JOURNAL OF MECHANICAL ENGINEERING AND SCIENCES
ISSN: 2289-4659 e-ISSN: 2231-8380
VOLUME 18, ISSUE 1, 2024, 9928 – 9948
DOI: https://doi.org/10.15282/jmes.18.1.2024.10.0785

REVIEW ARTICLE

Review of path planning of welding robot based on spline curve

G. He[1,2], H. H. Teo[1*], L. K. Moey[3]

[1] Centre for Advanced Engineering Design, Faculty of Engineering, Built Environment & Information Technology, SEGi University, 47810, Selangor, Malaysia
Phone: +60177882314; Fax.: +60177882314
[2] Centre for Robotics Innovation, Department of Electronic Engineering, Anhui Wenda University of Information Engineering, 231201 Hefei, Anhui, China
[3] Centre for Modelling and Simulation, Faculty of Engineering, Built Environment & Information Technology, SEGi University, 47810, Selangor, Malaysia

ABSTRACT - This paper assesses the efficacy of intelligent path planning for welding robots utilizing splines. Traditional path planning methods can result in inefficient and inaccurate welding operations. The study reviews current research and case studies to appraise the practical application of spline-based path planning across diverse industrial scenarios. It underscores the benefits of discovering the shortest path and reducing cycle time while acknowledging challenges such as calibration accuracy and sensitivity to sensor data noise. The introduction of artificial intelligence algorithms in automobile welding path planning enables a more precise replication of the body's design curve, ensuring the continuity and smoothness of the welding process. This, in turn, fosters further automation and optimization of the automotive welding manufacturing process. The current research concentrates on integrating intelligent optimization algorithms and spline curves to provide an efficient and intelligent method for welding path planning. Intelligent path planning based on spline curves demonstrates significant potential in enhancing welding efficiency, determining the shortest path, and holds promising applications in the broader research field of welding path planning.

ARTICLE HISTORY
Received : 24th Aug. 2023
Revised : 11th Dec. 2023
Accepted : 21st Dec. 2023
Published : 30th Mar. 2024

KEYWORDS
Welding robot
Spline curve
Path planning
Intelligent algorithm
Industrial automation

1. INTRODUCTION

To attain manufacturing competitiveness, high productivity, low cost, and superior quality, automated and robotic welding is widely employed in the manufacturing sector [1]. On the other hand, manual teaching pendant programming is still often used for industrial welding robot task planning, which can be time-consuming, especially when dealing with large-scale welding joints or seams [2]. Intelligent robotic welding systems are developed to accomplish desired welding results due to their flexibility, efficiency, and precision. A complete intelligent welding robot system can be divided into three levels which cooperate with each other to achieve efficient welding. Management decision-making level processes information and planning, the system optimization layer optimizes the routing and scheduling and the system control layer integrates the system and monitors the parameters (see Figure 1 [3]). The order in which welding routes are arranged throughout the welding process can significantly affect variables including path length, energy use, and welding deformation. Optimizing the welding path of the robot is usually transformed into a multi-objective optimization problem. Among them, this study focuses on the impact of weld tracking, parameter extraction, welding pool monitoring, initial point guidance and other factors on the work of welding robots. [4]. In this study, virtual robot software was used to simulate the gas metal arc welding robot, and the influence of arc speed, arc length and arc time on welding defects was deeply studied [5]. The framework of welding robot route optimization is depicted in Figure 2 [6].

For industrial robots, path planning is to find an optimal motion path that traverses all task points for the robot according to the task and ensure reasonable robot posture [7]. The design of a welding robot's path is seen as a multi-objective optimization problem, with the avoidance of obstacles serving as a constraint and the optimization of the path's length and energy consumption as optimization objectives [8]. Intelligent optimization methods are used to deal with these discrete multi-objective optimization issues and produce the desired optimization results. Therefore, the path planning ability of robot is improved and the path optimization technology is studied [9]. Geometric information can be retrieved from the welding path described in the workpiece CAD model to provide the welding path's position coordinates, and then a spline curve can be used to approximated these position values to build a smooth curve, which is an excellent way for welding path planning [10]. Saravanan used the NURBS curve to map the path of the robot manipulators so that it could satisfy various constraints [11]. The time optimal model for robot path planning is established by the use of Bezier curves based on the requirements of a tower crane standard dyne welding robot, which needs to plan its weld path perfectly throughout operation [12]. A study was conducted to generate smooth and time-optimal trajectories for industrial robots by dividing the path into multiple segments and connecting them using spline curves. This approach reduces computational complexity by automatically constraining jerks and eliminating the need for jerk constraints during the optimization process [13].

***CORRESPONDING AUTHOR | H. H. Teo** | ✉ hhteo@segi.edu.my

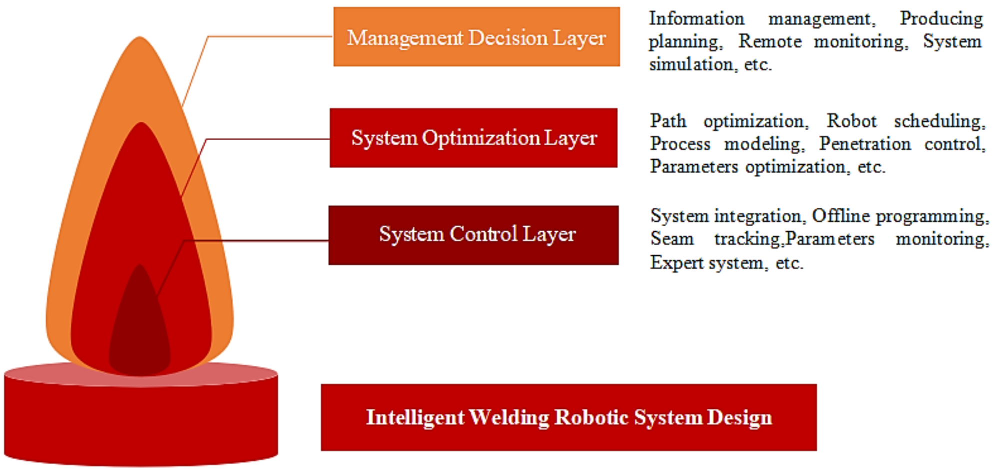

Figure 1. Intelligent welding robotic system purpose description [3]

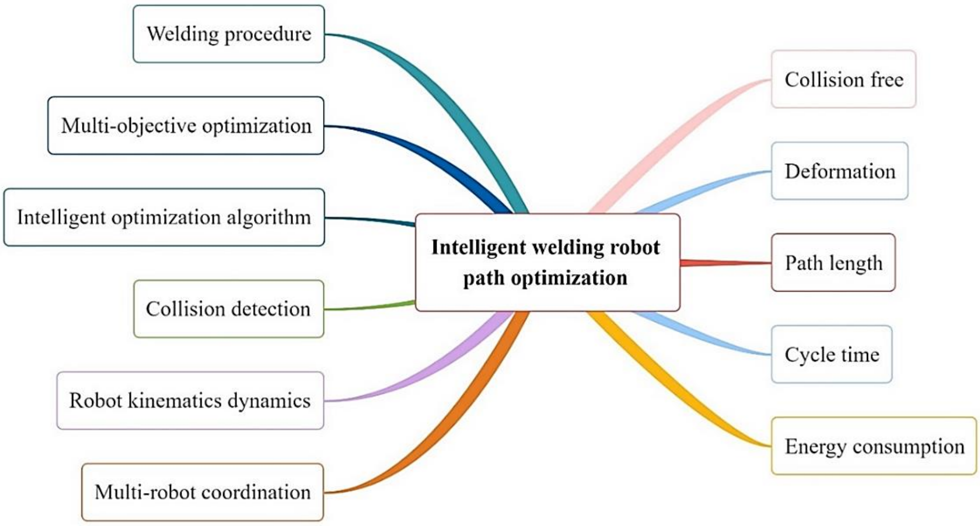

Figure 2. Intelligent path optimization structure for welding robot [6]

When the welds have simple shapes, such as straight lines and broken lines, mathematical models can actually describe them [14]. However, mathematical modeling for elaborate welds is time-consuming and only relevant to certain work-pieces [15]. During automotive welding, the body structure usually contains a variety of complex curved surfaces and shapes [16]. Welding of irregular curves is a challenging task. Spline curve as a smooth curve fitting method, which is widely used in path planning, brings many advantages to automotive welding. Therefore, the effective path planning method is of great significance to improve the intelligence of welding robots and improve the welding production efficiency and quality. Research efforts have concentrated on the development of autonomous and efficient path optimization methods to improve robot path planning skills.

2. PATH PLANNING METHOD BASED ON SPLINE CURVES

The challenge of path planning in robots is both substantial and intricate. The goal is to ascertain the optimal course of action for a robot, starting from a given point and reaching a destination, with the aim of efficiently completing tasks or precisely reaching the intended location. Approximation splines are commonly employed for constructing surfaces of objects, while interpolation splines are typically utilized for digital mapping and prove beneficial in robot path planning [17]. The subsequent sections will provide a detailed analysis of the application of spline curves in the context of path planning.

2.1 The Principle of Spline Curve

Spline curve is a smooth curve fitting technique widely used in mathematics and computer graphics. Its smoothness is determined by its mathematical construction, and curves are defined by a group of control points. These points are in charge of the curve's overall shape [18]. The polynomic curve is interpolated through known discrete data points so that all the points are connected into a smooth curve. It is simple to calculate, but easy to produce oscillation phenomenon. Bezier curve, B-spline curve and the Non-uniform Rational B-spline (NURBS) curve have more advantages. This type of curve can be defined by a set of control points, and the shape of the curve can be easily modified by changing the position and weight of these control points.

2.1.1 Polynomial interpolation

In the 1970s, with the continuous development of computer science and mathematical modeling techniques, the study of path planning gradually became more systematic and in-depth [19]. In this process, Polynomial Interpolation began to be widely used. It has the advantages of smoothness and differentiability, and is suitable for describing and controlling continuous motion trajectories. To control smoothness in path planning, Lagrange interpolation polynomials are commonly used. It is using Lagrange basis functions to approximate a given set of data points. Given $n+1$ distinct data points $P_j(x)(x_0, y_0), (x_1, y_1), \cdots, (x_n, y_n)$, the expression for the Lagrange interpolation polynomial, as shown in Eq. (1) [20]:

$$P(x) = \sum_{j=0}^{n} y_j \cdot P_j(x) \tag{1}$$

where $P_j(x)$ is the Lagrange basis function defined as Eq. (2) [20]:

$$P_j(x) = \prod_{k=0, k\neq j}^{n} \frac{x - x_k}{x_j - x_k} \tag{2}$$

where, $j = 0,1, \cdots, n$. Each $P_j(x)$ is an n degree polynomial, equal to 1 at x_j and 0 at other x_k (where $k \neq 0$).

One of the advantages of the Lagrange interpolation polynomial is its intuitive and easily understandable form. However, for large datasets, the computation and storage of the Lagrange interpolation polynomial can become complex, and potential numerical instability in numerical calculations should be taken into consideration. In recent years, Quadratic Polynomial and Membership Interpolation (QPMI) algorithm was recently proposed in a work by Chang and Huh [21]. This algorithm avoids Runge's phenomenon and the weakness of spline interpolation by using only the quadratic polynomials and membership functions to create a G^2 continuous path [22]. Chang et al. [23] use interpolation to present a collision free continuous G^2 path. However, their techniques necessitated rewriting smooth pathways and explicit collision detection checks, which can be costly in busy environments. Quintic polynomials are used in other methods [24].

