Motion Profile Control Algorithm and Corner Smoothing Technique for Trajectory Optimization of High-Precision Processing

Kseniia V. Zimenko, Maxim Ya. Afanasev,
Anastasiya A. Krylova, Sergey A. Shorokhov, Yuri V. Fedosov
ITMO University, St. Petersburg, Russia
zksenia@yahoo.com, amax@niuitmo.ru, {ananasn94, stratumxspb}@gmail.com, yf01@yandex.ru

***Abstract*—Processing accuracy in instrumental technology has always been of great importance. Producers of Computer Numeric Control (CNC) systems are constantly looking for novel solutions to achieve higher velocities and precision. However, most of the produced software algorithms are inaccessible to the general public. Hence the task to develop sufficient open source software arises. This paper aims to create a trajectory optimization algorithm, including feed rate control and a corner smoothing technique, which will allow effective high-speed and high-precision processing. It is intended to standardize the algorithm for application with both stepper and servo motor driven machines. The developed motion planning method is based on a cosine function to attain a smooth change of velocity that allows for vibration reduction. To achieve smooth corner processing, spline curves are applied to adjust the size and shape of a fillet and thus satisfy the required tolerance and maintain high velocities. The resulting algorithm is programmed and simulation tests are carried out. The final algorithm shows a smooth transition of velocities, which leads to vibration reduction and consequently to minimization of machining error. In corner smoothing the use of parametric curves demonstrates the ability to vary tolerance. As a result, a sufficient motion control algorithm is developed and can be used in CNC software.**

I. Introduction

Since the requirements for accuracy are continuously growing nowadays in the field of instrumentation technology, the main direction of development of CNC systems is increasing effectiveness of high-speed and high-precision processing. This can be achieved with the improvement of software algorithms, used for converting the given part of the program into tool movements.

The process of analyzing an input part program consists of several stages. It ranges from program interpretation, where all the given data like feed rate and trajectory coordinates is extracted, to acceleration/deceleration control and then to interpolation procedure. Interpolation process includes axis movement generation from block data given by an interpreter. The final step is position control, which main aim of which is to minimize the difference between interpolation positions and real positions received from an encoder [1]. The stage of acceleration/deceleration (acc/dec) control, which is applied to create a velocity profile and smooth tool movements.

Together with interpolation most reflects the accuracy of processing and thus it is important to apply effective algorithms that lead to minimal machining error. However, most of feed rate control methods in use, such as linear motion profile generation, for instance, result in high jerk and vibration, causing the degradation of the surface quality obtained during processing.

Apart from that, some methods that allow for an increase in processing speed lead to a divergence between programmed and real instrument path, namely machining error. Calculations show that this inaccuracy is directly proportional to feed rate. Hence, these methods lead to low accuracy during high-speed processing.

On the other hand, most of the CNC systems offered on the modern market provide software which allows high-precision processing even at high velocities. Yet these solutions are part of proprietary code that is for the most part inaccessible to third-party developers. Hence, the goal of this work is to create and implement an open source algorithm. Another task is handling the processing at trajectory corners. There are two main types of corner machining: full stop mode (G61) and continuous mode (G64) [1], [2].

The first mode means that the machine will follow the programmed path exactly, stopping at the corner. The second type results in a fillet, as it is impossible to create a sharp corner without reducing processing speed to zero. Both of modes result in machining error. In the first case the obtained surface quality decreases due to the high jerk caused by sudden changes of velocity. In the case of continuous mode, there is a discrepancy between the given and the resulting trajectory, as a sharp corner is replaced with a curve. The goal is to reduce error caused by velocity change and to maintain the required trajectory accuracy by regulating the size and shape of the resulting fillet.

To get a small enough fillet that corresponds to the demanded precision it is important to reduce the machining speed, compared to neighboring linear blocks. The problem is to find a way of calculating the corner velocity based on the data received from the corresponding blocks. One method, for example, suggests comparing the allowable acceleration on each axis to the actual velocity changes at the corner followed by a speed reduction if this value is exceeded [1]. However, the resulting feed rate does not depend on the size and shape of processed fillet. Another aim of this study is to create an algorithm universal enough for use on both stepper and servo motor driven machines. Most of the methods presented in recent years are specifically designed for servo driven CNC systems. This type of motor can achieve higher precision processing, as it is intended for applications where rapid acceleration and deceleration occur [3].

