MODELING, IDENTIFICATION AND CONTROL, 1987, VOL. 8, NO. 3, 125–135
doi:10.4173/mic.1987.3.1

The planning of straight line trajectory in robotics using interactive computer graphics

KESHENG WANG and TERJE K. LIEN†

Keywords: *robotics, trajectory planning, robot motion, simulation*

The planning of straight line trajectory using the interactive computer graphics simulation of robot manipulator movement is discussed. This new approach to straight line motion planning improves the 'bound deviation joint paths' developed by R. M. Taylor (1979). The new approach has three characteristics: (1) linear interpolation in joint space; (2) unequal intervals for interpolating knot points; (3) using interactive computer graphics to assure that the maximum deviation in the whole segment is less than the pre-specified values. The structure and mathematical basis of a computer program developed for this purpose are presented.

1. Introduction

In robotic applications it is often necessary for the manipulator to move along a planned path between goal points to carry out specific tasks or avoid obstacles, such as assembly, parts handling, welding and many other cases.

The development of manipulators requires both the provision of suitable formalisms to describe the motions to be made and the implementation of control strategies for carrying them out. The simplest approach is to recode the values of the joint variables which place the hand at particular desired points and then move the joints independently from one set of variables to the next. More sophisticated motion execution schemes frequently include an open loop trajectory component that generates intermediate target values for the joints. These schemes are often accompanied by more sophisticated means of describing the desired motion to be made. The development of programming languages in which the manipulator target points are described by transformations relating the coordinate system of the hand or tool to the coordinate system of the work station has been of particular interest. Motions in these languages are specified as a sequence of knot points which the controlled frame is to pass through. The joint variables corresponding to each cartesian knot point are computed by solving the inverse equations for the manipulator, which are then used by the motion execution programs. One drawback to this scheme is that it leaves the precise path taken by the manipulator between the knot points undefined. This makes programming more difficult since it is hard to predict just how many intermediate points should be added to a motion statement and complicates the development of model-based program automation tools.

One obvious motion strategy is to cause the hand or tool to move along a straight line path between the knot points. Whitney (1969) achieved differential straight line motion by multiplying the inverse Jacobean of the manipulator's joint to cartesian space transformation by a desired motion increment vector. By repeated evaluation of the inverse Jacobean, this technique can be used to produce long

Received 2 February 1987.
† Production Engineering Laboratory, SINTEF-NTH, N-7034 Trondheim, Norway

straight line motions. Paul (1975) implemented a more straightforward technique, in which intermediate cartesian space goals are evaluated every 100 ms during motion execution. The manipulator inverse equations are then solved to produce intermediate joint goals. It is obvious that the computation time is extensive and the value of deviation cannot be predicted. Taylor (1979) proposed a 'bounded deviation joint paths' approach which relies on a planning phase to interpolate enough intermediate points so that the manipulator can be driven in joint space without deviating more than a pre-specified amount from the desired path.

Taylor's algorithm rests upon a number of assumptions that are discussed only briefly. It is implicitly assumed that bounding the deviation between the joint midpoint and the cartesian midpoint guarantees that the deviation between the knot points will not be unacceptably large. The basis for the assumption is his claim that 'for many manipulator geometries . . . the maximum deviations occur at or near the segment midpoint'. The claim is not generally true (Brady 1982). In this paper we discuss a new approach which is based on Taylor's 'bounded deviation'. However, some refinements are presented which can ensure that the whole path of the hand of the manipulator will not deviate more than a pre-specified amount from a straight line cartesian path using the interactive computer graphics technique.

2. Straight line trajectory

Manipulator control language which has been recently developed typically specifies a sequence of points through which a tool affixed to the end effector of the manipulator is to pass. The effectiveness of such motion specification formalism is greatly increased if the tool moves in a straight line between the points specified by the user. The typical straight line motion path of the manipulator and the configuration of the manipulator is shown in Fig. 1. In this paper, we discuss the planning and execution of the cartesian straight line trajectory.

