International Journal of Industrial Engineering & Production Research June 2025, Vol. 36, No. 2: 156-169
DOI: 10.22068/ijiepr.36.2.2274

RESEARCH PAPER

# Determining Optimal Piecewise Polynomial Coefficients for the Electronic Cam in the CNC Sanding Mechanism

**Davood Nazari Maryam Abadi[1,*], Mohammad Bagheri[2]**

Received 18 January 2025; Revised 25 March 2025; Accepted 20 April 2025;
© Iran University of Science and Technology 2025

**ABSTRACT**
*In this paper, an optimal Electrical Cam (Ecam) profile is obtained by identifying the best breakpoint positions for piecewise polynomials using the cubic spline interpolation method. To achieve a curve that best tracks the reference Ecam curve, the breakpoint positions are determined using particle swarm optimization with random inertia weight (RNW-PSO). The previous programmable logic controller (PLC) used in the sanding mechanism was the DELTA DVP40ES2, utilizing the Ecam capability of DELTA ASD-A2 servo motors. To implement the Ecam function independently of the servo motor type, it has been integrated into a PLC, specifically the SIEMENS SIMATIC CPU 1215C. The optimized Ecam curve is then applied to a computer numerical control (CNC) sanding machine. Practical results demonstrate the effectiveness of the proposed method, showing improved sanding quality and better compliance with the reference curve.*

**KEYWORDS:** *Cubic spline interpolation; Ecam; CNC sanding machine; RNW–PSO.*

## 1. Introduction

The cam-follower is an efficient mechanism for generating irregular motion and is widely used in various automated equipment, such as textile machinery, CNC sanding machines, and other automatic devices. In order to achieve precise motion of the follower through the rotating cam, its shape must be correctly designed. As a core motion control system, the cam-follower mechanism is extensively used in modern high-speed automatic machines, including internal combustion engines [1, 2], indexing mechanism [3] and CNC machining center [4]. It is widely recognized that the dynamic performance of the follower is largely determined by the cam's driving profile, which, in turn, depends on the mathematical characteristics of the input motion law (commonly referred to as the cam curve) [5- 7]. Therefore, the optimal synthesis of cam curves with desirable kinematic and dynamic characteristics is crucial for the development of high-speed precision cam mechanisms.

Given the limitations of basic cam curves, researchers have continuously worked on developing nonstandard cam curves that offer optimized kinematic and/or dynamic performance [8-10]. Generally, the development of nonstandard, case-specific cam curves involves numerical approximation, interpolation, or optimization processes based on control points, nodal constraints, basis functions, and objective functions [11]. With predefined control points, a smooth cam curve with the desired flexibility can be readily achieved using approximation algorithms. However, in many cases, the approximated cam curve may deviate from the user-specified motion constraints at critical nodes. To address this, interpolation methods are often preferred for constructing cam curves that faithfully satisfy the arbitrary nodal constraints required by the user.

Interpolation methods used for cam motion design generally fall into two categories: continuous interpolation and piecewise interpolation. Continuous interpolation uses a single curve to connect all given nodes, while piecewise interpolation uses curve segments (splines) to connect two or more adjacent nodes. When only a few nodal constraints are involved (typically motion parameters at both ends of the rising stage), polynomial-based continuous interpolation (PBCI) methods, such as classic Lagrange, Newton, Hermite, and Bezier polynomial interpolations, are suitable for constructing a closed-form cam curve with good smoothness and high flexibility [12- 14]. However, as the number of nodal constraints increases, the degree of these PBCI cam curves also increases, leading to unwanted

* *Corresponding author: Davood Nazari Maryam Abadi*
*d.nazari@grandmaghsoud.com*

1. *Department of Electrical Engineering of the University of Applied Science and Technology, Grand Maghsoud Group, Mashhad, Iran*
2. *Department of Electrical Engineering of the University of Applied Science and Technology, Grand Maghsoud Group, Mashhad, Iran*

divergence and vibration issues [15].

To address this limitation, cam curve synthesis methods based on piecewise interpolation have been developed and have gained increasing interest in recent years [16]. Polynomial splines are among the most widely used tools for synthesizing piecewise cam curves. For instance, Yoon and Rao [17] used cubic splines to construct an optimized cam profile with reduced peaks in acceleration and jerk, though the resulting jerk curve exhibited discontinuities. Similar jerk discontinuities were observed in the cam profiles synthesized by Kim et al. [18] and Nguyen and Kim [19], who also employed cubic or quintic splines. In such cases, polynomial splines of higher degrees should be preferred to achieve higher-order continuity [20].

The non-uniform rational B-spline (NURBS) family, which includes Bezier splines and rational B-splines, offers another powerful approach for cam curve synthesis [7, 15, 21]. The primary advantage of this method is that it avoids the local divergence issues that occur with continuous interpolation. As a result, piecewise cam curves can achieve excellent interpolation accuracy, even when satisfying multiple displacement constraints across numerous nodes (up to 1000) [7]. Additionally, the kinematic and dynamic performance of these piecewise cam curves can be enhanced by optimizing the control points of the B-splines, as demonstrated by Qiu et al. [15].

An Electronic cam (Ecam) is typically used for coordinated control in various applications. Ecam is a technique that electronically replicates nonlinear motion, similar to the functionality achieved with mechanical cams. It is widely known as a master-slave system, where one or more slave axes follow a predetermined trajectory based on the position of a master's axis. Ecam encompasses velocity tracking control for the master motor and trajectory generation for the slave motor. Today, Ecam systems, which replace mechanical cams, are applied in areas such as robotic material handling, and nonlinear machining. These systems offer the advantage of software flexibility, allowing users to quickly switch between cam profiles, while also eliminating mechanical issues like backlash, friction, engagement debouncing, and structural resonance. In [24], a design for implementing an Ecam in a 3D feeding manipulator has been proposed. The Ecam mechanism is realized using a Keynes KV-7500 CPU and a KV-XH16ML motion module. Additionally, a prototype of a high-speed rotary synchronous cutting system, utilizing the electronic cam feature of the Yaskawa MP-3000 Series motion controller with a Sigma-7 servo system, was studied in [25]. The use of servomotors in industrial control systems, particularly as an alternative to classical mechanical mechanisms through position and speed synchronization, is analyzed in [26]. Additionally, a comprehensive methodology is presented for determining the dynamic parameters of the adopted kinematic system using electronic motion profiles. In [27], B-spline interpolation is employed to generate the basic motion path of the lower hook plate. Subsequently, a self-adaptive mutation-based particle swarm optimization method is proposed and applied to determine the optimal parameters of the quintic polynomial.

Particle Swarm Optimization (PSO) has been widely adopted for solving complex optimization problems due to its simplicity and efficiency. However, traditional PSO and its various improved versions, such as Inertia Weight PSO (IW-PSO), Local Best PSO (LBest-PSO), and Constriction Factor PSO (CF-PSO), often suffer from issues such as premature convergence, poor exploration capabilities, and stagnation in local optima. To overcome these challenges, RNW-PSO has been introduced, demonstrating superior performance over other PSO variants. Leveraging the superiority of RNW-PSO, the exact positions of breakpoints in the cubic spline interpolation method are determined, ensuring the accurate execution of the Ecam profile. Given RNW's advantages over other PSO variants, the proposed approach more precisely computes the cubic spline interpolation coefficients.

Based on the literature review, the specific challenges of CNC sanding machines include generating an Ecam profile independent of the servo drive type and designing an Ecam profile that best conforms to the reference curve. Despite the significance of sanding mechanisms in the cutlery and tableware industries, there has been limited research on the Ecam curve in CNC sanding machines. Therefore, optimizing the Ecam curve in the sanding mechanism can greatly improve sanding quality.

The Ecam mechanism is primarily supported by servo drives like the DELTA ASD-A3 and SIEMENS SIMOTION. Each of these servo

drives comes with its own dedicated software, which must be used to make changes to the cam profile. However, since servo drives can receive external pulses, it is possible to implement the Ecam mechanism independently of the specific drive type by sending pulses that correspond to the Ecam curve. In this study, the SIEMENS SIMATIC CPU 1215C was employed to implement the Ecam mechanism. To determine the equation of the curve passing through the points of the Ecam curve, the piecewise cubic spline interpolation method was utilized. The function that calculates the coefficients for these equations was developed using SIEMENS TIA Portal V19 software.