However, servo motors require more complicated controllers which make their price rise significantly. On the other hand, stepper motors have a lower price compared to servo motors and still allow for the attainment of sufficient machining accuracy. Low-end and middle-end machines are driven by stepper motors [4].

This study aims to develop a novel trajectory optimization algorithm, including a low jerk acc/dec control. The main criterion for trajectory optimization is the minimization of machining error during high-speed processing. The selected feed rate control method is based on applying the cosine function to achieve a smoother transition of velocity and avoid surplus vibrations [5].

Because for the most part programs consist of a large number of blocks, it is crucial to build a velocity profile with the use of a look-ahead algorithm, which takes into account the data of neighboring blocks while executing. It helps to avoid unnecessary decelerations on the edges of blocks and smooths the resulting motion profile. The look-ahead algorithm is used in the proposed method of acc/dec control.

The proposed algorithm also includes fillet generation at trajectory corners to avoid unnecessary decelerations. The ability to adjust the size and shape of said fillet is also taken into account, as it is important to satisfy the required processing accuracy. The corner smoothing technique includes using b-splines or parametric curves. This type of corner processing requires several values that can determine the size and shape of the generated blend. Two types of curves are compared in this paper: those with three and those with six control points. This paper also applies a method of feed rate reduction at corners which avoids exceeding the allowable acceleration value. This paper presents a description of the methodology, implementation and simulation results.

The paper is organized as follows. Section II is dedicated to an overview of related work. In Section III the reader can find a description and the results of the acc/dec control algorithm simulation, with examples provided. Section IV describes the proposed corner smoothing algorithm. Examples of fillet compositions are shown. A comparison between splines with three and six control points given. In Section V the example of the proposed algorithm execution is depicted. The obtained results are presented graphically. The final section is dedicated to an analysis of the results.

II. Related Work

As creating efficient acc/dec control is crucial to achieving smooth and precise tool motion, there have been a lot of algorithms developed. The linear or trapezoidal feed rate scheduling method [6–8] has the simplest implementations. On the other hand, this methodology results in large amount of vibrations and jerk and thus lower accuracy. This is caused by the abrupt change of velocity.

There are different ways to smooth the feed rate profile by applying s-shape [9], Sine-shape or polynomial algorithms. Hepeng Ni et al. [10] proposed an optimized s-shape algorithm of feed rate scheduling with round-off error compensation. Xu Du et al. [11] used the same type of feed rate profile and developed the s-shape motion control schedule. His algorithm was designed specifically for NURBS curve interpolation. Another s-shaped speed control schedule was proposed by Lin Wang et al. [12]. A look ahead algorithm was also applied to prevent surplus decelerations. Huaizhong Li et al. [5] presented the acc/dec control algorithm, which reduces jerk and vibration residuals. Their approach is based on the application of a cosine function to the acceleration calculation. It smoothes velocity change and thus leads to minimal vibrations. The algorithm allows real-time motion profile generation. However, there was no look ahead algorithm presented, and it is expected that block analysis is executed independently. This methodology is used in the present paper to smooth the acc/dec motion phases. Qingzhen Bi et al. [13] proposed a continuous-curvature smoothing algorithm, which includes the usage of a Bezier curve to blend the junctions of the linear tool path. A Bezier blend, which consists of three control points, is applied in the algorithm. It is possible to approximate the curve under a predefined tolerance. So it is possible to change the size of the blend to satisfy the accuracy requirements. However, due to the number of control points it is impossible to change the shape of the resulting curve. The present paper sets a goal of altering the shape of the curve and therefore requires a greater number of control points.

Burak Sencer et al. [14] demonstrated a way to blend two consecutive linear segments with a Bezier curve, which has six control points. With this method it is possible to adjust both the size and shape of the curve. However, in this case two values are required to specify the fillet, which leads to more complicated calculations. Apart from that the corner-velocity-finding algorithm presented by the researchers requires the use of curvature value, which leads to complicated calculations. The present paper uses a Bezier curve with six control points to adapt the resulting fillet to processing requirements; in other words, to change the size and shape under a predefined tolerance. However, the present work has a goal of simplifying the corner feed rate calculation. Any Bezier curve with a greater number of control points [15], [16] requires more than two values to determine the size and shape of the fillet. It will lead to more complicated calculations for users.

Therefore, after the related work analysis the task remains to combine the low jerk motion control, look-ahead algorithm and corner velocity generation with a corner smoothing technique to further minimize the velocity discontinuities and thereby reduce the discrepancy between the programmed and the real trajectories during high-speed processing.

