

Choice of Interpolation Method for an Arbitrary Arc Welding Curve

Olesja Minejeva¹, Zigurds Markovics²
^{1,2} Riga Technical University

Abstract – Practical application of the classical methods of interpolation, such as Lagrange's, Bessel's, Stirling's, Newton's and others, cannot always provide sufficient accuracy in the restoration of lines according to the requirements of modern technologies. In this paper, recommendations for the improvement of interpolation accuracy are given, including the transition to the method of cubic splines. The results of research are used to create a route for arc welding in order to increase the performance of industrial robot.

Keywords – Arbitrary curve, accuracy, cubic splines, degrees of polynomials, finite differences, interpolation, polynomials, welding.

I. INTRODUCTION

This paper presents issues that occur in the industrial robot applications during the arc welding process. One of the most important problems is to ensure and increase the precision, specifying robot's route of movement or weld seam.

Accuracy in the spot welding depends only on technical possibilities to determine the coordinates of the necessary point and on technical capabilities of robot to realise the coordinates of these points. In case of arc welding, it is necessary to apply the methods of interpolation.

As it is known, interpolation belongs to the class of approximate calculation, so questions of accuracy are especially topical.

II. PROBLEM STATEMENT

Robot control system, which is suitable for the spot welding, is developed in robotics and called "discrete positional control" system. The coordinates of necessary points, where a robot must stay and start welding, are defined.

Robot relocates grip with the weld mechanism from one position to another, according to how it is written in the program.

Block diagram of the control system is given in Fig. 1.

The coordinates of the welding positions are stored in the "memory of positions" block. The "discrete positioning" block compares the information with the actual state of the manipulator and provides its getting into the necessary state. After the receiving of such a signal, the "state transition" block gives permission for the further actions through the "sequence of the actions" block, "control of the position" block to a welding mechanism.

The control system becomes more complicated if there is an arc welding process with the uninterrupted seam. For this purpose, "contouring control system" is developed. The coordinates of the positioning points are input. The process goes across these points without stopping. Welding process must continue all the time, also between indicated points. Main question is – which route a welding mechanism should choose in the region between given points. The task of new block "Interpolator" is to calculate it (Fig. 2) [1].

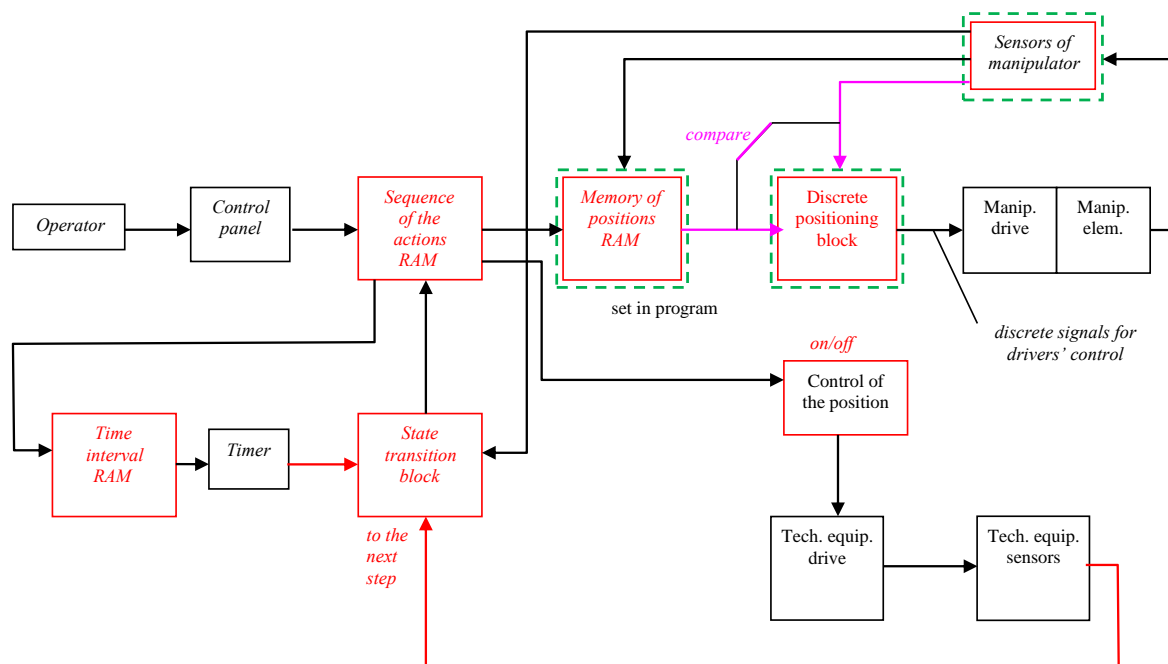


Fig. 1. Discrete positional control system.

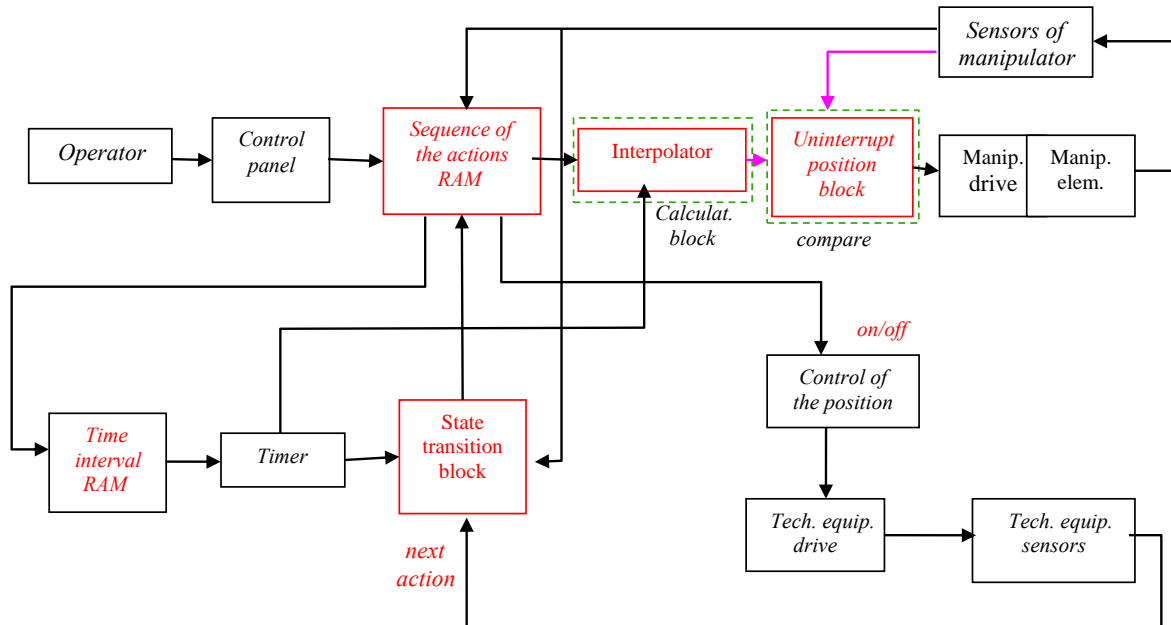


Fig. 2. Contouring control system.