Building on the research discussed earlier, this paper aims to present a method for the optimal design of an Ecam profile by selecting the best positions for breakpoints in piecewise polynomials, using the RNW-PSO algorithm. Implementation of optimization algorithm is performed by MATLAB R2022b software and the optimized breakpoints are used for creating the Ecam curve.

To predict the quality of the CNC machine, the effects of various machining input parameters—such as cutting speed, feed rate, depth of cut, and nose radius—on output parameters, including surface roughness and metal removal rate of C40 steel, are analyzed in [28] using an artificial neural network (ANN) approach. In comparison, to practically verify the efficiency of the proposed method for the CNC sanding mechanism of SAE 304 stainless steel—specifically in terms of compliance with the reference curve—the Renishaw Cyclone Digitizer machine has been used.

The key contributions to this paper are as follows:

1. The Ecam capability, which is exclusively supported by certain servo drives, is typically implemented using the SIEMENS SIMATIC CPU 1215C. The proposed method can be implemented on all servo drives capable of receiving external pulses, unlike [24-27], which relies on the specific Ecam capability of certain servo drives.
2. The exact locations of breakpoints in the cubic spline interpolation method are determined using RNW-PSO, ensuring the precise implementation of the Ecam profile. Given the advantages of RNW over other versions of PSO, the proposed method more accurately determines the coefficients of cubic spline interpolation. Compared to [27], which employs self-adaptive mutation-based particle swarm optimization, the proposed method achieves a closer match to the reference curve.
3. To check the quality of the CNC machine, the proposed Ecam design method is applied to a CNC sanding machine. Practical results demonstrate the superiority of the method in terms of improved sanding quality and its compliance with the reference curve.

The rest of the paper is organized as follows: section 2 describes the different parts of the CNC sanding machine. Section 3 explains the basics of the cubic spline interpolation method. In section 4, the optimization method using the RNW-PSO algorithm is explained. Simulation and practical results are presented in section 5 and finally the conclusion is expressed in section 6.

## 2. CNC Sanding Mechanism

The sanding mechanism is a crucial part of the cutlery and tableware production process. After the stainless steel sheet is punched by a press machine, the external edges of the punched item are rough and require sanding. Typically, sanding is done manually. However, designing a machine that automates the sanding process not only improves safety but also enhances quality and increases production output. The closer the designed curve matches the reference curve, the higher the sanding quality. Therefore, making the optimal design of the Ecam curve is essential to sanding performance. The CNC sanding machine and its schematic are shown in Fig. 1 and Fig. 3, respectively. The main components of the control system include the motor and inverter of the main table, servo motors connected to the sanders via ball screws, an encoder for the main table, an inductive sensor to detect items in the clamps, a human-machine interface (HMI), and a programmable logic controller (PLC). The main table has 10 clamps, each positioned 36 degrees apart. The combination of a servo motor, ball screw, and sander is referred to as a station. The item detection sensor is located at the zero-degree position of the table, while stations 1, 2, 3, and 4 are positioned at 36, 72, 96, and 108 degrees, respectively. The table rotates counterclockwise. The sanding mechanism is shown in Fig. 2. When an item is detected, its outer edge is sanded by each sander according to its Ecam curve. The first sanding is rough, and the

roughness progressively decreases with each subsequent sanding.

The optimized Ecam curves are stored in the HMI memory, allowing the operator to select curves for different items and modify the coordinates of the curve points if needed. After an item is selected, the curve coordinates are transmitted from the HMI to the PLC via the Modbus network platform, where the Ecam curve equations are calculated by executing the cubic spline interpolation function. The pulse received from the main table encoder serves as the input variable for the Ecam curve function, while the output variable is the pulse of the user unit (PUU). The PUU is a unit which is scaled by the electronic gearing of the servo drive.

**Fig.1. CNC sanding machine**

**Fig.2. Sanding mechanism**

Fig. 4 illustrates the Modbus RTU configuration between the servo drives and the PLC, the digital inputs and outputs used in the servo drives, and the motor encoder cable connecting each servo drive to its respective servo motor. The SIEMENS SIMATIC CPU 1215C, used in the sanding machine, features four high-speed counter outputs, which generate pulses applied to the pulse inputs of each servo drive in the stations. Also the position and speed of the main table are controlled via the analog output of the PLC module, which is fed into the analog input of the main table's servo drive.

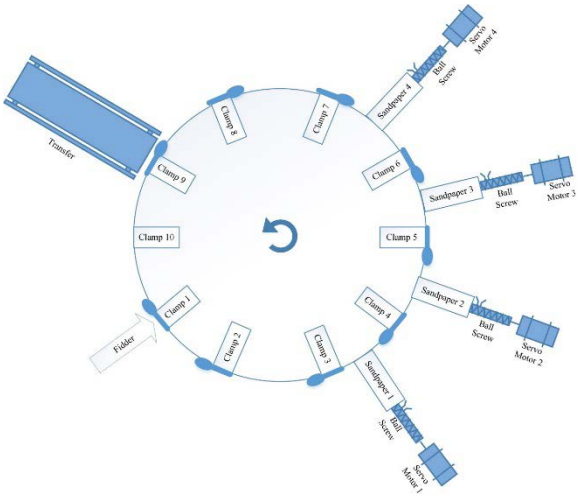



**Fig.3. Schematic of CNC sanding machine**

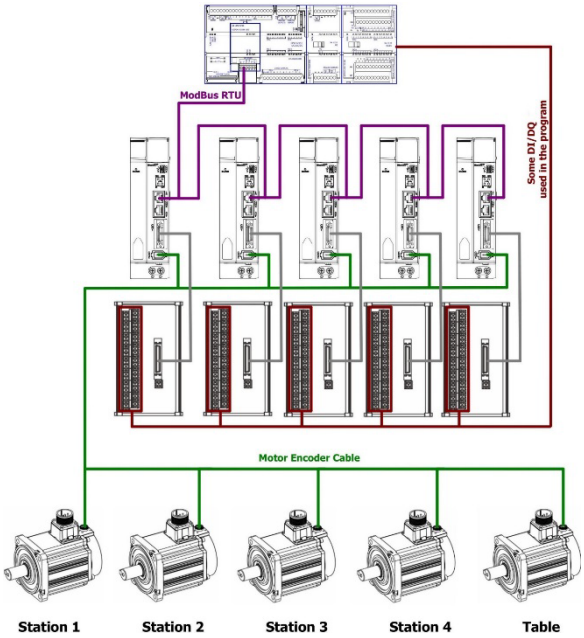



**Fig.4. Network configuration and digital input/output connections between the PLC and servo drives**

## 3. Piecewise Cubic Spline Interpolation

Piecewise polynomials are commonly used to define curves, as they are constructed from multiple segments. A key advantage of polynomials is their ability to maintain continuous derivatives of all

orders. Parametric polynomial fitting provides a global fit, where a single set of coefficients represents the entire dataset; however, this approach can sometimes result in erratic behavior. In contrast, piecewise polynomials offer smoother fits and greater flexibility, making them more effective across a broader range of datasets. Cubic spline interpolation, in particular, presents several advantages over other interpolation methods, especially regarding smoothness, accuracy, and computational efficiency:

- **Smoothness and continuity:** Cubic splines are piecewise polynomials that ensure continuity up to the second derivative. This results in a smooth interpolant without abrupt changes in slope or curvature, making it especially useful in applications requiring seamless transitions, such as computer graphics and data visualization. In contrast, linear interpolation offers only first-derivative continuity, leading to noticeable corners at data points.
- **Avoidance of Runge's phenomenon:** High-degree polynomial interpolation often suffers from oscillations between data points, known as Runge's phenomenon. Cubic spline interpolation mitigates this issue by employing low-degree (third-degree) polynomials over each interval, resulting in a more stable and accurate fit.
- **Computational efficiency:** Spline interpolation is computationally efficient, as it requires solving a tridiagonal system of equations—a process that can be completed in linear time relative to the number of data points. This efficiency makes it suitable for large datasets and real-time applications [29].
- **Reduced overshooting:** Compared to polynomial interpolation, cubic splines are less prone to overshooting between data points. This characteristic is crucial in applications where it is important to keep interpolated values within specific bounds, such as in physical simulations and engineering designs.