2.1.2 Cubic spline

An apparent characteristic of the Lagrange interpolation polynomial is that it passes through all given data points. However, in certain situations, especially when dealing with a large number of data points, using more numerically stable methods such as cubic spline interpolation may be more appropriate [25]. In the late 20th century, with the rise of robotics and path planning applications, Cubic spline became a common choice for generating smooth, continuous paths. In path planning, when using cubic spline interpolation, a cubic spline function can be employed between every two adjacent discrete points. Assuming a set of discrete points on the path as $(x_0, y_0), (x_1, y_1), \cdots, (x_n, y_n)$, where $x_0 < x_1 < \cdots < x_n$. The cubic spline function on each interval $[x_i,\ x_{i+1}]$ can be represented as Eq. (3) [26]:

$$s_i(t) = a_i + b_i(t - x_i) + c_i(t - x_i)^2 + d_i(t - x_i)^3 \tag{3}$$

where, t is the parameter, usually ranging within the interval $[x_i,\ x_{i+1}]$. The coefficients a_i, b_i, c_i, d_i are determined through interpolation conditions.

The interpolation conditions typically include, as shown in Eq. (4) [22]:

$$\begin{cases} s_i(x_i) = y_i \\ s_i'(x_i) = s_{i-1}'(x_i) \\ s_i''(x_i) = s_{i-1}''(x_i) \end{cases} \tag{4}$$

By satisfying these conditions, a system of $4(n-1)$ linear equations are obtained, where n is the number of discrete points. In this way, a linear system of equations is formed, and by solving this system, the coefficients for each interval can be determined. These coefficients define the cubic spline functions on each interval, forming a smooth interpolation for the entire path. In path planning, additional constraint conditions, such as minimum curvature or minimum velocity, are often incorporated based on the specific requirements of the problem to optimize the performance of the path planning.

The interpolation of cubic splines has been used extensively. For instance, the land surface temperatures was regulated using cubic spline interpolation [27]. Suspicious values in the temperature observation data can be more clearly highlighted by interpolating the temperature measurements for each altitude segment. Additionally, the manipulator's route smoothness was investigated and the UAV's trajectory was simulated using cubic spline interpolation [28]. Jianfang Lian et al. [29] proposed a path planning optimization method based on cubic spline interpolation, and proposed a chaotic adaptive particle swarm optimization to optimize control points in cubic spline interpolation. To maintain the smoothness of the robot's motion path.

2.1.3 Bezier curve

Since 1962, Pierre Bézier, an engineer at Renault Automobile Company in France, introduced a distinctive method for the approximate construction of parametric curves and surfaces in Computer-Aided Geometric Design (CAGD) [30]. Bezier curve combines modern approximation theory with geometry, which has become an important mathematical method of computer-aided geometric shape design [31]. The advantages are obvious. Bezier curve vector equation of order N, as shown in Eq. (5) [32].

$$\theta(u) = C_n^0(1-u)^n P_0 + C_n^1(1-u)^{n-1}uP_1 + \cdots + C_n^j(1-u)^{n-j}P_j + \cdots C_n^n u^n P_n$$

$$\theta(u) = \sum_{j=0}^{n} C_n^j (1-u)^{n-j} u^j P_j \tag{5}$$

where, P_0, $P_1 \cdots P_j$ is space vector, u is the parameter.

Bezier curve transition curve advantage, get the approbation of experts. It is applied to robot path planning. It is also confirmed that the curves for angular displacement, angular velocity, and angular acceleration are smooth [33]. The disadvantage is that Bezier curves and control polygons are far apart, not easy to control; Modifying any control vertex on the Bezier curve without local modifiability causes the entire Bezier curve to change [34]. So, the more control vertices, the more Bezier curves there are, the more unstable the curves become.

2.1.4 B-Spline curve

To address the drawbacks of the Bezier method, the B-spline approach is presented. In 1974, Gordon and Riesenfeld used the standard algorithm for parametric B-splines for the first time, based on an in-depth study of the Bezier method [35]. Every advantage of the Bezier method is available with the B-spline approach. The B-spline curve is close to the control polygon and is therefore simple to adjust. The ability to locally modify a curve is a characteristic of curves [36]. B-spline curve's control vertex changes only a few of the linked curve segments. Only the number of B-spline segments grows as the control vertices do, not the number of B-spline curves. Therefore, the B-spline method is getting more and more attention in CAGD.

B-spline is one of the geometric representation methods commonly used in CAD systems and has been widely used in the field of robot application. The equation of the B-spline of degree p as shown in Eq. (6) [37].

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u) d_i \qquad a \le u \le b \tag{6}$$

where, d_i is the distance of the control point and is represented as a scalar, $N_{i,p}(u)$ are canonical B-spline basis functions with p degrees, u is the parameter, $C(u)$ is represented as a scalar.

B-spline basis function recursive formula, is shown in Eq. (7) [38]:

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1, u_i \le u \le u_{i+1} \\ o, others \end{cases} \\ N_{i,k}(u) = \dfrac{u-u_i}{u_{i+k}-u_i} N_{i,k-1}(u) + \dfrac{u_{i+k+1}-u}{u_{i+k+1}-u_{i+1}} N_{i+1,k-1}(u) \\ Appoint\ \dfrac{0}{0} = 0 \end{cases} \tag{7}$$

From the above recursive formula, it can be seen that to calculate $N(u)$, u_i, $u_{i+1} \cdots u_{i+k+1}$ has $K+2$ nodes. The interval $[u_i, u_{i+k+1}]$ is called the supporting interval of $N_{i,k}(u)$. There are many studies on B spline. For roundabout exit traverse planners, a parametric B-spline-based path planning approach has been suggested by certain researchers [39]. According to certain academic research, high-degree B-spline interpolation has excellent Fourier features, little interpolation error, and reasonable computation durations, making it appropriate for using in image processing applications [40]. Some scholars studies multi-robot path planning algorithm, effectively addressing the problem by improving the artificial potential field method and using B-spline curve optimization [41]. Some scholars proposes a composite A-star and B-spline algorithm to generate C2 path that fits the motion constraints of the Autonomous Underwater Vehicle (AUV), eliminating the discontinuity caused by multi-segment paths [42]. A method for creating a continuous trajectory from a set of waypoints that reflect a static path is presented by certain researchers. Additionally, it outlines a technique for quickly altering the trajectory in reaction to a changing environment. [43]. Some scholars present Spline-based Convolutional Neural Networks (SCNNs), a variant of deep neural networks suitable for irregular structured and geometric inputs like graphs or meshes [44]. B-splines have good localization and continuity, and can effectively represent complex free curve surface shapes, so they are widely used. However, when B-spline interpolation or approximation is used for curve reconstruction, the parameterization of the data points may affect the final approximation results. In this case, the nodes in the B-spline can be adjusted or the method of finding better data points can be investigated.

2.1.5 Non-uniform rational B-spline curve

In 1975, Versprille of Syraeuse University first proposed the NURBS curve in his doctoral thesis [24]. The NURBS approach is a useful addition to the B-spline and Bezier methods. It has become one of the most popular techniques for describing curves and surfaces. One of the greatest advantages of NURBS curves is that they can represent straight lines, conic curves and free curves uniformly [45]. It also has some disadvantages, for example, improper weight selection may lead to poor parameterization, or even destroy the subsequent curve and surface structure. There's also a lack of some basic algorithms, like the integration algorithm. This brings great limitations to practical applications. Therefore, it is of great significance to further study NURBS method so as to improve and perfect NURBS method. The representation of the NURBS curve is shown in Eq. (8) [46].

$$F(u) = \frac{\sum_{i=0}^{n} w_i d_i N_{i,k}(u)}{\sum_{i=0}^{n} w_i N_{i,k}(u)}, u \in [0,1] \tag{8}$$

In the equation, u is the control variable of the curve with a range of range[0,1]; k is the number of repetitions, that indicates the strength of the interpolation basis function; w_i is the weight factor, which is the same as the number of control vertices, and it is necessary that $w_1 > 0$, $w_n > 0$ as well as all other weight factors be more than or equal to 0; $N_i,k(u)$ is called the basis function, which is chosen based on the node vector., it is shown in Eq. (9) [11]:

$$U=[u_0,u_1,\ldots,u_k,u_{k+1},\ldots,u_n,u_{n+1}, \ldots,u_{n+k+1}] \tag{9}$$

It only stipulates the selection method of the start and end node vectors, generally select $u_0 = u_1 = \cdots = u_k = 0, u_{n+1} = \cdots = u_{n+k+1} = 1$, how many middle n-k vectors can be selected in the two ways, the selection of the node sequence is different, and the shape of the NURBS curve is also different. $N_{i,k(u)}$ represents the basis function, which has many forms, the most common and easy understanding is the DeBoor-Cox recursive definition, which is similar to the recursive expression of B-spline, as show in Eq. (3) [47]. The NURBS method introduces weighting factors and denominators that enable practically accurate representation of standard analytic shapes, such as quadratic curved surfaces. It should be noted that most curved surfaces involved in industrial design or scientific research are not necessarily standard analytic shapes. For such free-form curved surfaces, the B-spline method is sufficient. The use of NURBS will introduce unnecessary parameters and complicate the process.

2.2 Comparison of Path Planning Methods based on Spline Curve

The predetermined path is used as the input condition for industrial robot path planning. This results in three output parameters, namely displacement, velocity and acceleration, and their relation to time. The two most fundamental path planning techniques are joint space path planning and Cartesian space path planning [48]. The former describes the path of the robot by changing the function of joint motion angle. Its advantage is that the calculation is simple and fast, and there is no need to solve the inverse kinematics. The disadvantage is that there is no way to predict the motion state of the robot from the starting position to the set end position. The latter describes the terminal pose motion of the robot in Cartesian space. Its benefits are obvious, and the end-effector's path form is simply to see. The substantial quantity of calculation is a drawback. Singularities could occur and inverse kinematics is necessary. One study showed that the optimization algorithm was combined with the travel time of the five-degree NURBS curve, and the reparameterization [46]. This path planning technique effectively lowers the robot's power consumption, accelerates execution, and enhances motion smoothness, as shown in Table 1.

Table 1. Path planning method based on spline curve

Method	Constraint	Control point flexibility	Smoothness	Angular displacement curve	Complexity	Reference
Polynomial interpolation	4	Low	Poor	Large displacement	Simple	[49, 50]
Cubic spline	>4	High	Poor	Large displacement	Complex	[51, 52]
Bezier Curve	4	High	Good	Control point positions	Simple	[31], [53]
B-spline curve	>4	High	Very good	Smooth and continuous	Complex	[37], [54]
NURBS	>4	High	Very good	Smooth and continuous	Complex	[38], [46]

From the above table, it can be seen that Cubic polynomial and Bezier Curve have a fixed number of control points and their adjustment flexibility is low. Higher order polynomial, B-spline curve and NURBS can adjust the position and weight of the control points, which can realize more flexible curve shape adjustment. Cubic polynomial has poorer smoothing, and there may be corner points and breaks. Higher order polynomial may generate oscillations and singular points, resulting in poorer smoothing. Bezier Curve has poorer smoothing. Higher order polynomial may produce oscillations and singularities, resulting in poor smoothing. Bezier Curve has better smoothing. The B-spline curve and NURBS have local control and can generate curves with good smoothness. In summary, Cubic polynomial and Bezier Curve are commonly used in computer graphics and simple curve fitting, while Higher order polynomial is suitable for surface modeling, data fitting and numerical simulation, etc. B-spline curve and NURBS have a wide range of applications in computer-aided design, surface modeling and path planning. B-spline curve and NURBS are widely used in computer-aided design, surface modeling, and path planning. It should be noted that the performance and applicability of different types of curves may be affected by specific application scenarios and the selection of control points, so the most suitable type of curve needs to be selected according to the actual application.