Contouring control provides movement along the continuous trajectories and contains positional control with the block of interpolation, which allows actualising uninterrupted paths between the positioning points according to the certain algorithm.

Contouring control system with interpolator principally is the same system of the positional control, but it is complemented by the special computing unit – interpolator, which calculates values between positioning points according to the set program. It means that the complexity of process execution is intended for the block of interpolation.

III. THE TASK OF INTERPOLATION

Process, when function's approximate intermediary values are calculated in the points that do not match the given points, is called interpolation (Fig. 3).

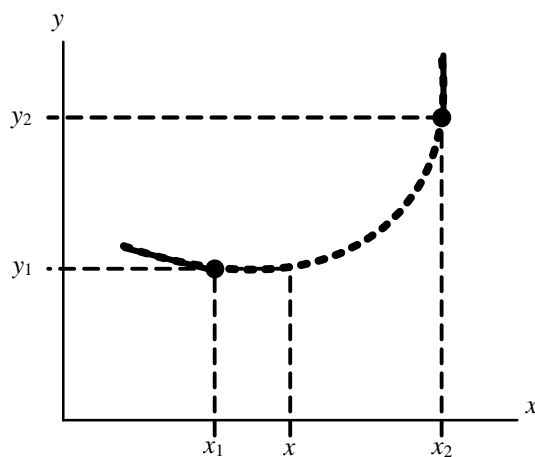


Fig. 3. Meaning of the interpolation.

Two function $y = f(x)$ values y_1 and y_2 in the points x_1 and x_2 are known. Other function's values x in the interval between x_1 and x_2 must be calculated using interpolation methods.

There are 3 ways how function can be set: analytical, graphic and tabular. In our case, we need to use the last one – tabular method. In general, the problem is following.

Suppose that for the discrete values of the argument $x_0, x_1, \dots, x_i, \dots, x_n$ corresponding values $y_0 = f(x_0), y_1 = f(x_1), \dots, y_i = f(x_i), \dots, y_n = f(x_n)$ are known from the practice.

Information is given in tabular form. The task is to find analytical expression $F(x)$ for the function being defined in the table. $F(x)$ values in the points $x_0, x_1, \dots, x_i, \dots, x_n$ have to match with the values of the defined function. Specified points x_0, x_1, \dots, x_n are called interpolation nodes.

Different function expressions can act as function $F(x)$. Combination of the power functions is widely used in practice. As a result, polynomial is formed

$$F(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (1)$$

If it is assumed as interpolation polynomial, parabolic interpolation and Lagrange's formulas are obtained calculating the function values.

If interpolated function $F(x)$ is a periodical function, then interpolation polynomial consists of the composition of $\sin x$ and $\cos x$ functions.

IV. INTERPOLATION IN ARC WELDING TASKS

As the contouring control and respectively interpolation block are used for arc welding, when robot performs uninterrupted welding along the necessary route, it is more convenient to set the route (input function) in the tabular form with the constant argument x step h , i.e. interpolation nodes are distributed uniformly. Then the form of interpolation poly-

mial and the process of the calculation of coefficients become easier.

Then interpolation nodes $x_0, x_1, x_2, \dots, x_i, \dots, x_n$ are expressed as follows:

$$x_0, x_0 + h, x_0 + 2h, \dots, x_0 + nh, \quad (2)$$

$$\text{or: } x_i = x_0 + ih, \quad i = 0, 1, 2, \dots, n,$$

where $h = \text{const}$, argument x step.

Table of the analytically unknown function (welding route) with x values (2) and corresponding y values

$$y_0 = f(x_0), y_1 = f(x_1), \dots, y_i = f(x_i), \dots, y_n = f(x_n) \quad (3)$$

can be obtained practically even from real welded parts by measurement.

Let us recall that the final task of interpolation is to calculate function $y = f(x)$ values in the points x , which are not fixed in the table, i.e. in the points between interpolation nodes x_i , when values of the interpolation polynomial in the nodes x_i coincide with the values of the given function.

V. THEORETICAL QUESTIONS OF THE INTERPOLATION

Let us turn to the classic mathematical methods [2]–[5]. If argument x_i values are set with the constant step h , then finite differences and special table (Table I) of the finite differences have to be used for the formation of interpolation polynomial.

The first order finite difference is difference between the function values in nearby nodes:

$$\Delta^I y_0 = y_1 - y_0 = f(x_1) - f(x_0)$$

$$\Delta^I y_1 = y_2 - y_1 = f(x_2) - f(x_1) \quad (4)$$

...

$$\Delta^I y_{n-1} = y_n - y_{n-1} = f(x_n) - f(x_{n-1})$$

$$\text{or } \Delta^I y_i = y_{i+1} - y_i$$

The second order finite difference is difference between the two first order finite differences:

$$\begin{aligned} \Delta^{II} y_0 &= \Delta^I y_1 - \Delta^I y_0 \\ \Delta^{II} y_{n-2} &= \Delta^I y_{n-1} - \Delta^I y_{n-2} \\ \Delta^{II} y_i &= \Delta^I y_{i+1} - \Delta^I y_i \end{aligned} \quad (5)$$

Similarly other higher-order differences can be found:

$$\Delta^k y_i = \Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i.$$

It is effectively to summarise the calculated finite differences in the special table. Some properties are reflected in the table:

- how the values of finite differences are found,
- what values of the differences are used for the calculating of the different interpolation polynomials. S – Stirling's, B – Bessel's, N_I and N_{II} – Newton's polynomials are shown here.

Additionally, argument x_0 and function y_0 values are located in the centre of the table. It allows showing how to handle the intervals at the beginning of table, when x_0 is the smallest value and further argument values keep rising (table part from x_0 downwards).

TABLE I
THE TABLE OF FINITE DIFFERENCES

x	y	$\Delta^I y$	$\Delta^{II} y$	$\Delta^{III} y$	$\Delta^{IV} y$
$x_{-3}=x_0-3h$	y_{-3}				
		$\Delta^I y_{-3}$			
$x_{-2}=x_0-2h$	y_{-2}		$\Delta^{II} y_{-3}$		
		$\Delta^I y_{-2}$		$\Delta^{III} y_{-3}$	
$x_{-1}=x_0-h$	y_{-1}		$\Delta^{II} y_{-2}$		$\Delta^{IV} y_{-3}$
		$\Delta^I y_{-1}$		$\Delta^{III} y_{-2}$	
x_0	y_0	$\Delta^I y_0$	$\Delta^{II} y_{-1}$	$\Delta^{III} y_{-2}$	$\Delta^{IV} y_{-2}$
			$\Delta^{II} y_0$	$\Delta^{III} y_{-1}$	$\Delta^{IV} y_{-1}$
$x_1=x_0+h$	y_1	$\Delta^I y_1$		$\Delta^{III} y_0$	
		$\Delta^I y_2$	$\Delta^{II} y_1$		
$x_2=x_0+2h$	y_2				
$x_3=x_0+3h$	y_3				

Diagrammatic annotations on the right side of the table:

- Red arrow pointing to $\Delta^{III} y_{-3}$ labeled N_{II}
- Green arrow pointing to $\Delta^{IV} y_{-2}$ labeled S
- Black arrow pointing to $\Delta^{IV} y_{-1}$ labeled B
- Red arrow pointing to $\Delta^{III} y_0$ labeled N_I

It can also be shown how to handle the intervals at the end of table when x_0 is the largest value and further argument values keep decreasing (table part from x_0 upwards).

