Trajectory Planning of a Biped Robot

Arun Dayal Udai & Dr. A. N. Sinha
Birla Institute of Technology, Mesra,
Ranchi – 835215

Abstract

In this paper, a real time algorithm for trajectory planning is presented for a biped robot legs. A fully functional robot with eight degree of freedom in two legs is proposed to be implemented. The legs are designed to walk on a flat floor on a path ahead. Each leg has 2 joints for hip, 1 for knee and 1 for ankle. A B – Spline curve is used to describe the path of the joint trajectories which satisfy the constraints of biped balancing and walking.

Introduction

Path planning is the process of calculating a path for a mobile robot to follow so that it can move from one pose (position and orientation) to another [3]. When path of manipulator end is decided in terms of position and orientation of the end effectors in Cartesian space schemes, it can be converted using inverse kinematics to joint space. A method has been proposed based on cubic spline trajectory planning of an industrial manipulator [1]. A constrained time efficient and smooth cubic spline trajectory generation method for industrial robots manipulators have been proposed by B. Cao [2]. An efficient optimal trajectory planner for multiple mobile robots designed by Jason [3] and Planning walking patterns for a biped robot by Qiang [4] uses cubic spline. These methods have an inherent disadvantage that the trajectory path cannot be locally modified, which is very much required when the path is to be altered because of any obstruction found. This also prevents from choosing the order of the curve if required. All are designed based on the fact that real time calculation is complex. Moreover, processing of complex path is slow due to heavy computations and numerical methods involved. However, with the advent of fast microcontrollers and processors, the problem may be overcome.

In this proposed method, B – Spline curve fitting is used for given set of via points or calculated set of points based on time optimization, minimum energy or upon encountering any obstacle. The method uses procedure described by Rogers [5] for fitting of B – Spline curve. The Cartesian path points are traced back to joint space, i.e. angles of the various joints, using inverse kinematics. The method considers various constraints of biped balancing and locomotion.

Parametric B – Spline Curve

The parametric equation [5] for B – Spline curve is given by

$$\mathbf{p}(u) = \sum_{k=0}^{M} \mathbf{p}_k B_{k,n}(u) \qquad u_{min} \le u \le u_{max} \text{ and } 2 \le n \le M + 1 \qquad (1)$$

The above equation may be summarized as below

- a knot vector $\mathbf{U} = \{ u_0, u_1, u_2, \ldots\ldots\}$.
- n is the order the B – Spline function (degree of curve is n – 1).
- $\mathbf{p}_k$ is the set of M + 1 control points
- $B_{k,n}(u)$ is the B – Spline blending function

The fundamental formula for the B – Spline blending function $B_{k,n}(u)$ is given by

$$B_{k,n}(u) = \frac{u - u_k}{u_{k+n-1} - u_k} B_{k,n-1}(u) + \frac{u_{k+n} - u}{u_{k+n} - u_{k+1}} B_{k+1,n-1}(u) \qquad k = 0, 1, 2 \ldots M.$$

Any, indeterminate forms like integer division by 0 or 0/0 is assumed as 0.

$$B_{k,1}(u) = \begin{cases} 1, & \text{if } u_k \le u \le u_{k+1} \\ 0, & \text{otherwise} \end{cases}$$

The u_k are again the knot values, which relates the parametric variable u to the $\mathbf{p_k}$ control points and may be defined in ascending numerical order. All the knot values together form the knot vector and integer knot are commonly used for convenience, each blending polynomial of order n is non zero over n intervals of knot vector and thus the total number of values in the vector set of point is M + n + 1.

B – Spline Curve fitting

In the previous section, it is required to know the set of control points to make the curve pass through a given set of points on trajectory. The defining control polygon is not always known a priori. However, points of the path are known. In such a case, the method described below to calculate the control polygon for a set of known data points [5] is implemented.

If set of *i* data points lies on the B – Spline curve with M + 1 control points, then each must satisfy equation (1)

Thus,

$$p_0(u_0) = B_{0,n}(u_0)p_0 + B_{1,n}(u_0)p_1 + \ldots\ldots\ldots\ldots + B_{M,n}(u_0)p_M$$

$$p_1(u_1) = B_{0,n}(u_1)p_0 + B_{1,n}(u_1)p_1 + \ldots\ldots\ldots\ldots + B_{M,n}(u_1)p_M$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$p_i(u_i) = B_{0,n}(u_i)p_0 + B_{1,n}(u_i)p_1 + \ldots\ldots\ldots\ldots + B_{M,n}(u_i)p_M$$

where M + 1 ≤ i. The above set of equations can be written as

[P] = [B][p]

where $[P]^T = [p_0(u_0)\ \ p_1(u_1) \ldots\ldots\ldots\ldots\ldots\ p_i(u_i)]$

$[p]^T = [p_0 \quad p_1 \quad \ldots\ldots \quad p_M]$

$$[B] = \begin{bmatrix} B_{0,n}(u_0) & B_{1,n}(u_0) & \ldots\ldots & B_{M,n}(u_0) \\ B_{0,n}(u_1) & B_{1,n}(u_1) & \ldots\ldots & B_{M,n}(u_1) \\ \vdots & & & \vdots \\ B_{0,n}(u_i) & B_{1,n}(u_i) & \ldots\ldots & B_{M,n}(u_i) \end{bmatrix}$$