Cubic splines are a common example of piecewise polynomial applications in the approximation and interpolation of data sets. Each segment is described by four coefficients, calculated using a cubic (third-degree) polynomial. This approach fits a unique cubic polynomial between every pair of data points. The piecewise polynomial (PP) method is used to represent the displacement of a cam follower through a series of polynomial expressions, with each polynomial corresponding to a segment of the profile. The displacement generated by a cam profile with $n$ steps can be represented using a polynomial function for each of the non-dwelling segments in the following form:

$$y_i(\theta) = \sum_{j=1}^{3} b_{ij}(\theta - \beta_i)^{j-1}, \forall\theta \in \{\beta_i \le \theta \le \beta_{i+1}\}, i = 1, \cdots, n-1 \quad (1)$$

Where $\theta$ represents the cam angle, $b_{ij}$ are the displacement coefficients, $k$ is the polynomial order, $\beta_i$ is the i-th breakpoint in the profile (rad) and $n$ is the number of breakpoints in the profile.

Velocity and acceleration of the displacement functions can be expressed similarly as:

$$v_i(\theta) = \dot{y}_i(\theta) = \sum_{j=2}^{k} b_{ij}(j-1).(\theta - \beta_i)^{j-2} \quad \forall\theta \in \{\beta_i \le \theta \le \beta_{i+1}\}, i = 1, \cdots, n-1 \quad (2)$$

$$a_i(\theta) = \ddot{y}_i(\theta) = \sum_{j=3}^{k} b_{ij}(j-1).(j-2).(\theta - \beta_i)^{j-3} \quad \forall\theta \in \{\beta_i \le \theta \le \beta_{i+1}\}, i = 1, \cdots, n-1 \quad (3)$$

where $v_i(\theta)$ is the polynomial function describing velocity for the i-th segment, $a_i(\theta)$ is the polynomial function describing acceleration for the i-th segment.

All the above functions are derived by differentiating the displacement with respect to $\theta$, assuming a constant angular velocity ($\omega$).

$$v(t) = \frac{\partial y}{\partial t} = \frac{\partial y}{\partial \theta}.\frac{\partial \theta}{\partial t} = v(\theta).\omega \quad (4)$$

## 3.1. Boundary conditions

A designer can typically specify the desired displacement values at each breakpoint. In general, these boundary conditions can be expressed as the following sets of $(n-1)$ equations:

$$y_i(\beta_i) = Disp_i, \quad i = 1, \cdots, n-1 \quad (5)$$

$$v_i(\beta_i) = Vel_i, \quad i = 1, \cdots, n-1;$$
$$a_i(\beta_i) = Accel_i, \quad i = 1, \cdots, n-1; \quad (6)$$

where $Disp_i$, $Vel_i$ and $Accel_i$ are constants. Note that in (6), each segment function is used to define only one of the boundaries of the segment.

## 3.2. Continuity conditions

Additional equations are derived to ensure continuity of the displacement function and its derivatives. The condition of continuity throughout the cam profile can be expressed as a set of equations evaluating the polynomial motion functions at the boundary points. In the case of continuous displacement of the follower (which is necessary for the manufacture of the cam)

continuity will be given by the following $(n-1)$ equations:

$$\begin{cases} y_i(\beta_{i+1}) = y_{i+1}(\beta_{i+1}), & i = 1, \cdots, n-2 \\ y_{n-1}(\beta_1) = y_1(\beta_1). \end{cases} \quad (7)$$

This ensures continuity at the end of the profile, and similar sets of equations can be derived for each displacement derivative.

$$\begin{cases} v_i(\beta_{i+1}) = v_{i+1}(\beta_{i+1}), & i = 1, \cdots, n-2 \\ v_{n-1}(\beta_1) = v_1(\beta_1), \end{cases}$$

$$\begin{cases} \alpha_i(\beta_{i+1}) = \alpha_{i+1}(\beta_{i+1}), & i = 1, \cdots, n-2 \\ \alpha_{n-1}(\beta_1) = \alpha v_1(\beta_1), \end{cases} \quad (8)$$

The expressions in (5) provide a set of equations that, expressed as a linear system of equations, can be used to determine the polynomial coefficients $b_{ij}$ of the cam displacement profile.

The process of determining the appropriate polynomial order can be easily demonstrated with an example. Consider the Ecam curve corresponding to a spoon, which consists of 21 breakpoints. The coordinates of these breakpoints are shown in Eq. (9).

$$y(\beta_1) = 0, y(\beta_2) = 13.66, \cdots y(\beta_{20}) = 15.13$$
$$y(\beta_{21}) = 0 \quad (9)$$

Additionally, the designer specifies that the velocity and acceleration at all breakpoints must be zero.

$$v_i(\beta_i) = 0, \quad i = 1, \cdots, n-1;$$
$$a_i(\beta_i) = 0, \quad i = 1, \cdots, n-1 \quad (10)$$

Finally, it is inherently required that the displacement function of any cam profile remains continuous throughout, as discontinuities would make manufacturing the cam impossible.

Polynomial order can be determined as follows. The designer would have $(n-1)$ equations relating to the displacement boundary conditions Eq. (9), $(n-1)$ equations ensuring continuity of the follower's displacement (7) and further $2(n-1)$ equations given by Eq. (10), and 2 equations for the natural boundary conditions. Therefore, the number of coefficients to solve, equals the number of equations, namely $4(n-1) = 4(21-1) = 80$.

## 3.3. Solving for the polynomial coefficients

Table. 1 lists the coordinates of each breakpoint to be used in Eq. (9), along with the corresponding velocity and acceleration values for each breakpoint to be applied in Eq. (10). A designer can typically define the desired displacement values at each breakpoint. For example, consider a follower rise and dwell cycle with a 21.52 mm amplitude, represented as a profile with 21 breakpoints, as shown in Table. 1, corresponding to a sample spoon. In Ecam design, a follower rise and dwell cycle describes a specific phase of the follower's motion as it interacts with the Ecam profile:

- Follower rise: The follower moves upward (away from the camshaft center) as the Ecam rotates.
- Dwell: The follower remains stationary at a constant position for a certain duration, even as the cam continues to rotate.

The term '21.52 mm amplitude' refers to the maximum displacement of the follower during the rise phase. Based on Table 1, in this example:

- The follower starts at 0 mm displacement at $\beta_0 = 0$.
- It rises to 21.52 mm at $\beta_{17} = 305.32°$ (the maximum amplitude).
- It then returns to 0 mm at $\beta_{20} = 360°$.

Throughout this complete cam rotation (360°), the follower rises to its peak displacement, may dwell at that height, and then returns to its initial position.

The procedure for determining the coefficients of a cubic spline polynomial for the sample example involves the following steps:

1. Establish the number of breakpoints required: $n = 21$;
- Note that the boundary condition for breakpoint 1 at 0° is the same as those for breakpoint 21 at 360°.
2. Boundary conditions are listed in Table. 1.
3. Based on boundary conditions, we will require 20 cubics. Sea Appendix A.
4. Since there are 80 coefficients, we must derive 80 equations to solve it. The splines must agree with the function (the Ecam-displacement) at the nodes (the Ecam-angle). Sea Appendix B.
5. The first and second derivatives of the cubics must agree at their shared nodes Sea Appendix C.
6. The final two equations come from the natural boundary conditions.

$$\ddot{S}_1(13.66) = 0 = 2b_{3,1} + 81.96b_{4,1}$$
$$\ddot{S}_{20}(360) = 0 = 2b_{3,20} + 109.38b_{4,20} \quad (11)$$

7. Write equations in matrix form, constructing a linear system in terms of the polynomial coefficients.

$$[\vec{M}].[\vec{b}_{i,j}] = 0 \quad (12)$$

8. Solving the system (12), each coefficient can be expressed as a function of the breakpoints $\beta_1 \cdots \beta_{n-1}$ and the values for the boundary conditions

$$\mathrm{Disp}_1 \cdots \mathrm{Disp}_{n-1}, \mathrm{Vel}_1 \cdots \mathrm{Vel}_{n-1}, \mathrm{Accel}_1, \cdots, \mathrm{Accel}_{n-1}$$