The main idea of interpolation is that function $y = f(x)$ is replaced by the n order polynomial

$P(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n$, whose values match with the function $y = f(x)$ values to given x values $x = x_0, x = x_1, \dots, x = x_n$ [6].

Geometrically it means that the line $y = f(x)$ is replaced by the n order parabola

$y = a_0x^n + a_1x^{n-1} + \dots + a_n$, which crosses $n + 1$ point on the line.

If only one initial point $M_0(x_0, y_0)$ is given, then zero-degree polynomial is obtained:

$$P(x) = y_0.$$

It means that $y = f(x)$ is replaced by the straight line AB , which crosses point M_0 (Fig. 4).

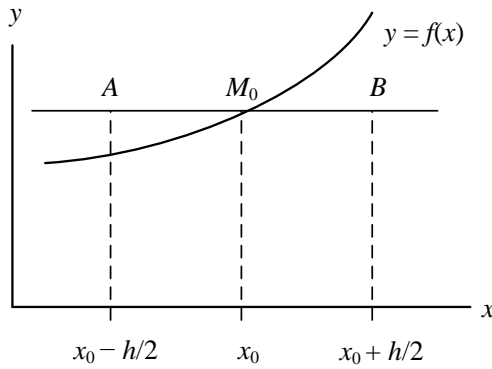


Fig. 4. Zero-degree polynomial.

If 2 values are given:

$$y_0 = f(x_0)$$

$y_1 = f(x_0 + h)$, then the first-degree polynomial is obtained:

$$P(x) = y_0 + \frac{y_1 - y_0}{h}(x - x_0), \quad (6)$$

expressing straight line M_0M_1 through the points $M_0(x_0, y_0)$, $M_1((x_0 + h), y_1)$ (Fig. 5).

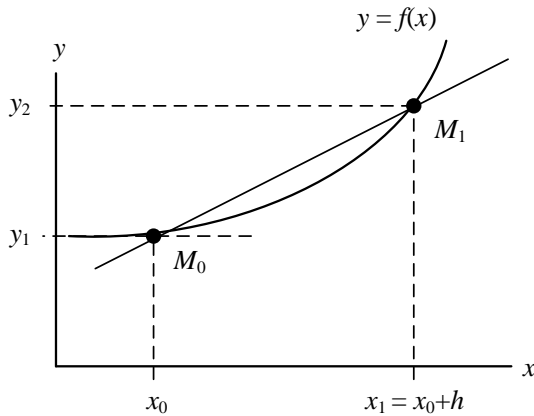


Fig. 5. First-degree polynomial.

If 3 argument values are given

$$y_0 = f(x_0)$$

$$y_1 = f(x_0 + h)$$

$y_2 = f(x_0 + 2h)$, then the second-degree polynomial is obtained:

$$P(x) = y_0 + \frac{y_1 - y_0}{h}(x - x_0) + \frac{y_2 - 2y_1 + y_0}{2h^2}(x - x_0)(x - (x_0 + h)) \quad (7)$$

Polynomial displays parabola with the vertical axis, crossing 3 points: $M_0(x_0, y_0)$, $M_1((x_0 + h), y_1)$, $M_2((x_0 + 2h), y_2)$ (Fig. 6).

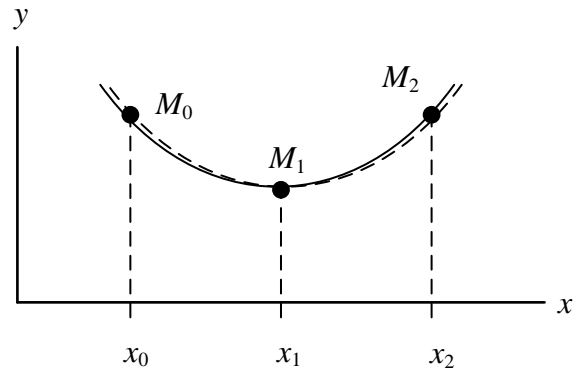


Fig. 6. Second-degree polynomial.

In general, interpolation polynomials are obtained [7]:

$$P_n(x) = y_0 + \frac{\Delta^1 y_0}{1!h}(x - x_0) + \frac{\Delta^2 y_0}{2!h^2}(x - x_0)(x - x_1) + \frac{\Delta^3 y_0}{3!h^3}(x - x_0)(x - x_1)(x - x_2) + \dots + \frac{\Delta^n y_0}{n!h^n}(x - x_0)(x - x_1) \dots (x - x_{n-1}) \quad (8)$$

This is the so-called first Newton's formula, which is abbreviated as NI. In practice, other version of this formula is used, when intermediate value is introduced. Here x is an argument current value for which a function value is calculated: $U = \frac{x - x_0}{h}$.

Thus,

$$P_n(x) = y_0 + \frac{U}{1!} \Delta^1 y_0 + \frac{U(U-1)}{2!} \Delta^2 y_0 + \frac{U(U-1)(U-2)}{3!} \Delta^3 y_0 + \dots + \frac{U(U-1) \dots (U-n+1)}{n!} \Delta^n y_0 \quad (9)$$

Subsequently, the second Newton's formula is:

$$\begin{aligned}
P_n(x) = & y_0 + \frac{\Delta^I y_{-1}}{1!h} (x - x_0) + \\
& + \frac{\Delta^{II} y_{-2}}{2!h^2} (x - x_0)(x - x_{-1}) + \\
& + \frac{\Delta^{III} y_{-3}}{3!h^3} (x - x_0)(x - x_{-1})(x - x_{-2}) + \dots + \\
& + \frac{\Delta^n y_{-n}}{n!h^n} (x - x_0)(x - x_{-1}) \dots (x - x_{-(n-1)})
\end{aligned} \quad (10)$$

$$\begin{aligned}
P_n(x) = & y_0 + \frac{U}{1!} \Delta^I y_{-1} + \frac{U(U+1)}{2!} \Delta^{II} y_{-2} + \\
& + \frac{U(U+1)(U+2)}{3!} \Delta^{III} y_{-3} + \dots + \\
& + \frac{U(U+1)(U+2) \dots (U+n-1)}{n!} \Delta^n y_{-n}
\end{aligned} \quad (11)$$

In literature, Newton's formulas can also be found in other transcription.