III. Motion profile generation

As the main aim of the proposed motion control algorithm is to minimize jerk and vibrations, induced by high speed change, it is critical to ensure the smooth transition of acceleration. The selected method is based on applying the cosine function to achieve a smooth transition of velocity and avoid surplus vibrations of motion [5]. The velocity profile can be divided into three stages: acceleration $t_{inter} \in [t_0, t_1]$, constant speed $t_{inter} \in [t_1, t_2]$ and deceleration $t_{inter} \in [t_2, t_3]$. The acceleration value at each interpolation time can be calculated by applying the following equations. The acceleration value during the speed escalating stage is obtained as follows, Eq. (1):

$$\begin{aligned} a &= \frac{A}{2}\cdot\left(1-\cos\left(\frac{2\pi}{T_{acc}}\cdot t\right)\right) \\ T_{acc} &= t_1 - t_0 = \frac{2F}{A} \end{aligned} \tag{1}$$

where F is the feed rate and A is the allowable acceleration. During the constant speed period acceleration equals zero. And in the deceleration stage it equals, Eq. (2):

$$\begin{aligned} a &= -\frac{D}{2}\cdot\left(1-\cos\left(\frac{2\pi}{T_{dec}}\cdot (t-t_2)\right)\right) \\ T_{dec} &= t_3 - t_2 = \frac{2F}{D} \end{aligned} \tag{2}$$

where D is the allowable deceleration. The speed and displacement values can be obtained by integration. The resulting acceleration and speed profiles are shown in Fig. 1.

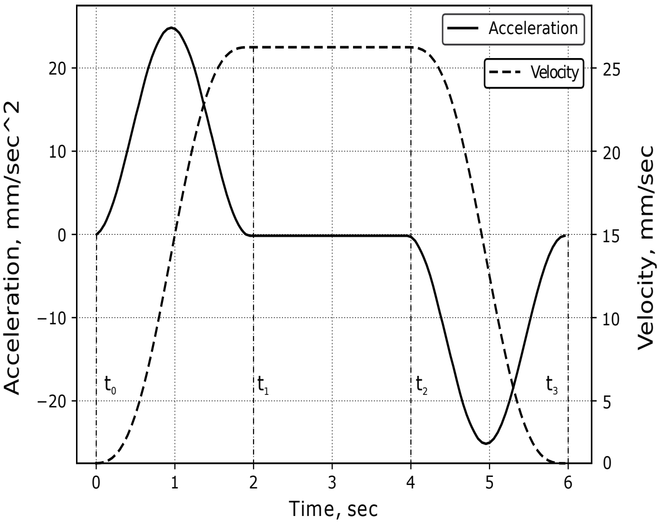
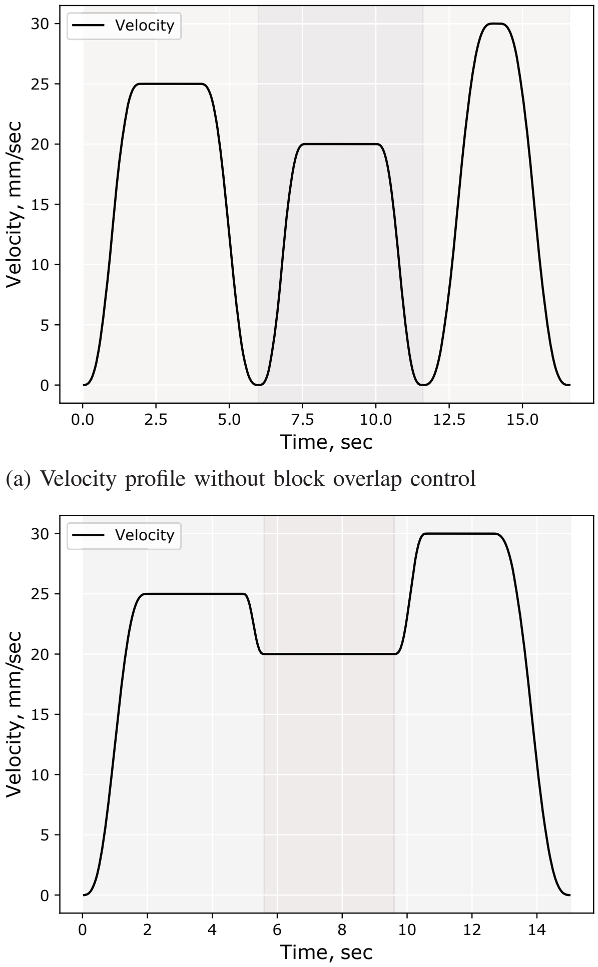


Fig. 1. Acceleration and velocity profiles.

