'87 KACC 1987.10.16～17

Construction of Minimum Time Joint Trajectory for an Industrial Manipulator Using FTM

∘H. C. CHO, Y. S. OH, H. J. SEONG, AND H. T. JEON

Dept. of Electronic Engineering, Chung-Ang Univ., Seoul, Korea

ABSTRACT

The path of an industrial manipulator in a crowded workspace generally consists of a set of Cartesian straight line path connecting a set of two adjacent points. To achieve the Cartesian straight line path is, however, a nontrivial task and an alternative approach is to place enough intermediate points along a desired path and linearly interpolate between these points in the joint space. A method is developed that determines the subtravelling- and the transition-time such that the total travelling time for this path is minimized subject to the maximum joint velocities and accelerations constraint. The method is based on the application of nonlinear programming technique, i.e., FTM (Flexible Tolerance Method). These results are simulated on a digital computer using a six-joint revolute manipulator to show their applications.

1. INTRODUCTION

An industrial manipulator is a computer controlled mechanical system that consists of a series of links connec ted together at joints driven by actuators and a hand which carries an object. There are a number of different motions which a manipulator can perform between any two points (called starting and destination) in the work space. The Cartesian straight line motion (CSLM) between these two points (if possible) seems to have certain distinct advan tages over any other motions. To obtain such motion is, however, a nontrivial task. A practical alternative is to approximate the Cartesian straight line path with a sequence of nonstraight line segments. To study this problem, Taylor [8] proposes an approximation method by placing enough inte rmediate points along the straight line path first. Subsquently, these successive points are lineraly interpolated in the joint space. In Taylor's method the positional and orientational deviations (resulted from these linear inter polations) shall remain below some specified deviation tole rances. In this paper Taylor's method will not be described but it is assumed that the intermediate points between a starting point X_s and a destination point X_d in the Cartesian space are already determined using this method.

To perform linear interpolations it is necessary to construct new joint-trajectories between adjacent segments to avoid discontinuity of joint velocities between successive points. Paul [6] and Taylor [8] propose a symmetric transition, where quadratic polynomials with constant accelerations are used at each transition. When these quadratic polynomial joint-trajectories are constructed, appropriate determination of subtravelling- and transition-time becomes a significant task from the optimal time joint-trajectory planning point of view. In this paper we are proposing a searching method that determines the subtravelling- and transition-time such that the total travelling time between X_s and X_d, subject to the maximum joint velocities and accelerations, is minimized.

2. STATEMENT OF MINIMIZATION PROBLEM

Referring to Fig. 1, $q^0(\in R^n)$ and $q^N(\in R^n)$ represent the joint coordinates of the manipulator having n joints at X_s and X_d in the Cartesian space, respectively, and are obtained through the inverse kinematic equation of the manipulator. Also q_i's $(i=1,2,....,N-1)$ are the intermediate points determined by Taylor's method. Having determined these intermediate points, the problem of determining the subtravelling time t_i $(i=1,2,...,N)$ and transition time τ_i $(i=1,2,...,N+1)$ to minimize the total travelling time (T) subject to the constraints can be briefly described as follows.

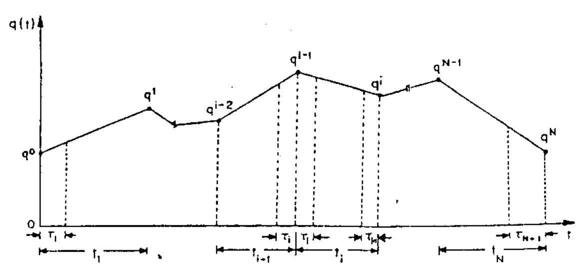

Fig. 1. Linear interpolation in the joint space

MINIMIZATION PROBLEM

Minimize: $T = \sum_{i=1}^{N} t_i$, (1)

Subject to:

$t^i \geq \max \{\Delta q_j^i/\omega_j\}$ for $i=1,2,...,N$, and $j=1,2,...,n$, (2)

$\dot{\omega}_j \geq \max \{\Delta q_j^1/t_1\tau_1, ..., | \Delta q_j^{i+1}/t_{i+1} - \Delta q_j^i/t_i | /2\tau_{i+1}, ..., \Delta q_j^N/t_N\tau_{N+1}\}$ for $j = 1,2,...,n$, and (3)

$\tau_i \leq \min \{t_i/K , t_{i-1}/K\}$ for $i = 1,2,...,N+1$. (4)

Here $\Delta q_j^i = |q_j^i - q_j^{i-1}|$, q_j^i is the jth-joint component of q^i, ω_j and $\dot{\omega}_j$ are the maximum velocity and acceleration of the jth joint, and K is a preselected scalar value to avoid the excessive deviations of the trajectories from the intermediate point during transition. It is to be noted that each joint trajectory consists of one linear polynomial during $t_i - (\tau_i + \tau_{i+1})$ and two different quadratic polynomials during each transition times τ_i and τ_{i+1}, respectively.