**Tab.1. Boundary conditions for piecewise polynomials of a spoon Ecam curve**

| Breakpoint no. | Ecam Angle β (deg) | Displacement (mm) | Velocity (mm/rad) | Acceleration (mm/rad$^2$) |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 13.66 | 6.45 | 0 | 0 |
| 2 | 31.90 | 6.13 | 0 | 0 |
| 3 | 50.12 | 5.81 | 0 | 0 |
| 4 | 68.35 | 5.50 | 0 | 0 |
| 5 | 86.57 | 5.18 | 0 | 0 |
| 6 | 104.81 | 4.86 | 0 | 0 |
| 7 | 123.04 | 4.55 | 0 | 0 |
| 8 | 141.26 | 4.23 | 0 | 0 |
| 9 | 159.50 | 3.91 | 0 | 0 |
| 10 | 177.72 | 3.59 | 0 | 0 |
| 11 | 195.50 | 3.31 | 0 | 0 |
| 12 | 214.17 | 3.12 | 0 | 0 |
| 13 | 232.41 | 3.02 | 0 | 0 |
| 14 | 250.64 | 6.84 | 0 | 0 |
| 15 | 268.86 | 16.39 | 0 | 0 |
| 16 | 287.08 | 20.87 | 0 | 0 |
| 17 | 305.32 | 21.52 | 0 | 0 |
| 18 | 323.55 | 19.66 | 0 | 0 |
| 19 | 341.77 | 15.13 | 0 | 0 |
| 20 | 360.00 | 0 | 0 | 0 |

## 4. Particle Swarm Optimization

The PSO algorithm is a partly new population-based heuristic optimization method which is based on a metaphor of social interaction, specifically bird flocking [23]. In PSO, each candidate solution is referred to as a 'particle.' Each particle in the swarm represents a potential solution to the optimization problem. If the solution comprises multiple variables, the particle can be represented as a vector of variables. In PSO, particles navigate through a multidimensional search space, adjusting their positions based on their momentum, as well as their personal and global experiences. Each particle utilizes its best-known position and that of its neighbors to move closer to an optimal solution. The suitability of each particle can be evaluated based on the cost function of the optimization problem. At each iteration, the velocity of each particle is calculated as follows:

$$v_i(k+1) = \omega v_i(k) + c_1 r_1 (P_{id} - x_i(k)) + c_2 r_2 \left(P_{gd} - x_i(k)\right) \quad (13)$$

Where k represents the current iteration step, ω is the inertia weight, $c_1$ and $c_2$ are the acceleration constants, $r_1$ and $r_2$ are random numbers within the range [0,1], $x_i(k)$ is the present position of the particle, $P_{id}$ is one of the best solutions this particle has achieved and $P_{gd}$ is one of the best solutions all the particles have achieved. After calculating the velocity, the new position of each particle is determined as follows:

$$x_i(k+1) = x_i(k) + v_i(k+1) \quad (14)$$

In PSO, (13) is used to calculate the new velocity according to its previous velocity, the distance of its current position from both its own personal best position and the global best position of the entire population. Then, the particle moves to a new position according to equation (14). This process continues until a stopping criterion is reached. Typically, the maximum number of iterations is used as the stopping criterion.

### 4.1. Random inertia weight particle swarm optimisation

While standard PSO (SPSO) offers significant improvements and achieves a high rate of convergence for certain problems, it also exhibits some limitations. It has been shown that SPSO struggles to find the best particle due to the absence of a speed control mechanism. Most of the procedures are tried to ameliorate the efficiency of SPSO by changeable inertia weight. The inertia weight is crucial for the effectiveness of PSO, as it balances the swarm's ability to explore globally and exploit locally. In PSO, the

inertia weight governs the influence of previous velocities on the current velocity, making its value important for regulating the trade-off between global (wide-ranging) and local (nearby) exploration. A larger inertia weight enhances global exploration, while a smaller inertia weight encourages local exploration to refine the search within the current area. A good balance between exploration and exploitation can cause increased performance and reduce the required iterations on average to find the optimum. Hence, to balance global exploration and local exploitation abilities of the swarm in [22] RNW–PSO is used and is obtained by changing (13) as below.

$$v_i(k+1) = r_0 v_i(k) + c_1 r_1 (P_{id} - x_i(k)) + c_2 r_2 \left(P_{gd} - x_i(k)\right) \quad (15)$$

where $r_0$ is a uniformly distributed random number inside the interval (0, 1), and other parameters are the same as before. Before starting the optimization process, a performance criterion or objective function must first be defined. Typically, heuristic algorithms like PSO only need to evaluate the objective function to guide their search, without requiring further information about the system. So, in this paper, the sum of squared errors (SSE) given in (16) is considered as an objective function.

$$SSE = \sum_{k=1}^{N} [y(k) - y_d(k)]^2 + \left[\frac{\partial y(k)}{\partial k} - \frac{\partial y_d(k)}{\partial k}\right]^2 + \left[\frac{\partial^2 y(k)}{\partial k^2} - \frac{\partial^2 y_d(k)}{\partial k^2}\right]^2 \quad (16)$$

where e is the tracking error, N is the number of given sampling steps, $y(k)$ and $y_d(k)$ are the curve obtained by the cubic spline interpolation method and the reference curve of an item, respectively.

To minimize the objective function, in each iteration, the breakpoints in piecewise polynomials are randomly chosen by the RNW–PSO algorithm. Then the program will be run. At the end of the run, the objective function value is computed and compared with the value from previous iterations. If the new value is better than previous values, new estimated values for the position of breakpoints are stored. This process continues until a stopping criterion is met. Upon completion, the RNW–PSO algorithm provides the best solution as the optimal result.

## 5. Simulation and Practical Results

In this section, the effectiveness of the proposed methodology is validated through both simulations and practical experiments. The objectives of the practical experiments are listed below:

1. Obtaining an E-Cam curve that best matches the reference curve.
2. Removing the roughness of SAE 304 stainless steel while minimizing deviations from the item's reference curve.

Initially, simulations were performed using MATLAB software to demonstrate the optimization process and evaluate the performance of the RNW–PSO algorithm in determining the optimal Ecam curve. Subsequently, practical results were obtained by implementing the optimized Ecam curve on a CNC sanding machine. Comparative analyzes between simulated and practical results highlight the advantages of the optimized Ecam curve in terms of its improved tracking accuracy, reduced velocity and acceleration differences, and superior sanding quality.

To simulate the proposed method, MATLAB software was used. The simulation was conducted on a personal computer with a Core i7 Duo processor, 3.06 GHz, 8 GB of RAM, running Windows 10. In each iteration, the RNW–PSO algorithm randomly selects the positions of the breakpoints to minimize the objective function. The parameters of the RNW–PSO algorithm in (15) are set as $c_1 = c_2 = 2$ and $r_1 = r_2 = 2$. Fig. 5 illustrates the curve equations of a sample item obtained using the cubic spline method. The vertical axis is measured in PUU (Pulse of User Unit), a unit scaled by the electronic gear, while the horizontal axis represents the angle in degrees. During each iteration, the breakpoints' positions are optimized by RNW–PSO, and the polynomial coefficients in (12) are calculated using the cubic spline function executed in MATLAB. The vertical axis is measured in PUU, a unit scaled by the electronic gear, while the horizontal axis represents the angle in degrees. The relationship between PUU and angle depends on the electronic gear ratio and the mechanical system configuration. Generally, this relationship can be expressed as:

$$\theta = \frac{PUU \times 360°}{\text{PUU per revolution}} \quad (17)$$

where θ is the rotational angle (in degrees), PUU is the commanded or feedback pulse count, and PUU per revolution represents the number of PUUs required for one full revolution of the servo motor or system. The PUU per revolution is determined by:

$$\text{PUU per revolution} = \text{Encoder Resolution} \times \text{Electronic Gear Ratio} \quad (18)$$

where Encoder Resolution is the number of pulses per revolution (PPR) generated by the encoder, and Electronic Gear Ratio (E-Gear) is the scaling factor applied to modify the pulse rate.

