Time optimal trajectory planning for a robot system Under torque and impulse constraints.

Bang-Hyun Cho, Jang-Myung Lee

Bang-Hyun Cho, Department of Electronics Engineering, Pusan National University, Busan, Korea
email: runfortop@pusan.ac.kr
Jang-Myung Lee, Department of Electronics Engineering, Pusan National University, Busan, Korea
email: jmlee@pusan.ac.kr

ABSTRACT

Moving a fragile object from an initial point to a goal location in minimum time without damage is pursued in this paper. In order to achieve the goal, first of all, the range of maximum acceleration and velocity are specified, which the manipulator can generate dynamically on the path that is planned *a priori* considering the geometrical constraints. Later, considering the impulsive force constraint of the object, the range of maximum acceleration and velocity are going to be obtained to keep the object safe while the manipulator is carrying it along the curved path. Finally, a time-optimal trajectory is planned within the maximum allowable range of the acceleration and velocity. This time optimal trajectory planning can be applied for real applications and is suitable for not only a continuous path but also a discrete path.

I. Introduction

For the higher productivity and profit, faster and safer manipulators have been required for the industrial automation. For this purpose, the manipulators have to follow the desired trajectory within the shortest time, which has been planned considering the minimum time and safety requirements. The first requirement for the optimal trajectory planning comes from the torque limits of the manipulator, which may specify the ranges of the acceleration and velocity along the desired path. Therefore if the manipulator moves along the desired path with the maximum acceleration and velocity, it becomes the minimum time trajectory. However solving the inverse dynamics of the manipulator is not easy since it also requires the inverse kinematics solutions a priori. Especially for a redundant manipulator, finding the optimal trajectory in real time among the numerous candidates is not realizable. As for the second requirement, the impulsive force should be considered not to cause the deformation of the object during the moving operation. That is, while the manipulator is following the trajectory, the impulsive force should be kept within the durable range that depends on the curvature of the path as well as the acceleration and velocity of the manipulator.

There are several examples where the impulsive force constraints are required: 1. We should reduce the speed of an automobile before it gets into the rapid curvature not to be upturned. 2. When a robot carries a dangerous object such as a cup of water, a big size of glass, or a bomb, it should be very carefully controlled not to cause an accident.

For the real applications, the algorithm should be not too complex to be implemented in a control system. Practically, the path can be represented a certain number of points, N, to be fit to memory limit and processor capability instead of being represented by a functional description. Therefore the optimal trajectory planning should be applicable for the discontinuous path.

In the paper, a cost function is defined in section II-1 to achieve the minimum time trajectory; the ranges of acceleration and velocity under the torque limit are obtained in section II-2; the safe ranges of acceleration and velocity for moving an object without violation of impulsive force limit are found in section II-3; an optimal trajectory planning algorithm is implemented within the two allowable ranges in section III-1; the algorithm is applied for the real time trajectory planning of a path in sections III-2; finally the optimal trajectory planning algorithm is verified through the real experiments in section IV. Concluding remarks are given in section V, expecting the result – safe and fast carrying algorithms - is helpful for the improvement of productivity as well as the control of manipulators.

II. Time Optimal Trajectory Planning

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 manipulator can be denoted as a position vector, $\vec{S}$, which starts from the starting point, l_0, to the end point, l_f. Now the arc length, $l\,(l = \int_{\vec{S}_0}^{\vec{S}} \|d\vec{S}\|,\ \vec{S}_0 \le \vec{S} \le \vec{S}_f)$, can be defined as

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

Now a cost function for the minimum time trajectory can be defined in terms of l as follows:

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

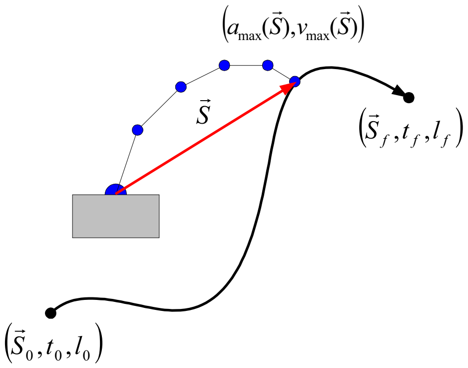


Fig. 1. Time optimal trajectory planning.