If M + 1 = i, the matrix [B] is a square and the defining polygon can be easily found by

$$[p] = [B]^{-1}[P] \qquad (2)$$

The resulting B – Spline with this M + 1 control point passes through each data points and the curve fit is obtained with C^{n-2} continuity everywhere.

A more natural looking smoother curve can be obtained by lesser number of control points. i.e M + 1 < i.

$$[P] = [B][p]$$

$$[B]^T [P] = [B]^T [B][p]$$

Or $$[p] = [\,[B]^T [B]\,]^{-1}[B]^T[P] \qquad (3)$$

If order n of the curve, the number of defining polygon vertices M + 1 and the parametric value along the curve u_i are known, the matrix [B] can be obtained. The curve is useful when there is no restriction on these parameters and can be arbitrarily assumed.

An approximate method for calculating the parametric value for the curve at the data points uses cord length between the data points. For i data points the parameter value at the m^{th} data point is

$$u_m = u_{\max} \frac{\sum_{r=2}^{m} |D_r - D_{r-1}|}{\sum_{r=2}^{i} |D_r - D_{r-1}|}, \qquad m \geq 2$$

Where $u_0 = 0$ and u_{max} is the maximum value of the knot vector

Trajectory Planning and Balancing

Once the path for ankle [as shown in figure (3)] is decided based on starting point, maximum lift and end point of the step, the intermediate points are calculated based on fitted B - Spline curve parameters i.e. the control points. The intermediate path can be modified if any obstruction is encountered, by modifying these control points or simply by fitting another curve over the shifted intermediate points.

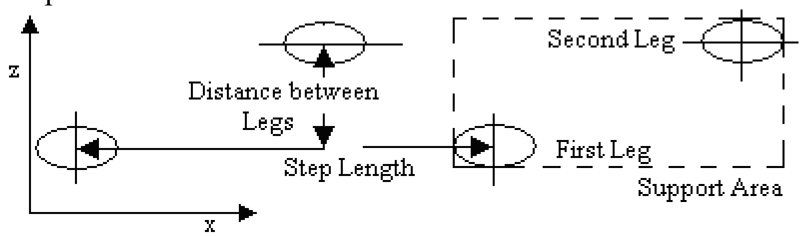

Figure (1): Plan view of path

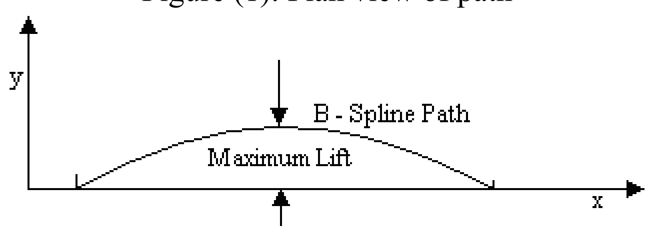

Figure (2): Side view and path of single leg

The hip path is assumed to be in a straight line and the legs are assumed to be hanging with this base line in motion. Support Area or Support polygon is the area surrounded by the corners of the feet [6]. This area is elementary for stability considerations. A statically balanced robot will have Normal Projection of Centre of Mass (NPCM) within the support area. This is also true for slow walking, where body of the robot is required to swing left – right so as to maintain the above condition. Zero Moment Point (ZMP) is the point on the ground surface about which the sum of all the moments of active forces is equal to zero [7]. For dynamic balancing the NPCM can lie outside the support area, in such case ZMP must be inside the support area. All these act as constraints to the body orientation angles, as well as the trajectory path. The foot is to be kept parallel to the ground during complete motion of the ankle joint along the trajectory.