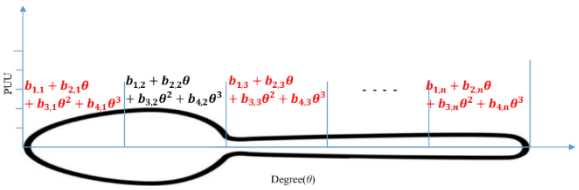



**Fig.5. Curve equations of a sample item obtained by the cubic spline method**

Ecam displacement curves with equally spaced breakpoints and optimized breakpoints are compared in Fig. 6 and Fig. 7 respectively.

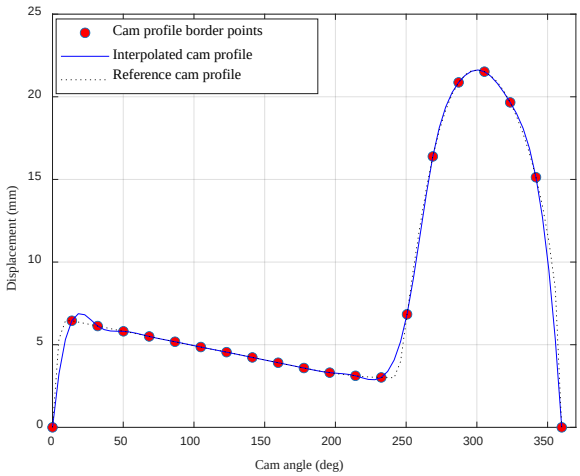



**Fig.6. Sample curve obtained by cubic spline interpolation method using breakpoints with eaqual border points**

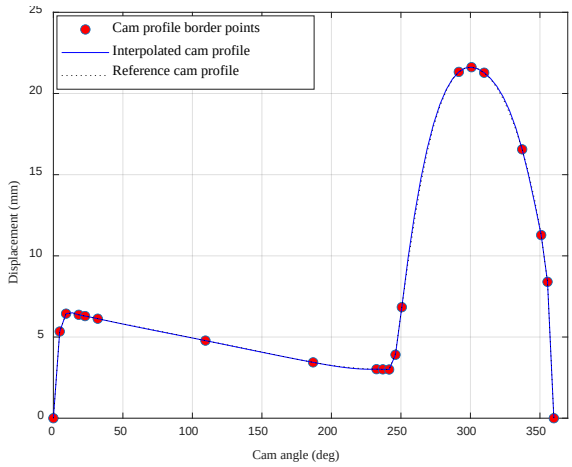



**Fig.7. Sample curve obtained by cubic spline interpolation method using optimized breakpoints**

It can be observed that, with the same number of breakpoints, cubic spline interpolation using optimized breakpoints provides better alignment with the reference curve.

Ecam velocity and acceleration curves with equally spaced breakpoints and optimized breakpoints are compared in Fig. 8 and Fig. 9 respectively, with the results summarized in Table. 2. The comparison clearly shows that cubic spline interpolation with optimized breakpoints exhibits smaller velocity and acceleration differences from the reference curve compared to interpolation with equally spaced breakpoints.

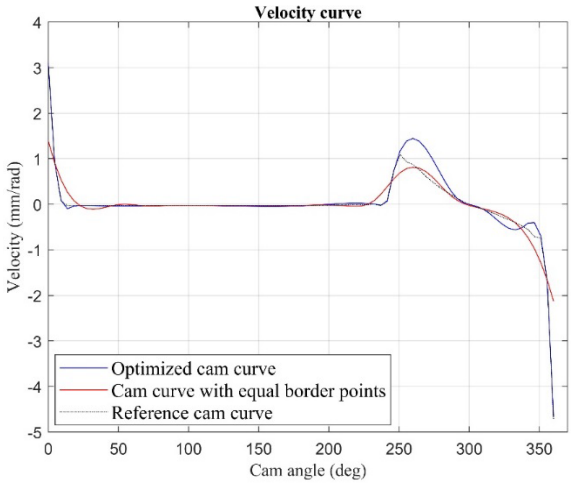



**Fig.8. Ecam velocity curves with equally spaced and optimized breakpoints**

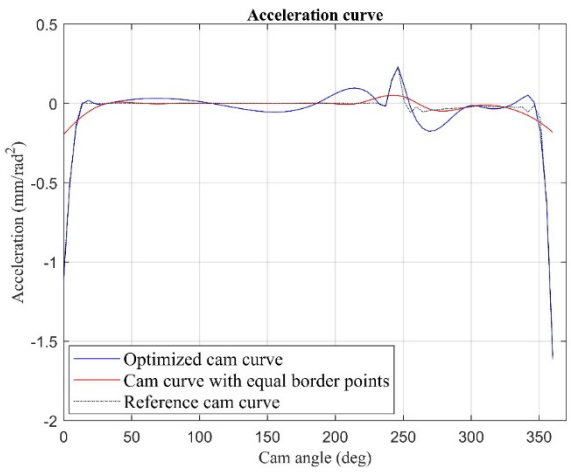


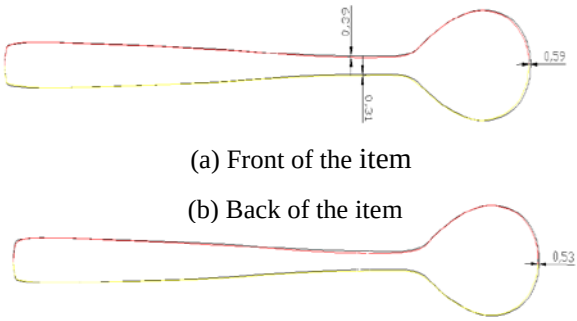

**Fig.9. Ecam acceleration curves with equally spaced and optimized breakpoints**

As shown in Fig. 10, to evaluate the performance of the proposed algorithm, the sanded item was scanned using the Renishaw Cyclone Digitizer, which generates the item's map in CAD format. The CAD profiles obtained from the Renishaw Cyclone Digitizer are shown in Fig. 11.

**Tab.2. Displacement, velocity and acceleration error**

| | Optimized Breakpoints | Equally Spaced Breakpoints |
|---|---|---|
| $[y(k) - y_d(k)]$ | 3.0302 | 16.8043 |
| $\left[\frac{\partial y(k)}{\partial k} - \frac{\partial y_d(k)}{\partial k}\right]$ | 5.4818 | 8.8227 |
| $\left[\frac{\partial^2 y(k)}{\partial k^2} - \frac{\partial^2 y_d(k)}{\partial k^2}\right]$ | 2.9213 | 4.3086 |

**Fig.10. Checking the accuracy of sanded item by renishaw cyclone digitizer**




(a) Front of the item

(b) Back of the item

**Fig.11. CAD profiles are obtained by the renishaw cyclone digitizer. Black line: reference profile, red line: Ecam with equal border points, yellow line: optimized Ecam**

The black lines represent the reference profile, while the yellow lines indicate the profile of the sample item sanded using the optimized Ecam curve. The red lines illustrate the CAD profile of the sample item sanded using the Ecam curve with equally spaced breakpoints. The deviation values from the reference curve, measured in millimeters, at the spoon head and spoon handle are presented in Table. 3.

## 6. Conclusion

This paper presented a novel approach for designing an optimal Ecam profile by determining the best breakpoints for piecewise cubic spline interpolation using the RNW–PSO algorithm. The proposed methodology was implemented on a CNC sanding machine to demonstrate its practical applicability. Simulation results confirmed that the optimized Ecam curve achieved superior tracking accuracy and minimized velocity and acceleration deviations compared to equally spaced breakpoints. Practical experiments further validated the effectiveness of the proposed method, showing improved sanding quality and better compliance with the reference curve. The integration of the optimized Ecam curve into a PLC system (SIEMENS SIMATIC CPU 1215C) using MATLAB software demonstrated the flexibility and scalability of the approach. By storing multiple Ecam curves in the HMI memory, the system allows for adaptability to various items and production requirements.