Apart from Newton's formulas, other formulas also exist, for example, Stirling's and Bessel's formulas [8]:

$$\begin{aligned}
S(x) = & y_0 + U \frac{\Delta^I y_0 + \Delta^I y_{-1}}{2} + \frac{U^2}{2} \Delta^{II} y_{-1} + \\
& + \frac{U(U^2-1)}{3!} \frac{\Delta^{III} y_{-2} + \Delta^{III} y_{-1}}{2} + \\
& + \frac{U^2(U^2-1)\Delta^{IV} y_{-2}}{4!} + \dots + \\
& + U^2(U^2-1) \dots \frac{(U^2-(n-1)^2)\Delta^{2n} y_{-n}}{(2n)!}
\end{aligned} \quad (12)$$

$$\begin{aligned}
B(x) = & y_0 + U \Delta^I y_0 + \frac{U(U-1)}{2} \frac{\Delta^{II} y_{-1} + \Delta^{II} y_0}{2} + \\
& + \frac{U(U-1)(U-0,5)}{3!} \Delta^{III} y_{-1} + \\
& + \frac{U(U^2-1)(U-2)}{4!} \frac{\Delta^{IV} y_{-2} + \Delta^{IV} y_{-1}}{2} + \dots + \\
& + \frac{(U-0,5)U(U^2-1) \dots (U^2-(n-1)^2)(u-n)}{4!} \Delta^{2n+1} y_{-1}
\end{aligned} \quad (13)$$

Error of interpolation is calculated using formulas [9]:

$$y(x) - N_I(x) = \frac{h^{(n+1)}}{(n-1)!} y^{(n+1)}(\xi) \cdot U(U-1) \dots (U-n) \quad (14)$$

$$y(x) - N_{II}(x) = \frac{h^{(n+1)}}{(n+1)!} y^{(n+1)}(\xi) \cdot U(U+1) \dots (U+n) \quad (15)$$

$$\begin{aligned}
y(x) - S(x) = & \frac{h^{(2n+1)}}{(2n+1)!} y^{(2n+1)}(\xi) \cdot \\
& \cdot U(U^2-1) \dots (U^2-n^2)
\end{aligned} \quad (16)$$

$$\begin{aligned}
y(x) - B(x) = & \frac{h^{(2n+2)}}{(2n+2)!} y^{(2n+2)}(\xi) \cdot \\
& \cdot U(U^2-1) \dots (U^2-n^2)(U-n-1)
\end{aligned} \quad (17)$$

ξ – intermediate value between the largest and the smallest numerical value in the sequence x_0, x_1, \dots, x_n , it can be in different formulas.

VI. PRACTICAL APPLICATIONS OF INTERPOLATION CLASSIC METHODS

For the practical use of interpolation methods, including the calculation of robot's route, it is helpful to follow some instructions.

y values of the necessary function are calculated for each interval of x values.

At every step (for each x value), x_0 and y_0 are taken from the nearest interpolation node point. For example, in the interval x_1, x_2 by way of x_0 becomes x_1 , in the interval x_2, x_3 – accordingly x_2 .

When computing in the bounds of one interval, only U values are changing in formulas. When computing in other intervals, U, y_0 and finite differences are changing in formulas.

Note that in N_I formula only y_0 differences exist, but in N_{II} formula – various y_j differences.

In practice, an important question is when one or another formula is better to use.

From the instructions in Table I, it is clear that finite differences, which are located on the right downward from y_0 , are included in N_I .

In N_{II} formula they are located on the right upwards. Moreover, the area of finite differences itself has wedge form – by increasing order, amount of differences is decreasing.

As a result, formula N_I cannot be used in interval at the bottom of the table, because there is a lack of difference values. Thus, N_I is used in intervals, which are located at the top of the table and in the centre. Formula N_{II} , contrariwise, is used in intervals, which are located at the bottom of the table and in the centre.

In the formulas $S(x)$ and $B(x)$ differences are located horizontally, parallel to the x axis, so they are well used for central intervals.

In literature, you can find references that $S(x)$ formula is better to use when $U \leq 0.25$, but $B(x)$ – when $0.25 \leq U \leq 0.75$.

Additionally, the calculation using Newton's, Stirling's or Bessel's formulas are available in MathCAD, MathLab and Mathematica programs.

VII. PRACTICAL APPLICATIONS OF NEWTON'S POLYNOMIALS

Let us suppose that an arbitrary line is given. It is shown in Fig. 7.

It is necessary to calculate finite differences for the points, which are given in the table to create Newton's polynomial. But we must follow the condition that input points are distributed uniformly, i.e. x values are given with the same step

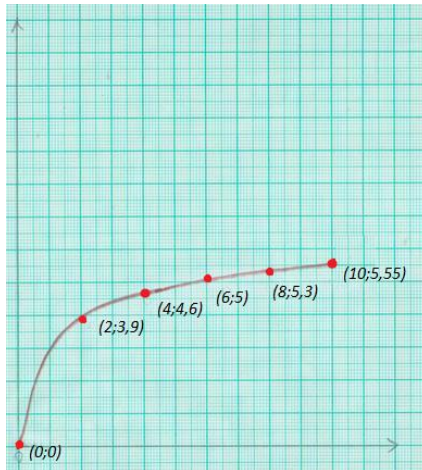


Fig. 7. Original line.

h (in this case $h = 2$). Input data for the given line are shown in Table II.

TABLE II
INPUT DATA

x	y
0	0
2	3.9
4	4.6
6	5
8	5.3
10	5.55

Applying formulas of the finite differences, as a result, forward differences are calculated (because x_0 was chosen at the beginning of the table) up to the fifth-order differences. (Table III).

Now, when the table of finite differences is obtained, the first Newton's formula can be used. As a result, fifth-degree polynomial is made:

$$P_5(x) = 0.000067x^5 - 0.0203x^4 + 0.2378x^3 + 1.3375x^2 + 1.6742x \quad (18)$$

Restored line repeats the form of original line (Fig. 8). Interpolation polynomial's accordance can be verified to points between interpolation nodes. For example, y coordinate for the point of the original line $x = 1$ is 2.95, but if it is calculated with the Newton's polynomial, coordinates of the point are $(1; 2.706)$.

Thus, error 2.44 mm appears. If we compare the results to other intermediate points, we obtain average error 0.0752 or 0.752 mm. It indicates that Newton's interpolation works quite accurately in this example.

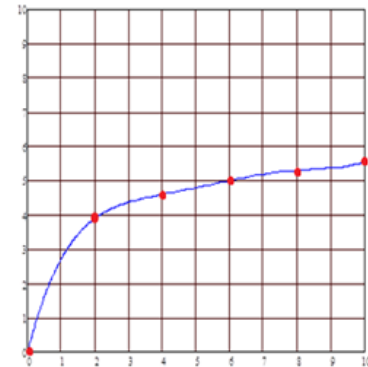


Fig. 8. Restored line.