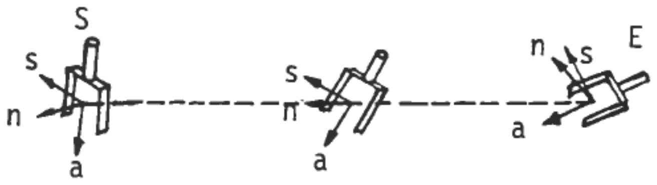

Figure 1. Straight line path of manipulator.

The reasons why a programmer or special planning system choose a cartesian straight line path for the end effector of the manipulator are basically

— In many applications for a manipulator, for example, inserting a pin, welding a seam on a rectangular plate, tracking a conveyor belt, we can see that most motion paths are either a single straight line segment or consist of several straight line segments.

— Straight line paths are relatively predictable and easily visualized.

— They give the shortest path, though not necessarily the fastest, between a starting point and an end point.

— Uniform motion along straight line paths minimizes inertial forces on the end effector of the manipulator and any object it may be carrying.

— The techniques for achieving trajectories that follow straight lines form a basis for which trajectories for achieving other space curves, such as circles, cones, and cycloids, can be developed.

Though there is a lot to be gained from adopting cartesian straight line trajectory, there are some difficulties in achieving this type of trajectory.

3. Cartesian motion and joint motion

The planning and execution of straight line motion trajectory of the manipulator, involves two different approaches: cartesian motion and joint motion (Paul 1979, 1981). Cartesian motion is a form of motion which is natural to cartesian coordinates with the end effector of the manipulator moving along straight lines and rotating about fixed axes in space. Joint motion is a form of motion which is linear in joint coordinates with the end effector of the manipulator neither moving along a straight line nor any other simple, well-defined path.

3.1 The characteristics of cartesian motion

(*a*) Advantages

— The motion between the trajectory segment end points is well and easily defined and thus is particularly suited to the initial and final trajectory segments.

— Smooth motion characteristics can be obtained which have constant velocity and constant acceleration without jerk or vibration.

(*b*) Disadvantages

— The cartesian motion does not consider the effect of joint motion when the trajectory is being planned.

— Cartesian motion breaks down whenever the manipulator becomes degenerate.

— The calculations involved may be too time-consuming.

3.2 The characteristics of joint motion

(*a*) Advantages

— From the point of view of the control system and the formulation of kinematics and dynamics, the most convenient description of the configuration is as a vector of joint variables.

— In joint motion, a large number of calculations is direct kinematic solution which is simple and easy.

(*b*) Disadvantages

— While the joint is interpolated linearly, it is not along the desired path of motion, such as, along a straight line or other simple well-defined paths.

— In joint motion, we can ensure that the movement of joints is smooth, but we can't ensure that the movement of the end effector of the manipulator is smooth.

To sum up, both cartesian motion and joint motion offer advantages as well as disadvantages. We should optimally produce a new strategy for motion planning which can combine the advantages of both forms of motion.

4. Motion between positions

We know that a motion to change from one point to another can be decomposed into a straight line translation and a rotation about a fixed axis in space. If we can find such a line and an axis, we can produce a controlled linear and angular velocity and also calculate the position and orientation of the end effector of manipulator at every interpolation knot point in a cartesian straight line path.

4.1. Intermediate transformation matrix T_D

The end effector of the manipulator is to move from point S (the starting point) to point E (the end point) along the straight line SE with a constant linear velocity, and rotate an angle ϕ about an axis with a constant angular velocity where the axis is a unit vector, we are given the position and orientation transformation matrix of point S and point E, with respect to the base coordinate system $OXYZ$ (see Fig. 2) as the following:

$$T_E = \begin{bmatrix} E_{nx} & E_{sx} & E_{ax} & E_{px} \\ E_{ny} & E_{sy} & E_{ay} & E_{py} \\ E_{nz} & E_{sz} & E_{az} & E_{pz} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$T_S = \begin{bmatrix} S_{nx} & S_{sx} & S_{ax} & S_{px} \\ S_{ny} & S_{sy} & S_{ay} & S_{py} \\ S_{nz} & S_{sz} & S_{az} & S_{pz} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

