraints can be represented as

$$B_v = \{v \mid v \in D_v \quad and \quad v \in I_v\}, \tag{16}$$

$$B_a = \{a \mid a \in D_a \quad and \quad a \in I_a\}. \tag{17}$$

That is, the ranges of the acceleration and velocity are specified at a point represented by position vector $\vec{S}$ on the path. Note that for the given $\vec{S}$, l and c are determined uniquely; the ranges of the acceleration and velocity are specified in terms of l and c.

To minimize the cost function, T, v must be maximally determined along the path from l_0 to l_f within the given ranges of acceleration and velocity. The algorithm to maximize v is described as follows:

AFTOT: Algorithm For Time Optimal Trajectory

Let's define the boundary between $v \in B_v$ and $v \notin B_v$ as $v_b(l)$.

Step 1: Form the trajectory formal from the position $l = l_0$ with $v = v_0$ to have the maximum acceleration, $a_{max} \in B_a$. If $\left|\frac{dv_b}{dl}\right| \le \left|\frac{dv}{dl}\right|$ at the boundary, $v_b(l)$, then keep the velocity as $v_b(l)$. Repeat this step until v is $v \notin B_v$, or $l = l_f$.

Step 2: Form the trajectory backward from the position $l = l_f$ with $v = v_f$ to have the maximum deceleration, $a_{min} \in B_a$. If $\left|\frac{dv_b}{dl}\right| \le \left|\frac{dv}{dl}\right|$ at the boundary, $v_b(l)$, then keep the velocity as $v_b(l)$. Repeat this step until v is $v \notin B_v$, or $l = l_0$.

Step 3: When the two positions of Step 1 and 2 meet in the middle of the path, the planning for the trajectory is complete.

Step 4: If the two positions do not meet at the same position on the phase plane (refer to Fig. 3), the trajectory stays at the boundary point (l_1, v_1). To find the point (l_2, v_2) satisfying $\left|\frac{dv_b}{dl}\right| = \left|\frac{dv}{dl}\right|$, the search proceeds along $v_b(l)$ from (l_1, v_1). Using (l_2, v_2) as the starting point, completes steps 1 and 2.

Step 5: If both trajectories meet, a time-optimal trajectory is generated. If not, go to Step 4.

When the optimal velocity curve is obtained as $v_{max}(l)$, the task execution time, for the task, t, along l can be obtained as

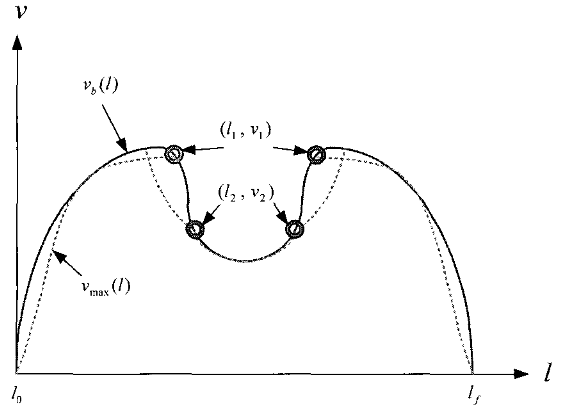


Fig. 3. Construction of v_{max} for time optimal trajectory.

$$t = \int_{t_0}^{t} \frac{1}{v_{max}(l)}\, dl, \quad l_0 \le l \le l_f. \tag{18}$$

From the equation, $l(t)$ also can be obtained adversely. Since $l(t)$ can be represented by $\vec{S}(t)$ from (1), $q(t)$ in (4) can be represented as

$$q(t) = f\left(\vec{S}(t)\right). \tag{19}$$

This completes the optimal trajectory planning. As along as the robotic manipulator keeps the joint trajectory, $q(t)$, the object can be brought to the final location safely within the minimum time. However for complex and long paths, the search process may be long enough to limit the application of this algorithm. In the following section, a solution is provided to guarantee that a real time application to determine general paths is possible.

3.2. Real-time trajectory planning

In practice, a trajectory must be planned for a given path in real-time, because the desired end position might change unpredictably. However, several cases might exist in which the trajectory planning requires too much time to process the task in real-time. For example, if the length of the path is long, the resulting plan would not be accurate.

To resolve this problem, a parallel processing algorithm is adapted to this approach. That is, the original path, l, is divided into n partitions, l_i, and for each l_i, $B_v^{\,i}$ and $B_a^{\,i}$ are obtained for trajectory planning. The partitioning process continues until real-time processing is possible within the capability of the controller. While the manipulator is performing the carrying operation in the interval $[l_i,\ l_{i+1}]$, the trajectory planning for the next partition $[l_{i+1},\ l_{i+2}]$, must be performed to enable continued motion. Even though this local optimal solution may not guarantee the best total performance, this provides a scheme for real-time trajectory planning[10]. Fig. 4. show the division algorithm for real tome control.