## References

[1]. Z. Zhang, F. Liu, P. Wang, R. Hu, B. Sun, Methodology to parametric design of cam profile for electronic unit pump, Energy, Vol. 139, (2017), pp. 170–183. DOI: 10.1016/j.energy.2017.07.142.

[2]. B. Hu, C. Zhou, S. Chen, Elastic dynamics modelling and analysis for a valve train including oil film stiffness and dry contact stiffness, Mech. Mach. Theory, Vol. 131, (2019) pp. 33–47. DOI: 10.1016/j.mechmachtheory.2018.09.009.

**Tab.3. Deviations from the reference curve**

| | Optimized Breakpoints | Equally Spaced Breakpoints |
|---|---|---|
| **Spoon Handle (mm)** | 0.31 | 0.39 |
| **Spoon Head (mm)** | 0.53 | 0.59 |

[3]. S. Ji, J. Zhao, Y. Zhang, An application of geodesics to the calculation of the rib-thickness of globoidal cam mechanisms, Mech. Mach. Theory, Vol. 87, (2015), pp. 163–176. DOI: 10.1016/j.mechmachtheory. 2015.01.003.

[4]. K. Verma, R. Belokar, V.K. Verma, K. Ntalianis, Track-based analysis for profile generation on globoidal cam in automatic tool changer of CNC machining center, Assembly Autom, Vol. 39, (2019), pp. 369–379. DOI: 10.1108/AA-08-2018-111.

[5]. S.P. Mall, S.K. Srivastava, Cam profile analysis and simulation using synthetic curves, International Conference on Advances in Mechanical, (2017), pp. 225–230. DOI: 10.1109/AMIAMS.2017.8069216.

[6]. C. Zhou, B. Hu, S. Chen, L. Ma, Design and analysis of high-speed cam mechanism using Fourier series, Mech. Mach. Theory, Vol. 104, (2016), pp. 118–129. DOI: 10.1016/j.mechmachtheory.2016.05.009.

[7]. A. Borboni, F. Aggogeri, I. Elamvazuthi, G. Incerti, P.L. Magnani, Effects of profile interpolation in cam mechanisms, Mech. Mach. Theory, Vol. 144, (2020), p. 103652. DOI: 10.1016/j.mechmachtheory.2019.103652.

[8]. H. Abderazek, A.R. Yildiz, S. Mirjalili, Comparison of recent optimization algorithms for design optimization of a cam-follower mechanism, Knowledge-Based Systems, Vol. 191, (2020), p. 105237. DOI: 10.1016/ j.knosys.2019.105237, 105237-105237.

[9]. F. Hamza, H. Abderazek, S. Lakhdar, D. Ferhat, A.R. Yıldız, Optimum design of cam-roller follower mechanism using a new evolutionary algorithm, Int. J. Adv. Manuf. Tech. Vol. 99, (2018), pp. 1267–1282. DOI: 10.1007/s00170-018-2543-3.

[10]. X. Shang, B.-Z. Xia, S.-Y. Ren, Design and optimization of cam connecting rod system in high-speed drives based on dynamic characteristics and mathematical model, P. I. Mech. Eng. K, J. Mul. Vol. 232, (2018), pp. 113–128. DOI: 10.1177/14644193177 11345.

[11]. J.W. Yu, H. Luo, J.Z. Hu, T.V. Nguyen, Y.T. Lu, Reconstruction of high-speed cam curve based on high-order differential interpolation and shape adjustment, Appl. Math. Comput. Vol. 356, (2019), pp. 272–281. DOI: 10.1016/j.amc.2019.03.049.

[12]. M. Chew, C.H. Chuang, Minimizing residual vibrations in high-speed cam-follower systems over a range of speeds, J. Mech. Design, Vol. 117, (1995), pp. 166–172. DOI: 10.1115/1.2826102.

[13]. M. Hidalgo-Martínez, E. Sanmiguel-Rojas, M. Burgos, Design of cams with negative radius follower using B´ezier curves, Mech. Mach. Theory, Vol. 82, (2014), pp. 87–96. DOI: 10.1016/j.mechmachtheory. 2014.08.001.

[14]. J.K. Jiang, Y.R. Iwai, H. Su, Minimizing and restricting vibrations in high-speed cam-follower systems over a range of speeds, Int. J. Appl. Mech. Vol. 74, (2007), pp. 1157–1164. DOI: 10.1115/1.2723812.

[15]. H. Qiu, C. Lin, Z. Li, H. Ozaki, J. Wang, Y. Yue, A universal optimal approach to cam curve design and its applications, Mech. Mach. Theory, Vol. 40, (2005), pp. 669–692. DOI: 10.1016/j.mechmachtheory. 2004.12.005.

[16]. L.K. Sahu, V.K. Kedia, M. Sahu, Design of cam and follower system using basic and synthetic curves: A review, Int. J. Innovative Sci. Eng. Technol. Vol. 3, (2016), pp. 362–372.

[17]. K. Yoon, S.S. Rao, Cam motion synthesis using cubic splines, J. Mech. Design 115 (1993) 441–446. DOI: 10.1115/1.2919209.

[18]. J. Kim, K. Ahn, S. Kim, Optimal synthesis of a spring-actuated cam mechanism using a cubic spline, P. I. Mech. Eng. C, J. Mec. Vol. 216, (2002), pp. 875–883. DOI: 10.11 77/095440620221600902.

[19]. V. Nguyen, D. Kim, Flexible cam profile synthesis method using smoothing spline curves, Mech. Mach. Theory, Vol. 42, (2007), pp. 825–838. DOI: 10.1016/j.mech machtheory.2006.07.005.

[20]. P. Mermelstein, M. Acar, Optimising cam motion using piecewise polynomials, Eng. Comput. Vol. 19, (2004), pp. 241–254. DOI: 10.1007/s00366-003-0264-0.

[21]. T.T.N. Nguyen, S. Kurtenbach, M. Husing, B. Corves, A general framework for motion design of the follower in cam mechanisms by using non-uniform rational B-spline, Mech. Mach. Theory, Vol. 137, (2019), pp. 374–385. DOI: 10.1016/j.mech machtheory.2019.03.029.

[22]. M.H. Khooban, A. Alfi, D. Nazari Maryam

Abadi, Control of a class of non-linear uncertain chaotic systems via an optimal Type-2 fuzzy proportional integral derivative controller, IET Science, Measurement and Technology, Vol. 7, Iss. 1, pp. 50–58, 2013. DOI: 10.1049/iet-smt.2012.0092.

[23]. X. Zhang,Y. Yang, Optimization of PID controller parameters using a hybrid PSO algorithm. Int. J. Dynam. Control, Vol. 12, (2024), pp. 3617–3627. DOI: 10.1007/ s40435-024-01455-y.

[24]. C. Lu, W. Houng, C. Chang, S. Yen, C. Tseng, T. Su, Stamping System with Automated Feeding Based on Programmable Logic Controller Electronic Cam Function, Sensors and Materials, Vol. 34, No. 3, (2022), pp. 1105–1117. DOI: 10.18494/ SAM. 2022.3478.

[25]. A. Setiadi, H. J. Berchmans, Development of Electronic Cam Motion Control for Synchronous Cutting System, Conference on Management and Engineering in Industry (CMEI), (2024).

[26]. K. Wójkowski, K. Talaśka, D. Wilczyński, Mathematical Model of the Electronic Cam in Terms of Application in a Dosing Machine, Processes, Vol. 12(9), (2024), p. 1909.3. DOI: 10.3390/pr12091909.

[27]. C. Ming, X Chi, Z. Sun, Y. Sun, Design of electronic cam for lower hook mechanism of fishing net-weaving machine based on polynomial fitting, Textile Research Journal, Vol. 92, (2022), pp. 1748–1759. DOI: 10.1177/004051752110687.

[28]. S. A. Rizvi, W. Ali, An artificial neural network approach to prediction of surface roughness and material removal rate in CNC turning of C40 steel, IJIEPR, Vol. 32(3), (2021), pp.1-10. DOI: 10.22068/ ijiepr.32.3.7.