3.MINIMUM SUBTRAVELLING AND TRANSITION TIMES

The minimization of T subject to (2) to (4) leads to the application of a constrained nonlinear programming method. Most of these methods are based on the strict restriction of search points, i.e., converting every infeasible search point violating the constraints into the feasible one meeting the constraints. This fact results in a large computational time and slow convergence for the desired optimal solution. The Flexible Tolerance Method [2] (FTM), is selected to deal with this problem, since in FTM, every infeasible search point is converted into the near-feasible point (which will be defined later) that releases the above restriction and therefore improves the computational speed.

In order to apply FTM, we define the sum of infeasibility function (SIF), which denotes the degree of the constraint violation of a search point Z_m^k (indicating the mth-search point at the kth-search point at the kth-search stage), as follows:

$$SIF(Z_m^k) = [\sum_{i=1}^{N} v\{(t_{m,i}^k - a_i)^2\} + \sum_{j=1}^{n} v\{(\dot{\omega}_j - b_j)^2\} + \sum_{i=1}^{N+1} v\{(c_i - \tau_{m,i}^k)^2\}]^{1/2}. \quad (5)$$

Here,

$$v\{x^2\} = \begin{cases} 0 & \text{if } x \geq 0 \\ x^2 & \text{if } x<0 \end{cases} \quad (6a)$$

$$Z_m^k = [\ t_{m,1}^k ,..., t_{m,N}^k, \tau_{m,1}^k ,..., \tau_{m,N+1}^k\ ] , \quad (6b)$$

$$a_i = \max_{1 \leq j \leq n} \{\Delta q_j^i/\omega_j\} , \quad (6c)$$

$$b_j = \max \{\Delta q_j^1/t_{m,1}^k\tau_{m,1}^k, ..., | \Delta q_j^{i+1}/t_{m,i+1}^k - \Delta q_j^i/t_{m,i}^i | /2\tau_{m,i+1}^k, ..., \Delta q_j^N/t_{m,N}^k\tau_{m,N+1}^k\}, \quad (6d)$$

$$c_i = \min \{t_{m,i}^k/K, t_{m,i-1}^k/K\}. \quad (6e)$$

Also a tolerance function ψ^k at the kth-search stage, which is apositive nonincreasing function of (2N+2) search points, is defined as follows:

$$\psi^k = \min \{ \psi^{k-1} , U^k \} , \quad (7)$$

$$U^k = \sum_{i=1}^{2N+2} [\sum_{j=1}^{2N+1} (Z_{ij}^k - Z_{2N+3,j}^k)]^{1/2} /(2N+2), \quad (8)$$

$$Z_{2N+3,j}^k = [\sum_{i=1}^{2N+2} Z_{ij}^k - Z_{l,j}^k] / (2N+1), \quad \text{for } j = 1,2,...,2N+1. \quad (9)$$

Here U^k denotes the average distance from Z_i^k $(i=1,2,...,2N+2)$ to Z_{2N+3}^k, Z_{2N+3}^k is the centroid of the (2N+1) search points excluding Z_l^k at the kth-search stage, Z_l^k is the search point which has the largest value (T_l^k) of the (2N+2) values of T, and ψ^0 (<<1) is selected initially. Also Z_{ij}^k, $Z_{2N+3,j}^k$ and Z_{lj}^k are the jth component of the corresponding Z_i^k, Z_{2N+3}^k and Z_l^k. Thus the condition for Z_m^k to be a near-feasible point at the kth-search stage is defined as follows.