To affirm that this classic interpolation method works well for all lines of welding seams, it is to be checked for other examples changing a form of original line, its length, count of inflection points etc. Newton's interpolation was applied to many different lines, and some regularity, which showed that this interpolation method was not universal, was noticed.

Let us suppose that a welding trajectory has a more complicated form (Fig. 9).

Interpolation nodes, which are painted in red (Fig. 9), are distributed evenly with the step $h = 2$. Thus, Newton's polynomial is obtained using finite differences.

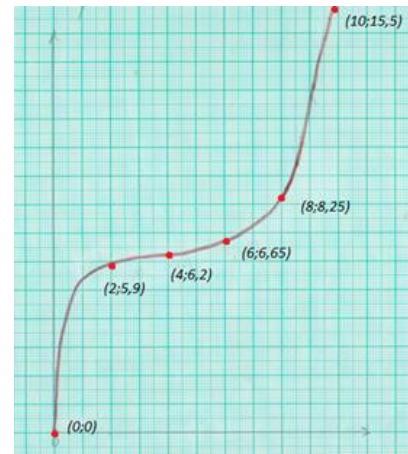


Fig. 9. Original line.

TABLE III
FINITE DIFFERENCES

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$
0	0	3.9	-3.2	2.9	-2.7	2.55
2	3.9	0.7	-0.3	0.2	-0.15	
4	4.6	0.4	-0.1	0.05		
6	5	0.3	-0.05			
8	5.3	0.25				
10	5.55					

Now we can plot a graph of the resulting function using the program MathCAD 14. As a result, we obtain the line shown in Fig. 10.

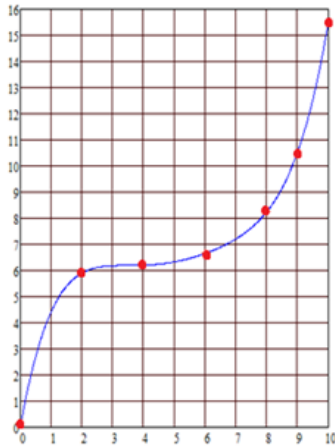


Fig. 10. Restored line.

Resulting line repeats the trajectory of the original line. Now we must check how much the line deviates from the original trajectory. For this purpose, we take points between interpolation nodes through each 0.5 mm. In order to conclude a possible deviation of the interpolation, it is necessary to calculate differences between the original coordinates of points and values, which are obtained with interpolation.

If we compare the results of interpolation and original coordinates, we obtain values, which range from 0 to 1.774, i.e. maximum line deviation from the necessary trajectory is more than one centimetre (17.74 mm). These results do not correspond to the required accuracy for arc welding processes. Doing experiments with a variety of such lines, we can conclude that the line deviates from the original trajectory mostly in the places, where the line is increasing or decreasing abruptly.

Now we need to check that the result will be obtained if we use Newton's interpolation and a welded trajectory has still a more complex form, for example, line with several inflection points.

Let us suppose that welding seam looks as follows:

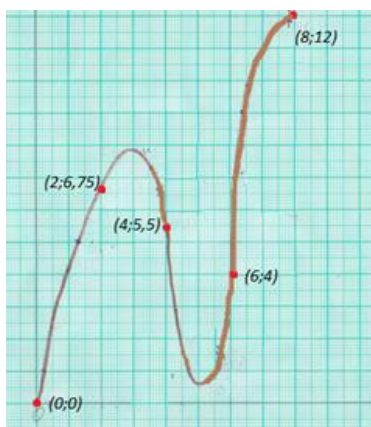


Fig. 11. Original line.

In this case, the interpolation step remains the same as in the previous lines, i.e. $h=2$. When we replace all necessary values in the first Newton's interpolation formula, we derive such a function:

$$f(x) = 0.0052x^4 + 0.099x^3 - 1.7396x^2 + 6.42x \quad (19)$$

Calculating this function in MathCAD-14, the following graph is obtained:

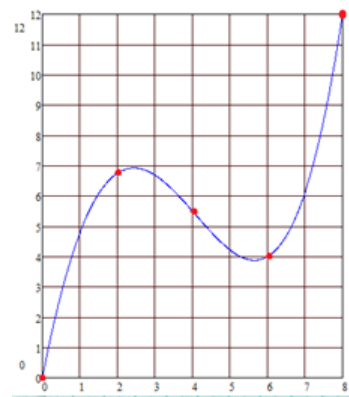


Fig. 12. Restored line.

The form of restored line (Fig. 12) does not match the original line (Fig. 11), so interpolation is incorrect. To improve it, let us take a twice smaller step between the interpolation nodes ($h = 1$) to clarify the points between interpolation nodes.

If we decrease the step between nodes, the number of nodes is increasing. Thus, as Prof. Iltiņš has noticed [10], the degree of interpolation polynomial also increases and degree of polynomial, which is larger than fifth, is not used in practice. As a result there are "extra" curves that do not correspond to the original trajectory. In Fig. 13 curves, which should not appear, are marked in green.

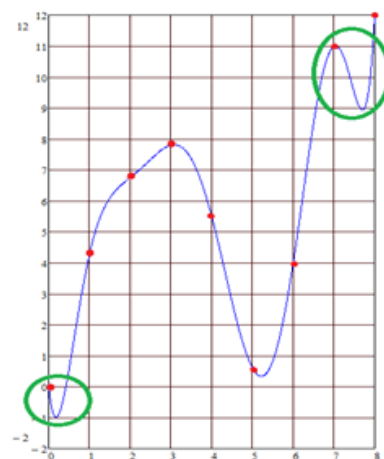


Fig. 13. Obtained line with "extra" curves.

If we take even a smaller step ($h = 0.5$), we obtain this line:

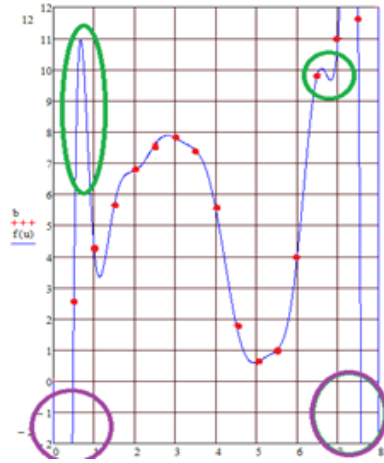


Fig. 14. Obtained line with “extra” curves.

From Fig. 14, it is clear that the graph of the interpolation polynomial considerably deviates from nodes and tends to infinity near to the end points of interpolation interval (in Fig. 14, regions are marked in purple). Also two curves, marked in green in Fig. 14, appear. They show that a deviation from a necessary trajectory is very large.