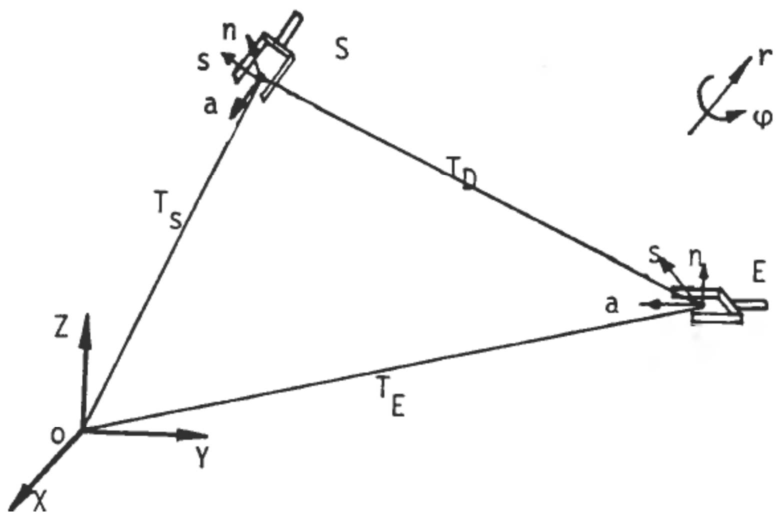

Figure 2. The position and orientation transformation matrix T_S, T_D and T_E.

We can choose an intermediate transformation matrix T_D representing a translation and a rotation. The motion represented by T_D will correspond to a constant linear velocity and a constant angular velocity.

In the base coordinate system, we can get the value of T_D from the matrix T_S and T_E.

The matrix T_E can be expressed as

$$T_E = T_S T_D \tag{3}$$

Premultiplying T_S^{-1}, we obtain

$$T_D = \mathrm{T}_S^{-1} T_E$$
$$= \begin{bmatrix} S_n E_n & S_n E_s & S_n E_a & S_n(E_p - S_p) \\ S_s E_n & S_s E_s & S_s E_a & S_s(E_p - S_p) \\ S_a E_n & S_a E_s & S_a E_a & S_a(E_p - S_p) \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} D_{nx} & D_{sx} & D_{ax} & D_{px} \\ D_{ny} & D_{sy} & D_{ay} & D_{py} \\ D_{nz} & D_{sz} & D_{az} & D_{pz} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

Equation (4) can also be changed into the following

$$R_D = R_S^{-1} R_E \tag{5}$$

4.2. *Evaluation of the rotation axis r and angle ϕ of the transformation matrix*

We can get the rotation matrix about an arbitrary axis r (Lee 1983)

$$R_{r,\phi} = \begin{bmatrix} r_x^2 v_\phi + c_\phi & r_x r_y v_\phi - r_z s_\phi & r_x r_z v_\phi + r_y s_\phi \\ r_x r_y v_\phi + r_z s_\phi & r_y^2 v_\phi + c_\phi & r_y r_z v_\phi - r_x s_\phi \\ r_x r_z v_\phi - r_y s_\phi & r_y r_z v_\phi + r_x s_\phi & r_z^2 v_\phi + c_\phi \end{bmatrix} \tag{6}$$

where

$$v_\phi = 1 - \cos\phi \qquad c_\phi = \cos\phi \qquad s_\phi = \sin\phi$$

Because the rotation matrix R_D is equal to $R_{r,\phi}$, we can get the solution for r and ϕ as

$$\phi = \tan^{-1}\left[\frac{[(D_{ny} - D_{sx})^2 + (D_{ax} - D_{nz})^2 + (D_{sz} - D_{ay})^2]^{1/2}}{D_{nx} + D_{sy} + D_{az} - 1}\right] \tag{7}$$