$$NF(Z_m^k) = (\psi^k - SIF(Z_m^k)) \geq 0. \quad (10)$$

It is to be noted that FTM is fundamentally based on the Flexible Polyhedron Method [5] (FPM). At the initial stage of the FTM, a flexible polyhedron is constructed with the (2N+2) search points or vertices (determined from (19) by selecting an initial search point Z_1 and substituting Z_m^k, Z_1 and H_m for X_i^m, X^m and H_i, respectively). At each search stage, the search point with the largest T is replaced by one with a smaller T. Such replacement constructs a new flexible polyhedron for the next search. When it is impossible to find a point with smaller T than before, then (2N+1) search points are collapsed into one single point with this T that is the final solution to this minimization problem. During the search, every infeasible search point is converted into the corresponding near-feasible point through minimization of SIF by the FPM. This procedure is detailed in the Appendix. The ψ^k reduces fast and finally zero out because of its definition of being a positive nonincreasing function of the (2N+2) search points. This fact indicates the feasibility of the final solution and convergence of the FTM. The final solution represents the determination of t_i $(i = 1,2,...,N)$ and τ_i $(i = 1,2,...,N+1)$ for the minimum travelling time from X_s to X_d. The above procedure is detailed in the following algorithm.

Algorithm 1:

Step 1. Choose $\delta_1, \delta_2, \delta_3, \epsilon, \psi^0$, and Z_1. Set $k = 0$. Obtain Z_m^k $(m = 1,2,...,2N+2)$ from (19) by substituting Z_m^k, Z_1 and H_m for X_i^m, X^m and H_i, respectively, and compute T_m^k (the value of T at Z_m^k) for $m = 1,2,...,2N+2$.

Step 2. m=1. Compute $NF(Z_m^k)$. If $NF(Z_m^k) < 0$, then find another Z_m^k by the FPM. Repeat this step with $m = m+1$ until $m = 2N+2$.

Step 3. Find Z_l^k, Z_s^k, T_l^k and T_s^k. Compute Z_{2N+3}^k by (9) and ψ^k by (7) to (8). If $\psi^k \leq \epsilon$, then stop. Otherwise continue. *Comment*: Z_s^k is the search point which has the smallest value (T_s^k) of the (2N+2) values of T.

Step 4. Reflect Z_l^k through Z_{2N+3}^k as follows.

$$Z_{2N+4}^k = Z_{2N+3}^k + \delta_1(Z_{2N+3}^k - Z_l^k), \quad (11)$$

where $\delta_1(>1)$ is the reflection coefficient. If $NF(Z_{2N+4}^k) < 0$, then find another Z_{2N+4}^k by the FPM. If $T_{2N+4}^k \leq T_s^k$, then go to Step 5. If $T_{2N+4}^k > T_i^k$ for all $i \neq k$, then go to Step 6. Otherwise, set $Z_l^k = Z_{2N+4}^k$ and go to Step 2 with $k=k+1$.

Step 5. Expand the vector $(Z^k_{2N+4} - Z^k_{2N+3})$ as follows.

$$Z^k_{2N+5} = Z^k_{2N+3} + \delta_2(Z^k_{2N+4} - Z^k_{2N+3}), \quad (12)$$

where δ_2 (>1) is the expansion coefficient. If $NF(Z^k_{2N+5})<0$, then find another Z^k_{2N+5} by the FPM. Set $Z^k_l = Z^k_{2N+4}$. If $T^k_{2N+5} < T^k_s$, then set $Z^k_l = Z^k_{2N+5}$. Go to Step 2 with k=k+1.

Step 6. If $T^k_{2N+4} < T^k_l$, then set $Z^k_l = Z^k_{2N+4}$. Contract the vector $(Z^k_l - Z^k_{2N+3})$ as follows.

$$Z^k_{2N+6} = Z^k_{2N+3}+\delta_3(Z^k_l - Z^k_{2N+3}), \quad (13)$$

where $0 < \delta_3 < 1$ is the contraction coefficient. If $NF(Z^k_{2N+6}) < 0$, then find another Z^k_{2N+6} by the FPM. If $T^k_{2N+6} > T^k_l$, then go to Step 7. Otherwise, set $Z^k_l = Z^k_{2N+6}$ and go to Step 2 with k=k+1.

Step 7. Reduce all the Z^k_m as follows.

$$Z^k_m = Z^k_s + 0.5(Z^k_m - Z^k_s), \quad \text{for } m = 1,2,\ldots,2N+2 . \quad (14)$$

Go to Step 2 with k=k+1.

