International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 12 (2019), pp. 2077-2080

Trajectory Planning of Robots – Curve Fitting Method

Dr. Prasanna L Kaujala

JNTUH College of Engg. Manthani, Telangana, India.

Abstract

A robot has to be designed keeping two aspects, one is the motion of robots and the other is an obstacle. The motion is made smooth if the robots encounter curved shaped obstacles and are made to trace a straight line motion if it encounters a rectangular polygonal-shaped obstacle. The paper deals with the method to use the curve fitting technique to ensure smoothness and an algorithm to ensure the minimum time consumption, moving amongst the static regular and curvilinear obstacles, deviating from its pre-planned trajectory to avoid the collision. Generally, a smooth trajectory is a mandate when the robots are moving around humans. So this approach is a useful tool for such applications.

Keywords: trajectory, curve fitting, motion, obstacle, collision.

I. INTRODUCTION

Multiple robot motion generally has two approaches. A centralized approach and a decentralized approach. A Centralized approach has the pre planned paths in order to move from an initial location to target location and is aware of the path. Therefore, centralized approach ensures the movement of robot from an initial to target position even if any static obstacles hinders its path. But if there is any change in the position of the robot or of the obstacle, the algorithm used has to modify the paths increasing the computational effort, while maintaining the robot achieve its target. Consequently, the computational effort will make a minor delay as compared to the earlier paths. Decentralized approach designs the paths dynamically, so any change does not alter the computational effort, but every robot in the workspace being considered meets the target or not, is not guaranteed. A balanced optimized value must therefore be chosen between the smooth computational effort and reaching the goal with minimum time.

II. EXISTING METHODOLOGIES

Ref [1], approach is the smoothing technique used using polynomial interpolation and have also used spline, Bezier curves. But in this paper only one robot's motion is considered at a time. Ref [2], proposes a study based on the estimation of an equation of curved path which is used as a robot trajectory. A nearness diagram algorithm and S shaped Bezier curve is used as a tool to form the smooth trajectory. But only one robot is considered with multiple obstacles. Ref [3], exhibits a smooth path using potential field method algorithm and curve based continuous path algorithm. The paper ensures the safety factors for critical applications considering the path planning among multiple obstacles of a single robot. Ref [4], is a practical approach for generating motion paths with continuous steering for car like mobile robots. B spline curve is leveraged to model the vehicle's path, considering a single robot.

Ref [5], Merges to the quadratic polynomial interpolation with 3 points. The polynomial expands and merges to G2 continuity. The quadratic polynomials are combined with the membership functions, which affect the whole path because of variations in the membership, considering a single robot.

All the above methods results in a smooth curved path without touching the obstacle. The methodology used in this paper is using the curve fitting technique with multiple robots.

III. PATH PLANNING SCHEME

The robots are depicted as a rectangular shape with a certain area and each robot in the process is given an initial position and an approaching target location. The rectangular workspace of the robot when moving from the original position to the target position is tested for intersection with the area of the obstacles or intersection with the area of the workspace of the other robots in every incremental value either in the x direction or y direction. Every other moving robot is treated as a moving obstacle, having a rectangular workspace area.

A. How collision is detected ?

The line intersection command in the Java applet is used to detect the intersection with the circumferential lines of obstacles between all the boundary lines of the robot's workspace. The robot's orientation is determined from the original to the final position based on the slope of the motion. If the slope is positive, it is determined that the robot movement must travel upward and forward. If the slope is negative, it is determined that the robot goes down and forward. The zero slope infers that the motion is parallel.. The robot travels in the specified direction based on the slope and such a movement is calculated by small increments both in X and Y direction :

$$X_{inc} = \frac{ctL - crL}{10} \qquad \text{Eq 1}$$

Where ctL and crL are the left positional coordinates of the robot's target and initial corners respectively.

$$Y_{inc} = \frac{ctT - crT}{X_{inc}} \qquad \text{Eq 2}$$

Where ctT and crT are the top position coordinates of the

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 12 (2019), pp. 2077-2080

target and initial corners of the robot's workspace respectively.