$$r_x = \frac{D_{sz} - D_{ay}}{2\sin\phi} \tag{8}$$

$$r_y = \frac{D_{ax} - D_{nz}}{2\sin\phi} \tag{9}$$

$$r_z = \frac{D_{ny} - D_{sx}}{2\sin\phi} \tag{10}$$

4.3. *Evaluation of the position and orientation of a given knot point*

As described above, we wish to move the end effector of the manipulator along a straight line path from point S to point E in time T. We consider this path as consisting of the translation of the end effector of the manipulator from SP to EP, coupled with the rotation of the end effector of the manipulator orientation part from *SR* to *ER*. If we suppose that the time of motion is *t* at a given knot point *P* for the uniform motion, the displacement and rotation parts of the end effector at time *t* are given by

$$\eta = t/T \tag{11}$$

$$R(t) = SR \,.\, ROT(r, \eta \,.\, \phi) \tag{12}$$

$$P(t) = SP + \eta \,.\, \Delta p \tag{13}$$

where

$$\Delta p = EP - SP \tag{14}$$

and $ROT(r, \phi)$ is a rotation by ϕ about r required to reorient *SR* into *ER*

$$ROT(r, \phi) = SR^{-1} \,.\, ER \tag{15}$$

5. Transition between path segments in joint coordinate

If we assume that we have computed a set of joint variables θ_i from the cartesian knot points P_i. Then we can use θ_i as the point for a joint motion interpolation strategy shown in Fig. 3. If linear interpolation is made between successive knot points, this will give constant velocities in the joints. But it must be noted that there is a difference in velocity between successive segments. In order to avoid the discontinuity of velocity, we should make a smooth transition between path segments. Here the linear function with parabolic transition is used for the path with knot points. There is a linear function connecting the knot points, and parabolic transition regions are added around each knot point. The notions are used as follows: $\dot{\theta}_{ij}$ is the constant velocity during the linear portion; $\ddot{\theta}_i$ is the acceleration during the transition at point *i*; t_i is the duration of the transition region at knot point *i*; t_{ij} is the duration of the linear portion between points *i* and *j*; T_{ij} is the overall duration of the segment connecting points *i* and *j*.

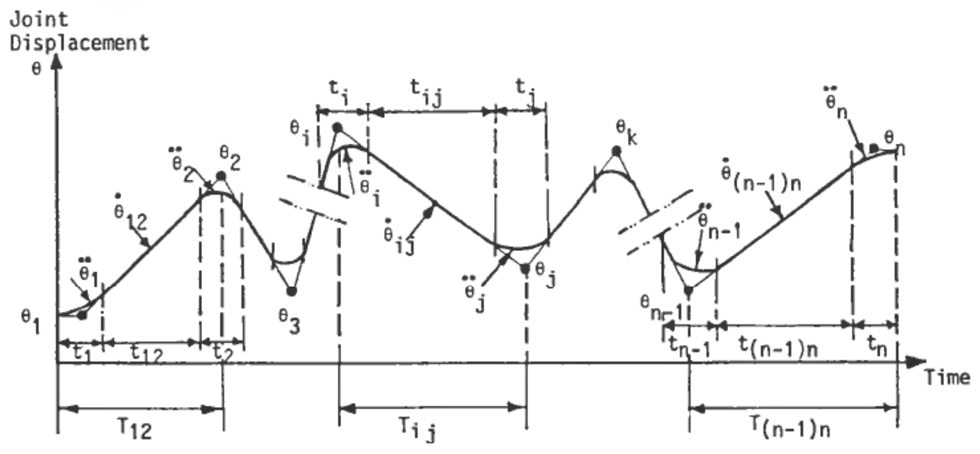

Figure 3. Multisegment linear path with smooth transition.

In general, given all the knot points θ_i, $i = 1, 2, \ldots, n$, and the desired duration T_{ij}, and the magnitude of acceleration to use at each path point θ_i. We can compute the transition time. There are three sets of equations namely (Craig 1986):

For the interior path points,

$$\dot{\theta}_{ij} = \frac{\theta_j - \theta_i}{T_{ij}} \tag{16}$$

$$\ddot{\theta}_j = \text{SGN}\,(\dot{\theta}_{jk} - \dot{\theta}_{ij})|\ddot{\theta}_j|, \tag{17}$$

$$t_j = \frac{\dot{\theta}_{jk} - \dot{\theta}_{ij}}{\ddot{\theta}_j} \tag{18}$$

$$t_{ij} = T_{ij} - \tfrac{1}{2}t_i - \tfrac{1}{2}t_j \tag{19}$$

For the first segment, we solve for t_1 by equating two expressions for the velocity during the linear phase of the segment:

$$\frac{\theta_2 - \theta_1}{T_{12} - \frac{1}{2}t_1} = \ddot{\theta}_1 t_1 \tag{20}$$