[29]. C. Martin, P. Enqvist, J. Tomlinson, Z. Zhang, Linear Control Theory, Splines and Interpolation, In: Computation and Control IV. Progress in Systems and Control Theory, Vol 20, Birkhäuser Boston, 1995. DOI: 10.1007/978-1-4612-2574-4-18.

**Appendix A: Cubic spline equations**

| | |
|---|---|
| $S_1(x) = b_{1,1} + b_{2,1}(x-0) + b_{3,1}(x-0)^2 + b_{4,1}(x-0)^3$ | $S_{11}(x) = b_{1,11} + b_{2,11}(x-177.72) + b_{3,11}(x-177.72)^2 + b_{4,11}(x-177.72)^3$ |
| $S_2(x) = b_{1,2} + b_{2,2}(x-13.66) + b_{3,2}(x-13.66)^2 + b_{4,2}(x-13.66)^3$ | $S_{12}(x) = b_{1,12} + b_{2,12}(x-195.50) + b_{3,12}(x-195.50)^2 + b_{4,12}(x-195.50)^3$ |
| $S_3(x) = b_{1,3} + b_{2,3}(x-31.90) + b_{3,3}(x-31.90)^2 + b_{4,3}(x-31.90)^3$ | $S_{13}(x) = b_{1,13} + b_{2,13}(x-214.17) + b_{3,13}(x-214.17)^2 + b_{4,13}(x-214.17)^3$ |
| $S_4(x) = b_{1,4} + b_{2,4}(x-50.12) + b_{3,4}(x-50.12)^2 + b_{4,4}(x-50.12)^3$ | $S_{14}(x) = b_{1,14} + b_{2,14}(x-232.41) + b_{3,14}(x-232.41)^2 + b_{4,14}(x-232.41)^3$ |
| $S_5(x) = b_{1,5} + b_{2,5}(x-68.35) + b_{3,5}(x-68.35)^2 + b_{4,5}(x-68.35)^3$ | $S_{15}(x) = b_{1,15} + b_{2,15}(x-250.64) + b_{3,15}(x-250.64)^2 + b_{4,15}(x-250.64)^3$ |
| $S_6(x) = b_{1,6} + b_{2,6}(x-86.57) + b_{3,6}(x-86.57)^2 + b_{4,6}(x-86.57)^3$ | $S_{16}(x) = b_{1,16} + b_{2,16}(x-268.86) + b_{3,16}(x-268.86)^2 + b_{4,16}(x-268.86)^3$ |
| $S_7(x) = b_{1,7} + b_{2,7}(x-104.81) + b_{3,7}(x-104.81)^2 + b_{4,7}(x-104.81)^3$ | $S_{17}(x) = b_{1,17} + b_{2,17}(x-287.08) + b_{3,17}(x-287.08)^2 + b_{4,17}(x-287.08)^3$ |
| $S_8(x) = b_{1,8} + b_{2,8}(x-123.04) + b_{3,8}(x-123.04)^2 + b_{4,8}(x-123.04)^3$ | $S_{18}(x) = b_{1,18} + b_{2,18}(x-305.32) + b_{3,18}(x-305.32)^2 + b_{4,18}(x-305.32)^3$ |
| $S_9(x) = b_{1,9} + b_{2,9}(x-141.26) + b_{3,9}(x-141.26)^2 + b_{4,9}(x-141.26)^3$ | $S_{19}(x) = b_{1,19} + b_{2,19}(x-323.55) + b_{3,19}(x-323.55)^2 + b_{4,19}(x-323.55)^3$ |
| $S_{10}(x) = b_{1,10} + b_{2,10}(x-159.50) + b_{3,10}(x-159.50)^2 + b_{4,10}(x-159.50)^3$ | $S_{20}(x) = b_{1,20} + b_{2,20}(x-341.77) + b_{3,20}(x-341.77)^2 + b_{4,20}(x-341.77)^3$ |

**Appendix B: Implementation of boundary conditions in cubic spline equations**

| | |
|---|---|
| $0 = S_1(0) = b_{1,1}$ | $3.59 = S_{11}(177.72) = b_{2,11}$ |
| $6.45 = S_1(13.66)$ | $3.31 = S_{11}(195.50)$ |
| $6.45 = b_{1,1} + 13.66b_{2,1} + 186.59b_{3,1} + 2548.89b_{4,1}$ | $3.31 = b_{1,11} + 17.78b_{2,11} + 316.12b_{3,11} + 5620.76b_{4,11}$ |
| $6.45 = S_2(13.66) = b_{2,2}$ | $3.31 = S_{12}(195.50) = b_{2,12}$ |
| $6.13 = S_2(31.90)$ | $3.12 = S_{12}(214.17)$ |
| $6.13 = b_{1,2} + 18.24b_{2,2} + 332.69b_{3,2} + 6068.40b_{4,2}$ | $3.12 = b_{1,12} + b_{2,12}(x - 195.50) + b_{3,12}(x - 195.50)^2 + b_{4,12}(x - 195.50)^3$ |
| $6.13 = S_3(31.90) = b_{2,3}$ | $3.12 = S_{13}(214.17) = b_{2,13}$ |
| $5.81 = S_3(50.12)$ | $3.02 = S_{13}(232.41)$ |
| $5.81 = b_{1,3} + 18.22b_{2,3} + 331.96b_{3,3} + 6048.46b_{4,3}$ | $3.02 = b_{1,13} + 18.24b_{2,13} + 332.69b_{3,13} + 6068.40b_{4,13}$ |
| $5.81 = S_4(50.12) = b_{2,4}$ | $3.02 = S_{14}(232.41) = b_{2,14}$ |
| $5.50 = S_4(68.35)$ | $6.84 = S_{14}(250.64)$ |
| $5.50 = b_{1,4} + 18.23b_{2,4} + 332.33b_{3,4} + 6058.42b_{4,4}$ | $6.84 = b_{1,14} + 18.23b_{2,14} + 332.33b_{3,14} + 6058.42b_{4,14}$ |
| $5.50 = S_5(68.35) = b_{2,5}$ | $6.84 = S_{15}(250.64) = b_{2,15}$ |
| $5.18 = S_5(86.57)$ | $16.39 = S_{15}(268.86)$ |
| $5.18 = b_{1,5} + 18.22b_{2,5} + 331.96b_{3,5} + 6048.46b_{4,5}$ | $16.39 = b_{1,15} + 18.22b_{2,15} + 331.96b_{3,15} + 6048.46b_{4,15}$ |
| $5.18 = S_6(86.57) = b_{2,6}$ | $16.39 = S_{16}(268.86) = b_{2,16}$ |
| $4.86 = S_6(104.81)$ | $20.87 = S_{16}(287.08)$ |
| $4.86 = b_{1,6} + 18.24b_{2,6} + 332.69b_{3,6} + 6068.40b_{4,6}$ | $20.87 = b_{1,16} + 18.22b_{2,16} + 331.96b_{3,16} + 6048.46b_{4,16}$ |
| $4.86 = S_7(104.81) = b_{2,7}$ | $20.87 = S_{17}(287.08) = b_{2,17}$ |
| $4.55 = S_7(123.04)$ | $21.52 = S_{17}(305.32)$ |
| $4.55 = b_{1,7} + 18.23b_{2,7} + 332.33b_{3,7} + 6058.42b_{4,7}$ | $21.52 = b_{1.17} + 18.24b_{2,17} + 332.69b_{3,17} + 6068.40b_{4,17}$ |
| $4.55 = S_8(123.04) = b_{2,8}$ | $21.52 = S_{18}(305.32) = b_{2,18}$ |
| $4.23 = S_8(141.26)$ | $19.66 = S_{18}(323.55)$ |
| $4.23 = b_{1,8} + 18.22b_{2,8} + 331.96b_{3,8} + 6048.46b_{4,8}$ | $19.66 = b_{1,18} + 18.23b_{2,18} + 332.33b_{3,18} + 6058.42b_{4,18}$ |
| $4.23 = S_9(141.26) = b_{2,9}$ | $19.66 = S_{19}(323.55) = b_{2,19}$ |
| $3.91 = S_9(159.50)$ | $15.13 = S_{19}(341.77)$ |
| $3.91 = b_{1,9} + 18.24b_{2,9} + 332.69b_{3,9} + 6068.40b_{4,9}$ | $15.13 = b_{1,19} + 18.22b_{2,19} + 331.96b_{3,19} + 6048.46b_{4,19}$ |
| $3.91 = S_{10}(159.50) = b_{2,10}$ | $15.13 = S_{20}(341.77) = b_{2,20}$ |
| $3.59 = S_{10}(177.72)$ | $0 = S_{20}(360.00)$ |
| $3.59 = b_{1,10} + 18.22b_{2,10} + 331.96b_{3,10} + 6048.46b_{4,10}$ | $0 = b_{1,20} + 18.23b_{2,20} + 332.33b_{3,20} + 6058.42b_{4,20}$ |