If we increase the number of nodes and degree of polynomial, the result becomes very corrupted, and interpolation error is very big. To use lower-degree Newton's polynomial for interpolation, we can divide input data table into several parts, i.e. into intervals, which are separated by inflection points as recommended [10].

Let us divide the original line into three parts: the first part – $x \in [0; 3]$; the second part – $x \in [3; 5]$; the third part – $x \in [5; 7.5]$. For each part it is necessary to create the corresponding Newton's polynomial.

As a result, three third-degree Newton's polynomials are obtained. Constructing graphs in one coordinate plane, the restored line (like in Fig. 15) is obtained.

Resulting line repeats the trajectory of original line. In Fig. 15, the function of interpolation polynomial of the first part is shown by the blue line, the second part – by the green dotted line, the third part – by the purple dotted line. Resulting line (combining three parts) is painted in red. This line has small deviations. For example, in the first line segment coordinate $y = 2.6$ corresponds to value $x = 0.5$, but inserting this x value in interpolation polynomial, we obtain $y = 2.362$. In this case, difference is 2.38 mm, which is acceptable. But there are segments, where deviation is too large. For example, if $x = 7.5$ then welding seam must go through the coordinate $y = 11.6$, but the result of the interpolation is a point with coordinates $(7.5; 12.85)$. It is a very big error (1.25 cm) for welding operations.

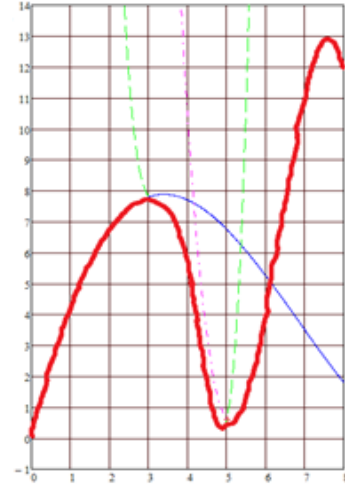


Fig. 15. Resulting line.

To liquidate this error and improve interpolation, it is possible to choose other nodes, other step between the interpolation nodes in the table. Also there is an opportunity to divide the line into smaller segments, in this way using several interpolation polynomials, but this operation complicates the process of computing. Therefore, Newton's interpolation is not effective for complicated lines, but it can be used when large details are welded and trajectories of welding seams can have a rougher error.

VIII. INTERPOLATION WITH THE CUBIC SPLINES

In practice, another method of interpolation is widely used – interpolation with the cubic splines. A piecewise-polynomial function, which is defined in the interval $[a, b]$ and has some continuous derivatives in this interval, is called spline.

We can imagine a concept of spline as a flexible ruler, which is stretched so that it crosses all points on which line is constructed [11]. In case of cubic splines, interpolation function between every couple of points is third-degree polynomial, which can be written as follows:

$$f(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + (x - x_{i-1})^3, \quad (20)$$

where $x_{i-1} \leq x \leq x_i$.

If function $f(x)$ is a continuous function, which is defined in the interval $[a, b]$, then it is divided in the interval:

$$a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b \quad (21)$$

Coefficients of the polynomial (20) in each interval can be determined by the conditions in nodes. It is clear that a polynomial value in the nodes must match the data in the table, consequently

$$y_{i-1} = f(x_{i-1}) = a_i, \quad (22)$$

where $1 \leq i \leq N$, and similarly

$$y_i = f(x_i) = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3, \quad (23)$$

where $h_i = x_i - x_{i-1}$ (h – step between interpolation nodes).

To construct a spline (according to its definition), additional conditions are required, i.e. continuous derivatives must be in the interpolation's interval $[a, b]$. Thus, we need to find the first and the second derivatives. The first derivative is:

$$f'(x) = b_i + 2c_i(x - x_{i-1}) + 3d_i(x - x_{i-1})^2, \quad (24)$$

where $x_{i-1} \leq x \leq x_i$.

The second derivative is:

$$f'' = 2c_i + 6d_i(x - x_{i-1}), \quad (25)$$

where $x_{i-1} \leq x \leq x_i$.

These derivatives must be continuous (to make a line smoother) in all points, including the interpolation nodes. Continuity conditions of the first and the second derivatives are

$$f'_i(x_i) = f'_{i+1}(x_i) \quad (26)$$

and

$$f''_i(x_i) = f''_{i+1}(x_i). \quad (27)$$

According to the continuity conditions (26) and (27), turning to the point x_i from the i -th to the $(i + 1)$ -th spline, and taking into account the first (24) and the second (25) derivative formulas, we obtain the following expressions:

$$b_{i+1} = b_i + 2c_i h_i + 3d_i h_i^2, \quad (28)$$

where $1 \leq i \leq N-1$, so the coefficient c in the $(i + 1)$ -th node is calculated by the formula:

$$c_{i+1} = c_i + 3d_i h_i, \quad (29)$$

where $1 \leq i \leq N-1$.

It is also necessary to set additional conditions at the line ending points, i.e. in the points x_0 and x_n . In general, these conditions depend on the concrete task. More often an assumption that the ends of a spline are free is applicable [12]. If the line is not "fixed" in the points, which are not within the interval $[x_0, x_n]$, then it is described by the straight line equation, i.e. by the first-degree polynomial. Thereby, in accordance with the continuity conditions of the spline second derivatives (27) in the ends of interval, we obtain the following relationships:

$$f'''(x_0) = c_1 = 0 \quad (30)$$

and

$$f'''(x_n) = c_n + 3d_n h_n = 0. \quad (31)$$

Expressions (22), (23), (28), (29), (30), (31) form the system of linear equations. Necessary spline coefficients a_i , b_i , c_i , d_i can be found solving this system. To solve this system, it is necessary to transform it in such a form that only coefficients c_i remain unknown. Coefficients a_i are obtained by the equation (22). It follows from equations (29), (30) and (31) that

$$d_i = \frac{c_{i+1} - c_i}{3h_i}, \quad (32)$$

where $1 \leq i \leq N-1$, and

$$d_n = -\frac{c_n}{3h_n}. \quad (33)$$

Now let us place expressions (32) and (33) in the formula of cubic polynomial (23), simultaneously replacing $a_i = y_{i-1}$ according to expression (22). As a result of all transformations, equations of b_i and b_n are obtained:

$$b_i = \frac{y_i - y_{i-1}}{h_i} - \frac{1}{3} h_i (c_{i+1} + 2c_i), \quad (34)$$

where $1 \leq i \leq N-1$, and

$$b_n = \frac{y_n - y_{n-1}}{h_n} - \frac{2}{3} h_n c_n. \quad (35)$$

Now, when equations (34) and (35) of the coefficients b_i and b_n are known, they can be used in equation (28) to get rid of these coefficients in the system of equations. Coefficient d_i can also be replaced according to expressions (32) and (33). Then only coefficients c_i remain in the system of equations. Finally, equations in such a form are obtained

$$h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} = 3\left(\frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-1}}\right), \quad (36)$$

where $2 \leq i \leq N$.

