

DEVELOPMENT OF A MATHEMATICAL MODEL OF A TRIAXIAL INTERPOLATION FOR A CNC MACHINE IMPLEMENTED IN CONTROL SOFTWARE

Petar Panayotov¹, Goran Goranov²

¹ TU - Gabrovo, jimp@abv.bg

² TU - Gabrovo, ph.d.g.goranov@gmail.com

Abstract

The CNC machine digitally interpolates the commands and thus controls the cutting tool in the working area of the machine. The operating parameters of the CNC can be changed by loading programs. The present paper proposes the development of a new mathematical model for three-axis control by method of a single impulse direct calculation method. The biggest advantage of the proposed method is that it can describe absolutely any mechanics - there are no restrictions. It can also work when there is a degree of uncertainty more degrees of freedom.

Keywords: CNC control, mathematics model, synchronization, PC software

INTRODUCTION

The most common difficulty in reading and executing G-code generated by CAD-CAM software is synchronizing the motors to provide tool motion that is as close as possible to the theoretical set point. Any deviation from the set can damage the tool or the machine's motors. There are different types of syncs. A part of them performs synchronization of two or more engines by controlling all engines. This type of synchronization is used in CNC machines. The supply of pulses is simultaneous to all motors with a certain frequency or change in the modes of acceleration and deceleration of the motors. This control is performed by (motion) controllers or PC systems with appropriate software. According to the proposed methodology, software was developed for a personal computer running on the Windows operating system. In this type of synchronization, there are feedback systems and non-feedback systems. Depending on the dynamics and price of the machines, different control options may be preferred. In scientific research, there are many examples of how to optimize the process with more precise and smooth control of the CNC machine. [1,3,4,5,6]

THE MATH MODEL



Fig.1 Block diagram of the process and control machine

A method by direct calculation of a single pulse.

Through this method, the controller does not generate a frequency and instead calculates the occurrence of each pulse necessary to move along any of the three axes X,Y,Z. This method considers the implementation of an analytical model of a CNC machine. As mentioned, the G-code that is generated by CAD-CAM programs, gives information about the coordinates to which the tool must reach and the speed for this movement. Since, as we mentioned, the motors driving CNC machines are most often controlled with pulses, the linear movement of the tool along any of the axes is discrete. The discretization corresponds to the linear displacement step of the CNC machine and it is different depending on the mechanics and configuration of the machine.

$\Delta X_{step} [mm]$ - Linear displacement for one pulse along the X axis
 $\Delta Y_{step} [mm]$ - Linear displacement for one pulse along the Y axis
 $\Delta Z_{step} [mm]$ - Linear displacement for one pulse along the Z axis

Due to the presence of such a discretization, the tool can never follow the "ideal" line in space specified by the G-code. If it is taken into account that the current position of the tool is the zero point, then at a precise moment in time the tool can move to one of the following points:

- $(0,0,0)$ $(\Delta X_{step},0,0)$ $(-\Delta X_{step},0,0)$
- $(0,\Delta Y_{step},0)$ $(\Delta X_{step},\Delta Y_{step},0)$ $(-\Delta X_{step},\Delta Y_{step},0)$
- $(0,-\Delta Y_{step},0)$ $(\Delta X_{step},-\Delta Y_{step},0)$
- $(-\Delta X_{step},-\Delta Y_{step},0)$
- $(0,0,\Delta Z_{step})$ $(\Delta X_{step},0,\Delta Z_{step})$
- $(-\Delta X_{step},0,\Delta Z_{step})$
- $(0,\Delta Y_{step},\Delta Z_{step})$ $(\Delta X_{step},\Delta Y_{step},\Delta Z_{step})$
- $(-\Delta X_{step},\Delta Y_{step},\Delta Z_{step})$
- $(0,-\Delta Y_{step},\Delta Z_{step})$ $(\Delta X_{step},-\Delta Y_{step},\Delta Z_{step})$
- $(-\Delta X_{step},-\Delta Y_{step},\Delta Z_{step})$
- $(0,0,-\Delta Z_{step})$ $(\Delta X_{step},0,-\Delta Z_{step})$
- $(-\Delta X_{step},0,-\Delta Z_{step})$
- $(0,\Delta Y_{step},-\Delta Z_{step})$ $(\Delta X_{step},\Delta Y_{step},-\Delta Z_{step})$
- $(-\Delta X_{step},\Delta Y_{step},-\Delta Z_{step})$
- $(0,-\Delta Y_{step},-\Delta Z_{step})$ $(\Delta X_{step},-\Delta Y_{step},-\Delta Z_{step})$
- $(-\Delta X_{step},-\Delta Y_{step},-\Delta Z_{step})$