**Appendix C: First and second derivatives of cubic spline equations with substituted boundary conditions**

| | |
|---|---|
| $\dot{S}_1(13.66) = b_{2,1} + 27.32b_{3,1} + 559.78b_{4,1}$<br>$b_{2,2} = \dot{S}_2(13.66)$ | $\ddot{S}_1(13.66) = 2b_{3,1} + 81.96b_{4,1}$<br>$2b_{3,2} = \ddot{S}_2(13.66)$ |
| $\dot{S}_2(31.90) = b_{2,2} + 36.48b_{3,2} + 998.09b_{4,2}$<br>$b_{2,3} = \dot{S}_3(31.90)$ | $\ddot{S}_2(31.90) = 2b_{3,2} + 109.44b_{4,2}$<br>$2b_{3,3} = \ddot{S}_3(31.90)$ |
| $\dot{S}_3(50.12) = b_{2,3} + 36.44b_{3,3} + 995.90b_{4,3}$<br>$b_{2,4} = \dot{S}_4(50.12)$ | $\ddot{S}_3(50.12) = 2b_{3,3} + 109.32b_{4,3}$<br>$2b_{3,4} = \ddot{S}_4(50.12)$ |
| $\dot{S}_4(68.35) = b_{2,4} + 36.46b_{3,4} + 996.99b_{4,4}$<br>$b_{2,5} = \dot{S}_5(68.35)$ | $\ddot{S}_4(68.35) = 2b_{3,4} + 109.38b_{4,4}$<br>$2b_{3,5} = \ddot{S}_5(68.35)$ |
| $\dot{S}_5(86.57) = b_{2,5} + 36.44b_{3,5} + 995.90b_{4,5}$<br>$b_{2,6} = \dot{S}_6(86.57)$ | $\ddot{S}_5(86.57) = 2b_{3,5} + 109.32b_{4,5}$<br>$2b_{3,6} = \ddot{S}_6(86.57)$ |
| $\dot{S}_6(104.81) = b_{2,6} + 36.48b_{3,6} + 998.09b_{4,6}$<br>$b_{2,7} = \dot{S}_7(104.81)$ | $\ddot{S}_6(104.81) = 2b_{3,6} + 109.44b_{4,6}$<br>$2b_{3,7} = \ddot{S}_7(104.81)$ |
| $\dot{S}_7(123.04) = b_{2,7} + 36.46b_{3,7} + 996.99b_{4,7}$<br>$b_{2,8} = \dot{S}_8(123.04)$ | $\ddot{S}_7(123.04) = 2b_{3,7} + 109.38b_{4,7}$<br>$2c_{3,8} = \ddot{S}_8(123.04)$ |
| $\dot{S}_8(141.26) = b_{2,8} + 36.44b_{3,8} + 995.90b_{4,8}$<br>$b_{2,9} = \dot{S}_9(141.26)$ | $\ddot{S}_8(141.26) = 2b_{3,8} + 109.32b_{4,8}$<br>$2b_{3,9} = \ddot{S}_9(141.26)$ |
| $\dot{S}_9(159.50) = b_{2,9} + 36.48b_{3,9} + 998.09b_{4,9}$<br>$b_{2,10} = \dot{S}_{10}(159.50)$ | $\ddot{S}_9(159.50) = 2b_{3,9} + 109.44b_{4,9}$<br>$2b_{3,10} = \ddot{S}_{10}(159.50)$ |
| $\dot{S}_{10}(177.72) = b_{2,10} + 36.44b_{3,10} + 995.90b_{4,10}$<br>$b_{2,11} = \dot{S}_{11}(177.72)$ | $\ddot{S}_{10}(177.72) = 2b_{3,10} + 109.32b_{4,10}$<br>$2b_{3,11} = \ddot{S}_{11}(177.72)$ |
| $\dot{S}_{11}(195.50) = b_{2,11} + 35.56b_{3,11} + 948.38b_{4,11}$<br>$b_{2,12} = \dot{S}_{12}(195.50)$ | $\ddot{S}_{11}(195.50) = 2b_{3,11} + 106.68b_{4,11}$<br>$2b_{3,12} = \ddot{S}_{12}(195.50)$ |
| $\dot{S}_{12}(214.17) = b_{2,12} + 37.34b_{3,12} + 1045.70b_{4,12}$<br>$b_{2,13} = \dot{S}_{13}(214.17)$ | $\ddot{S}_{12}(214.17) = 2b_{3,12} + 112.02b_{4,12}$<br>$2b_{3,13} = \ddot{S}_{13}(214.17)$ |
| $\dot{S}_{13}(232.41) = b_{2,13} + 36.48b_{3,13} + 998.09b_{4,13}$<br>$b_{2,14} = \dot{S}_{14}(232.41)$ | $\ddot{S}_{13}(232.41) = 2b_{3,13} + 109.44b_{4,13}$<br>$2b_{3,14} = \ddot{S}_{14}(232.41)$ |
| $\dot{S}_{14}(250.64) = b_{2,14} + 36.46b_{2,14} + 996.99b_{3,14}$<br>$b_{2,15} = \dot{S}_{15}(250.64)$ | $\ddot{S}_{14}(250.64) = 2b_{3,14} + 109.38b_{4,14}$<br>$2b_{3,15} = \ddot{S}_{15}(250.64)$ |
| $\dot{S}_{15}(268.86) = b_{2,15} + 36.44b_{3,15} + 995.90b_{4,15}$<br>$b_{2,16} = \dot{S}_{16}(268.86)$ | $\ddot{S}_{15}(268.86) = 2b_{3,15} + 109.32b_{4,15}$<br>$2b_{3,16} = \ddot{S}_{16}(268.86)$ |
| $\dot{S}_{16}(287.08) = b_{2,16} + 36.44b_{3,16} + 995.90b_{4,16}$<br>$b_{2,17} = \dot{S}_{17}(287.08)$ | $\ddot{S}_{16}(287.08) = 2b_{3,16} + 109.32b_{4,16}$<br>$2b_{3,17} = \ddot{S}_{17}(287.08)$ |
| $\dot{S}_{17}(305.32) = b_{2,17} + 36.48b_{3,17} + 998.09b_{4,17}$<br>$b_{2,18} = \dot{S}_{18}(305.32)$ | $\ddot{S}_{17}(305.32) = 2b_{3,17} + 109.44b_{4,17}$<br>$2b_{3,18} = \ddot{S}_{18}(305.32)$ |
| $\dot{S}_{18}(323.55) = b_{2,18} + 36.48b_{3,18} + 996.99b_{4,18}$<br>$b_{2,19} = \dot{S}_{19}(323.55)$ | $\ddot{S}_{18}(323.55) = 2b_{3,18} + 109.38b_{4,18}$<br>$2b_{3,19} = \ddot{S}_{19}(323.55)$ |
| $\dot{S}_{19}(341.77) = b_{2,19} + 36.44b_{2,19} + 995.90b_{4,19}$<br>$b_{2,20} = \dot{S}_{20}(341.77)$ | $\ddot{S}_{19}(341.77) = 2b_{3,19} + 109.32b_{4,19}$<br>$2b_{3,20} = \ddot{S}_{20}(341.77)$ |

Follow this article at the following site:

Davood Nazari Maryam Abadi, Mohammad Bagheri. "Determining Optimal Piecewise Polynomial Coefficients for the Electronic Cam in the CNC Sanding Mechanism" IJIEPR 2025; 36 (2): 156-169
URL: http://ijiepr.iust.ac.ir/article-1-2274-en.html