The proposed method is best suited for individual blocks; however, a typical control program consists of a large number of blocks. Consequently, when used on several segments as a whole, the algorithm returns a feed profile like that shown in Fig. 2a. Evidently, the reductions of feed rate on the border of blocks are unnecessary and lead to exceeded vibrations and jerk resulting in a lower quality of obtained surface.

Hence, the special look-ahead algorithm is built-in. While generating the velocity profile it considers the consecutive segments and adjusts the speed profile according to the maximum reachable feed rate value on these segments. Thus, feed profile like the one depicted in Fig. 2b can be obtained using this technique.



(a) Velocity profile without block overlap control

(b) Velocity profile with block overlap control

Fig. 2. The result of look-ahead algorithm application

IV. Corner smoothing algorithm

The only way to create an accurate corner is to stop the instrument completely. However, this results in intermittent motion of the tool that leads to a deterioration in the quality of the obtained surfaces and an increase in machining time. To raise the quality of the processing result and velocity maintenance it is necessary to use smoothing algorithms. There are two ways of implementing fillets using the corner processing mode: using arcs or splines, or parametric curves. Changing the angle to a circular arc is easy in implementation. However, there is no opportunity for the user to change the form and size of the arc.

To implement smoothing at corners with the ability to adjust the parameters of the fillet the study suggests the algorithm which uses splines, a Bezier curve in particular. As all smoothing methods result in machining inaccuracy, the advantage of said algorithm lies in its error value control. And the less deceleration there is at the corner, the bigger the fillet and consequently the processing inaccuracy will be. And in each case it is possible to regulate the correlation between the corner speed and error. However, for precise spline formation parameters are needed in a quantity corresponding to the number of control points used.

Two adjacent linear segments are blended with a Bezier curve. The resulting blend in comparison to the initial trajectory is shown in Fig. 3. The simplest method of Bezier curve formation is to use three control points P_{start}, P_{trans} and P_{end}. Implementation requires one parameter: transition length L_t depicted in Fig. 3. In this case, it is possible to adjust the size of the curve by changing the L_t parameter which functions as scaling factor. However, the shape of the resulting blend is constant, which limits processing possibilities.

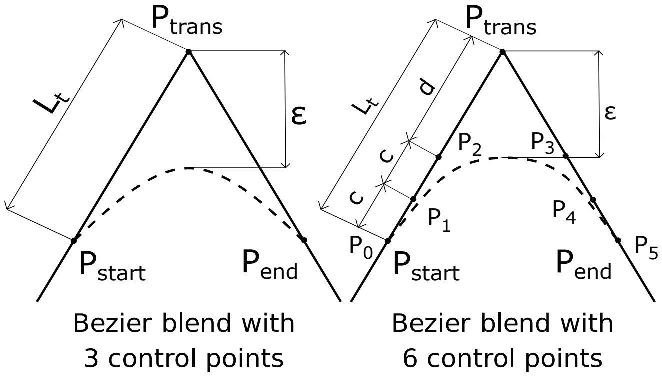


Fig. 3. Bezier curve

For example, two fillets have been obtained. For the first curve, shown in Fig. 4a, the L_t parameter is set to 2 mm, and for the second, depicted in Fig. 4b, the L_t parameter equals 5 mm. Evidently, the size of the blend has changed; however, the curvature stays the same. This is shown in Fig. 4c.