The tool move pattern looks like this-fig.1

Building the model in this way, at a precise moment in time, the tool can move to one of the marked points from 1 to 26, which we will call discrete points. These are, in fact, all combinations of the possibilities of supplying pulses to the three axes at the same moment in time $t + \Delta t$.

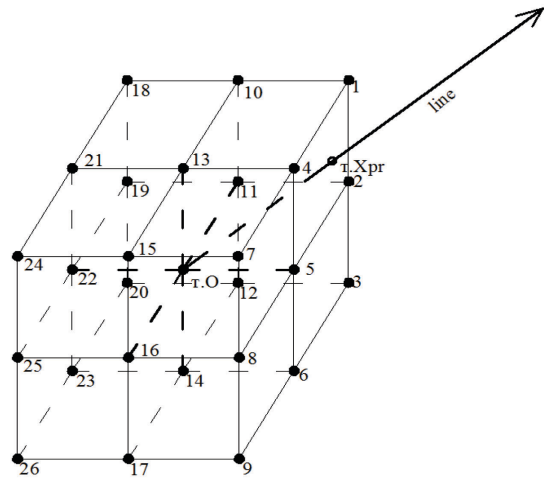


Fig.2. Possible displacement points for one pulse

Following the end point that the tool must reach and starting from p.O, the coordinates of the two points form a straight line that can be described by its equation [2]:

$$\frac{x - x_0}{x_{end} - x_0} = \frac{y - y_0}{y_{end} - y_0} = \frac{z - z_0}{z_{end} - z_0}, \quad (1)$$

Since p.O is taken as the origin of the OXYZ coordinate system, therefore $x_0 = 0, y_0 = 0, z_0 = 0$

Equation (1) takes the form:

$$\frac{x}{x_{end}} = \frac{y}{y_{end}} = \frac{z}{z_{end}}, \quad (2)$$

The equation of the line can also be represented in a parametric form:

$$\begin{cases} x = \lambda.t \\ y = \mu.t, \quad (3) \text{ where} \\ z = \nu.t \end{cases}$$

A quantity $t \in (-\infty, \infty)$ is called a parameter.

$$\lambda, \mu, \nu \text{ are zero vector coordinates } l = \begin{bmatrix} \lambda \\ \mu \\ \nu \end{bmatrix}$$

Because the line passes through p.O, from where $\lambda = x_{end}, \mu = y_{end}, \nu = z_{end}$ the parametric equation can be represented in the form:

$$\begin{cases} x = x_{end} \cdot t \\ y = y_{end} \cdot t \\ z = z_{end} \cdot t \end{cases}, \quad (4)$$

The space around the point O that can be reached by the tool for one pulse, forms a parallelepiped or if the displacement steps are equal, forms a cube, the sides of which form six planes. Since every plane is defined by three non-collinear points, it can be said that every plane can be defined by the following points:

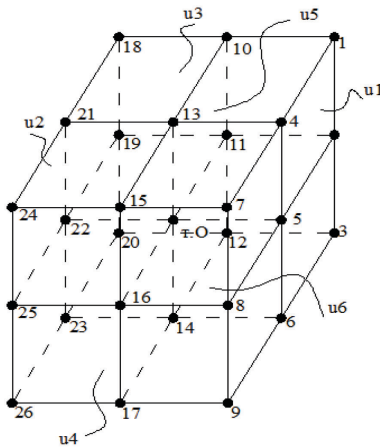


Fig.3. Planes formed by the possible displacement points

- $v_1 \rightarrow p.5, p.1, p.7$
- $v_2 \rightarrow p.22, p.18, p.24$
- $v_3 \rightarrow p.11, p.1, p.18$
- $v_4 \rightarrow p.16, p.24, p.7$
- $v_5 \rightarrow p.13, p.24, p.7$
- $v_6 \rightarrow p.14, p.26, p.9$