In Fig. 1, v represents the magnitude of velocity. When the manipulator velocity, v, is maximized while it is kept under manipulator torque and impulsive force limits for keeping the object safe, the cost function becomes minimum. Furthermore inverse kinematics solution of the manipulator is necessary to control the manipulator. However, in this paper, we are not going to handle the problem in detail.

2. Manipulator torque limit.

The maximum acceleration and velocity range of the manipulator at a certain point on the path is going to be obtained in this section.

Robot dynamics of n degrees of freedom can be represented as follows:

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

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 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, having

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

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

Dynamic equations along the path then becomes

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

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, 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 Eq. (6).

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

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

For the existence of $\dot{v}$, $\min[UB_i] - \max[LB_i] \ge 0$ and $v \ge 0$ should be satisfied. That is, $UB_i(l,v) - LB_j(l,v) \ge 0$ 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 the Eq.'s (6) and (7) as follow:

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

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 $v \in D_v$ [1].

3. Impulsive force limit

To move the object as fast as possible without damaging the object, the impulsive force limit should be kept with the assumption of stable grasping of the object by the manipulator, which 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 consists of two components on the curve: tangential and normal components.

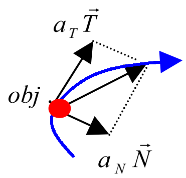


Fig. 2. Acceleration of an object.

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

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

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

Now the impulse force limit can be represented as

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

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 effects 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 $c \ne 0$,

$$\begin{cases} I_v = \left\{ v \mid 0 \le v \le \sqrt{\frac{a_{obj\_max}}{c}} \right\} & (14a) \\ I_a = \left\{ a \mid \sqrt{a_T^2 + a_N^2} \le a_{obj\_max} \right\} & (14b) \end{cases}$$

otherwise,

$$\begin{cases} I_v = \left\{ v \mid all \;\; v \right\} & (15a) \\ I_a = \left\{ a \mid |a_T| \le a_{obj\_max},\ a_N = 0 \right\} & (15b) \end{cases}$$

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

III. Time Optimal Trajectory

So far the ranges of acceleration and velocity for an object moving along a path have been searched. In this section, the time optimal trajectory is constructed within the specified ranges.

1. Construction of optimal trajectory

In the section II-2, D_v and D_a are obtained; II-3 provides the ranges of I_v and I_a in terms of c. To minimize the execution time, the velocity, v, should be kept as large as possible while the constraints of D_v, D_a, I_v, and I_a are satisfied all together. The common region can be represented as

$$B_v = \left\{ v \mid v \in D_v \;\; and \;\; v \in I_v \right\} \tag{16}$$

$$B_a = \left\{ a \mid a \in D_a \;\; and \;\; a \in I_a \right\}. \tag{17}$$

That is, the ranges of the acceleration and velocity are specified at a point represented by the 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 should be maximally determined along the path with the given ranges of acceleration and velocity from l_0 to l_f. The algorithm for maximizing 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 forward 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 $v_b(l)$. Proceed 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 $v_b(l)$. Proceed 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 trajectory planning terminates.

Step 4. If 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). And to find the point (l_2, v_2) satisfying $\left|\frac{dv_b}{dl}\right| = \left|\frac{dv}{dl}\right|$, the search proceeds

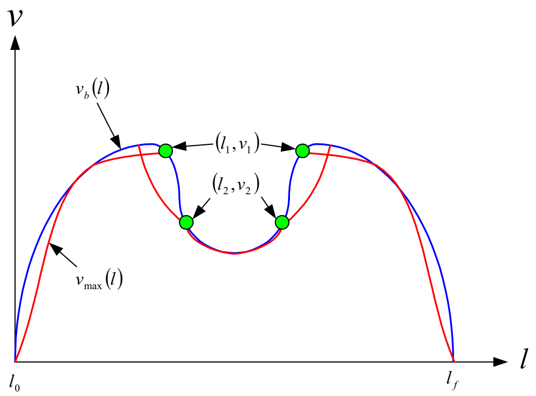


Fig. 3. Construction of v_{max} for time optimal trajectory.

along $v_b(l)$ from (l_1, v_1). From (l_2, v_2) proceed Step 1 forward, and proceed Step 2 backward.