3. INTELLIGENT ALGORITHMS FOR PATH PLANNING

Scholars are continually improving path planning systems based on simple algorithms. Various path planning algorithms arise one after another over a long period of investigation by different scholars. Ioan et al. [55], proposed a feasible spatial partitioning method to solve the path generation problem in multi-obstacle environments. Riazi et al. [56], using intelligent algorithms to reduce energy consumption and peak power of industrial robots. Palmieri et al. [57], compared three bio-heuristic algorithms (firefly algorithm, particle swarm optimization algorithm and artificial bee algorithm), and the simulation results showed that in complex scenarios, strategies based on fireflies usually have superior performance and can reduce energy waste. Welding robot path optimization mainly includes the following work: optimization algorithm, obstacle avoidance strategy, multi-robot cooperative operation, parameter optimization. In order to solve the problem of local convergence and small distribution range of non-inferior solutions in discrete multi-objective optimization problems, it is necessary to integrate some optimization strategies with intelligent algorithms to obtain ideal results.

3.1 Classification and Description of Intelligent Algorithms

Although there are many different kinds of intelligent algorithms, they can be broadly categorized into three groups: Biological swarm intelligence techniques, evolutionary algorithms (EA) and other algorithms, as shown in Figure 3.

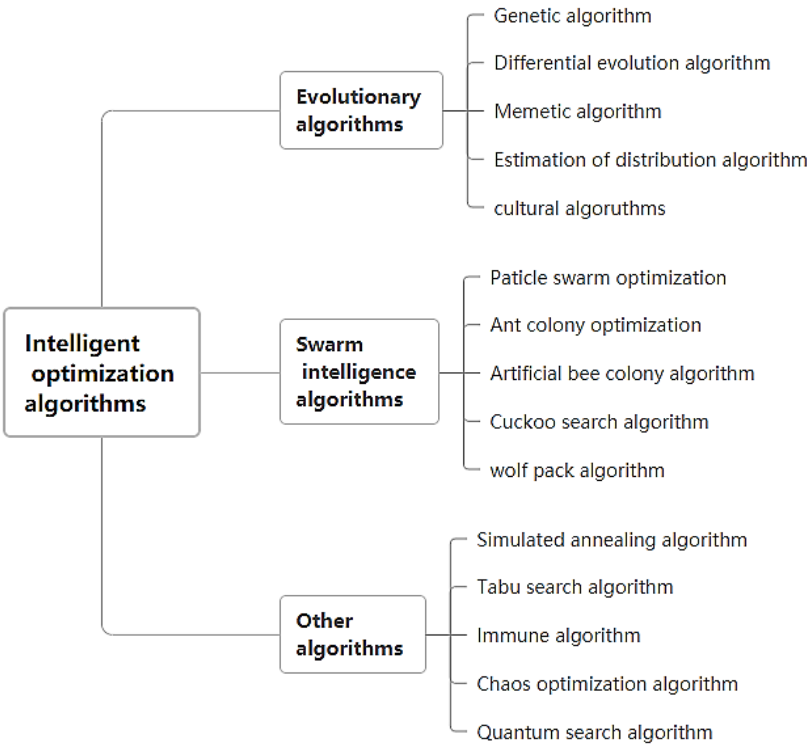

Figure 3. Intelligent algorithms for welding robot [3]

3.1.1 Genetic algorithm

By fitting tests in nature, Genetic Algorithm (GA) replicate Darwin's theory of survival [58]. Holland [59] proposed this technique in 1975. Chromosome representation, fitness selection, and biological heuristic operators make up the core elements of GA. Holland also created a novel component called inversion, which is frequently used in the creation of genetic algorithms [60]. Normally, chromosomes are represented as binary strings. Each locus (a specific region on the chromosome) in chromosomes contains two potential alleles (different variations of the gene)- 0 and 1. Chromosomes are thought of as points in the solution space. These are dealt with by utilizing genetic operators to substitute their populations iteratively. Each chromosome in a population is given a value applying the fitness function [61]. The fundamental procedures are selection, crossover, and mutation, which gradually bring the problem closer to the ideal solution. To plan the best faster path, genetic algorithm has been applied often [62]. In one study, genetic algorithms played an important role in the digital twin system of welding path planning for arc welding robots in the welding of ship sub-components, improving welding efficiency and quality by optimizing and selecting the optimal welding path [63]. Yimei Zhang [64] aims to discover a solution to the path planning problem of robots that addresses the problems of slow convergence and ease of locally optimal fall off, and proposes an adaptive selection technique based on the assessment of population diversity level. The feasibility and effectiveness of the proposed algorithm are verified by simulation in grid environment. The genetic algorithm has been widely applied to a wide range of issues because it was proposed earlier. The advantages of a genetic algorithm include parallel search, simplicity and universality, great robustness, etc.

3.1.2 Differential evolution

Differemtial Evolution (DE) is an evolutionary class of algorithms that mimic the process of natural selection and evolution. The optimal solution is found by iteratively improving the population of candidate solutions. Introduced by Kenneth Price and Rainer Storn in 1995 [65]. It is a stochastic, population-based optimization technique, which is suitable for solving optimization problems in continuous spaces. The DE algorithm operates in a population of candidate solutions (individuals), within the search space. It starts with a random initial population of solutions and proceeds through a series of generations. In each generation, new candidate solutions are generated by mutation, recombination (crossover) and selection operations. DE makes few assumptions about the problem, requires only a few parameter adjustments, and is relatively easy to implement [66]. It excels in handling complex, multimodal and nondifferentiable objective functions and is suitable for global optimization problems. However, DE also has some drawbacks. For example, it converges slowly when dealing with high dimensional spaces or complex optimization spaces. Sensitive to parameter selection (e.g., variation rate, crossover rate, and population size). Performs poorly on noisy objective functions or problems with discrete variables.

3.1.3 Memetic algorithm

Memetic Algorithm (MA) is an evolutionary optimization algorithm that integrates genetic algorithms with local search techniques [67]. It performs genetic operations on individuals during evolution and conducts local searches to enhance solution quality [68]. In a specific application, a hybrid path planning method based on the meme algorithm was proposed to address the intricate path planning problem in the machining of the body in white surface. Operating within the meme algorithm framework, the method eliminates redundant nodes and performs post-smoothing processing to obtain a smooth, collision-free optimal path set between solder joints. It constructs an objective function for traversing all solder joints with the shortest path length and the highest smoothness. Simulation results demonstrate the effectiveness of the hybrid path planning method based on the meme algorithm in optimizing the path of a spot welding robot [69]. MA is primarily applied to combinatorial optimization problems (such as the Traveling Salesman Problem (TSP) and the boxing problem), continuous optimization problems (including parameter optimization and function optimization), and model tuning problems. By combining global exploration and local search strategies, MA improves the search efficiency and convergence speed of the algorithm. Its combination of global and local search makes it more likely to find superior solutions, making it applicable to various optimization problems. However, it comes with the drawback of requiring careful parameter tuning due to the abundance of parameter settings.

3.1.4 Estimation of distribution algorithm

Estimation Distribution Algorithm (EDA) guides the search process by modeling the distribution of solutions in the problem space [70]. EDA achieves an efficient search for good solutions by constructing a model of the probability distribution of the solutions and generating new candidate solutions based on the model [71]. A study has proposed an approach for addressing the path planning problem of Autonomous Underwater Vehicles (AUVs) in dynamic environments using the Learning Fixed Height Histogram (LFHH) method based on the EDA. By integrating EDA with LFHH and employing a smoothing technique, the method facilitates a faster discovery of feasible paths for AUVs in complex dynamic underwater environments [72]. This method is independent of gradient information, making it well-suited for addressing non convex, multi-peaked, and high-dimensional optimization challenges. It offers valuable insights into the distribution of solutions within the problem space. Nevertheless, EDA demands meticulous attention to model construction and parameter settings, limiting its application to scenarios like combinatorial optimization, parameter search in optimization, and fine-tuning of machine learning models.

3.1.5 Cultural algorithms

Cultural algorithm (CA) introduces the concepts of cultural inheritance and social learning, combining individual evolution with the propagation of culture [73]. In this study, CA play a crucial role in dynamically adjusting parameter weights for different path-planning mechanisms based on the performance of each mechanism, contributing to an overall improvement in path-planning effectiveness [74]. These algorithms enhance the search capabilities, allowing broader exploration of solution spaces, particularly in optimization problems involving interactions among individuals. Despite their effectiveness in addressing complex optimization challenges such as parameter tuning and machine learning model selection, drawbacks include prolonged convergence times, intricate algorithm design, and the need for careful parameter settings.

3.1.6 Particle swarm optimisation

Particle Swarm Optimization (PSO) is an algorithm that takes inspiration from how birds forage [62]. When flying towards a target meal, a bird constantly emulates the flock's top performer in terms of velocity direction and size. For time-optimal Delta robot path planning, some researchers suggested an improved particle swarm optimization approach. It also realizes obstacle avoidance height control ability [75]. Some academics proposed a direct route planning method based on fuzzy reward and punishment theory and enhanced PSO. Its goal is to improve the efficiency of path planning for specific mission objectives [76]. According to the operation requirements of a citrus harvesting robot, a new optimization technique for harvesting sequence planning has been proposed. Inverse kinematics scheme selection and sequential programming dynamic programming are both realized. The results of simulations demonstrate that this strategy saves energy [77]. Some researchers proposed a PSO adjusted PID controller. It is used to regulate the orientation of the flexible manipulator both directly and indirectly in order to lessen the manipulator's vibration [78]. Some researchers proposed the genetic algorithm and the discrete particle swarm optimization algorithm as intelligent welding path optimization techniques to optimize the welding robot's path. By optimizing operator selection, GA achieves the highest generation adapted to the specific [79]. The simplicity of use and quick convergence of the particle swarm algorithm are advantages. It is frequently utilised in a variety of applications, including system identification, neural network training, combinatorial optimisation issues, multi-objective constrained optimisation, and others.

3.1.7 Ant colony algorithm

The fundamental principle of the Ant Colony Algorithm (AC) is derived from the observation that ants eventually discover the shortest path by releasing pheromones and using the size of the pheromone concentration while searching for food over time [80]. The traveler path planning problem was the main application for the ant colony method at first. Due to the local region or neighborhood's tiny size and constrained scope inside the optimization problem, the search efficiency is highly evident. However, the ant colony algorithm will encounter the issue of sluggish convergence when the constraints on the problem are relaxed or the number of swarms is very high [81]. Ye Xuan et al. [80] incorporated energy loss, processing time, and path smoothness as optimization objectives in their study on coverage path design. They employed an enhanced ant colony algorithm in their research and proposed two approaches to prevent the algorithm from converging to a local optimum, thus enhancing the system's global search capability. The DL-ACO algorithm, a successful double-layer ant colony optimization method for autonomous robot navigation, was created by certain researchers. Two concurrently running ant colony algorithms make up the DL-ACO. The findings indicate that this approach can produce more collision-free paths [82]. A variety of methods of improvement have been put forth in recent years by scholars from various nations to increase the convergence of their algorithms. This technique has been used to solve issues related to vehicle scheduling, large-scale integrated circuit design, and robot collaboration in various ways.