The equation of a plane passing through three points, has the following form:

$$\begin{vmatrix} x - x_1 & y - y_1 & z - z_1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{vmatrix} = 0, \quad (5)$$

Since each of the formed planes is parallel to at least two of the coordinate axes, it is possible to use a plane segment equation having the form:

$$\frac{x}{m} + \frac{y}{n} + \frac{z}{p} = 1, \quad (6)$$

where

m, n, p - sections of the plane along the x, y, z axis respectively.

Taking the segments of the planes, all the equations of the six discrete planes formed can be written:

$$v_1 \Rightarrow \frac{x}{\Delta X_{step}} = 1, \quad (7)$$

$$v_2 \Rightarrow \frac{x}{-\Delta X_{step}} = 1, \quad (8)$$

$$v_3 \Rightarrow \frac{y}{\Delta Y_{step}} = 1, \quad (9)$$

$$v_4 \Rightarrow \frac{y}{-\Delta Y_{step}} = 1, \quad (10)$$

$$v_5 \Rightarrow \frac{z}{\Delta Z_{step}} = 1, \quad (11)$$

$$v_6 \Rightarrow \frac{z}{-\Delta Z_{step}} = 1, \quad (12)$$

To find the intersection of the line with each of the planes, the parametric equation of the line (4) and the equations of the six planes.

$$v_1 \Rightarrow \Delta X_{step} = x_{end} \cdot t \Rightarrow t = \frac{\Delta X_{step}}{x_{end}},$$

$$x_1 = \Delta X_{step}, y_1 = y_{end} \cdot t, z_1 = z_{end} \cdot t$$

$$v_2 \Rightarrow -\Delta X_{step} = x_{end} \cdot t \Rightarrow t = \frac{-\Delta X_{step}}{x_{end}},$$

$$x_2 = -\Delta X_{step}, y_2 = y_{end} \cdot t, z_2 = z_{end} \cdot t$$

$$v_3 \Rightarrow \Delta Y_{step} = y_{end} \cdot t \Rightarrow t = \frac{\Delta Y_{step}}{y_{end}},$$

$$x_3 = x_{end} \cdot t, y_3 = \Delta Y_{step}, z_3 = z_{end} \cdot t$$

$$v_4 \Rightarrow -\Delta Y_{step} = y_{end} \cdot t \Rightarrow t = \frac{-\Delta Y_{step}}{y_{end}},$$

$$x_4 = x_{end} \cdot t, y_4 = -\Delta Y_{step}, z_4 = z_{end} \cdot t$$

$$v_5 \Rightarrow \Delta Z_{step} = z_{end} \cdot t \Rightarrow t = \frac{\Delta Z_{step}}{z_{end}},$$

$$x_5 = x_{end} \cdot t, y_5 = y_{end} \cdot t, z_5 = \Delta Z_{step}$$

$$v_6 \Rightarrow -\Delta Z_{step} = z_{end} \cdot t \Rightarrow t = \frac{-\Delta Z_{step}}{z_{end}},$$

$$,x_6 = x_{end} \cdot t, y_6 = y_{end} \cdot t, z_6 = -\Delta Z_{step}$$

After finding the intersection of the line with each of the planes, the coordinates of 6 intersection points are found. If the line passes through the corners of the parallelepiped, then all six points will belong to its surface, and three by three will coincide. If the line passes through the edge of the parallelepiped, 4 of the points will belong to the surface of the cube as two by two will coincide and the other two will be outside the volume, which means that the tool cannot reach them in one pulse. If the line does not pass through an edge or corner of the parallelepiped, only two of the puncture points will belong to the surface, the remaining 4 will be outside the volume or outside the range of displacement for one pulse. Since the tool performs a discrete movement, it cannot be positioned at the puncture point. Therefore, it is necessary to find the nearest discrete point to the puncture point. Since the line has no direction, it can be considered as a zero vector

$$l = \begin{bmatrix} \lambda \\ \mu \\ \nu \end{bmatrix}$$

In order to determine the directions to each discrete point, it is necessary to find the vectors passing through each of the discrete points or all the discrete vectors formed by OT and the discrete points belonging to the surface of the parallelepiped.

