Finding an optimal solution for the shortest path in a 3 D space

T.C. Manjunath

Atria Institute of Technology, Bangalore-24, Karnataka, India.

Abstract
This paper deals with the determination of a shortest path between two points in a 3D space using an articulated robot.

I. INTRODUCTION

SHORTEST path between two points in a 3D space is a straight line motion & is defined as the motion along a straight line or movement of a rigid body along a straight line and represents the shortest distance between the two points in the 3D workspace of any robot.
The straight line motion from the source (pick) to the goal (place) covered in a specific amount of time is known as the straight line trajectory, i.e., if temporal information is added to the straight line path by specifying the times at where the gripper or tool-tip is along the straight line path, then the straight line path gets converted into a straight line trajectory [1].
Straight-line motion is always required in TCS R^6. By controlling all the joints in a coordinated manner, the tool-tip can be made to move along a straight-line path. If the distance between the adjacent points in the joint space R^n is approximately small, then a straight-line path or trajectory in the TCS R^6 can be designed. How we get straight-line motion is to use the IK equations.
The applications of straight line motions are conveyor belt operations, straight line seam arc welding, inserting peg into a hole, threading a nut onto a bolt, performing screw transformations, for inserting electronic components onto PCB, doing robotic manipulation from above the object.
The paper is organized in the following sequence.
A brief introduction about the straight line motion was presented in the previous paragraphs along with the applications. A review of the straight line motion concepts is presented in the section 2. The bounded deviation algorithm used in the paper is discussed in section 3. A mathematical formulation of the simulation study is depicted in section 4 followed by the simulation results in section 5. This section is followed by the conclusions and the references.

II. STRAIGHT LINE MOTION CONCEPTS

Consider the Fig. 1. Let w^0 and w^1 be the two points in the space between which the robot has to draw a straight line. Here, we use the following parameters as [2].

w^0 : is the source point (initial point) ; i.e., the TCV at the point 0 ;
w^1 : is the goal point (final point) ; i.e., the TCV at the point 1 ;
w^0 and w^1 : both are (6 × 1) vectors in TCS, R^6.
T : is the total time taken to move from w^0 to w^1, i.e., the total time taken to traverse the path, obviously $T > 0$.
$\Gamma = \{ w^0, w^1 \}$ = path taken by tool.

The equation for SL path / straight line trajectory w(t) of the tool as shown in the Fig. 1 is represented by an equation of 1st degree or of 1st order, i.e., no squared terms in the expression, i.e., we are writing an expression for the straight line path or trajectory in terms of the SDF and the TCV [2].

$$w(t) = [1 - S(t)]\, w^0 + s(t)\, w^1 \quad ; 0 \le t \le T \quad (1)$$

where s(t) is a differentiable Speed Distribution mapping Function [SDF] which maps (0 , T) into (0 , 1) & is given by [2]

$$s(t) = \frac{t}{T} \quad (2)$$

At the start of the trajectory, t = 0 ;
i.e., s (0) = 0 (start ; initial ; pick ; source point)

$$\therefore, \quad w(t) = [1 - 0]\, w^0 + 0\, w^1 = w^0 \quad (3)$$

.... corresponds to start of the path [2].

At the end of the trajectory, t = T,
i.e., s (T) = 1
i.e., the end ; goal ; destination ; place ; final point.

$$\therefore, \quad w(t) = [1 - 1]\, w^0 + 1\, w^1 = w^1 \quad (4)$$

.... corresponds to end of the path [2].

Hence, it is verified that the Eq^n (1) is of the first order or first degree, i.e., a straight line equation of the form y = mx + c, where c is the intercept, m is the slope.

III. BOUNDED DEVIATION ALGORITHM [BDA] AND ITS BASIC WORKING PRINCIPLE

BDA is an algorithm, which is used to obtain an approximated straight-line motion in TCS R^6 by using an articulated robot by selecting the number of knot points properly [4], minimizing them and distributing them along the trajectory in an optimal manner [2].