3.1.8 Artificial bee colony algorithm

Artificial Bee Colony (ABC) algorithm simulates the search strategy of bees when searching for food [83]. To achieve the best time and distance, Savsani [84] presented a path-planning strategy based on an enhanced artificial bee colony algorithm. The artificial bee colony approach was improved by combining the proper learning algorithm with the joint Angle, angular velocity, and running time as parameters. According to the experimental data, the updated artificial bee colony algorithm may successfully optimize the travelling distance and time. The search and updating of candidate solutions is achieved by recruiting the roles of worker bees, observer bees, and scout bees. ABC is simple to apply, easy to implement, and requires fewer parameters to be tuned. However, it exists a slow convergence speed and is not suitable for complex problems. It is mainly applied in machine learning model parameter tuning and path planning.

3.1.9 Cuckoo search algorithm

Cuckoo Search Algorithm (CS) simulates the foraging behavior of cuckoo populations for new nests, utilizing random generation and updating of nests for solution search [85]. Wang et al. [86] proposed a hybrid path planning strategy for unknown 3D environments by combining differential evolution algorithms with CS to accelerate global convergence. This faster convergence enhances the exploration capabilities of aerial robots in three-dimensional environments. Xie et al. [87] demonstrated three-dimensional environment exploration using the CS algorithm, specifically addressing air path planning. The enhanced CS model integrates differential evolution to streamline the cuckoo selection process, allowing the bird to act as an agent in determining the optimal course of action. While simple and easy to implement, it exhibits sensitivity to parameter settings and requires careful tuning, resulting in slow convergence for complex problems. CS finds application in scenarios like parameter tuning and vehicle path planning, showcasing some global search capabilities.

3.1.10 Grey wolf optimizer

Grey Wolf Optimizer (GWO) emulates the collaborative strategy observed in wolf packs during the predation process, conducting solution searches by modelling the behaviour of leaders and followers within the pack [88]. Dewangan et al. [89] demonstrated that the Enhanced GWO algorithm exhibits superior capabilities in avoiding local optimality and exploring the search space. Ge et al. [90] proposed a hybrid algorithm by combining GWO with the Fruit Fly Optimization (FFO) algorithm to enhance local optimal solutions. Kamalova et al. [91] utilized a set of boundary points as input parameters, employing the global waypoint control technique of boundary-based exploration to generate points in unexplored areas without sensor signal transmission. GWO demonstrated superior searching performance compared to the PSO algorithm. It possesses both global and local search capabilities, making it applicable to various optimization problems. However, its sensitivity to parameter settings requires careful adjustment. GWO is predominantly utilized in function optimization, such as tuning machine learning model parameters.

3.1.11 Simulated annealing algorithm

Kirkpatrick et al. [92] presented simulated annealing algorithm, a meta-heuristic method based on a single solution. It can be compared to a hill-climbing strategy that iteratively seeks to enhance the most recent answer in terms of the goal function. The algorithm will accept better moves while rejecting fewer wealthy moves, which will increase the probability of eliminating the local optimal solution. The simulated annealing algorithm is simple to implement and requires fewer parameters in the algorithm. It is well adapted to obtain better global search results in nonlinear problems involving multiple constraints with large spatial diversity [93]. However, the simulated annealing algorithm is computationally intensive and takes a long time by randomly exploring and accepting inferior solutions. The initial solution has a large influence on the results, which tends to reduce the convergence speed and accuracy of the algorithm.

3.1.12 Immunization algorithm

The immunization algorithm is based on the principle of biological immunity, where the antigen corresponds to the problem to be solved and all possible solutions are antibodies [52]. Diversity does not need to be actively preserved because cells divide asexually, which means that even the slightest difference will be multiplied exponentially after numerous differentiations. The fact that self-equilibrium regulation makes use of concentration regulation mechanisms to either encourage or inhibit the maintenance of population variety is its biggest benefit [94]. A path planning method for underwater vehicles based on immune genetic algorithm is proposed in this paper. This method combines genetic algorithm and immune algorithm to optimize the path planning. Simulation results show the effectiveness of the proposed method [95]. The algorithm is better protected from entering the local ideal solution thanks to this design, which considerably enhances the local optimal capability and has a significant impact on the promotion and suppression of the concentration of the antibody population. The presence of memory cells can greatly improve the search speed and operational efficiency.

3.1.13 Tabu search algorithm

Tabu search is a local search-based optimization algorithm designed to prevent getting trapped in local optima by incorporating a "taboo table" [96]. This table records explored solutions and specific move operations, guiding the search process for improved exploration of the solution space. Khaksar et al. [97] integrated tabu search for intelligent sampling, employing two strategies—uniform sampling and Gaussian sampling—in the path planning process. Testing in diverse environments resulted in relatively shorter path lengths, faster running speeds, and lower memory and computation requirements. Lee et al. [98] introduced a distance-constrained multi-robot task path planning algorithm, combining mixed tabu search and 2-opt path planning. This algorithm excels in both path optimization and runtime, producing shorter, smoother paths with reduced energy consumption for multiple robots coordinating tasks. It effectively avoids local optimal solutions and exhibits some global search capabilities. However, it requires careful parameter adjustment due to its numerous settings. It is commonly utilized in combinatorial optimization problems like the TSP and graph colouring problems.

3.1.14 Chaos optimization algorithm

The chaos optimization algorithm explores solutions by incorporating chaotic mapping to generate a sequence of random numbers as candidate solutions in the search space [99]. In a related study, a novel approach to robot trajectory planning is introduced using a genetic chaotic optimization algorithm. This method employs a quintic polynomial to interpolate position nodes in the joint space, creating a trajectory model for the robot's motion. Subsequently, a genetic chaotic optimization algorithm, merging genetic and chaotic algorithms, is applied. The study demonstrates, through simulation and analysis, that this approach, considering velocity, acceleration, and acceleration constraints, achieves a smooth and time-optimal trajectory for the robot's end-effector [100]. The algorithm leverages the randomness and sensitivity of the chaotic nature to enhance search diversity. However, the algorithm's performance is influenced by the parameter settings and the choice of the chaotic map. It is commonly applied to parameter estimation problems in signal processing.

3.1.15 Quantum search algorithm

Quantum search algorithm simulates the characteristics of quantum superposition and coherence, employing "quantum bits" to search for solutions and reach the target solution in fewer iterations [101]. Another study introduces a robot path planning algorithm based on the Quantum-inspired Evolutionary Algorithm (QEA), tailored for large-scale optimization problems. Operating in a discretized environment, the QEA efficiently approximates optimal robot paths, surpassing the performance of traditional genetic algorithms in both static and dynamic scenarios. With a runtime of approximately 2 seconds, the proposed QEA demonstrates notable efficiency in addressing robot path planning optimization [102]. The quantum search algorithm excels at finding the target solution with fewer iterations. However, its implementation and understanding are relatively complex, requiring some knowledge of quantum computing. It is commonly applied to function optimization, particularly for optimization problems involving complex functions.

Table 2. Optimization algorithm for path planning of welding robots

Algorithm	Global Optimization Capability	Local Optimization Capability	Model Complexity	Convergence Speed	Precision	Efficiency	Stability	Recent Developments	Reference
Genetic algorithm	Good	Good	Low to Moderate	Low	Relatively Low	High	Unstable	Improved crossover and mutation operations	[103], [104], [105], [58], [60]
Ant colony algorithm	High	High	High	Moderate	Some precision	High	Unstable	Addressing local optima and convergence speed issues	[103], [104], [105], [80], [82]
Particle swarm optimization	Moderate	High	Low to Moderate	High	High	High	Unstable	Chaotic particle swarm algorithms and other enhancements	[103], [104], [105], [6], [106]
Artificial bee colony	High	High	Moderate to High	Moderate to High	Relatively Low	High	High	Improved versions focusing on convergence speed and global search capability	[103], [104], [105]
Wolf pack	High	High	Low to Moderate	High	Some precision	Effective	Unstable	Fine-tuning of algorithm parameters and improvements for large-scale problems	[103], [104], [105]
Tabu search	Fair	High	Low to Moderate	Moderate	High	Excellent in local search	Moderate	Enhanced neighborhood structures and adaptive parameter tuning	[103], [104], [105]
Simulated annealing	Fair	High	Low to Moderate	Moderate	Some precision	High	Moderate	Research focused on improving convergence speed and avoiding premature convergence	[103], [104], [105], [92], [94]
Immune algorithm	High	Moderate	Low	Low	Some precision	High	High	Research on adaptive parameter tuning and improvements for large-scale problems	[103], [104], [105], [52], [107]
Differential Evolution	High	Moderate	Low	Moderate	High	High	High	Improvements in mutation strategies and parameter tuning	[104], [65, 66]

Table 2 (cont.)

Algorithm	Global Optimization Capability	Local Optimization Capability	Model Complexity	Convergence Speed	Precision	Efficiency	Stability	Recent Developments	Reference
Memetic Algorithm	High	High	High	High	High	Combines global and local search	Moderate	Research on adaptive evolution and self-adaptive parameters	[104], [68]
Estimation of Distribution	High	Low	Moderate	Low	Some precision	High	High	Enhancements in distribution estimation and model updates	[104], [76, 77]
Cultural Algorithms	High	High	High	Moderate	Some precision	High	Moderate	Research on adaptive features and parameter tuning	[104], [73]
Artificial Bee Colony	Moderate	Low	Low	Moderate	Some precision	High	High	Improvements focused on convergence speed and global search capability	[104], 83]
Cuckoo Search	High	Low	Moderate	High	Some precision	High	Moderate	Fine-tuning of algorithm parameters and adaptive evolution	[104, 85]
Grey Wolf optimizer	High	Moderate	Moderate	High	Some precision	Effective	High	Fine-tuning of algorithm parameters and improvements for large-scale problems	[105, 88]
Tabu Search	High	High	Moderate	High	High	Excellent in local search	High	Enhanced neighborhood structures and adaptive parameter tuning	[105, 96]
Chaos Optimization	Moderate	Low	Low	Moderate	Some precision	Some global search	Low	Integration with deep learning and research on self-adaptive parameters	[105, 99]
Quantum Search	High	Moderate	High	High	Some precision	Potential for global search	High	Performance improvements and developments in quantum computing technology	[105, 101]

3.2 Comparison of Common Intelligent Algorithms for Path Planning of Welding Robots

Among numerous intelligent algorithms, genetic algorithms, particle swarm optimization, and traditional techniques are predominantly employed in the path planning of welding robots. Each algorithm comes with its own set of advantages and limitations, making it crucial for researchers and engineers to meticulously evaluate and choose the most suitable algorithm tailored to the specific requirements of welding tasks. In this comparative analysis, factors such as computational efficiency, convergence speed, and search capability will be examined. The objective of this study is to offer valuable insights into the selection of optimal algorithms that align with the unique demands of welding robot path planning. The performance of each optimization algorithm is compared in detail, as shown in Table 2.