Two conditions must be taken into consideration: equation (30) and, if $i=n$, then, observing spline free end condition, this equality is valid:

$$c_{n+1} = 0. \quad (37)$$

In this way, $n-1$ equations like (36) together with conditions (30) and (37) create the system of linear equations to determine coefficients c_i . Coefficients d_i and b_i are computed when c_i is found from the formulas (32) and (34), coefficients a_i are equal to the function values in the interpolations nodes according to formula (22).

There are three unknown variables with sequential indices c_{i-1} , c_i , c_{i+1} in each equation (36). Therefore, matrix of the system of equations is tridiagonal, i.e. nonzero elements are located only in the main diagonal and in two nearby diagonals. This system is solved with the tridiagonal matrix algorithm or Thomas algorithm, which is a special kind of Gauss method [13].

Thus, function $s(x)$, which satisfies the following conditions, is called spline, which corresponds to the given function $f(x)$:

- 1) function $s(x)$ in each interval $[x_{i-1}, x_i]$, where $i = 1, 2, \dots, n$, is cubic polynomial;
- 2) the first and the second derivatives of function $s(x)$ are continuous in the interval $[a, b]$;
- 3) $s(x_i) = f(x_i)$, where $i = 0, 1, 2, \dots, n$.

The third condition is called interpolation condition. Spline, which is defined by the conditions 1)–3), is called interpolation cubic spline [14].

IX. PRACTICAL USES OF CUBIC SPLINES

Let us consider once again the example, where welding seam is a monotone line, which sharply increases (Fig. 16).

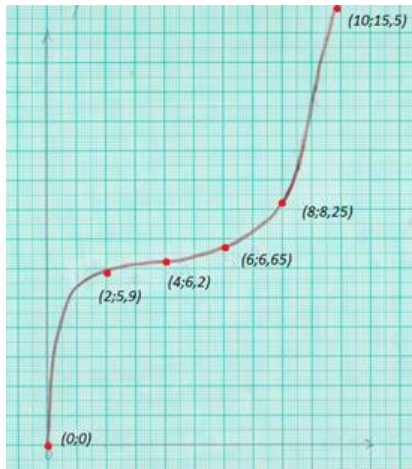


Fig. 16. Original line.

When using the first Newton's formula, the largest errors occur in places, where line coordinates are changing abruptly. Thus, in these places we should take more nodes to make the line go on the right trajectory. The input data for interpolation with the cubic splines are displayed in Table IV.

TABLE IV
INPUT DATA

x	y
0	0
0.15	2.7
0.5	4.5
1	5.4
2	5.9
4	6.2
6	6.65
8	8.25
8.5	9.5
9	11.7
9.5	13.8
10	15.5

As a result, using MathCAD 14 built-in functions, the resulting line is obtained (Fig. 17). To estimate the accuracy of interpolation, we can use mathematical indicators from probability theory: dispersion and mean square deviation. In total,

18 random points between the interpolation nodes were tested. These points are: (0.7;5), (0.9;5.3), (1.3;5.6), (1.7;5.8), (2.8;6.1), (3.6;6.2), (4.7;6.3), (5.5;6.5), (6.7;7), (7.5;7.6), (8.2;8.5), (8.4;9.1), (8.7;10.3), (8.9;11.2), (9.1;12.2), (9.3;13), (9.7;14.5), (9.9;15.3). Using the spline function, we can compute y values to chosen x coordinates.

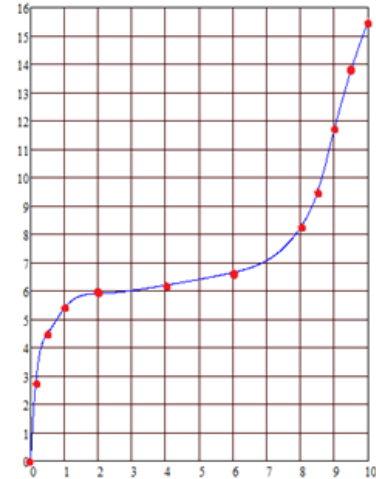


Fig. 17. Line that is obtained by cubic spline interpolation.

From the example of monotone line, it is clear that cubic splines give a more accurate result than Newton's polynomial.

Now let us suppose that the form of welding seam is a line with several increasing and decreasing intervals (Fig. 18).

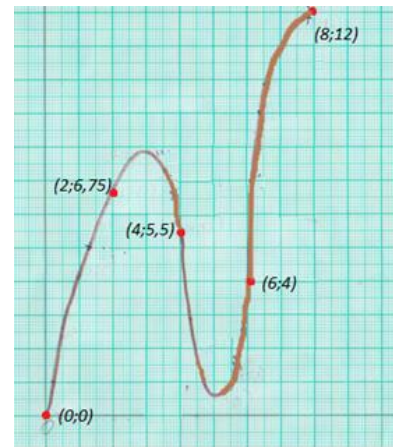


Fig. 18. Trajectory of welding seam.

It is essential to choose nodes so that in places, where y values are changing abruptly in a small interval of x values, nodes should be placed more often to make the line more accurate. Taking into account these conditions, line, which is displayed in Fig. 19, is obtained, where nodes are painted in red.

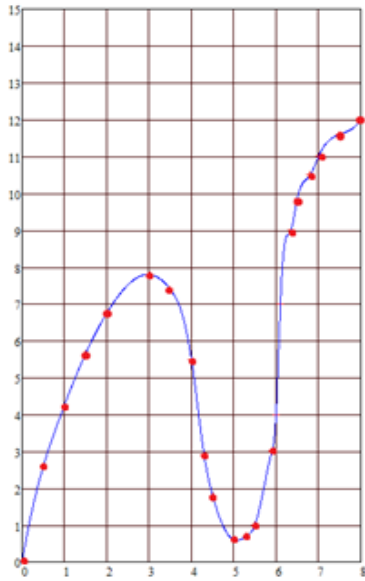


Fig. 19. The result of interpolation with cubic splines.

The figure shows that the line is a little bit misshapen in some places. We can take random points and check if obtained coordinates match the necessary trajectory to evaluate accuracy. It is necessary to take intermediate points between interpolation nodes and compare their original coordinates with the results of interpolation. If distances between the nodes are not too large, it is enough to take one or two points. Average difference between original data and the obtained data by the interpolation is 0.088 cm or 0.88 mm. We also should calculate additional values from the statistics to conclude about the data spread around the necessary seam trajectory. In this example, dispersion is 3.246×10^{-3} . In practice, mean square deviation is used for data processing. Mean square deviation is equal to 0.057 cm or 0.57 mm.

Resulting interpolation is implemented by the method of cubic splines and the industrial robot IRB 1600. The result is presented in the program RobotStudio (Fig. 20).