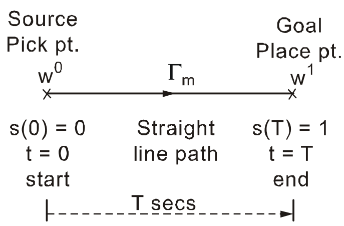

Fig. 1 : Straight line motion, a graphical representation

A. Principle of BDA

It is easier to produce a straight line motion (SL path or SL trajectory) in case of xyz, PTP, cylindrical, polar or spherical or SCARA or Stanford robots. But, in case of articulated robots, it is very difficult to obtain a straight line motion in the TCS. All the joints has to be activated simultaneously in a coordinated manner in order to make the tool-tip to move in a straight line. For achieving a straight line motion in the TCS, the following procedure is used [2].

Inverse kinematics equations [2] has to be solved at each point after minimizing the number of knot points on the trajectory and distributing them along the trajectory in an optimal manner. The concept of the BDA algorithm used is shown in the Fig. 2.

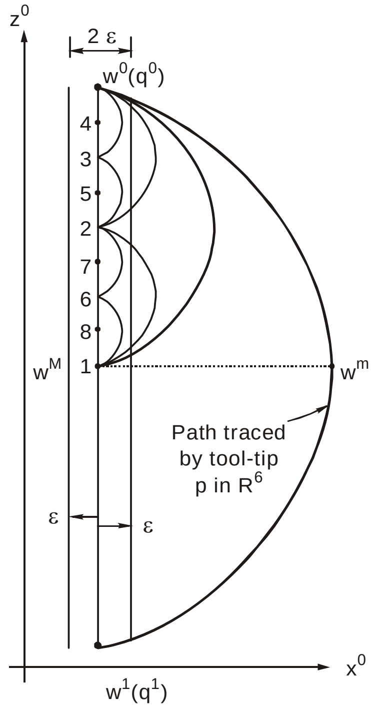

Fig. 2 : Interpolation of joint space approximation to the straight line motion

B. Bounded Deviation Algorithm for Obtaining Straight Line Motion [2]

1. Select a tolerance limit (called as threshold value) for straight line motion as $\varepsilon > 0$ [8].
2. Given, the start point and end point of trajectory as w^0 and w^1 [TCV's at the starting and ending points], use the inverse kinematics equations to compute q^0, q^1 ; i.e., the joint vectors associated with $\{w^0, w^1\}$.
   $w^0 \rightarrow \text{IKP} \rightarrow q^0$;
   $w^1 \rightarrow \text{IKP} \rightarrow q^1$.
3. Compute the joint space mid-point as
   $$q_m = \frac{q_0 + q_1}{2}$$
4. Use the information from q^m and tool configuration vector w to find the equivalent TC space mid-point as
   $w^m = w(q^m)$
   i.e., using q^m, find the TCV, $w(q^m)$; substitute q^m in the TCV of that particular robot which is used to obtain the straight line motion and obtain w^m.
5. Find the exact TCS mid-point as
   $$w^M = \frac{w^0 + w^1}{2}$$
6. If the error or deviation $\| w^m - w^M \| \le \varepsilon$; then, stop.
7. Else, insert w^M as a exact knot point between w^0 and w^1. Now, the trajectory is broken up into two parts, viz., $\{w^0, w^M\}$ and $\{w^M, w^1\}$.
8. Repeat the steps (1 to 6) recursively to the newly generated trajectory segments $\{w^0, w^M\}$ and $\{w^M, w^1\}$ till all the newly generated trajectory segments are within limit of ε.

IV. A SIMULATION STUDY

A simulation is performed on a five axis articulated robot which was designed and fabricated in the college laboratory as shown in Fig. 4 [10]. One pass of BDA is shown analytically here to find a joint space knot point for approximating the following straight-line trajectory [8]. We consider the tool configuration vectors at the starting point and the ending point to be specified by the user as [7]

$$w^0 = [\, 600 \,,\ 0 \,,\ 250 \,,\ 0 \,,\ 0 \,,\ -2]^T$$
$$w^1 = [\, 600 \,,\ 0 \,,\ 50 \,,\ 0 \,,\ 0 \,,\ -2]^T$$

The physical dimensions of the designed system are as follows.

$d = [495.2 \quad 0 \quad 0 \quad 0 \quad 368.2\,]^T$ and

$a = [\,0 \quad 457.2 \quad 457.2 \quad 19 \quad 0\,]^T$?