By comparing their convergence speed and model complexity properties. The category of fast convergence speed and relatively low model complexity includes algorithms such as DE, GA, and ABC. These algorithms are able to converge to a better relatively quick solution during the optimization process, while their models are relatively simple and suitable for the fast solution of some problems. Algorithms with medium convergence speed and model complexity include CA, CS, TS, PSO, and WPA. These algorithms are able to strike a balance between optimization speed and model complexity, approaching the optimal solution relatively quickly while handling a certain degree of problem complexity. Among the algorithms with slower convergence speed and higher model complexity can be found EDA, Chaos Optimization, Quantum Search, AC, SC, and Immune algorithm, etc. These algorithms may require more computational resources and time but have the potential to handle more complex problems and find high-quality solutions.

4. INTELLIGENT PATH PLANNING APPLICATION OF WELDING ROBOT BASED ON SPLINE CURVE

Intelligent path planning is an important field in the application of industrial robots. Its goal is to generate an optimal or near-optimal motion trajectory for robots. Spline curve is a mathematical tool that can be used to describe smooth and continuous trajectories, so it is widely used in path planning of industrial robots. In this part, the application of a single curve in path planning is reviewed [108]. Then, an intelligent path planning method based on spline curve is listed, which uses the spline curve to model the trajectory of the robot and optimizes the spline curve through intelligent algorithms (such as ant colony algorithm, genetic algorithm, etc.) to find the optimal or near-optimal trajectory that meets various constraints to realize the intelligent path planning of the robot.

4.1 Bezier Curve Compensates the Motion Deviation of FSW Robot

An novel method for solid state welding of aluminum alloys utilizing robots is robotic friction stir welding (RFSW) [109]. The robot joints would experience elastic deformation of up to 10 kN during friction stir welding (FSW), which is a drawback of the method and can result in tool deviation and improper alignment [110]. Due to high forces, lateral tool deviations of several millimeters can be observed [111]. This results in the need to make necessary adjustments to the off-line planned paths to achieve high precision. Therefore, deflection compensation should be set up for the path planning of RFSW [112]. There are both online and offline deflection compensation strategies available. Online compensation technique is based on feedback control and uses sensors [113] or deflection models [114] for real-time compensation. It has been proposed that a path deviation compensation method for robot offline welding based on Bezier curve [108]. In this example, a Bezier curve is utilized to generate an optimized path with deflection using the y and z coordinate values taken from the CAD model. As shown in Figure 4, the effectiveness of Bezier curve in robot path planning is proved.

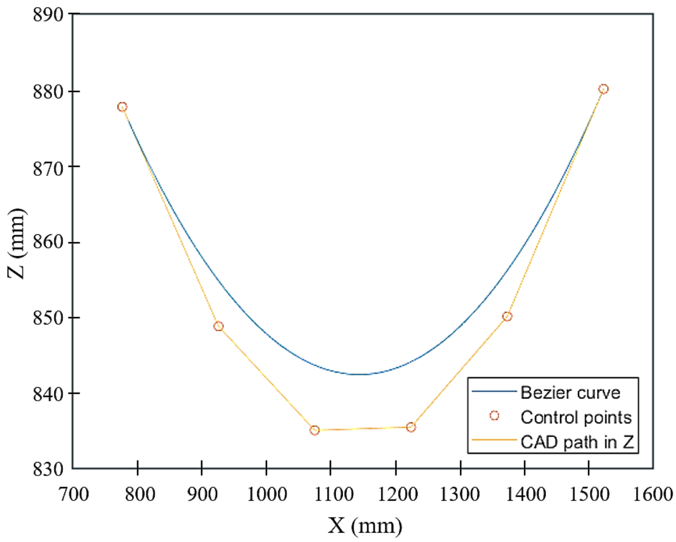

Figure 4. Path comparison in the z direction of the robot [53]

Off-line compensating techniques are based on deflection knowledge prior to welding. Typically, the tried-and-true approach is used. Before conducting the second experiment, the first must be completed, including any deviations being investigated and the data being integrated. This is a labor-intensive and time-consuming approach. Another offline feed-forward control method involves simulating the robot's deflection with a stiffness analysis, adding it to the desired path, and then finding the optimal path. These technologies are used in robot milling [115]. Cubic curve and quintic curve are used respectively to plan the optimal path between adjacent nodes with velocity, acceleration and pulsations constraints, and the two different methods are compared and analysed [116]. Comparison of experimental results between Bezier curve and robot master, please refer to [33].

Therefore, the optimal position and orientation of the welding tool in the sinusoidal welding route are offline generated using spline curve approximation techniques. A smooth path can be created because the spline curve function is infinitely differentiable. This welding path's low order approximation, which was obtained, is accurate enough. NURBS curves can be researched for other extremely complex pathways.

4.2 Application of the B-Spline Curve in the Directional Path Planning

A smooth path ensures that the motion follows a continuous curve with no breakpoints. This is one of the key points of path planning research [117]. Optimization techniques are proposed in [118] and [24] to allow smooth trajectories to be generated through the desired direction. A spline approach that roughly minimizes angular acceleration is given for creating direction trajectories in [34]. In [119], the curvature interpolation algorithm is also presented. Their applications are more potential due to their computational complexity and robustness. In other words, the approximated frame orientation or stated configuration interpolation smooths the angular velocity and its time derivative. This ensures that the robot's joint motor torque remains consistent. This method is the industry standard for constructing paths between two directions because it is simple to calculate. This example is the path planning of an industrial robot polishing a car fender. With the research goal of forming a smooth direction path planning. Car fender path generated by the path point, please refer to [120]. The experiment's parameters are computed using B-splines and are based on the exponential mapping of quaternions. Each B-spline results from a quaternion or quaternion derivative in the needed direction. From the experimental results, it can be seen that the B-spline path passes through each point accurately, please refer to [120]. The angular acceleration is significantly reduced, as shown in Figure 5.

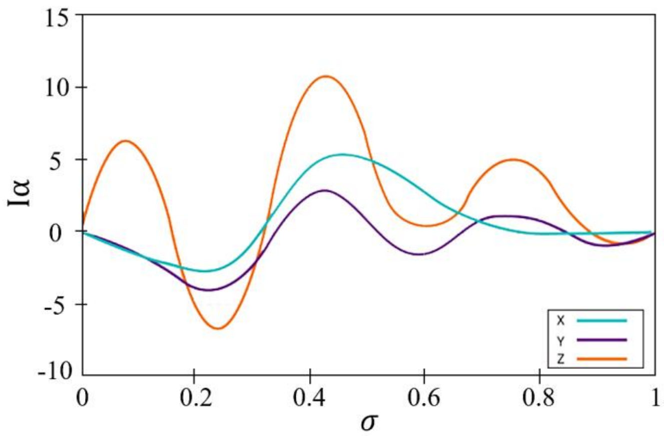

Figure 5. Angular acceleration of the planned path [120]

Other, the different approach is proposed in [10], using spherical linear interpolation（SLERP） in combination with mixing functions, such as Bezier curves or B-spline curves, for achieve precise initial and terminal directions (quaternions). In [121], an incremental path planning method based on SLERP is proposed. In this method, continuous tangents are obtained by calculating the intermediate points, but only a continuous curve of C1 is obtained. This is not accurate enough for robotic applications. Some researchers using cubic B-spline curve interpolation, planned the time-smoothness comprehensive optimal path suitable for welding or grasping robots [122]. A method using B-spline parameterized rotation to achieve smooth path direction planning has proposed [123]. Some researchers proposed a virtual node interpolation method in which B-spline curves were incorporated into the virtual node interpolation. This approach is particularly suitable for closed pathways to construct an ideal B-spline curve, which is utilized to generate smoother and shorter paths [124]. Each B-spline is computed in terms of the desired direction of the quaternion or quaternion derivative and is an optimized scheme. It can be seen that control points are computed by B-spline curves through a specified set of directions. This method is to control the smoothness of the path effectively.

4.3 Application of Intelligent Path Planning based on B-Spline Curve

With the development of artificial intelligence technology, the path planning of industrial robots will be more intelligent and autonomous, and can better adapt to complex working environments and task requirements [125]. It is a good idea to use intelligent control method to complete the path planning of welding robot to improve the working efficiency of welding robot and reduce the working time [126]. The path planning problem of welding robot can be regarded as a travel promotion problem, and each position node is regarded as a welding point. The traditional path optimization method of weld sequence selection based on experience has large time consumption and limited accuracy. In this example, the Ant Colony Algorithm (ACO) is used for parameter optimization, which improves the global search performance of the algorithm [127]. Then the B-spline curve interpolation method is combined with genetic algorithm to carry out segmentation optimization, which greatly reduces the time of path planning.

Under the premise of determining the welding points, when the robot traverses all the welding points, the ant colony algorithm proposes the correct path within the constraints according to the principle of the shortest foraging behaviour of the biological ant colony [128]. The path selection strategy of the initial ant colony algorithm has limitations. Therefore, in the iterative calculation process of the algorithm, the path selection of ant colony tends to be the path with higher pheromone concentration. In order to ensure the stability of the algorithm in complex computational convergence, this study combined the global pheromone with the path length model to calculate the pheromone increment. The probabilistic selection of iterative calculation is introduced into the interference factor to improve the global path optimization capability, as shown in Eq. (10) [129].

$$P_{ij}^{a}(t) = \begin{cases} \tau_{ij}(t) \times \left(\dfrac{1}{d_{ij}(t)}\right)^{\mu \times e^{\mu/t}} & , \mu \times e^{\mu/t} \geq P_0 \\ P_{ij}^{a}(t), & \mu \times e^{\mu/t} < P_0 \end{cases} \tag{10}$$

where, μ is the random interference scale factor, and its value range is [0,1], $\mu \times e^{\mu/t}$ is the inverse index value of the disturbance factor; $P_{ij}^{a}(t)$ is the likelihood that the $\tau_{ij}(t)$ ant t chooses j from the location to the place i at the moment, and t denotes the information concentration.

After solving the problem of shortest path, next solve the problem of shortest time. First, according to the basis function of cubic B-spline curve, as shown in Eq. (7). The overall equation table of B-spline curve is derived, as shown in Eq. (11) [130].

$$P(t) = \sum_{i=0}^{k} V_i N_{i,k}(u) \tag{11}$$

where, V_i is the arrangement number of the curve's vertices, i represents the position point number, k is dimensions of a spline curve, and u value interval is [0, 1].

The fixed point trajectory of welding robot is planned by cubic B-spline curve. The time optimization function of path planning is determined as the objective function of genetic algorithm, and the objective function is shown in Eq. (12) [107]. n is the welding point, so the trajectory of the welding robot is divided into $n-1$ time periods, and the time periods are represented as T_i.

$$T_{min} = \min\ [T_1 + T_1 + \cdots + T_{i-1}] = \min \sum_{i=1}^{n-1} T_i \tag{12}$$

where, n is the welding point, hence the trajectory of the welding robot is divided into $n-1$ time periods, and the time periods are represented as T_i.