Step 5. If all trajectories meet, 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, t, along l can be obtained as

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

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

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

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 is within the minimum time. However the search process may take longer time for a complex and long path, which may limit the application of this algorithm. In the following section, the solution is provided to guarantee the real time application for general paths.

2. Real time trajectory planning

In practice, the trajectory planning needs to be done for the given path in real time since the desired goal position may change dynamically. However there may exist several cases where the trajectory planning takes too much time to process the task in real time. For an example, if the length of path is long, the trajectory planning couldn't be accurate.

To resolve this problem, the parallel processing algorithm is adopted in 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 the trajectory planning. The partition process may continue until the real time processing becomes possible within the capability of the controller. While the manipulator is performing the

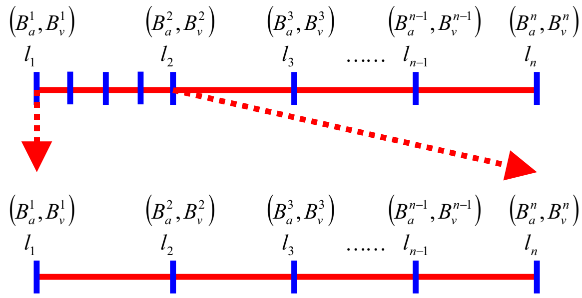


Fig. 4. Partitioned trajectory planning for real time processing.

carrying operation in the interval $[l_i, l_{i+1}]$, the trajectory planning for the next partition, $[l_{i+1}, l_{i+2}]$, should be done for a continue motion. Even though this local optimal solution may not guarantee the best total performance, this provides a scheme for the real time trajectory planning.

IV. Experiments and Discussion

For experiments, a moving path is given as

$$z = 0.2\sin\left(\frac{\pi}{0.4}x\right) + 0.43 \;,\; y = 0.42\,, \tag{20}$$
$$-0.4 \le x \le 0.4$$
$$(\text{unit}: \; m)$$

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

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

And the object mass is assumed to be 1 Kg with the allowable impulsive force of 1 N. Therefore the maximum allowable acceleration becomes $a_{obj\_max} = 1\,m/s^2$.

For the motion, a redundant 4 joint manipulator has been selected in our experiments, which has four links of each length (22.6cm, 26cm, 27cm, 9.5cm) from first link to fourth link and 2 Kg mass. Following the algorithm explained in this paper, I_a, I_v, D_a, and D_v are obtained for all the 81 points on the path shown as Fig. 6(a). The maximum velocity, v_{max}, is obtained as the result of the time optimal trajectory planning and it is shown in Fig. 6(b).

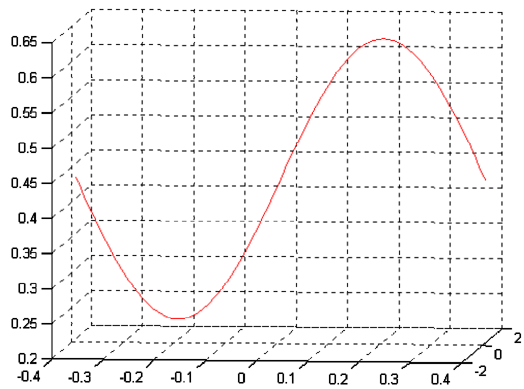


Fig. 5. Moving path of an object.