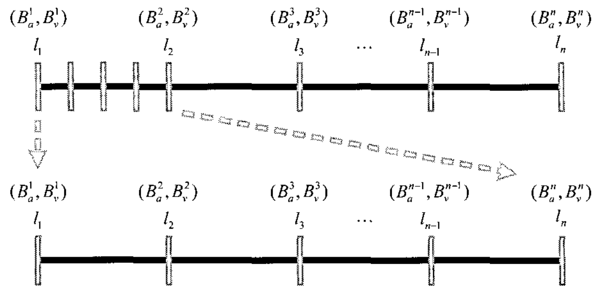


Fig. 4. Partitioned trajectory planning for real time processing.

4. EXPERIMENTS AND DISCUSSION

For the experiments, a moving path is given as

$$z = 0.2\sin\left(\frac{\pi}{0.4}x\right) + 0.43,\ \ y = 0.42,\qquad -0.4 \le x \le 0.4. \tag{20}$$

The path of moving object is shown in Fig. 5.

For the sake of simplicity, the path is represented by eighty one discrete values of x as follows:

$$x = -0.4 + 0.01n, \quad n = 0,1,2,\cdots,80. \tag{21}$$

The object mass is assumed to be 1 Kg with an allowable impulsive force of 1N. Therefore the maximum allowable acceleration is $a_{obj\_\max} = 1 m/s^2$.

For the motion, a redundant 4-joint manipulator was selected which has four links, of 22.6 cm, 26 cm, 27 cm, and 9.5 cm, respectively, and of 2 Kg mass. Using the algorithm explained in this paper, I_a, I_v, D_a and D_v were obtained for all 81 points on the path, as shown in Fig. 6. The maximum velocity, $v_{\max}$, was obtained as a result of the time-optimal trajectory planning, and is shown in Fig. 7.

As can be seen in Fig. 7, the trajectory initially follows the maximum acceleration curve, until it hits and follows boundary D_v. When the trajectory hits boundary, I_v, it follows the boundary again. This implies that the robot manipulator should reduce velocity when it moves into a high-curvature region in order to reduce the impulsive force, whereas it moves at maximum velocity along the low curvature path.

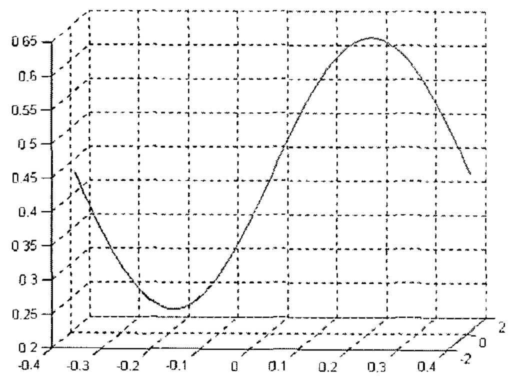


Fig. 5. Moving path of an object.

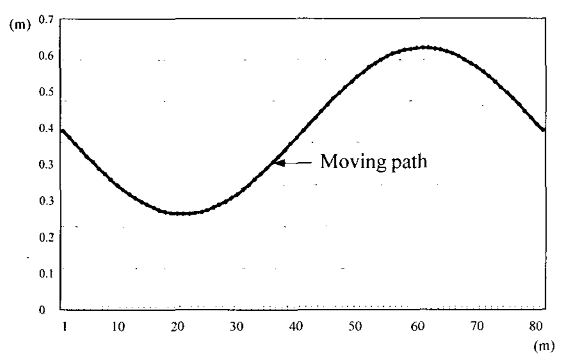


Fig. 6. Object path.

With the trajectory planning, $v_{\max}$, the joint angles of the robotic manipulator are obtained for each point, and shown in Fig. 8. The differentiation of the joint angles provides the joint velocities, and those are shown in Fig. 9.

As shown in Fig. 8, all of the joint angles have continuous values. Note that the changes in the joint angles are very similar. The inverse kinematics equations are solved by ORTIKS (optimal real-time

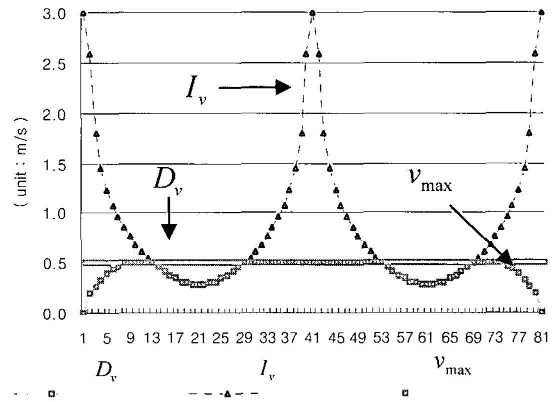


Fig. 7. $v_{\max}$ graph ($a_{obj\_\max}$ 1, D_v ≤ 0.5).

