Application of Genetic Algorithm for Trajectory Planning of Two Degrees of Freedom Robot Arm With Two Dimensions

Aphirak Khadwilard
Faculty of Engineering, Rajamangala University of Technology
Lanna Tak Campus, Tak, Thailand 63000.
Pupong Pongcharoen
Faculty of Engineering, Naresuan University, Phitsanulok, Thailand 65000.
Corresponding author email: Pupongp@yahoo.com; Pupongp@nu.ac.th

Abstract

In this work, the trajectory planning problem of a two-link robot arm in a two dimensional workspace containing obstacles was considered. The objective of this research was to apply genetic algorithm (GA) to minimise the rotation angles of a two-joint robot arm. The algorithm was used to find interior points, which were then used to interpolate the robot trajectory by avoiding obstacles. The methods for point-to-point interpolation including linear and cubic spline interpolation methods were investigated. However, due to the nature of stochastic search conducted within the GA, the investigation of this study was not only focused on the quality of the solution itself but also on the investigation of appropriate setting of GA parameters.

Half fractional factorial experimental design was carried out with five replications. The results were then analyzed using a general linear form of analysis of variance and main effect plots. It was found that the experimental results obtained by using the cubic spline interpolation method generally produced better results than those of using the linear spline interpolation method. The analysis of variance on the results suggested some GA parameters were found to be statistically significant factors.

Keywords: Robot Manipulator; Trajectory Planning; Path Interpolation; Genetic Algorithm.

1. Introduction

The roles of a robot arm such as for assembling, welding, picking and/or placing items are widely used for manufacturing products in advanced industries. In a typical working station, the movement of the arm may be obstructed by other facilities or the product itself. The path estimation of the robot arm, called trajectory planning, is one of the most challenging tasks for engineers. The main problem is to find an optimal trajectory path from a starting point to a target by minimising the rotation angles of robot links while avoiding obstacles. Minimising rotation angles for each robot link will result in shortening the movement of the robot tip and reducing energy consumption for motors and controllers.

The trajectory planning problem has been studied over the last decade. Tian and Collins [10] studied the trajectory planning problem for a two-degree-of freedom robot in a workspace with point obstacles. The goal is to move the end-effector of the robot from a given starting point to a target while avoiding the point obstacles. They applied GA to search interior points between the starting point and the target. The Hermite cubic interpolation method is used to estimate the path. Garg and Kumar [8] presented the formulation and application of a strategy for the determination of an optimal trajectory for a multiple robotic configuration. Simulated annealing (SA) and GA have been used as the optimisation techniques. Yue et.al [7] studied the problem of point-to-point trajectory planning for flexible redundant robot manipulators (FRM) in joint space. A trajectory interpolation method to minimise vibration and executing time of a point-to-point motion was based on GA. Nearchou [6] modified GA to solve the inverse kinematics problem of redundant robots operating in environments with obstacles. The goal of this algorithm is to simultaneously minimise the end-effector's positional error and the robot's total joint displacements, while the robot is safely moving in its entire free space. Nearchou and Aspragathos [2] solved the problem of point-to-point motion of redundant robot manipulators working in environments with obstacles considered. The problem is formulated as a constrained optimisation problem and is solved using a method based on GA. From the aforementioned work, both interpolation methods and the parameters' settings of genetic algorithm were however assigned in an ad hoc fashion. Furthermore, a comparison of interpolation methods has not been investigated.

The objective of this work was to apply GA to minimise robot links' angles by finding the optimal interior points, which were then used to interpolate the robot trajectory by avoiding obstacles. The appropriate settings of GA parameters including a combination of population size and the number of generations, probabilities of crossover and mutation and chromosome selection methods were also investigated.

This paper first reviews the characteristics of the robot arm trajectory problem including interpolation methods. Section 3 describes the genetic algorithm for robot arm trajectory planning. Section 4 presents the experimental design and analysis of the experimental results followed by the conclusions in section 5.

2. Robot Arm Trajectory Problem (Ratp)

In this work, we consider a two degrees of freedom robot arm with two dimensions shown in Fig. 1.