In order to optimize the output of fixed-point path in the shortest time, the fitness function of genetic algorithm and cross mutation operator are used to optimize the trajectory planning of cubic spline interpolation. A penalty function is introduced into the trajectory planning of the total B-spline curve equation to determine the fitness function of the genetic algorithm, as shown in Eq. (13) [127].

$$F(t) = \frac{1}{T_{min} + \sigma \bar{P}(t)} \tag{13}$$

where, σ is the punishment factor, $\bar{P}(t)$ is the penalty item. For the specific mathematical expression of the penalty item, as shown in Eq. (14) [127].

$$\bar{P}(t) = \sum_{m=1}^{G} |\max\,(\theta_m^{max}, \theta_m(t))| + \sum_{m=1}^{G} |\max\,(\omega_m^{max}, \omega_m(t))| + \sum_{m=1}^{G} |\max\,(\rho_m^{max}, \rho_m(t))| \tag{14}$$

The experiment compared the joint angle and angular velocity of a robot using both the optimized joint algorithm and the traditional B-spline curve difference method. Figure 6 illustrates a broken line trend, indicating that the joint angle and angular velocity under the joint algorithm are consistently lower than those achieved with the B-spline curve difference method. It is observed that the maximum angular difference of joint angle is 3.7 radians, and the maximum angular difference of angular velocity is 2.2 radians. These results demonstrate that the optimized scheduling provided by the joint algorithm effectively reduces joint vibration amplitudes and stabilizes starting and stopping speeds during the task. Consequently, the optimization algorithm proves to enhance the welding quality of the robot while simultaneously reducing its energy consumption.

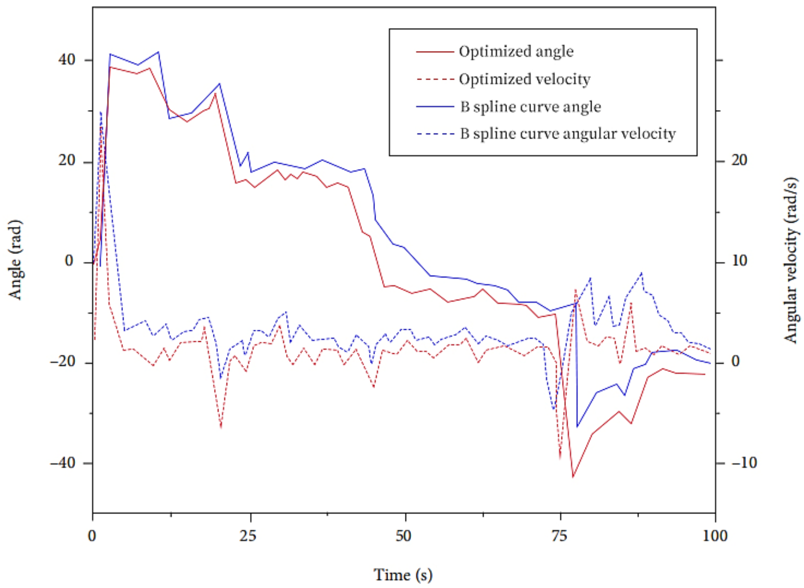

Figure 6. Comparison of robot performance between single B spline curve method and optimized joint algorithm [127]

In the realm of industrial robot path planning, the combined utilization of intelligent algorithms and spline curves not only facilitates the determination of the most efficient welding path but also empowers the robot to navigate through intricate and challenging working environments and constraints with agility. This integrated approach serves to significantly improve overall work efficiency in welding operations.

5. CONCLUSIONS

Through a comprehensive evaluation, it is found that the use of spline curves in welding robot path planning offers advantages over traditional methods. The continuous and flexible representation of the path results in smoother welding transitions, ultimately improving welding quality and productivity. Previous research and case studies have shown the success of spline curve welding path planning in various industrial settings. Particularly, in the automotive industry, spline curves play a crucial role in design, manufacturing, and repair processes. Spline curves accurately replicate the design curves of car bodies, ensuring the continuity and smoothness of welding processes. Intelligent algorithms play a key role in welding robot path planning by optimizing path planning and control strategies through extensive data preprocessing and prediction. These algorithms enable efficient searches for optimal solutions, particularly in finding spline curves that meet specified constraints in curve fitting and path planning. Future research will focus on four main aspects.

i) The research concentrates on optimizing B-spline curves by exploring techniques specifically tailored for control points and weights. The study delves into the integration of B-spline curves with diverse global fitting techniques, ensuring a comprehensive approach. Additionally, the application of B-spline curves under various path planning strategies is examined, accounting for dynamic environments and collision avoidance to enhance their practical utility.

ii) The primary objective is to seamlessly integrate B-spline technology with human-computer interaction. This integration aims to create a more intuitive and efficient path planning process, specifically designed for industrial welding robots. By prioritizing the user experience through HCI, the research aims to enhance the overall functionality of path planning systems.

iii) Intelligent algorithms remain a central area of exploration, with a continuous investigation into different hybrid algorithms. The emphasis is on addressing practical path planning problems and facilitating cooperative path planning between robots. The research also prioritizes the application of intelligent algorithms for path optimization,
particularly in high-dimensional complex environments, ensuring adaptability and efficiency.

iv) The overarching development direction emphasizes the integration of both spline curves and intelligent algorithms for path planning. This fusion aims to enhance the overall efficiency and adaptability of industrial welding robot systems. By laying a solid foundation for further advancements in the field, the integration seeks to provide a holistic and synergistic approach to path planning improvements.

6. ACKNOWLEDGMENTS

The project is funded by SEGi University, under the SEGi Internal Research Fund, Grant No.: SEGiUni/VC/RIMC/11-4-2019-24/95.

7. REFERENCES

[1] E. J. Lima, A. Q. Bracarense, “Robotic shielded metal arc welding,” *Welding Journal*, vol. 89, no. 8, pp. 166–171, 2010.

[2] T. S. Hong, M. Ghobakhloo, W. Khaksar, “Robotic welding technology,” *Comprehensive Materials Processing*, vol. 6, pp. 77–97, 2014.

[3] X. Wang, X. Zhou, Z. Xia, X. Gu, “A survey of welding robot intelligent path optimization,” *Journal of Manufacturing Processes*, vol. 63, no. 4, pp. 14–23, 2021.

[4] L. Yang, Y. Liu, J. Peng, “Advances techniques of the structured light sensing in intelligent welding robots: A review,” *International Journal of Advanced Manufacturing Technology*, vol. 110, no. 3–4, pp. 1027–1046, 2020.

[5] Asefa, Robsan Abebe, Liu, C. Hua, “Study of simulation technology for gas metal arc welding robot using robot studio,” *International Journal of Engineering Research and Technology*, vol. 9, no. 4, pp. 249–253, 2020.

[6] X. Wang, Y. Yan, X. Gu, “Spot welding robot path planning using intelligent algorithm,” *Journal of Manufacturing Processes*, vol. 42, no. 1, pp. 1–10, 2019.

[7] L. Wang, Q. Wu, F. Lin, S. Li, D. Chen, “A new trajectory-planning beetle swarm optimization algorithm for trajectory planning of robot manipulators,” *IEEE Access*, vol. 7, pp. 154331–154345, 2019.

[8] Y. K. Liu, Y. M. Zhang, “Supervised learning of human welder behaviors for intelligent robotic welding,” *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 3, pp. 1532–1541, 2017.

[9] J. Ogbemhe, K. Mpofu, “Towards achieving a fully intelligent robotic arc welding: A review,” *Industrial Robot*, vol. 42, no. 5, pp. 475–484, 2015.

[10] M. J. Kim, M. S. Kim, S. Y. Shin, “A general construction scheme for unit quaternion curves with simple high order derivatives,” in *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 369–376, 1995.

[11] R. Saravanan, S. Ramabalan, P. Sriram, C. Balamurugan, “Non-uniform rational B-spline-based minimum cost trajectory planning for autonomous robots,” *International Journal of Intelligent Systems Technologies and Applications*, vol. 9, no. 2, pp. 121–149, 2010.

[12] Yu, Xi, W. Zhu, L. Xu, “Real-time motion planning and trajectory tracking in complex environments based on Bézier curves and nonlinear MPC controller,” in *Chinese Control And Decision Conference (CCDC)*, pp. 1540–1546, 2020.

[13] M. Oberherber, H. Gattringer, A. Müller, “Successive dynamic programming and subsequent spline optimization for smooth time optimal robot path tracking,” *Mechanical Sciences*, vol. 6, no. 2, pp. 245–254, 2015.

[14] S. M. Manladan, F. Yusof, S. Ramesh, M. Fadzil, Z. Luo, S. Ao, “A review on resistance spot welding of aluminum alloys,” *International Journal of Advanced Manufacturing Technology*, vol. 90, no. 1–4, pp. 605–634, 2017.

[15] J. Xu, G. Zhang, Z. Hou, J. Wang, J. Liang, Z. Bao, et al., “Advances in multi-robotic welding techniques: A review,” *International Journal of Mechanical Engineering and Robotics Research*, vol. 9, no. 3, pp. 421–428, 2020.

[16] A. Billard, D. Kragic, “Trends and challenges in robot manipulation,” *Science*, vol. 364, no. 6446. pp. 72–86, 2019.

[17] F. De Tisi, M. Rossini, “Behavior of the beta-splines with values of the parameters beta2 negative,” *Computer Aided Geometric Design*, vol. 9, no. 6, pp. 419–423, 1992.

[18] H. Chen, F. Thomas, L. Xiongzi, “Automated industrial robot path planning for spray painting a process: A review,” in *IEEE International Conference on Automation Science and Engineering*, pp. 522–527, 2008.

[19] H. Bruyninckx, “Splines for "Optimal " nonholonomic robot motion planning,” *Computer-Aided Design*, pp. 1–10, 1997.

[20] E. W. Weisstein, “Lagrange Interpolating Polynomial,” Wolfram MathWorld [Online]. Available: http://mathworld.wolfram.com/LagrangeInterpolating.

[21] U. Y. Huh, S. R. Chang, “A G2 continuous path-smoothing algorithm using modified quadratic polynomial interpolation,” *International Journal of Advanced Robotic Systems*, vol. 11, no. 1, pp. 1–11, 2014.

[22] J. F. Epperson, “On the runge example,” *The American Mathematical Monthly*, vol. 94, no. 4, pp. 329–341, 1987.

[23] S. R. Chang, U. Y. Huh, “A collision-free G2 continuous path-smoothing algorithm using quadratic polynomial interpolation,” *International Journal of Advanced Robotic Systems*, vol. 11, pp. 1–10, 2014.

[24] J. Tan, Y. Xing, W. Fan, P. Hong, “Smooth orientation interpolation using parametric quintic-polynomial-based quaternion spline curve,” *Journal of Computational and Applied Mathematics*, vol. 329, pp. 256–267, 2018.

[25] G. Walberg, “Cubic spline interpolation: A review," *Technical Report*, no. CUCS-389-88, pp. 1-14, 1988.

[26] L. László, "Cubic spline interpolation with quasiminimal B-spline coefficients," *Acta Mathematica Hungarica*, vol. 107, no. 1–2, pp. 77–87, 2005.

[27] B. Parmentier, B. J. McGill, A. M. Wilson, J. Regetz, W. Jetz, R. Guralnick, et al., "Using multi-timescale methods and satellite-derived land surface temperature for the interpolation of daily maximum air temperature in Oregon," *International Journal of Climatology*, vol. 35, no. 13, pp. 3862–3878, 2015.