Then $\dot{\theta}_{12}$ and t_{12} can easily be computed:

$$\ddot{\theta}_1 = \text{SGN}\,(\theta_2 - \theta_1)|\ddot{\theta}_1|, \tag{21}$$

$$t_1 = T_{12} - \sqrt{\left(T_{12}^2 - \frac{2(\theta_2 - \theta_1)}{\ddot{\theta}_1}\right)}, \tag{22}$$

$$\dot{\theta}_{12} = \frac{\theta_2 - \theta_1}{T_{12} - \frac{1}{2}t_1}, \tag{23}$$

$$t_{12} = T_{12} - t_1 - \tfrac{1}{2}t_2 \tag{24}$$

For the last segment,

$$\frac{\theta_{n-1} - \theta_n}{T_{(n-1)n} - \frac{1}{2}t_n} = \ddot{\theta}_n t_n \tag{25}$$

$$\ddot{\theta}_n = \text{SGN}\,(\theta_{n-1} - \theta_n)|\ddot{\theta}_n|, \tag{26}$$

$$t_n = T_{(n-1)n} - \sqrt{\left(T_{(n-1)n}^2 + \frac{2(\theta_n - \theta_{n-1})}{\ddot{\theta}_n}\right)}, \tag{27}$$

$$\dot{\theta}_{(n-1)n} = \frac{\theta_n - \theta_{n-1}}{T_{(n-1)n} - \frac{1}{2}t_n}, \tag{28}$$

$$t_{(n-1)n} = T_{(n-1)n} - t_n - \tfrac{1}{2}t_{n-1}. \tag{29}$$

The above can be used to solve for the transition times and velocities for the multi-segment path.

In this linear function with parabolic transition strategy, note that the knot points are not reached unless the manipulator comes to a step. Often, when acceleration capability is sufficiently high, the paths will come quite close to the desired knot point. If we wish to pass through a point by coming to a stop, the knot point is simply repeated in the path specification.

6. Point interpolation method

Linear interpolation in joint coordinates between the start and end configurations is very efficient in implementation. However, as we have noted, it does not achieve a cartesian straight line path. Another feature is that if the start and end configurations correspond to nearby points in space, linear interpolation in joint coordinate may depart from a cartesian straight line path by an acceptably small amount, see Fig. 4.

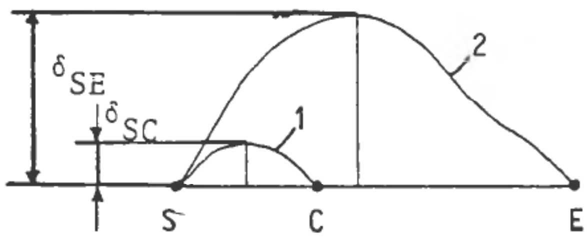

Figure 4. A comparison of δ_{SE} with δ_{SC}. SE: cartesian straight line path motion. Curve 1: joint coordinate motion corresponding to θ_S and θ_C; curve 2: joint coordinate motion corresponding to θ_S and θ_E.

From this figure it is obvious that since knot point C is much nearer point S than point E, the deviation δ_{SC} is less than δ_{SE}. Using this principle, we only need to choose enough knot points which can be interpolated in a cartesian straight line motion path, to enable the end effector of the manipulator to approximately move along a straight line with an acceptable deviation. Considering an arbitrary motion segment, $S \rightarrow E$ we have specified the maximum acceptable deviations:

$$\delta_P(t) \leqslant \delta_P^{\max}; \ \delta_R(t) \leqslant \delta_R^{\max}$$

where $\delta_P(t)$ and $\delta_R(t)$ express the deviations corresponding to the position and orientation of the end effector of the manipulator, and

$$\delta_P(t) = |P_j(t) - P_c(t)|$$
$$\delta_R(t) = \text{angle part of } |R_c(t)^{-1} \, . \, R_j(t)|$$

where

$P_j(t)R_j(t)$ express the position and orientation of the joint space motion

$P_c(t)R_c(t)$ express the position and orientation of the cartesian space motion.