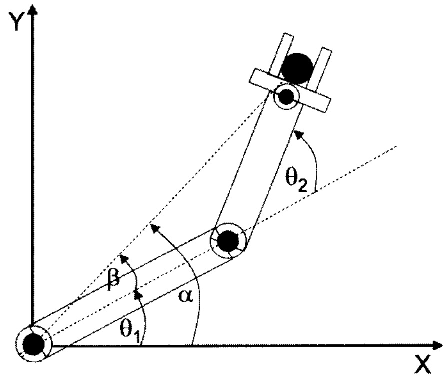

Fig. 1 Two degrees of freedom robot arm

We can use trigonometry to obtain the following expressions for the coordinates of the elbow and the hand. For the elbow, $x_{elbow} = L_1 \cos \theta_1$ and $y_{elbow} = L_1 \sin \theta_1$. The hand's coordinates (x,y) are given by:

$$x = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2)$$
$$y = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2)$$

These two equations can be solved for θ_1 and θ_2 in terms of x and y so that we can determine the elbow and arm angles necessary to place the hand at a specified position given by x and y. We omit the details of the solution and just state the results.

$$R^2 = x^2 + y^2$$

ผิดพลาด! วัตถุไม่สามารถถูกสร้างจากการแก้ไขรหัสเขต
ข้อมูล

ผิดพลาด! วัตถุไม่สามารถถูกสร้างจากการแก้ไขรหัสเขต
ข้อมูล

ผิดพลาด! วัตถุไม่สามารถถูกสร้างจากการแก้ไขรหัสเขต
ข้อมูล

ผิดพลาด! วัตถุไม่สามารถถูกสร้างจากการแก้ไขรหัสเขต
ข้อมูล

In Fig. 2, P_{start} and P_{target} are the given start and target points of the robot end-effector. P_{01}, P_{02}, and P_{03} represent the point obstacles in three different areas. $P_{interior}$ is the interior point, which is used to formulate the trajectory path of the end-effector of the robot arm.

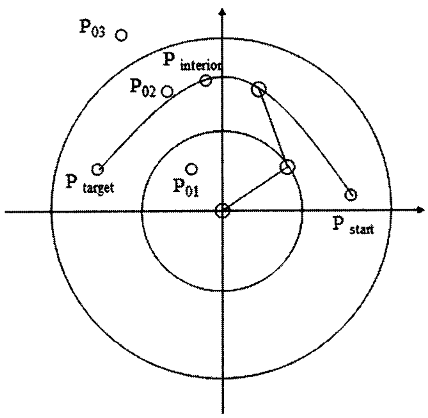

Fig. 2 Two degrees of freedom robot arm in work space

There are three cases for obstacles that we must consider when planning the trajectory. Let the distance from the obstacle to the origin of the robot be L_0. L_1 and L_2 are the length of links 1 and 2, respectively. The three cases are as follows:

(1) $L_0 < L_1$
(2) $L_1 < L_0 < L_1 + L_2$
(3) $L_0 > L_1 + L_2$

In the first case, the position of the obstacle has radial L_0 less than L_1. Thus this case is impossible for trajectory planning for a robot arm. If we meet this case, we will not consider it. The second case is the case that the obstacle position is between radial L_1 and $L_1 + L_2$. This case is possible for trajectory planning. Thus, in this work we consider this case only. The third case is when obstacle position L_0 is greater than the sum of $L_1 + L_2$. This position has no effect on trajectory planning. Thus the trajectory planning does not need to consider these obstacles.

In this work, linear and cubic spline interpolation methods were used to construct path. The concept of the spline originated from the drafting technique of using a thin, flexible strip (called a spline) to draw smooth curves through a set of points. The process is depicted in Fig. 3 for a series of five pins (called knots). In this technique, the drafter places paper over a wooden board and hammers nails or pins into the paper (and board) at the location of the data points. A smooth cubic curve results from interweaving the strip between the pins. Hence, the name "cubic spline" has been adopted for polynomials of this type.

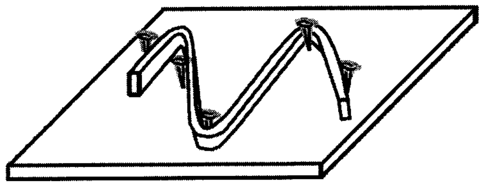

Fig. 3 The drafting technique of using a spline to draw smooth curves.

Linear Splines

The simplest connection between two points is a straight line. The first-order splines for a group of ordered data points can be defined as a set of liner functions,

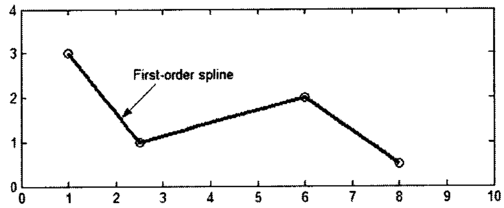

Fig.4 Spline fits of a set of four points. Linear spline

$$f(x) = f(x_0) + m_0(x - x_0) \quad x_0 \le x \le x_1$$
$$f(x) = f(x_1) + m_1(x - x_1) \quad x_1 \le x \le x_2$$
$$\vdots$$
$$f(x) = f(x_{n-1}) + m_{n-1}(x - x_{n-1}) \quad x_n \le x \le x_{n-1}$$

where m_i is the slope of the straight line connecting the points:

ผิดพลาด! วัตถุไม่สามารถถูกสร้างจากการแก้ไขรหัสเขตข้อมูล

Cubic Splines

The objective in cubic splines is to derive a third-order polynomial for each interval between knots. On the interval $x_i<x<x_{i+1}$ the cubic spline $s_3(x)_j$ is given by a cubic polynomial of the form:

$$S_3(x)_j = a_{j0} + a_{j1}(x - x_i) + a_{j2}(x - x_i)^2 + a_{j3}(x - x_i)^3$$
$$x_i < x < x_{i+1}$$

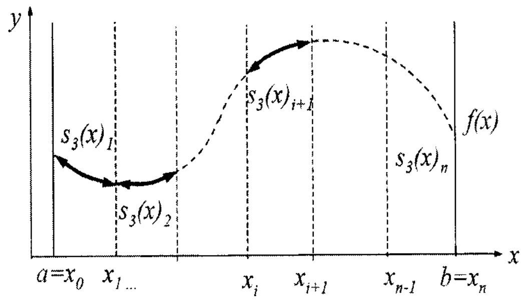

Fig.5 $f(x)$ over the interval $x_i<x<x_{i+1}$ is approximated by a cubic spline.

3. Genetic Algorithm For Robot Arm Trajectory (Garat)

Genetic Algorithm (GA) is adaptive random search method. In dealing with function optimisation, the minimum/maximum of a function is found based on a variable beginning with one or more starting points. GA evolves a set of points. The basic element of a GA is the artificial individual. Similar to a natural individual, an artificial individual consists of a chromosome and a fitness value. The fitness of an individual describes how well an individual is adapted to nature. It determines the individual's likelihood for survival and mating. Every change of the chromosome leads to a change of the individual fitness. In this work (searching a minimum of the rotation angles) an artificial individual consists of a value of R and θ, which is the position of interior points to construct paths and find inverse kinematics (all θ_1 and θ_2), for each point on the path. Fig. 6 shows the general procedure of Genetic Algorithms, which is adopted from Gen and Cheng [3]. The procedure consists of the population initialisation, genetic operations, fitness measure, and chromosome selection.

Chromosome Representation and Initialisation

The underlying GA implementation works based on a binary string representation of a number (0 and 1). Each chromosome is separated into two sectors. The first sector represents a radius (R), which is in solution space. The second sector represents an angle (θ) that is between an angle of a start point and an obstacle. The two sectors define a candidate interior point. In the following, an example of binary code is shown in Fig. 7. A 20 bit length of positive integer number is used for simplification.

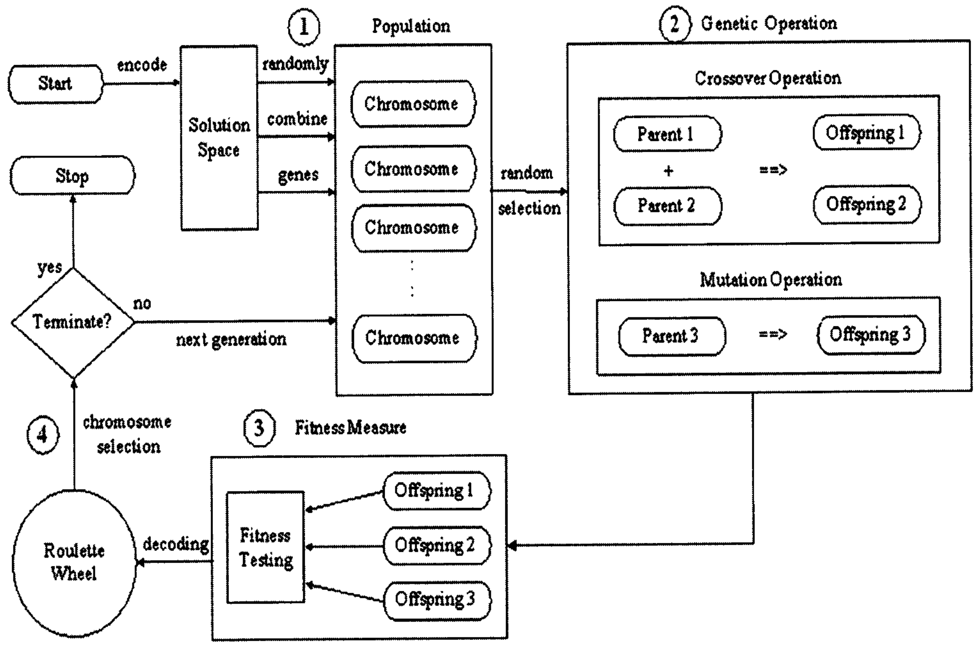

Fig. 6 A structure of Genetic Algorithm [9].