When moving in the pre-determined path that may change dynamically with any motion, the workspace of the robot may cross the workspace of the obstacle. Since the obstacles are the fixed obstacles and the obstacles ' location is also static, the robots are not to collide with the boundary of the obstacles workspace. The equation (2) decides whether the robot movement in upward or downward. The negative value of the equation instructs the robot to move in the upward direction and vice versa. While this motion is maintained, it may deviate its path if during its motion the static obstacle stands. The deviated trajectory also depends on the size and shape of the obstacle.

If the obstacle is a polygonal shape, it is practically inscribed in a rectangle and the rectangle's four corners define the obstacle's form. When the robot moves in the upward direction, the obstacle is counteracted, the robot's path changes and the changed path shows the robot's movement from its present position to the left and top of the obstacle, and the path proceeds to its destination.

Likewise, when the robot moves downward (where Yinc is positive), counteracts the obstacle, then the robot's trajectory transitions from its previous path to the left and bottom of the obstacle, and then the path proceeds to its destination, as shown in Figure 1 to Figure 4.

B. Curve Fitting Technique

If the obstacle is having a periphery of a curve, then the equation is found out by using one of the curve fitting methods. When the points of the curve may not be equispaced, lagrange interpolation method is preferred. Ref [1] shows the polynomial interpolation, which gives a path smoothing effect. But due to all the known points or coordinates of the static obstacle, which are not equally spaced and has more convolutions, lagrange interpolation method is preferred to form the smooth trajectory. To illustrate, an arbitrary curvature obstacle is taken in the workspace, and the robot is allowed to deviate its path around the periphery, touching the tangent of the offset curve, and moves to its destination. The tangent of the curve is taken to minimize the movement of the robot and thus decreasing its time to move to the destination. The lagrange interpolation generalized formula is :

$$f(x) = \frac{(x-x_1).......(x-x_n)}{(x_0-x_1).....(x_0-x_n)} f_0 + ... \frac{(x-x_0).......(x_n-x_{n-1})}{(x_n-x_0).....(x_n-x_{n-1})} f_n$$

$$\sum_{i=0}^{n} \left(\prod_{j=0,\, j \neq i}^{n} \frac{x - x_j}{(x_i - x_j)} \right) f_i$$

The obstacle considered has 8 segments whose some points are known and therefore an equation of each of the segment is formulated.

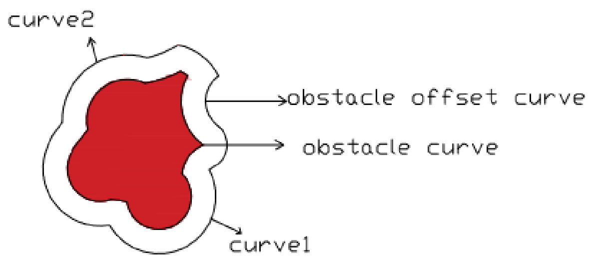

Fig 1: The obstacle with its curved shape and offset distance maintained, so that robot moving on the offset curve do not touch the obstacle.

The equation of 1st and 2nd curve is illustrated as

$$y = -0.02\, x^2 - 8.868\, x - 0.87$$

$$y = -4.2X10^{-3}\, x^3 + 8.86\, x^2 - 5753.7\, x + 1233557$$

Likewise, all such equations are formed using lagrange interpolation formula. Once the two dimensional equation is formed and for every incremental movement of robot, it synchronizes its movement with the x and y coordinates of the equation of that particular curve, and therefore a smooth curve is traced. For minimal distance, the extreme tangent point is taken on the curve, that comes in the way of the movement of the robot. Therefore, the derivative and the gradient will fetch the equation of the tangent. The slope of the tangent can be found by the instant position and final position of the robot's coordinates. The tangent is only required for the forward and downward motion because the reference position of the robot is left and top coordinate. While moving forward and downward, there is every chance of collision of the robot's workspace with the obstacle's workspace. Therefore an offset distance is maintained around the obstacle, of the size of the robot and is accordingly planned. The Fig 5 & Fig 6 shows the movement of all of the robots around the periphery of the obstacle, without touching it, deviating its path from the regular one, and reaching the destination, if moved simultaneously.