In order to reduce the computation time, we should take the minimum numbers of knot points.

The generation of an 'optimal' set of intermediate knot points requires a good characterization of the path deviation functions $\delta_P^{\max}$ and $\delta_R^{\max}$. These functions depend on the particular manipulator being used and can be quite complicated. We should adopt an approximate midpoint method to compare the deviation in order to simplify the calculating procedure. As the maximum deviations do not always occur at or near the midpoint of the segment, after we obtain an approximation of the knot point, it is necessary to use computer graphics technology to check all the deviations along the segment.

The algorithm for calculating the interpolation knot points is as follows:

Step 1 Calculate the joint variables J_S and J_E, corresponding to the starting point S and the end point E, respectively.

Step 2 Calculate the joint coordinate midpoint, $J_M = J_S + \frac{1}{2}(J_E - J_S)$, use J_M to calculate M (M is in the joint space motion path)

Step 3 Calculate the midpoint X of the cartesian straight line motion path.
$XP = SP + 1/2(EP - SP)$
$XR = SR \,.\, \mathrm{Rot}\,(r, \phi/2)$

Step 4 Calculate the deviation between M and X
$\delta_P = |MP - XP|$
$\delta_R =$ angle part of $|XR^{-1} \,.\, MR|$

Step 5 If $\delta_P \leqslant \delta_P^{\max}$; $\delta_R \leqslant \delta_R^{\max}$, then proceed to step 6 otherwise we can reduce the segment, that is, make the midpoint X as the end point E, and apply steps 1–5 for segment SE (SX).

Step 6 The interpolation knot point is obtained. In the remaining segment repeat steps 1–6 in order to find another interpolation knot point until all the interpolation knot points are found for the whole segment SE.

Step 7 Use computer graphics technology to display the joint space motion and the cartesian straight line motion. If the condition is satisfied, then the task is finished.

Otherwise, interpolate additional knot points in a straight line until condition $\delta_P \leqslant \delta_P^{\max}$; $\delta_R \leqslant \delta_R^{\max}$ is satisfied.

7. Interactive computer graphics simulation

The interactive computer graphics simulation system is shown in Fig. 5.

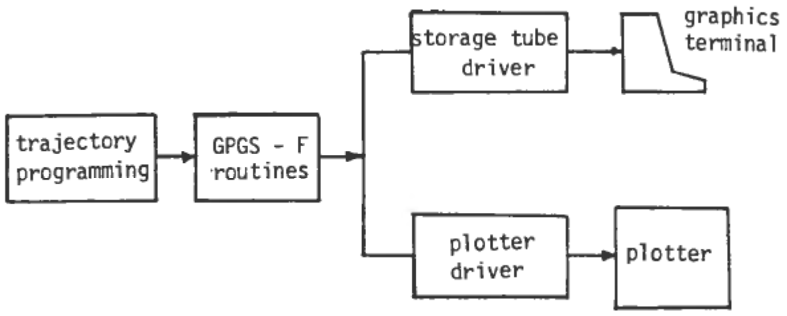

Figure 5. Computer graphics simulation system.