[28] A. Ravankar, A. A. Ravankar, Y. Kobayashi, Y. Hoshino, C.-C. Peng, "Path smoothing techniques in robot navigation: state-of-the-art, current and future challenges," *Sensors*, vol. 18, pp. 10–30, 2018.

[29] J. Lian, W. Yu, K. Xiao, W. Liu, "Cubic spline interpolation-based robot path planning using a chaotic adaptive particle swarm optimization algorithm," *Mathematical Problems in Engineering*, vol. 2020, p. 1849240, 2020.

[30] G. Farin, "Class a Bézier curves," *Computer Aided Geometric Design*, vol. 23, no. 7, pp. 573–581, 2006.

[31] Dai, Zhiwu, X. Sheng, J. Hu, H. Wang, D. Zhang, "Design and implementation of Bèzier curve trajectory planning in DELTA parallel robots," in *Intelligent Robotics and Applications: 8th International Conference*, pp. 420–430, 2015.

[32] C. Wang, X. Wang, H. Zheng, "Trajectory planning for a welding robot based on the Bezier curve," *International Journal of Control and Automation*, vol. 8, no. 2, pp. 351–362, 2015.

[33] I. Škrjanc, G. Klančar, "Optimal cooperative collision avoidance between multiple robots based on Bernstein-Bézier curves," *Robotics and Autonomous Systems*, vol. 58, no. 1, pp. 1–9, 2010.

[34] I. G. Kang, F. C. Park, "Cubic spline algorithms for orientation interpolation," *International Journal for Numerical Methods in Engineering*, vol. 46, no. 1, pp. 45–64, 1999.

[35] H. Prautzsch, W. Boehm, M. Paluszny. Bézier and B-spline Techniques, 1st ed. Berlin: Springer, 2002.

[36] Y. L. Kuo, C. C. Lin, Z. T. Lin, "Dual-optimization trajectory planning based on parametric curves for a robot manipulator," *International Journal of Advanced Robotic Systems*, vol. 17, no. 3, pp. 1–14, 2020.

[37] Y. Chen, "Solving robot trajectory planning problems with uniform cubic B-splines," *Optimal control applications and methods*, vol. 12, pp. 247–262, 1992.

[38] Y. Li, T. Huang, D. G. Chetwynd, "An approach for smooth trajectory planning of high-speed pick-and-place parallel robots using quintic B-splines," *Mechanism and Machine Theory*, vol. 126, pp. 479–490, 2018.

[39] H. Cao, M. Zoldy, "Implementing B-spline path planning method based on roundabout geometry elements," *IEEE Access*, vol. 10, pp. 81434–81446, 2022.

[40] T. M. Lehmann, C. Gönner, K. Spitzer, "Addendum: B-spline interpolation in medical image processing," *IEEE Transactions on Medical Imaging*, vol. 20, no. 7, pp. 660–665, 2001.

[41] Z. Wu, W. Su, J. Li, "Multi-robot path planning based on improved artificial potential field and B-spline curve optimization," in *Chinese Control Conference (CCC)*, pp. 4691–4696, 2019.

[42] Z. Wang, X. Xiang, J. Yang, S. Yang, "Composite Astar and B-spline algorithm for path planning of autonomous underwater vehicle," in *IEEE 7th International Conference on Underwater System Technology*, pp. 1–6, 2017.

[43] T. Arney, "Dynamic path planning and execution using B-splines," in *Third International Conference on Information and Automation for Sustainability*, pp. 1–6, 2007.

[44] M. Fey, J. E. Lenssen, F. Weichert, H. Muller, "Splinecnn: Fast geometric deep learning with continuous b-spline kernels," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 869–877, 2018.

[45] L. Piegl, "On NURBS: A survey," *IEEE Computer Graphics and Applications*, vol. 11, no. 1, pp. 55–71, 1991.

[46] Wu, Guanglei, W. Zhao, X. Zhang, "Optimum time-energy-jerk trajectory planning for serial robotic manipulators by reparameterized quintic NURBS curves," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 235, no. 19, pp. 4382–4393, 2021.

[47] R. Sevilla, S. Fernández-Méndez, A. Huerta, "NURBS-enhanced finite element method (NEFEM) Ruben," *International Journal for Numerical Methods in Engineering*, vol. 76, no. 1, pp. 56–83, 2008.

[48] X. Guan, J. Wang, "Trajectory planning theory and method of industrial robot," in *3rd International Conference on Computer Research and Development*, pp. 340–343, 2011.

[49] Q. Li, W. Wu, J. Xiang, H. Li, C. Wu, "A hybrid robot for friction stir welding," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 229, no. 14, pp. 2639–2650, 2015.

[50] Z. Shen, Y. Ding, A. P. Gerlich, "Advances in friction stir spot welding," *Critical Reviews in Solid State and Materials Sciences*, vol. 45, no. 6, pp. 457–534, 2020.

[51] J. Li, C. C. Moul, W. Zhang, "Hoping grey goes green: air pollution's impact on consumer automobile choices," *Marketing Letters*, vol. 28, no. 2, pp. 267–279, 2017.

[52] D. Mcguire. Computational Biology Modeling and Analysis, 1st ed. New York, USA: Callisto Reference, 2021.

[53] K. Kolegain, F. Leonard, S. Chevret, A. Ben Attar, G. Abba, "Off-line path programming for three-dimensional robotic friction stir welding based on Bézier curves," *Industrial Robot*, vol. 45, no. 5, pp. 669–678, 2018.

[54] D. Meike, "Industrial robot path optimization approach with asynchronous FLY-BY in joint space," in *IEEE International Symposium on Industrial Electronics*, pp. 911–915, 2011.

[55] D. Ioan, S. Olaru, I. Prodan, F. Stoican, S. I. Niculescu, "From obstacle-based space partitioning to corridors and path planning. A convex lifting approach," *IEEE Control Systems Letters*, vol. 4, no. 1, pp. 79–84, 2020.

[56] S. Riazi, O. Wigstrom, K. Bengtsson, B. Lennartson, "Energy and peak power optimization of time-bounded robot trajectories," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 646–657, 2017.

[57] N. Palmieri, X. S. Yang, F. De Rango, S. Marano, "Comparison of bio-inspired algorithms applied to the coordination of mobile robots considering the energy consumption," *Neural Computing and Applications*, vol. 31, no. 1, pp. 263–286, 2019.

[58] D. N. Mudaliar, N. K. Modi, "Unraveling travelling salesman problem by genetic algorithm using m-crossover operator," in *International Conference on Signal Processing, Image Processing & Pattern Recognition*, pp. 127–130, 2013.

[59] C. Duan, S. Luo, "Research on welding robot path planning based on genetic algorithm," in *IEEE International Conference on Control, Electronics and Computer Technology*, pp. 1401–1407, 2023.

[60] M. H. S. Helal, C. T. Fan, D. Y. Liu, S. M. Yuan, "Peer-to-Peer based parallel genetic algorithm," in *International Conference on Information, Communication and Engineering (ICICE)*, pp. 535–538, 2018.

[61] Z. Michalewicz, "Evolutionary algorithms for constrained parameter optimization problems," *Evolutionary Computation*, vol. 4, no. 1, pp. 1–32, 1996.

[62] J. J. Kim, , S. Member, J. Lee, "Trajectory optimization with particle swarm optimization for manipulator motion planning," *IEEE transactions on industrial informatics*, vol. 11, no. 3, pp. 620–631, 2015.

[63] X. Wang, Y. Hua, J. Gao, Z. Lin, R. Yu, "Digital twin implementation of autonomous planning arc welding robot system," *Complex System Modeling and Simulation*, vol. 3, no. 3, pp. 236–251, 2023.

[64] Y. Zhang, "Research on robot path planning based on improved genetic algorithm," in *14th International Conference on Advanced Computational Intelligence (ICACI)*, pp. 273–277, 2022.

[65] M. Pant, H. Zaheer, L. Garcia-hernandez, A. Abraham, "Differential evolution: A review of more than two decades of research," *Engineering Applications of Artificial Intelligence*, vol. 90, no. 1, pp. 1–24, 2020.

[66] K. R. Opara, J. Arabas, "Differential Evolution: A survey of theoretical analyses," *Swarm and Evolutionary Computation*, vol. 44, pp. 546–558, 2018.

[67] F. Neri, C. Cotta, "Memetic algorithms and memetic computing optimization: A literature review," *Swarm and Evolutionary Computation*, vol. 2, pp. 1–14, 2012.

[68] Q. H. Nguyen, Y. S. Ong, N. Krasnogor, "A study on the design issues of memetic algorithm," *IEEE Congress on Evolutionary Computation*, pp. 2390–2397, 2007.

[69] H. Zhao, B. Zhang, J. Sun, L. Yang, H. Yu, "Spot-welding path planning method for the curved surface workpiece of body-in-white based on a memetic algorithm," *The International Journal of Advanced Manufacturing Technology*, vol. 117, no. 9–10, pp. 3083–3100, 2021.

[70] F. Wang, Y. Li, A. Zhou, K. Tang, "An estimation of distribution algorithm for mixed-variable newsvendor problems," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 3, pp. 479–493, 2020.

[71] M. Hauschild, M. Pelikan, "An introduction and survey of estimation of distribution algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 3, pp. 111–128, 2011.

[72] R. Liu, S. Member, Z. Chen, "Intelligent path planning for AUVs in dynamic environments: An EDA-based learning fixed height histogram approach," *IEEE Access*, vol. 7, pp. 185433–185446, 2019.

[73] T. Mantere, J. Koljonen, "Solving and analyzing Sudokus with cultural algorithms," in *2008 IEEE Congress on Evolutionary Computation*, 2008, pp. 4053–4060.

[74] T. Palazzolo, C. Zhang, S. Saad, R. G. Reynolds, "Learning the impact of group structure on optimal herd path planning with cultural algorithms," in *2022 IEEE Congress on Evolutionary Computation (CEC)*, 2022, pp. 1–9.

[75] C. Liu, G. Cao, Y. Qu, Y. Cheng, "An improved PSO algorithm for time-optimal trajectory planning of Delta robot in intelligent packaging," *The International Journal of Advanced Manufacturing Technology*, vol. 107, pp. 1091–1099, 2019.

[76] X. Lv, Z. Yu, M. Liu, G. Zhang, L. Zhang, "Direct trajectory planning method based on IEPSO and fuzzy rewards and punishment theory for multi-degree of freedom manipulators," *IEEE Access*, vol. 7, no. 1, pp. 20452–20461, 2019.

[77] Y. Wang, J. Ma, Z. Zhang, H. Zhao, J. Teng, Y. He, et al., “Harvesting sequence planning of a harvesting robot based on the principle of minimum energy consumption,” *The Journal of Engineering*, vol. 16, pp. 1447–1456, 2018.

[78] H. Hizarci, S. Ikizoglu, “Position control of flexible manipulator using PSO-tuned PID controller,” in *Innovations in Intelligent Systems and Applications Conference*, 2019, pp. 1–5.

[79] Y. Tong, Z. Meng, L. I. Jingwei, L. I. Dongbo, W. Yulin, “Research on intelligent welding robot path optimization based on GA and PSO algorithms,” *IEEE Access*, vol. 6, pp. 65397–65404, 2018.

[80] X. Ye, L. Luo, L. Hou, Y. Duan, Y. Wu, “Laser ablation manipulator coverage path planning method based on an improved ant colony algorithm,” *Applied Sciences (Switzerland)*, vol. 10, no. 23, pp. 1–19, 2020.