4. SIMULATION RESULTS

The proposed scheme that determines the subtravelling (t_i) and transition (τ_i) times for minimizing the total travelling time (T) is simulated using a six-joint revolute manipulator such as PUMA 560 (Unima tion Inc., U.S.A) mechanical manipulator. In order to determine the intermediate points by Taylor's method, the starting points X_s, the destination point X_d, maximum positio nal (ϵ_P) and orientational (ϵ_R) deviation tolerances are chosen as follows.

$$X_s = \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 6 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$X_d = \begin{bmatrix} 1 & 0 & 0 & 14 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 18 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 17 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (15)$$

$$\epsilon_P = 0.02 \text{ inch}, \qquad \epsilon_R = 0.02 \text{ rad}. \quad (16)$$

Table 1 shows the intermediate points (in the joint space) determined by the kinematic equation of PUMA 560 manipulator and the Taylor's algorithm. Table 2 shows the maximum velocities and accelerations which we select. Here we choose K (a preselected parameter for small deviations of the trajectory around the intermediate point) and the initial search point in second as follows.

$$K = 4, \quad (17)$$

$$Z_1=[2.0,\ 1.0,\ 1.0,\ 1.0,\ 1.0,\ 1.0,\ 1.0,\ 2.0,\ 0.5,\ 0.2,\ 0.2,\ 0.2,\ 0.2,\ 0.2,\ 0.2,\ 0.2,\ 0.5] . \quad (18)$$

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
q^0	-0.28	-0.69	1.24	0.02	-0.55	0.28
q^1	-0.05	-0.42	1.13	0.07	-0.71	-0.01
q^2	0.09	0.26	1.04	0.13	-0.79	-0.19
q^3	0.26	-0.02	0.86	0.24	-0.91	-0.41
q^4	0.36	0.18	0.69	0.33	-0.99	-0.56
q^5	0.43	0.37	0.51	0.41	-1.08	-0.66
q^6	0.51	0.77	0.12	0.53	-1.26	-0.76
q^7	0.54	0.18	0.19	0.95	-0.74	-1.46
q^8	0.56	0.01	0.24	0.83	-0.94	-1.30

Table 1. Selection of intermediate points.

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
ω_j	1.2	0.9	1.0	0.8	1.0	0.9
q^0	8.0	7.5	8.2	4.4	6.2	5.7

Table 2. Maximum joint velocities (rad/sec) and accelerations (rad/sec^2)

With these numerical values, the FTM has been run on the VAX 8300 computer system. The final solution to our minimization problem is tabulated in Table 3. As is seen from this table, the total travelling time (T) between X_s and X_d is reduced from the initial choice of 12.4sec to 4.1386sec. It is to be noted that the final solution may be a local minimum value satisfying the nonlinear con straints. The convergence speed of the FTM generally depends on the selection of δ_1, δ_2 and δ_3 and W of (19) in Appendix, but these have little effect on the final solution.

i	τ^*_i	t^*_i
1	0.2167	0.4687
2	0.0535	0.2538
3	0.0621	0.3670
4	0.0876	0.8748
5	0.0743	0.3373
6	0.0813	0.3648
7	0.0910	0.9675
8	0.1163	0.5047
9	0.3154	

Table 3. The minimum subtravelling and transition times.

5. CONCLUSIONS

When a Cartesian straight line path is approximated by linear interpolations in the joint space, determination of every subtravelling- and transition time for minimizing the total travelling time between any two points is very important for high speed performance of a mechanical manipulator. To address this issue an optimal time joint-trajectory planning is proposed that yields the minimum total travelling time between two points. This optimal time joint-trajectory planning is very effective in the case where the intermediate points are closely located. The proposed scheme needs relatively large computational time. But this computational time is not critical since the trajectory planning is performed off-line. These results are applied to a six-joint revolute mechanical manipulator by computer simula tions to demonstrated their applications.

APPENDIX

The flexible polyhedron method (FPM) has been developed by Nelder and Mead [5]. The procedure of finding a near-feasible point is explained in the following.

Step 1. Determine δ_1, δ_2, δ_3, ϵ and W. Set m=0. Replace $X^m \in R^{2N+1}$ with the infeasible search point Z^k_m. Obtain the initial (2N+2) vertices as follows.

$$X^m_i = X^m + H_i \quad \text{for } i=1, 2, ..., 2N+2, \tag{19a}$$

$$H = \begin{bmatrix} 0 h_1 h_2 \ldots h_2 \\ 0 h_2 h_1 \ldots h_2 \\ \vdots \\ 0 h_2 h_2 \ldots h_1 \end{bmatrix}, \tag{19b}$$

$$h_1 = \frac{(\sqrt{2N}+\sqrt{2N+1})W}{\sqrt{2}(2N+1)}, \tag{19c}$$

$$h_2 = \frac{(\sqrt{2N+1}-1)W}{\sqrt{2}(2N+1)}. \tag{19d}$$

Here X^m is the origin vertex, H_i is the ith-column vector of $H \in R^{(2N+1)X(2N+2)}$, and W is a prespecified initial distance between two vertices. *Comment:* m denotes the number of search point.