$$p.1 \rightarrow l_1 = \begin{bmatrix} \Delta X_{step} \\ \Delta Y_{step} \\ \Delta Z_{step} \end{bmatrix}$$

$$p.2 \rightarrow l_2 = \begin{bmatrix} \Delta X_{step} \\ \Delta Y_{step} \\ 0 \end{bmatrix}$$

etc. for 26 points lying on the surface of the parallelepiped.

The zero vector will coincide in direction with a number of discrete vectors defined by the discrete points. To determine which discrete vector to move the tool along, it is necessary to compare the zero vector with all discrete vectors that match the direction of the zero vector.

It is necessary to find the magnitude of the zero vector and the magnitude of the vector

from the origin of the coordinate system to the point of the puncture, which has the form:

$$l_{len} = \sqrt{\lambda^2 + \mu^2 + \nu^2}$$

$$l_{len} = \sqrt{x_{pr}^2 + y_{pr}^2 + z_{pr}^2}, \quad (13)$$

The magnitude of the discrete vectors has the form:

$$l_{len1} = \sqrt{\Delta X_{step}^2 + \Delta Y_{step}^2 + \Delta Z_{step}^2}$$

$$l_{len2} = \sqrt{\Delta X_{step}^2 + \Delta Y_{step}^2 + 0}$$

etc. for all discrete vectors.

This discrete vector that is closest to the punch vector is the closest discrete tool displacement to the actual path. This is how it is determined to which of the motor drivers to send an impulse to carry out the necessary displacement. When the puncture hits the middle of the quarter of a given plane, the magnitude of the puncture vector will have the same difference as four of the discrete vectors. In this case, no single discrete vector can be determined, so any matching vector can be chosen.

Since all calculations are made for a coordinate system with origin t.0, after moving the tool to one of the discrete points, it is necessary to translate the coordinate system to the new point. This will change the values of x_{end} , y_{end} , z_{end} . The zero vector is different after each coordinate system translation.

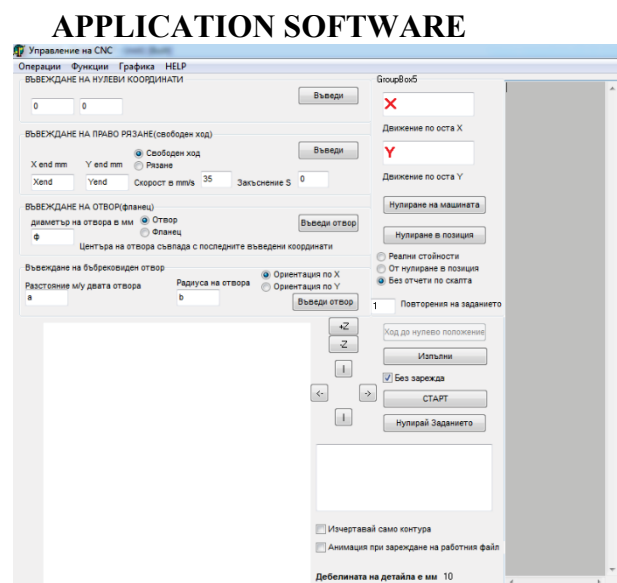


Fig.4. Software implemented on the basis of a method by direct calculation of a single impulse.

Figures 4 and 5 show the developed software according to the proposed methodology. The software works under Windows operating system. The buttons “I” and “->” serve to move the machine in real time along its axis X and Y.

The developed software is implemented on a CNC machine for plasma metal cutting. Due to the type of operation of the machine, the software implements the presented mathematical model along two axes X and Y, since the movements are only in the plane. The software generates the necessary G-code and executes it.