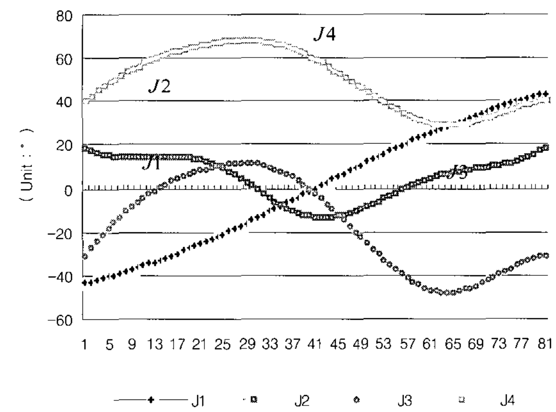


Fig. 8. Joint angles for the path.

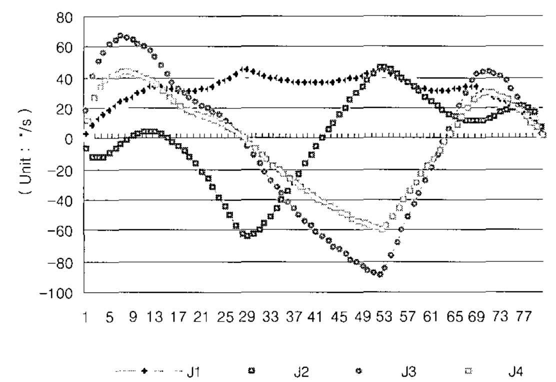
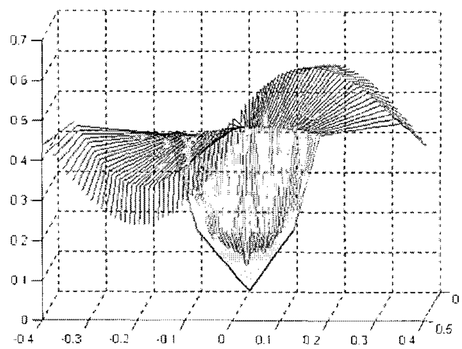
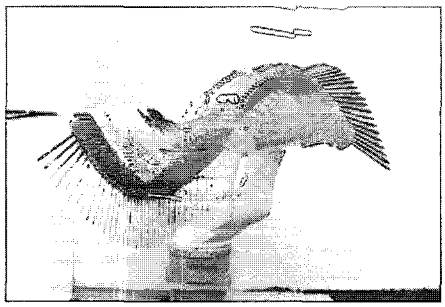


Fig. 9. Joint velocities of $v_{\max}$.

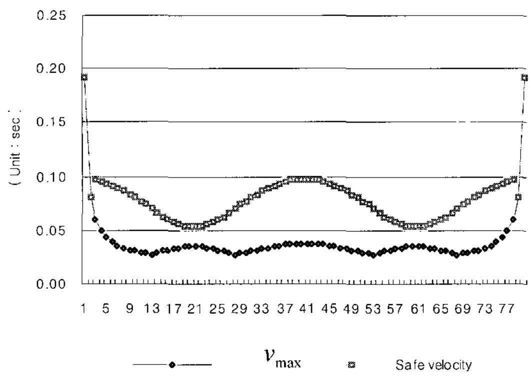


(a) Simulation results.

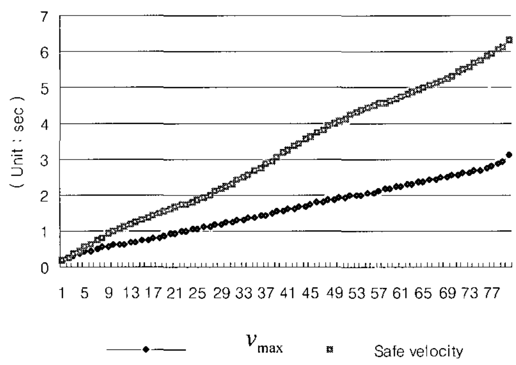


(b) Experimental results.

Fig. 10. Comparison of simulations and experiments.



(a) Interval time.



(b) Accumulative time.