When simultaneous motions of the robots are considered, the curve traced by many robots might be using the same trajectory points. Since more than one robot's trajectory might be same, priority has to be given to the robot whose task is to be accomplished on top. Some identification for the robots based on the priorities is assigned to the robot, and highest prioritized robot will move first on the specified path. Some reviews are taken from Ref [6] which gives the priority assignment to utilize the search space for all the robots and minimize the overall path length and gives an idea to choose the best minimum path length. Following this approach accordingly, further development of prioritized robot is chosen to move first on the specified trajectory, as shown in Fig 6 & 7. The figure shows the robot numbered (0),(1) & (3), moves on the same trajectory. The robot number is identified as the priority list. Therefore, Robot (0) will move first, robot (1) will move next and lastly robot (3) will move on the same

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 12 (2019), pp. 2077-2080

trajectory in continuation, so that these three robots do not collide with each other.

Whenever a robot encounters an obstacle, the robot will have two alternate paths, one path is generated while moving above the obstacle and another one is generated while moving below from the obstacle, to reach its target coordinates. If there are more than one obstacle, more than two paths will exist. Ref [7] gives the recursive algorithm for collision avoidance, which finds the shortest path if two or more obstacles are present, among multiple robot motions.

C. Movement of Robots in Specified Path

The figure shows the paths taken for every robot, if all the robots start moving simultaneously in (i) the absence of an obstacle and (ii) in the presence of an obstacle.

Two aspects are considered for the obstacle

(i) Rectangular or polygonal shape

(ii) Curved shape.

The figures shows the movement of robots moving simultaneously, in the presence of rectangular obstacle and arbitrarily curved shaped obstacle. The equations of curved shape is found out to be according to lagrange interpolation curve fitting method:

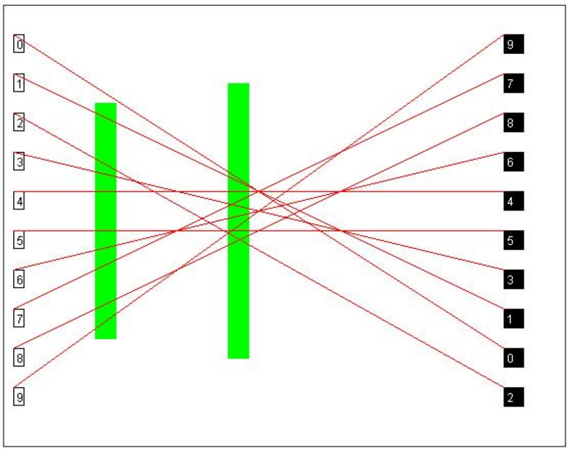

Fig 2 : Robots with initial and final ,positions

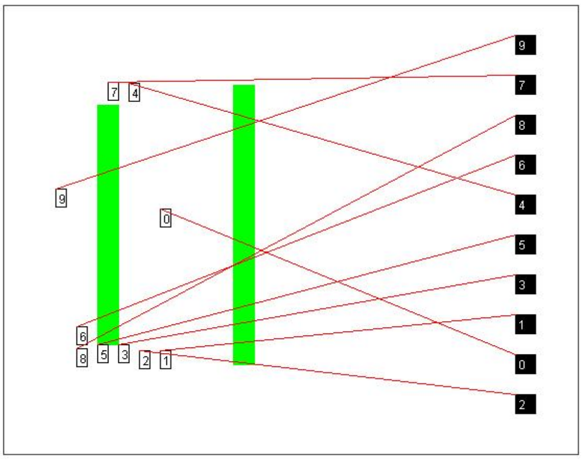

Fig 3 : Robots counteracted with the 1st obstacle and encountering with the 2nd one.