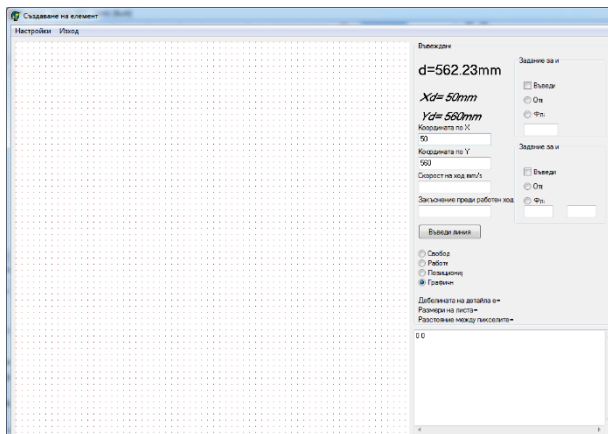


Fig.5. Create an item

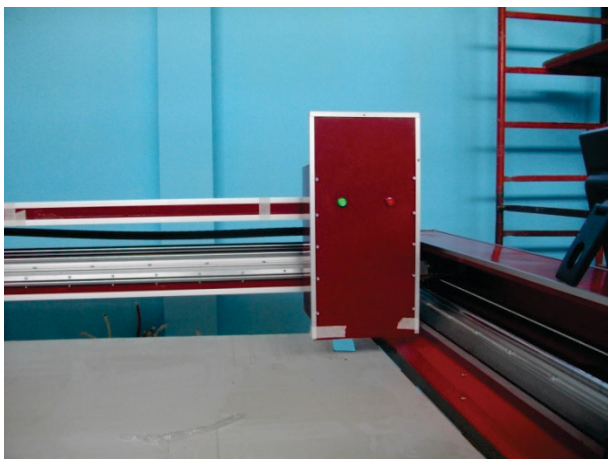


Fig.6. Plasma cutting machine

The fig.6 shows a plasma cutting machine using control software by the direct calculation method of a single pulse.

CONCLUSION

The presented mathematical model can also be applied to machines with different types of

mechanics. For machines performing circular movements or a combination of linear and circular axes. The advantage of the method is very good synchronization of all engines and maximum approximation to the ideal tractor. Through the model, trajectories of functions can also be realized, which can be set through their equations. This can result in minimizing the generated G-code.

The need for large computing power is a disadvantage of the model. Because each pulse is calculated immediately before its occurrence. If a higher generated frequency is needed, the calculations can be done before the pulses are applied. This generates an array of binary code that is executed at the appropriate frequency.

REFERENCE

- [1] Prodanov, P. I. M. Simeonov. "Control system of stepper motors in a machine with digital program control". Unitech 2011, November 18 – 19, 2011, Gabrovo, p.I-230 – p.I.234.
- [2] Georgi Kamenarov. REFERENCE BOOK - HIGHER MATHEMATICS
- [3] Mikhov M., M. Zhilevski, Analysis of a Multi-Coordinate Drive System Aiming at Performance Improvement, Proceedings of the International Conference “Research and Development in Mechanical Industry”, Vol. 2, pp. 1102-1107, Vrnjacka Banja, Serbia, 2012, ISBN 978-86-6075-037-4.
- [4] Mikhov M., M. Zhilevski, A. Spiridonov, Modeling and Performance Analysis of a Spindle Electric Drive with Adaptive Speed Control, Journal Proceedings in Manufacturing Systems, Vol. 7, No. 3, pp. 153-158, Bucharest, Romania, 2012, ISSN 2067-9238.
- [5] Syed Hassaan Ahmed; Nauman Razzaq; Zaman Malik; Umair Qadeer; Iqra Sarfraz; Aasim Sharif, Design & fabrication of MATLAB based solar powered CNC machine, 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE)2017, Electronic ISBN:978-1-5386-0484-7
- [6] Yusri Yusof; Kamran Latif, Development of new open soft-CNC system, International Conference on Computer, Communications, and Control Technology (I4CT) 2015, Electronic ISBN:978-1-4799-7952-3
- [7] He Sicong; Huang Dagui; Feng Daiwei; Huang Xiaogang, IEEE International Conference on Mechatronics and Automation 2008, Print ISBN:978-1-4244-2631-7.