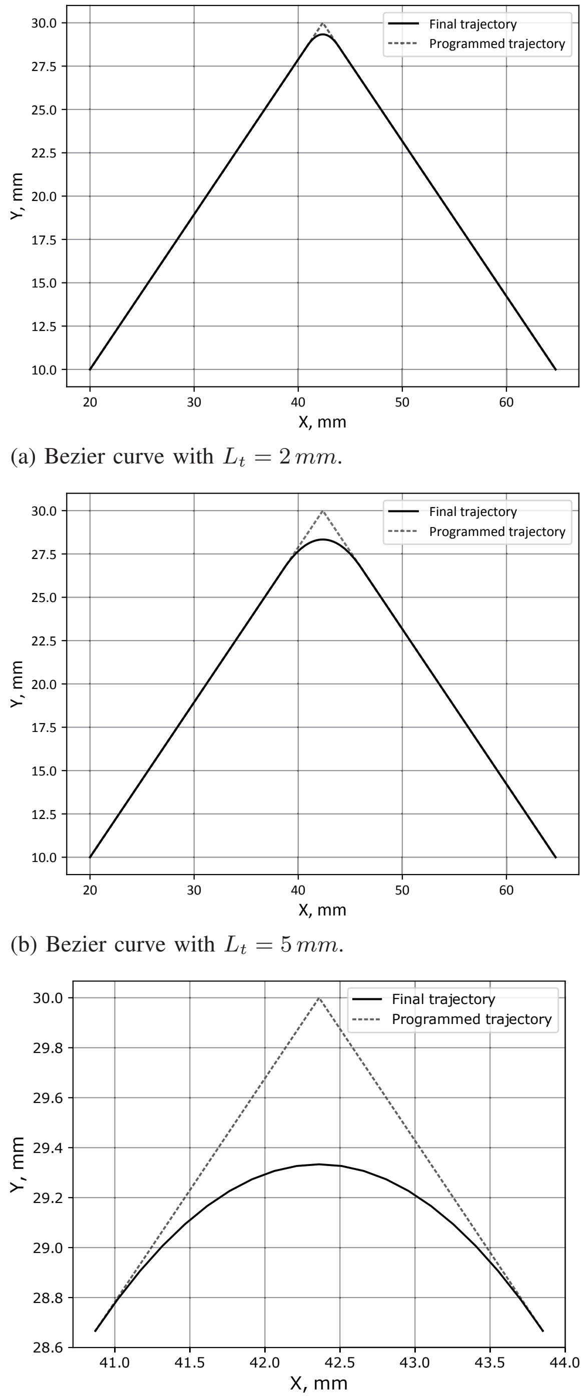
Recent studies, including Burak Sencer's [14], suggest using six control points for the Bezier blend. The spline segment can be defined in Cartesian coordinates as, Eq. (3):

$$B(t) = \begin{bmatrix} B_x(u) \\ B_y(u) \\ B_z(u) \end{bmatrix} = \left\{ \begin{array}{l} (1-u)^5 P_0 + \\ + 5u(1-u)^4 P_1 + \\ + 10u^2(1-u)^3 P_2 + \\ + 10u^3(1-u)^2 P_3 + \\ + 5u^4(1-u) P_4 + \\ + u^5 P_5 \end{array} \right\}$$

$$P_0 = \begin{bmatrix} P_{0,x}(u) \\ P_{0,y}(u) \\ P_{0,z}(u) \end{bmatrix}, \ldots, P_5 = \begin{bmatrix} P_{5,x}(u) \\ P_{5,y}(u) \\ P_{5,z}(u) \end{bmatrix} \tag{3}$$

where $P_0, P_1, \ldots, P_5$ are control points and u changes from 0 to 1. It also requires two values to clarify the size and shape of a curve. In addition to transition length, ratio n defined by Eq. (4) is applied. The resulting curve is shown in Fig. 5.

$$n = \frac{c}{d} \tag{4}$$



(a) Bezier curve with $L_t = 2\,mm$.

(b) Bezier curve with $L_t = 5\,mm$.

(c) Bezier curve shape with 3 control points.

Fig. 4. Bezier blend with 3 control points

With different values for segments c and d various curvatures of the fillet can be achieved. Thus both size and form of the curve are alterable. Other, higher order Bezier blends require more parameters to determine the curve, which leads the corresponding algorithm to unnecessary complications. Therefore this study applies a smoothing algorithm with six control points. Two curves were obtained with the same transition length of 2 mm, but with ratio n equal to 0.9 and 0.3.

Fillets are depicted in Fig. 5 respectively. Obviously, the curvature of the blend has changed, which allows customizations to match the required precision.