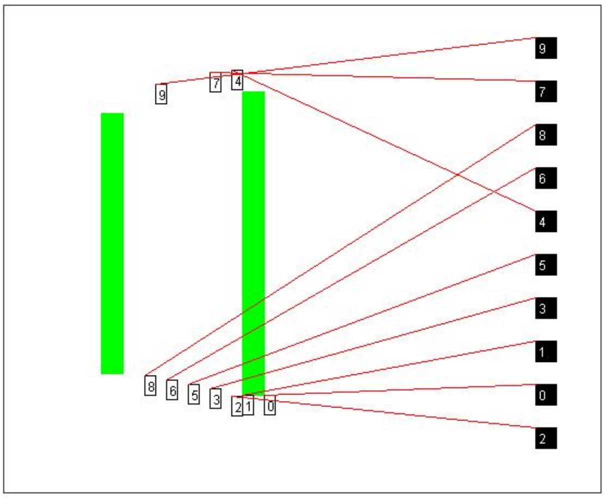

Fig 4: Robots changing its path due to hindrance by the 2nd obstacle.

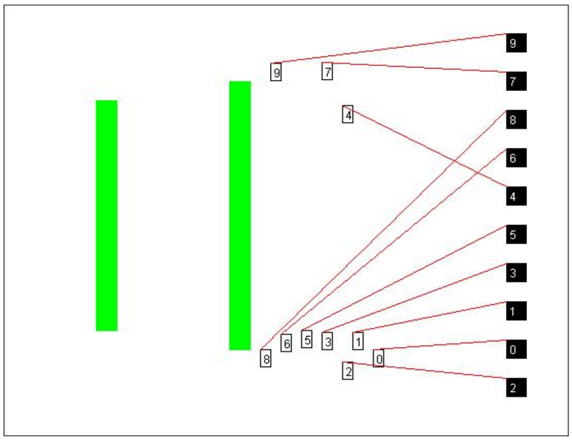

Fig 5: Robots cleared the blockage by the obstacles and reaching its destination.

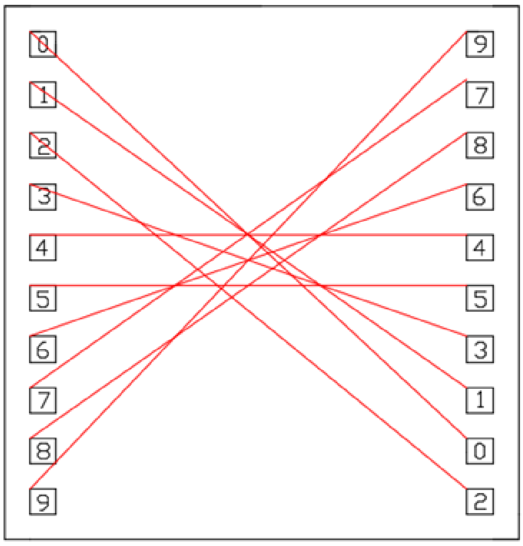

Fig 6: Paths of robots from initial point to destination point.

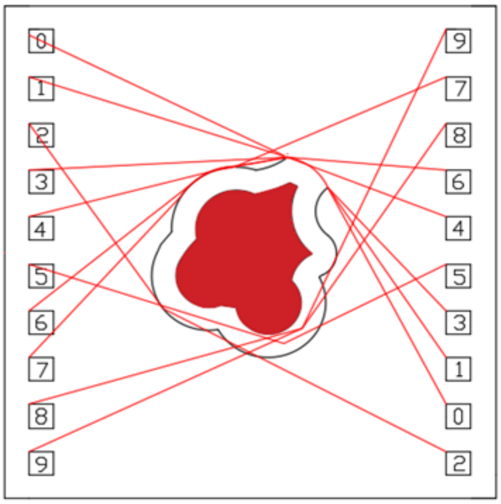

Fig.7: Deviated paths of the robots, when an irregularly curve shaped obstacle exists.