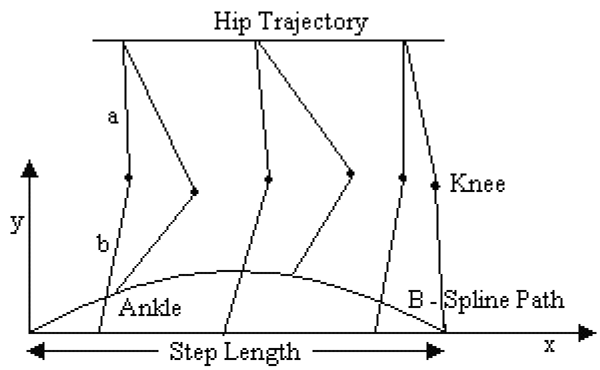

Figure (3): Forward Motion with leg movements for one step.

Once the trajectory is generated with the above constraints, it is converted to joint space parameters i.e. the individual angles of various joints using inverse kinematics.

Input Data

a = 20.32cm, b = 21.59cm, Step length = 20.32cm
Maximum lift at the centre of the path = 5.08cm

Results

Taking origin at the starting point of the trajectory, the three points on the curve can be taken as (0, 0), (10.16, 5.08) and (20.32, 0), and the matrix for the points on the curve will be

$$[P] = \begin{bmatrix} 0 & 0 \\ 10.16 & 5.08 \\ 20.32 & 0 \end{bmatrix}$$

The trajectory is assumed to be quadratic i.e. order (n) = 3,
Calculating for number of control points (M + 1) = 3, i.e. M = 2
Considering the B – Spline curve with open uniform knot vector i.e. multiplicity at the ends equals order of curve, for the trajectory path.
The knot vector will have (M + n + 1) elements, and can be written as [0 0 0 1 1 1].
B – Spline basis function matrix using the above parameters can be calculated as

$$[B] = \begin{bmatrix} B_{0,3}(0) & B_{1,3}(0) & B_{2,3}(0) \\ B_{0,3}(0.5) & B_{1,3}(0.5) & B_{2,3}(0.5) \\ B_{0,3}(1.0) & B_{1,3}(1.0) & B_{2,3}(1.0) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0.25 & 0.5 & 0.25 \\ 0 & 0 & 1 \end{bmatrix}$$

Calculating for control points using equation (2)

$$[p] = [B]^{-1}[P]$$

$$[p] = \begin{bmatrix} 1 & 0 & 0 \\ 0.25 & 0.5 & 0.25 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 \\ 10.16 & 5.08 \\ 20.32 & 0 \end{bmatrix}$$

$$[p] = \begin{bmatrix} 0 & 0 \\ 10.16 & 10.16 \\ 20.32 & 0 \end{bmatrix}$$

With this control points five equidistant parametric points on the curve using equation (1) can be calculated at u = 0, 0.25, 0.5, 0.75 and 1.0.

u	**x**	**y**
0	0	0
0.25	5.08	3.81
0.5	10.16	5.08
0.75	15.24	3.81
1.0	20.32	0

These points (x, y) are the input parameters for the inverse kinematics of the robot biped mechanism and converted to the individual joint angles.

Conclusion

A B–Spline trajectory is proposed to be used for online trajectory generation due to following advantageous features.

- A B – Spline curve offers a single parametric representation for wide range of curves.
- The curve fully lies within the convex hull. If any obstacle is found, it becomes easier to redefine the curve.
- A curve fit is obtained with C^{n-2}continuity everywhere which gives it a more natural looking curve with no wandering about the mean path.
- It is also easier to calculate the general path of the curve which the curve always follows.

References

[1] Shafaat Ahmed Bazaz, Bertand Tondu, "3 – Cubic Spline for On-line Cartesian space trajectory planning of an industrial manipulator", IEEE, 1998, AMC '98 – COIMBRA, pp: 493 – 498.

[2] B. Cao, G. I. Dodds, G. W. Irwin, "Constrained time-efficient and smooth cubic spline trajectory generation for industrial robots", IEE Proc. – Control Theory Appl., Vol. 144, No. 5, September 1997, pp: 467 – 475.

[3] Jason Thomas, Alan Blair, Nick Barnes, "Towards an efficient optimal trajectory planner for multiple mobile robots", Accessed through Google search 2005.

[4] "Planning Walking Patterns for a Biped robot", Qiang Huang, Kazuhito Yokoi, Shuuji Kajita, Hirojiko Arai, Noriho Koyachi, Kazuo Tanie, IEEE Transaction on Robotics and Automation, Vol. 17, No. 3, June 2001.

[5] D. F. Rogers, J. Alan Adams "Mathematical elements of Computer Graphics", McGraw Hill, 1990.

[6] Jochen Zimmermann, “Balancing of Biped Robot using Force Feedback”, document from University of Western Australia.

[7] I. J. Marshall, “Active Balance Control for a Humanoid Robot”, School of Information Technology and Electrical Engineering, University of Queensland, Australia, Dorrington 2002.