Step 2. Find X^m_l corresponding to the largest SIF and X^m_s corresponding to the smallest SIF. Compute the centroid X^m_{2N+3} as follows.

$$X^m_{2N+3,j} = \left(\sum_{i=1}^{2N+2} X^m_{i,j} - X^m_{lj}\right)/2N+1 \quad \text{for } j = 1,2,...,2N+1 . \tag{20}$$

Step 3. If $NF(X^m_s) \geq 0$, then set $Z^k_m = X^m_s$ and stop. Otherwise, reflect X^m_l through the X^m_{2N+3} as follows.

$$X^m_{2N+4} = X^m_{2N+3} + \delta_1(X^m_{2N+3} - X^m_l), \tag{21}$$

where δ_1 (>0) is the reflection coefficient.

Step 4. If $SIF(X^m_{2N+4}) < SIF(X^m_s)$, then go to Step 5. If $SIF(X^m_{2N+4}) > SIF(X^m_i)$ for all $i \neq l$, then go to go to Step 6. Otherwise, set $X^m_l = X^m_{2N+4}$ and go to Step 2 with m=m+1.

Step 5. Expand the vector $(X^m_{2N+4}-X^m_{2N+3})$ as follows.

$$X^m_{2N+5} = X^m_{2N+3} + \delta_2 (X^m_{2N+4} - X^m_{2N+3}), \tag{22}$$

where δ_2 (> 1) is the expansion coefficient. If $SIF(X^m_{2N+5}) < SIF(X^m_s)$, then set $X^m_l = X^m_{2N+4}$. Otherwise, set $X^m_l = X^m_{2N+4}$. Go to Step 2 with m=m+1.

Step 6. If $SIF(X^m_{2N+4}) < SIF(X^m_l)$, then set $X^m_l = X^m_{2N+4}$. Contract the vector $(X^m_l - X^m_{2N+3})$ as follows.

$$X^m_{2N+6} = X^m_{2N+3} + \delta_3(X^m_l - X^m_{2N+3}), \tag{23}$$

where $0 < \delta_3 < 1$ is the contraction coefficient. If $SIF(X^m_{2N+6}) > SIF(X^m_l)$, then go to Step 7. Otherwise, set $X^m_l = X^m_{2N+6}$ and go to Step 2 with m=m+1.

Step 7. Reduce all the X^m_i as follows.

$$X^m_i = X^m_s + 0.5(X^m_i - X^m_s) \quad \text{for } i = 1,2,...,2N+2. \tag{24}$$

Go to Step 2 with m=m+1.

REFERENCE

[1]. Gottfried, B. S., and Weisman, J., Introduction to Optimization Theory, Prentice-Hall, NJ, 1973.

[2]. Himmelblau, D. M., Applied Nonlinear Programming McGraw Hill, New York, 1972.

[3]. Lin, C. S., Chang, P. R., and Luh, J. Y. S., 'Formulation and optimization of cubic polynomial joint trajectories for industrial robots', IEEE Trans. Auto. Contr., AC-28, 1066-1074(1983)

[4]. Luh, J. Y. S., Walker, M. W., and Paul, R. P. C., 'On-line computational scheme for mechanical manipulators', ASME Trans. J. Dyn. Syst. Measurement Contr., 102, 69-76(1980).

[5]. Nelder, J. A., and Mead, R., 'A simplex method for function minimization', Computer J., 7, 308-313(1964).

[6]. Paul, R. P. C., 'Manipulator Cartesian path control', IEEE Trans. Syst. Man Cybern., SMC-9, 702-711(1979).

[7]. Paviani, D., and Himmelblau, D. M., 'Constrainted nonlinear optimization by heuristic programming', Operations Research, 17, 872-882(1969).

[8]. Taylor, R. H., 'Planning and execution of straight line manipulator trajectories', IBM J. Res. and Develop., 23, 424-436(1979).