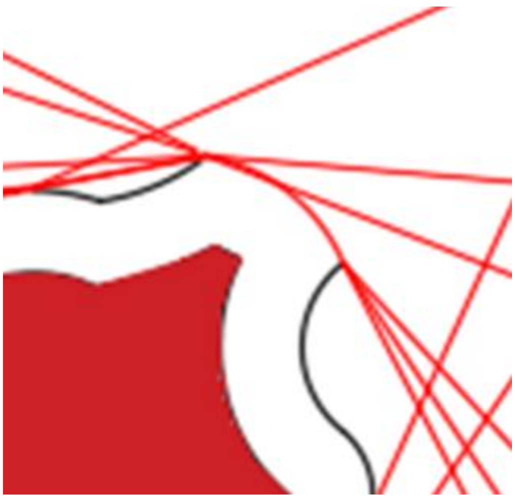

Fig 8: 3 Robots Numbered 0,1, & 3 are following the same curve while moving from initial to final targets.

IV. RESULTS

The results are obtained by the curve fitting method when simultaneous motion of multiple robots traces around the polygonal or curved obstacle. The lagrange interpolation method is used to fit the motion of the robot in a curve so as to ensure smoothness and making the robot to move in minimal path distance if it encounters a curved shaped obstacle. The figures illustrates the motion with the specified trajectory avoiding the collision of the robots with other robots and the static obstacles.

V. FUTURE SCOPE

In the present work, the robots have the credential to move simultaneously. The dynamic behavior can be seen even if the obstacle is added or a robot is added instantaneously. It is limited to 2D program where the third dimension is not included.. So the common challenge would be to create the same environment in a cluttered or irregular periphery. The future scope of this work would be appreciated if the third dimension, irregular boundary and velocity component is considered with multiple robots.

REFERENCES

[1] Abhijeet Ravankar, Ankit A Ravankar, Yukinori Kobayashi. Chao Chung Peng, "Path Smoothing techniques in Robot Navigation: State-of-the-Art, Current and Future Challenges", Sensors(Basel), Vol 18(9), Sept 2018.

[2] Abhinash, Ashwani Kumar Jha, "Decision of robotics trajectory based on curve fitting techniques", International Journal of Scientific Engineering & Research, Vol 2, Issue 2, Feb 2014.

[3] Khuram Shahzad, Sohail Iqbal, Peter Bloodsworth, "Points based safe path planning of continuum robots", International Journal of Advanced Robotics Systems, doi : 10.5772/60857, July 2015.

[4] Mohammed Elbanhawi, Milan Simic, Reza N Jazar, "Continuous Path smoothing for Car Like Robots Using B Spline Curves", Journal of Intelligent and Robotic Systems, Vol 80, pp 23-56, Dec 2015.

[5] Uk-Youl Huh and Seong-Ryong Chang, "A G^2 Continuous Path-smoothing Algorithm Using Modified Quadratic Polynomial Interpolation", International Journal of Advanced Robotics Systems, Doi: 10.5772/57340, Jan 2014.

[6] Ram Kishan dewangan, Anupam Shukla & W. Wilfred Godfrey, "Path Planning for Multiple Robots by Priority Assignment", Computational Intelligence : Theories, Applications & Future Directions – Vol II, pp 161-172, Sept 2018.

[7] K Prasanna Lakshmi, "Collision Avoidance Recursive Algorithm", International Journal of Mechanical Engineering and Robotics Research, Vol. 3, No.1, pp 312-319, Jan 2014.