Fig. 11. Execution time with optimal velocity, v_{max} and with safe velocity.

inverse kinematics solution) [11], which equalizes the distribution of joint variations. ORTIKS is relatively free from limitation in the workspace and joint limits as well as from singularities.

The experiments were performed to validate the algorithm with the FARAMAN-AT1 robot, and the results were compared with the simulation results, as shown in Fig. 10. Fig. 11(a) represents the time spent on each interval and Fig. 11(b) represents the cumulative time when the manipulator moves along a time-optimal trajectory path, and with an initial acceleration to maintain the velocity within the safe region without planning.

Since the impulsive force limit is $a_{obj\_max} = 1 m/s^2$ and the distance for the first interval is $0.018607 m$, the safe velocity is maintained at $0.19 m/s$. With the optimal velocity, the execution time is only $T = 3.13s$, while it is $T = 6.31s$ with the safe velocity. This clearly demonstrates the importance of time-optimal trajectory planning.

5. CONCLUSIONS

In this paper, a safe and fast carrying algorithm is proposed for a robotic manipulator. The impulsive force limit of an object and the torque limit of a manipulator were incorporated into trajectory planning to minimize the execution time. The feasibility of the algorithm was verified both by simulations and experiments. The algorithm can be further applied to planning the trajectories of cooperative robots as well as coordinating robots with humans, because it always maintains the tracks of the impulsive force.

REFERENCES

[1] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Trans. on Automatic Control*, vol. 30, no. 6, pp. 531-541, June 1985.

[2] K. G. Shin and N. D. Mckay, "A dynamic programming approach to trajectory planning of robotic manipulators," *IEEE Trans. on Automatic Control*, vol. 31, no. 6, pp. 491-500, June 1986.

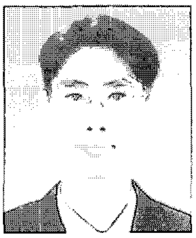
[3] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *Int. J. Robotics Res.*, vol. 4, no. 3, pp. 3-17, 1985.

[4] J. E. Slotine, "Improving the efficiency of time-optimal path-following algorithms," *IEEE Trans. on Robotics and Automation*, vol. 5, no. 1, pp. 118-124, February 1989.

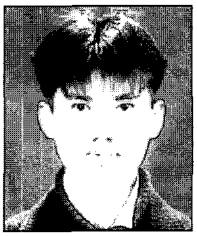
[5] Z. Shiller, "On singular time-optimal control along specified paths," *IEEE Trans. on Robotics and Automation*, vol. 10, no. 4, pp. 561-566, August 1994.

[6] S. B. Moon and S. Ahmad, "Time-optimal trajectories for cooperative multi-manipulator systems," *IEEE Trans. Syst. Man and Cybernetics*

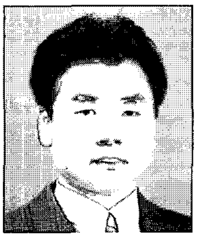
vol. 27 no. 2, April 1997.
[7] A. Klein and C. H. Huang, "Review of pseudo-inverse control for use with kinematically redundant manipulators," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, pp. 245-250, 1983.
[8] G. S. Chirikjian and J. W. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *IEEE Trans. Robot. Automat.*, vol. 10, pp. 343-354, June 1994.
[9] S. Ma and S. Hirose, "Development of a hyper-redundant multijoint manipulator for maintenance of nuclear reactor," *Int. J. Advanced Robot.*, vol. 9, no. 6, pp. 281-300, 1995.
[10] B . K. Choi and D. W. Kim, "Bounded deviation joint path algorithms for piecewise cubic polynomial trajectories," *IEEE Trans. Syst., Man, Cybern*, vol. 20, no. 3, pp. 725-733, May 1990.
[11] C.-H. Wang and J.-G. Horng, "Constrained minimum-time path planning for robot manipulators via virtual knots of the cubic B-spline functions," *IEEE Trans. on Automatic Control*, vol. 35, no. 5, pp. 573-577, May 1990.



Bang Hyun Cho received the B.S. degree in Electronics Engineering from Pusan National University in 2002 and received the M.S degree from Pusan National University in 2004. His research interests include nonlinear control, robotics, and intelligent system.



Byoung Suk Choi received the B.S. degree in Electronics Engineering from Pusan National University in 2004. His research interests include robot control, localization of robot, and sensor network



Jang Myung Lee received the B.S. degree in Electronics Engineering from Seoul National University in 1980. He received the M.S. degree in Electronics Engineering from Seoul National University in 1982 and received the Ph.D. in Electronics Engineering from University of Southern California in 1990.