The GPGS-F is a general purpose graphic system which is developed by NORSIGD (Norwegian Association for Computer Graphics). The system is made up of sub-routines which can be called from FORTRAN. The program is run for straight line trajectory planning on storage tube Tektronix 4014, using the interactive way to communicate with the computer to display different perspective view of the straight line traces. If the deviation is greater than the prespecified values, then another knot needs to be interpolated (Newman and Sproll 1983). The example is given as follows:

The positions and orientations of the starting point S and the end point E:

$$T_S = \begin{bmatrix} 0.00000 & -1.00000 & 0.00000 & -0.10000 \\ 0.00000 & 0.00000 & 1.00000 & 0.90000 \\ -1.00000 & 0.00000 & 0.00000 & 0.00000 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_E = \begin{bmatrix} 0.00000 & 0.00000 & -1.00000 & -0.20000 \\ 0.00000 & -1.00000 & 0.00000 & 0.80000 \\ -1.00000 & 0.00000 & 0.00000 & 0.20000 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The maximum deviation of position: $\delta_P^{\max} = 0.001$ (meter)
The maximum deviation of orientation: $\delta_R^{\max} = 0.05$ (radian)

Then we need to interpolate only six knot points, these are:

$$X_1 = \begin{bmatrix} 0.00000 & -0.99518 & -0.09802 & -0.10625 \\ 0.00000 & -0.09802 & 0.99518 & 0.89375 \\ -1.00000 & 0.00000 & 0.00000 & 0.01250 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$X_2 = \begin{bmatrix} 0.00000 & -0.96043 & -0.27852 & -0.11792 \\ 0.00000 & -0.27852 & 0.96043 & 0.88203 \\ -1.00000 & 0.00000 & 0.00000 & 0.03594 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$X_3 = \begin{bmatrix} 0.00000 & -0.90333 & -0.42894 & -0.12822 \\ 0.00000 & -0.42894 & 0.90333 & 0.87178 \\ -1.00000 & 0.00000 & 0.00000 & 0.05645 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$X_4 = \begin{bmatrix} 0.00000 & -0.74837 & -0.66328 & -0.14617 \\ 0.00000 & -0.66328 & 0.74837 & 0.85383 \\ -1.00000 & 0.00000 & 0.00000 & 0.09233 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$X_5 = \begin{bmatrix} 0.00000 & -0.59254 & -0.80554 & -0.15963 \\ 0.00000 & -0.80554 & 0.59254 & 0.84037 \\ -1.00000 & 0.00000 & 0.00000 & 0.11925 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$X_6 = \begin{bmatrix} 0.00000 & -0.31181 & -0.95014 & -0.17981 \\ 0.00000 & -0.95014 & 0.31181 & 0.82019 \\ -1.00000 & 0.00000 & 0.00000 & 0.15963 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

8. Conclusions

This paper has discussed an approach to straight line motion, which will improve the 'Bounded deviation joint paths' method. This new approach has four characteristics:

(1) Linear interpolation in joint space is very efficient to implement.

(2) We take unequal intervals for the interpolation knot points so that we can avoid the resonant vibration which is caused when the interpolation rate is equal to or close to the resonant frequency of the manipulator.

(3) We only use the deviation of the midpoint for precomputation and take the last check to assure that the maximum deviation in the whole segment is less than the specified values.

(4) The trajectory planning can be done off-line and executed on-line.

References

BRADY, M. (1982). *Robot motion* (MIT press), pp. 231–236.

CRAIG, J. J. (1986). *Introduction to robotics mechanics & control* (Addison-Wesley), pp. 205–209.

GPGS-F users guide (1984). NORSIGD. (TAPIR, Trondheim-NTH, Norway).

LEE, C. S. G. (1983). *Fundamentals of robotics* (Addison-Wesley), pp. 2.10–2.11.

NEWMAN, W. M. and SPROLL, R. F. (1983). *Principles of interactive computer graphics* (McGraw-Hill).

PAUL, R. P. (1981). *Robot manipulators: mathematics, programming and control* (MIT Press).

PAUL, R. P. (1979). Manipulator cartesian path control. *IEEE Trans on Systems, Man and Cybernetics*, **9**, pp. 702–711.

PAUL, R. P. (1975). *Manipulator path control*. Proc. IEEE Int. Conf. Cybernetics and Society, New York, pp. 147–152.

TAYLOR, R. H. (1979). Planning and execution of straight line manipulator trajectories. *IBM Journal of Research and Development*, **23**, pp. 424–436.

WHITNEY, D. E. (1969). Resolved motion rate control of manipulators and human prostheses. *IEEE Trans. Man-Machine Systems*, **10**, pp. 47–53.