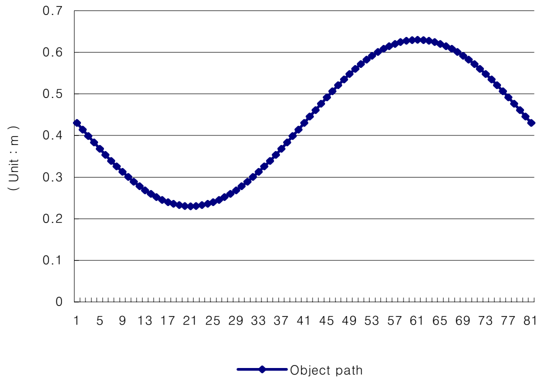


Fig. 6. (a) Object path.

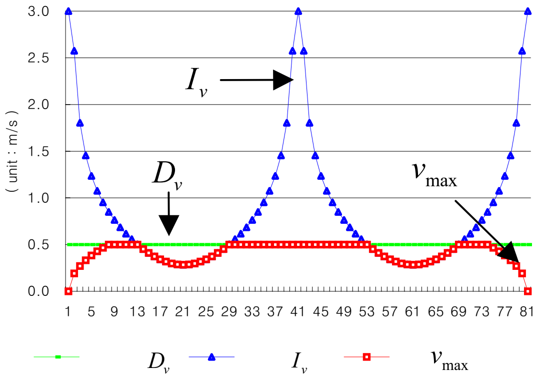


Fig. 6 (b) v_{max} graph ($a_{obj\_max} = 1,\ D_v \le 0.5$).

As shown in Fig. 6(b), the trajectory follows the maximum acceleration curve initially until it hits and follows the boundary D_v. When the trajectory hits the boundary, I_v, it again follows the boundary. This implies that the robot manipulator should reduce the velocity when it gets into the high curvature region to reduce the impulsive force, while it moves with the maximum velocity along the low curvature path.

With the trajectory planning, v_{max}, the joint angles of the robotic manipulator are obtained for each point, and shown in Fig. 7(a). The differentiation of joint angles provides the joint velocities, and shown in Fig. 7(b).

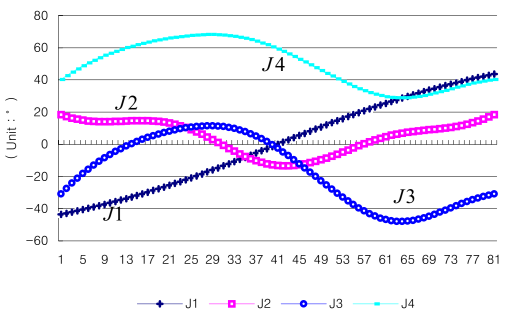


Fig. 7 (a). Joint angles for the path.

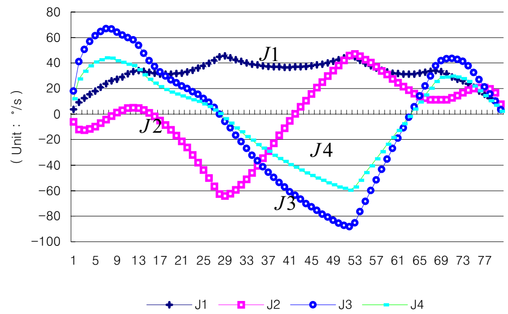
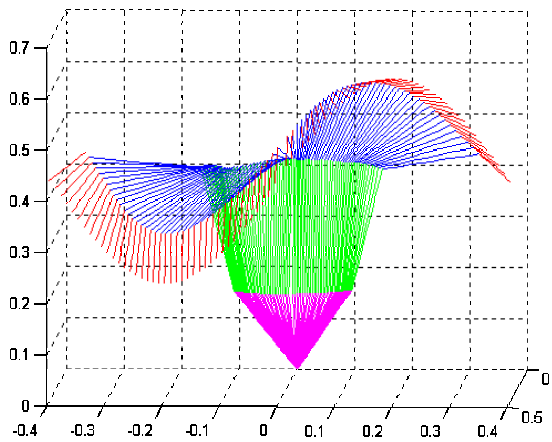