[81] X. Wu, Z. Xie, “Forward kinematics analysis of a novel 3-DOF parallel manipulator,” *Scientia Iranica*, vol. 26, no. 1B, pp. 346–357, 2019.

[82] H. Yang, J. Qi, Y. Miao, H. Sun, J. Li, “A new robot navigation algorithm based on a double-layer ant algorithm and trajectory optimization,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8557–8566, 2018.

[83] J. C. Bansal, H. Sharma, S. S. Jadon, “Artificial bee colony algorithm : A survey,” *International Journal of Advanced Intelligence Paradigms*, vol. 5, no. 1/2, pp. 123–159, 2016.

[84] P. Savsani, R. L. Jhala, V. J. Savsani, “Optimized trajectory planning of a robotic arm using teaching learning based optimization (TLBO) and artificial bee colony (ABC) optimization techniques,” in *IEEE International Systems Conference(SysCon)*, pp. 381–386, 2013.

[85] M. Mareli, B. Twala, “An adaptive Cuckoo search algorithm for optimisation,” *Applied Computing and Informatics*, vol. 14, no. 2, pp. 107–115, 2018.

[86] G. Wang, L. Guo, H. Duan, H. Wang, L. Liu, M. Shao, “A hybrid metaheuristic DE/CS algorithm for UCAV three-dimension path planning,” *The Scientific World Journal*, vol. 2012, pp. 1–12, 2012.

[87] C. Xie, H. Zheng, “Application of improved cuckoo search algorithm to path planning unmanned aerial vehicle,” in *Intelligent Computing Theories and Application: 12th International Conference*, pp. 722–729, 2016.

[88] Y. U. Xiu-wu, Y. U. Hao, L. Yong, X. Ren-rong, “A clustering routing algorithm based on wolf pack algorithm for heterogeneous wireless sensor networks,” *Computer Networks*, vol. 167, pp. 1–10, 2020.

[89] R. Kishan, D. Anupam, S. W. Wilfred, “Three dimensional path planning using Grey wolf optimizer for UAVs,” *Applied Intelligence*, vol. 49, no. 6, pp. 2201–2217, 2019.

[90] F. Ge, K. Li, W. Xu, Y. Wang, “Path planning of UAV for oilfield inspection based on improved grey wolf optimization algorithm,” in *Chinese Control and Decision Conference (CCDC)*, pp. 3666–3671, 2019.

[91] A. Kamalova, K. D. Kim, S. G. Lee, “Waypoint mobile robot exploration based on biologically inspired algorithms,” *IEEE Access*, vol. 8, pp. 190342–190355, 2020.

[92] S. Kirkpatrick, C. D. Gelatt, P. M.Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[93] H. Miao, Y. Tian, “Dynamic robot path planning using an enhanced simulated annealing approach,” *Applied Mathematics and Computation*, vol. 222, pp. 420–437, 2013.

[94] B. A. Prakash, H. Tong, N. Valler, M. Faloutsos, C. Faloutsos, “Virus propagation on time-varying networks: Theory and immunization algorithms,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 99–114, 2012.

[95] S. Ren, “Underwater robot path planning methods based on immunization genetic algorithms,” in *2nd International Conference on Information Engineering and Computer Science*, pp. 1–4, 2010.

[96] D. M. Jaeggi, G. T. Parks, T. Kipouros, P. J. Clarkson, “The development of a multi-objective Tabu Search algorithm for continuous optimisation problems,” *European Journal of Operational Research*, vol. 185, no. 3, pp. 1192–1212, 2008.

[97] W. Khaksar, T. S. Hong, M. Khaksar, O. R. E. Motlagh, “Sampling-based tabu search approach for online path planning,” *Advanced Robotics*, vol. 26, no. 8–9, pp. 1013–1034, 2012.

[98] M. T. Lee, B. Y. Chen, Y. C. Lai, “A hybrid tabu search and 2-OPT path programming for mission route planning of multiple robots under range limitations,” *Electronics (Switzerland)*, vol. 9, no. 3, p. 534, 2020.

[99] C. Rim, S. Piao, G. Li, U. Pak, “A niching chaos optimization algorithm for multimodal optimization,” *Soft Computing*, vol. 22, no. 2, pp. 621–633, 2018.

[100] Q. Zhang, M. Yuan, R. Song, “Robot trajectory planning method based on genetic chaos optimization algorithm,” in *18th International Conference on Advanced Robotics (ICAR)*, pp. 602–607, 2017.

[101] P. R. Giri, V. E. Korepin, “A review on quantum search algorithms,” *Quantum Information Processing*, vol. 16, no. 12, pp. 1–36, 2017.

[102] L. Gao. R. Liu, F. Wang, W. Wu, B. Bai, S. Yang, L. Yao, “An advanced quantum optimization algorithm for robot path planning,” *Journal of Circuits, Systems and Computers*, vol. 29, no. 8, pp. 1–8, 2020.

[103] L. Yang, J. Qi, D. Song, J. Xiao, J. Han, Y. Xia, “Survey of robot 3D path planning algorithms,” *Journal of Control Science and Engineering*, vol. 2026, p. 7426913, 2016.

[104] P. Raja, “Optimal path planning of mobile robots: A review,” *International Journal of the Physical Sciences*, vol. 7, no. 9, pp. 1314–1320, 2012.

[105] M. N. Zafar, J. C. Mohanta, “Methodology for path planning and optimization of mobile robots: A review,” *Procedia Computer Science*, vol. 133, pp. 141–152, 2018.

[106] Z. H. Liu Weirui, “Application of improved particle swarm optimization algorithm to forward position solution of parallel mechanism,” *Machinery Design & Manufacture*, vol. 22, pp. 181–183, 2014.

[107] S. Katoch, S. S. Chauhan, V. Kumar, “A review on genetic algorithm: Past , present , and future,” *Multimedia Tools and Applications*, vol. 80, pp. 8091–8126, 2020.

[108] K. Kolegain, F. L´eonard, S. Chevret, A. Ben Attar, G. Abba, “Robotic friction stir welding path planning with deflection compensation using B-splines,” in *ROMANSY 22–Robot Design, Dynamics and Control: Proceedings of the 22nd CISM IFToMM Symposium*, pp. 264–271, 2019.

[109] B. T. Gibson, D. H. Lammlein, T. J. Prater, W. R. Longhurst, C. D. Cox, M. C. Ballun, et al., “Friction stir welding: process, automation, and control,” *Journal of Manufacturing Processes*, vol. 16, pp. 56–73, 2014.

[110] R. S. Mishra, Z. Y. Ma, “Friction stir welding and processing,” *Materials science and engineering: R: Reports*, vol. 50, no. 1–2, pp. 1–78, 2005.

[111] F. Leali, F. Pini, M. Ansaloni, “Integration of CAM off-line programming in robot high-accuracy machining,” in *Proceedings of the IEEE/SICE International Symposium on System Integration*, pp. 580–585, 2013.

[112] A. Silva-Magalhães, L. Cederqvist, J. De Backer, E. Håkansson, B. Ossiansson, G. Bolmsjö, “A friction stir welding case study using emperature controlled robotics with a HPDC cylinder block and dissimilar materials joining,” *Journal of Manufacturing Processes*, vol. 46, no. 8, pp. 177–184, 2019.

[113] P. Kah, M. Shrestha, E. Hiltunen, J. Martikainen, “Robotic arc welding sensors and programming in industrial applications,” *International Journal of Mechanical and Materials Engineering*, vol. 10, no. 13, pp. 1-16, 2015.

[114] S. Briot, I. Bonev, “Pantopteron: A new fully decoupled 3DOF translational parallel robot for pick-and-place applications,” *Journal of Mechanisms and Robotics*, vol. 1, no. 2, pp. 1–9, 2009.

[115] Y. Li, Q. Xu, “Kinematic analysis of a 3-PRS parallel manipulator,” *Robotics and Computer-Integrated Manufacturing*, vol. 23, no. 4, pp. 395–408, 2007.

[116] A. Gasparetto, V. Zanotto, “Advances in engineering software optimal trajectory planning for industrial robots,” *Advances in Engineering Software*, vol. 41, no. 4, pp. 548–556, 2010.

[117] L. Liu, C. Chen, X. Zhao, Y. Li, “Smooth trajectory planning for a parallel manipulator with joint friction and jerk constraints,” *International Journal of Control, Automation and Systems*, vol. 14, no. 4, pp. 1–15, 2016.

[118] A. H. Barr, B. Currin, S. Gabriel, J. F. Hughes, “Smooth interpolation of orientations with angular velocity constraints using quaternions,” *ACM SIGGRAPH Computer Graphics*, vol. 26, no. 2, pp. 313–320, 1992.

[119] O. A. Bauchau, S. Han, “Interpolation of rotation and motion,” *Multibody System Dynamics*, vol. 31, no. 3, pp. 339–370, 2014.

[120] M. Neubauer, A. Muller, “Smooth orientation path planning with quaternions using B-splines,” in *IEEE International Conference on Intelligent Robots and Systems*, pp. 2087–2092, 2015.

[121] E. B. Dam, M. Koch, M. Lillholm, "Quaternions, interpolation and animation," *Technical Report*, University of Copenhagen, no. DIKU-TR-98/5, 1998.

[122] I. Noreen, “Collision free smooth path for mobile robots in cluttered environment using an economical clamped cubic B-Spline,” *Symmetry*, vol. 12, no. 9, pp. 1–13, 2020.

[123] J. Mei, F. Zhang, J. Zang, Y. Zhao, H. Yan, “Trajectory optimization of the 6-degrees-of-freedom high-speed parallel robot based on B-spline curve,” *Science Progress*, vol. 103, no. 1, pp. 1–26, 2020.

[124] J. M. Escorcia-Hernandez, H. Aguilar-Sierra, O. Aguilar-Mejia, A. Chemori, J. H. Arroyo-Nunez, “An intelligent compensation through B-spline neural network for a delta parallel robot,” in *6th International Conference on Control, Decision and Information Technologies*, pp. 361–366, 2019.

[125] H. Zhang, W. Lin, A. Chen, “Path planning for the mobile robot : A review,” *Symmetry*, vol. 10, no. 10, pp. 1–17, 2018.

[126] L. Luo, T. Guo, K. Cui, Q. Zhang, “Trajectory planning in robot joint space based on improved quantum particle swarm optimization algorithm,” *Applied Sciences*, vol. 7031, no. 13, pp. 8–20, 2023.

[127] Y. Gao, Y. Zhang, “Path optimization of welding robot based on ant colony and genetic algorithm,” *Journal of Applied Mathematics*, vol. 2022, pp. 1–11, 2022.

[128] M. Mahi, Ö. K. Baykan, H. Kodaz, “A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem,” *Applied Soft Computing Journal*, vol. 30, pp. 484–490, 2015.

[129] D. Zhao, L. Liu, F. Yu, A. A. Heidari, M. Wang, H. Chen, et al., “Opposition-based ant colony optimization with all-dimension neighborhood search for engineering design,” *Journal of Computational Design and Engineering*, vol. 9, no. 3, pp. 1007–1044, 2022.

[130] J. Huang, P. Hu, K. Wu, M. Zeng, “Optimal time-jerk trajectory planning for industrial robots,” *Mechanism and Machine Theory*, vol. 121, pp. 530–544, 2018.