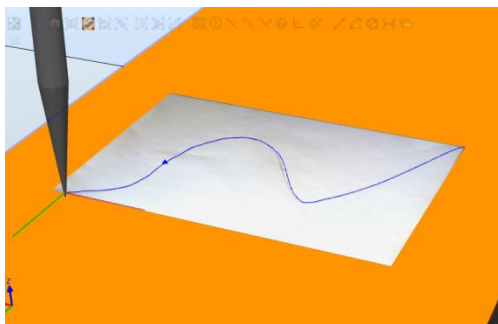


Fig. 20. Obtained moving path of the robot IRB 1600.

RobotStudio in-built function Point-to-Point can be used to evaluate the accuracy of line regeneration. This function shows the distance between two points. In Fig. 21, the enlarged trajectory of robot's moving (blue line) and an original line (black line) are shown. The distance between these two lines was measured with the function Point-to-Point in the

place, where the interpolated line deviates from the initial trajectory most of all.

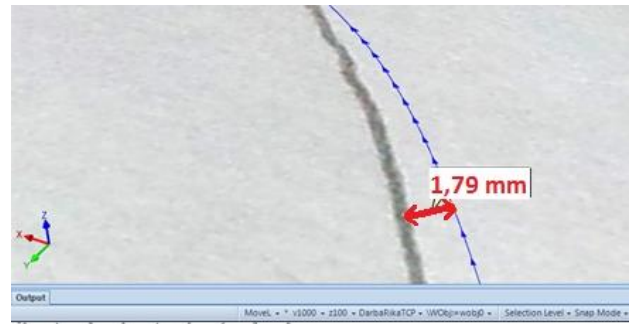


Fig. 21. The robot's movement deviation from the present line.

Therefore, the movement of robot's work tool along the points computed by the interpolation was visualised in the RobotStudio virtual area. Thus, imitation of welding process for the given line was implemented. The result shows that interpolation gives sufficiently precise line regeneration, and the robot repeats trajectory of the present line.

X. CONCLUSION

1. Having examined plenty of examples and comparing the results of two methods of interpolation, we can conclude that Newton's polynomial makes interpolation with significant inaccuracies.
2. It is necessary to do a lot of previously considered stages (to divide the line into several parts, to change interpolation step etc.) to make result more accurate.
3. Application of cubic splines makes the process of interpolation more convenient and easier. This method is universal for regeneration of different kinds of welding seams according to input points given in table form.
4. When trajectory of a necessary line is restored by the interpolation, this interpolation block can be used for the robot's contouring control.
5. Created interpolation block has also been used for the real industrial robot's IRB 1600 control, so it can be effectively used for arc welding operations.

REFERENCES

- [1] Jurevich, E.I. *Osnovy robototekhniki*. Leningrad: Mashinostroe-nie, 1985. 271 s.
- [2] Bronshtejn, I.N., Semendjaev, K.A. *Spravochnik dlja inzhenerov i uchashhihsja VTUZov*. Moskva: Nauka, Lejpcig: Tojbnier, 1981. 719 s.
- [3] Zviedris A. *Datorrealizācijas matemātiskās metodes*. Rīga: RTU, 2004. 76 lpp.
- [4] Kalitkin, N.N. *Chislennye metody*. Moskva: Nauka, 1978. 512 s.
- [5] Macokin, A.M., Sorokin, S.B. *Chislennye metody: Kurs lekcij*. Novosibirsk: Novosibirskij gosudarstvennyj institut, 2006. 132 s.
- [6] K.E. Atkinson, *An Introduction to Numerical Analysis*. John Wiley & Sons, Inc., New York, 1978, p. 587.
- [7] D.B. Small, J.M. Hosack, *Explorations in Calculus with a Computer Algebra System*. McGraw – Hill, USA, 1990, p. 226.
- [8] Vigodskis M. *Augstākās matemātikas rokasgrāmata*. Rīga: Liesma, 1968. 948 lpp.
- [9] P.J. Hartley, A.Wynn- Evans, *A Structured Introduction to Numerical Mathematics*. Stanley Thrones Ltd, England, 1979, p. 456.

- [10] Iltiņa, M., Iltiņš I., Veģere S. *Skaitliskās metodes un to realizācija ar datorprogrammu MATHEMATICA5*. Mācību līdzeklis. Rīga: RTU, 2012. 121 lpp.
- [11] J. H. Ahlberg, E.N. Nilson, J.L. Walsh, "The Theory of Splines and Their Applications", *Mathematics in Science and Engineering*, vol. 38. New York: Academic Press, 1967, p. 284.
- [12] Nikulin, E.A., *Kompjuterņaja geometrija i algoritmy mashinnoj grafiki*. Sankt-Peterburg: BHV-Peterburg, 2003. 560 s.
- [13] W.T. Lee, "Tridiagonal Matrices: Thomas Algorithm", *Scientific Computation*, University of Limerick, [Online]. Available: http://www3.ul.ie/wlee/ms6021_thomas.pdf
- [14] J. Rowell, *Mathematical Modelling with MathCAD*. Addison-Wesley Publishing Company, Inc. 1990, p. 157.

Oļesja Minejeva was born in 1993. She received the degree of B. sc. ing. in 2015 from Riga Technical University. She is a Master student at the Institute of Computer Control, Automation and Computer Engineering, Riga Technical University.
E-mail: o.minejeva@inbox.lv

Zigurds Markovics, Dr. habil. sc. ing., is a Professor (since 1993) of the Institute of Computer Control, Automation and Computer Engineering, Faculty of Computer Science and Information Technology, Riga Technical University.
He is the author of 148 scientific publications and 3 monographs.
His research interests include: computer control systems, artificial intelligence systems, robotics.
He is a member of the Latvian Association of Professors and Latvian Association of Scientists.
Address: 2, Daugavgrivas Str., Riga, LV-1007, Latvia.
E-mail: Zigurds.Markovics@rtu.lv

Oļesja Minejeva, Zigurds Markovičs. Interpolācijas metodes izvēle patvaļīgai loka metināšanas līknei

Prakse ir pierādījusi, ka modernajās tehnoloģijās interpolācijas metožu klasiskā mantojuma (Lagranža, Beseļa, Stirlingam Ņūtona u.c.) metodes ne vienmēr var nodrošināt vajadzīgo līknes restaurācijas precizitāti. Darbā analizētas interpolācijas metodes, sniegti ieteikumi precizitātes uzlabošanai, tajā skaitā pārejai uz kubisko splineu metodi. Pētījums pieskaņots metināšanas trajektorijas izveidošanai industriālā robota izpildījumā.

Олеся Минеева, Зигурдс Маркович. Выбор метода интерполяции для произвольной траектории дуговой сварки

Практическое применение классических методов интерполяции, таких, как методы Лагранжа, Бесселя, Стирлинга, Ньютона и др., не всегда обеспечивает достаточную точность реставрации кривых согласно требованиям современных технологий. В настоящей работе на основе анализа классического наследия методов интерполяции даны рекомендации по улучшению точности интерполяции при создании маршрута непрерывной сварки в робототехнике.