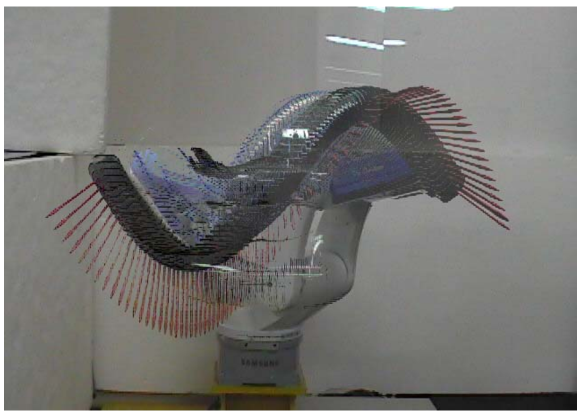
Fig. 7 (b). Joint velocities of v_{max} .

As shown in Fig. 7(a), all of the joint angles have continuous values. And also notice that the changes of the joint angles are very similar. The inverse kinematics equations are solved by the **ORTIKS** (optimal real-time inverse kinematics solution)[12] that aims at the equal distribution of the joint variations. The ORTIKS is relatively free from the workspace and joint limits as well as singularities.

To make it more realistic, the real experiments are performed with FARAMAN-AT1 robot, and the results are compared with the simulation results as shown in Fig. 8.

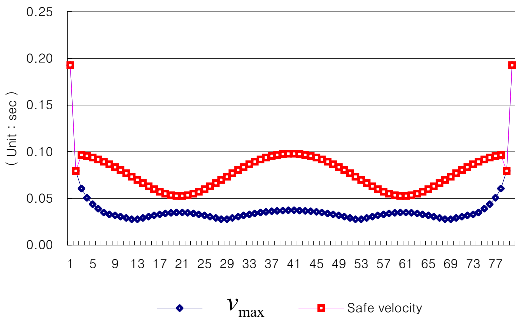


(a) Simulation results.

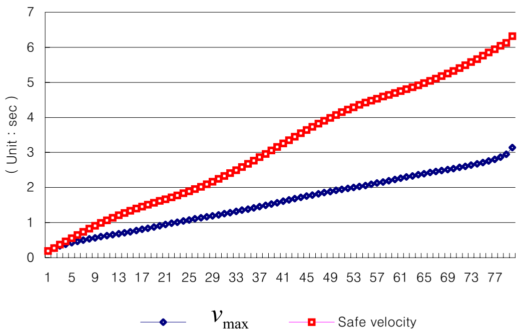


(b) Experimental results.

Fig. 8. Comparison of simulations and experiments.



(a) Interval time.



(b) Accumulative time.

Fig. 9. Execution time with optimal velocity, v_{max} and with safe velocity.

Fig. 9(a) represents the time spent for each interval and (b) represents the accumulative time, when the manipulator moves along the path with the optimal trajectory planning and with initial acceleration to keep 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 kept as $0.19\,m/s$.

With the optimal velocity, the execution time is only $T = 3.13\,s$, while it is $T = 6.31\,s$ with the safe velocity. This clearly demonstrates the importance of the time optimal trajectory planning.

V. Conclusion

A safe and fast carrying algorithm is proposed in this paper for a robotic manipulator. The impulsive force limit of an object and torque limit of a manipulator have been incorporated in the trajectory planning to minimize the execution time. Feasibility of the algorithm is verified both by simulations and experiments. The algorithm can be further applied for the trajectory planning of cooperative robots as well as coordinating robots with human, since it always keep 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. Automat. Contr.,* vol. AC-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. 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. Robotics And Automation*, vol. 5, no. 1, pp. 118-124, Feb., 1989

[5] Zvi. Shiller, “On singular time-optimal control along specified paths,” *IEEE Trans. Robotics And Automation*, vol. 10, no. 4, pp. 561-566, Aug., 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-redund-ant 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-June 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. Automat. Contr.*, vol. 35, no. 5, pp. 573 -577, May 1990

[12] Bang Hyun Cho and Jang Myung Lee, “Optimal real-time inverse kinematics solution for a redundant manipulator” IJCAS submitted.