The tool configuration deviation at the joint-space midpoint is obtained as follows [6]. The general inverse kinematic equations [2] for any five axis articulated robot is [2]

Base angle θ_1 (rotary) :

$$q_1 = \tan^{-1}\left(\frac{w_2}{w_1}\right)$$

GTP angle θ_{234} :

$$q_{234} = \tan^{-1}\left(\frac{-(C_1 w_4 + S_1 w_5)}{-w_6}\right)$$

$$= -\text{arc} \tan 2\,(-b_0\,, -w_6)$$

Intermediate variables are :

$$b_1 = C_1 w_1 + S_1 w_2 - a_4 C_{234} + d_5 S_{234}$$
$$b_2 = d_1 - a_4 S_{234} - d_5 C_{234} - w_3$$

Shoulder angle θ_2 (rotary) :

$$q_2 = \tan^{-1}\left(\frac{(a_2 + a_3 C_3)\, b_2 - (a_3 S_3)\, b_1}{(a_2 + a_3 C_3)\, b_1 + (a_3 S_3)\, b_2}\right)$$

Elbow angle θ_3 (rotary) :

$$q_3 = \cos^{-1}\left(\frac{b_1^2 + b_2^2 - a_2^2 - a_3^2}{2\, a_2\, a_3}\right)$$

Tool pitch angle θ_4 (rotary) :

$$q_4 = q_{234} - q_2 - q_3$$

Tool roll angle θ_5 (rotary) :

$$q_5 = \pi\, ln \sqrt{{w_4}^2 + {w_5}^2 + {w_6}^2}$$

Find the joint vectors q^0 and q^1 associated with w^0 and w^1 by using the above inverse kinematic equations.

Give w^0 as the input to IK algorithm of a five axis articulated robot and calculate q^0 [5], [4]. We get the parameters base angle, global tool pitch angle, intermediate variables, elbow angle, shoulder angle, tool pitch angle & the tool roll angle w.r.t. the TCV w^0 [2], [3].

Then, give w^1 as the input to IK algorithm of a five axis articulated robot and calculate q^1 [2]. We get the parameters base angle, global tool pitch angle, intermediate variables, elbow angle, shoulder angle, tool pitch angle & the tool roll angle w.r.t. the TCV w^1 [2].

Then, compute the joint space midpoint of the trajectory, i.e., q^m as the average of q^0 and q^1.

V. SIMULATION RESULTS

A graphical user interface program in C / C++ is developed and the simulation results are shown in the Fig. 3. Similarly, many points were given as inputs to the generated code which in turn was given as input to the controller and then the robot. The code controls the robot to come from the home position, start at the point A and then move to the specified point B in a straight line.

Fig. 3 : Another pass of the algorithm

VI. CONCLUSION

A simulation of an efficient method of the BDA was demonstrated in this paper using an articulated robot which was designed and fabricated in the college laboratory. A method of computing the straight line motion between two given points in a 3D space using an articulated robot is demonstrated in this paper. Analytical method is also show here for convenience along with the simulation study. The real time implementation of the same was also carried out using a indigenously developed 5 axis articulated robot in the college. The mathematical results & the experimental results / simulated results shows the effectiveness of the developed method [2].

REFERENCES

[1] Craig J, *Introduction to Robotics* : *Mechanics, Dynamics & Control,* Addison Wessely, USA, 1986.
[2] Robert, J. Schilling, *Fundamentals of Robotics - Analysis and Control*, PHI, New Delhi.
[3] Klafter, Thomas and Negin, *Robotic Engineering*, PHI, New Delhi.
[4] Fu, Gonzalez and Lee, *Robotics: Control*, *Sensing*, *Vision and Intelligence*, McGraw Hill.
[5] Groover, Weiss, Nagel and Odrey, *Industrial Robotics*, McGraw Hill.
[6] Ranky, P. G., C. Y. Ho, *Robot Modeling, Control & Applications*, IFS Publishers, Springer, UK.
[7] Crane, Joseph Duffy, *Kinematic Analysis of Robotic Manipulators*, Cambridge Press, UK.
[8] Manjunath, T.C., (2005), *Fundamentals of Robotics*, Fourth edn., Nandu Publishers, Mumbai.
[9] Manjunath, T.C., (2005), *Fast Track to Robotics*, Second edn., Nandu Publishers, Mumbai.
[10] William Burns and Janet Evans, (2000), Practical Robotics - Systems, Interfacing, Applications, Reston Publishing Co.