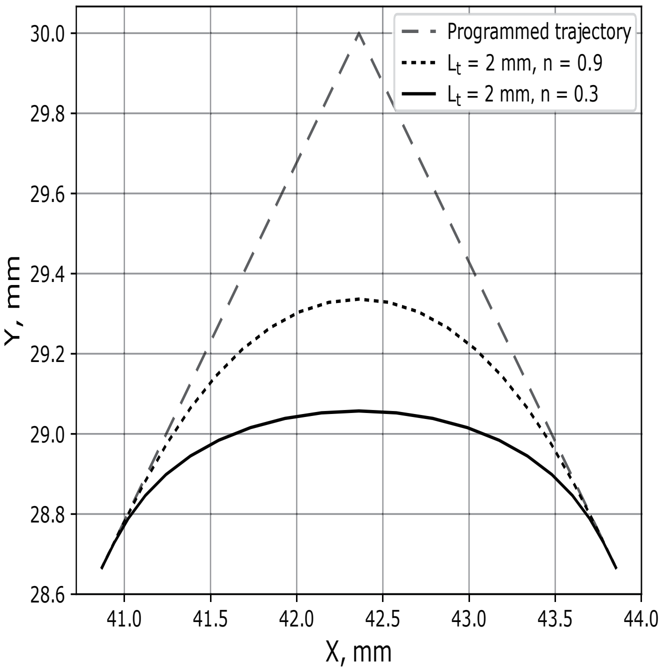


Fig. 5. Bezier blend with 6 control points

The next task is to create an algorithm which controls the feed rate value at trajectory corners. It is necessary to reduce the commanded feed rate for the corresponding linear segment to avoid exceeding the acceleration limit.

This study suggests comparing the allowable acceleration value on each axis with the actual speed difference on the border of two segments. Firstly, the velocity change is calculated as follows [1]:

$$\Delta V_A = F_2 \cdot \frac{A_{E2} - A_{S2}}{L_2} - F_1 \cdot \frac{A_{E1} - A_{S1}}{L_1} \tag{5}$$

where L_i is the length of block N_i, F_i is the feed rate of block N_i, A_{Ei} and A_{Si} is A-axis coordinate. Then the allowable acceleration ΔV_{mA} of A-axis is determined. And next, the velocity change ratio is given by Eq. (6) [1].

$$Q = min\left\{\frac{\Delta V_{mx}}{\Delta V_x}, \frac{\Delta V_{my}}{\Delta V_y}, \frac{\Delta V_{mz}}{\Delta V_z}\right\} \tag{6}$$

where ΔV_A is the real speed difference on the A-axis. If the ratio Q is greater than 1, then there is no violation of the allowable speed change. If the Q value is lower than 1, than the feed rate on the corner should be reduced by multiplying the initial value by Q. The resulting feed rate value is used to generate the speed profile.

V. The demonstration of the final algorithm execution

The resulting algorithm creates spline trajectories at corners, calculates allowable corner feed rates and builds a low jerk velocity profile.

The proposed approach has been implemented using the Python programming language of version 3.6, charts are made with the Matplotlib plotting library of version 3.0.0 and mathematical calculations are made using Numpy.

The algorithm execution requires input data including coordinates of trajectory segments, feedrate values at each block, allowable acceleration and deceleration, parameters for spline building (transition length L_t and ratio n).

For execution example a trajectory with four segments, divided by different corners is used. The input data for the computer simulation is given below.

```
Trajectory P, mm = [
    [20, 10, 0],
    [60, 40, 0],
    [160, 40, 0],
    [170, 20, 0]
];
Feedrate F, mm/sec  = [30, 30, 18, 25];
The allowable acceleration A,  mm/s = 15;
The allowable deceleration D, mm/s = 15;
Transition length Lt mm = [4, 3, 3];
Ration n = [0.9, 0.8, 0.94].
```

The proposed algorithm starts with analyzing the given data. Spline trajectories are created separately with machining error calculation. The machining error at corners is estimated as follows:

$$\epsilon = ||P_{trans} - B|_{u=0.5}|| \tag{7}$$

Applying Eq. (7) the following ϵ values are obtained: $\epsilon_1 = 0.63$ mm, $\epsilon_1 = 0.82$ mm, $\epsilon_1 = 1.2$ mm. The whole trajectory is built afterwards, shown in Fig. 6.

Next, feedrate on splines is calculated. The resulting velocities equal to $F_{c_1} = 25.0$ mm/s; $F_{c_2} = 12.3$ mm/s and $F_{c_3} = 8.2$ mm/s, where F_{c_i} is the velocity at i-th corner. All of the obtained speed values are smaller than the initial commanded feedrates. It prevents the violation of the allowable acceleration and deceleration.

Next, to apply the velocity profile generation segments' lengths, acceleration, constant speed and deceleration time are calculated. Finally, motion profile is generated with block overlap considered. All the resulting plots are depicted in Fig. 6.

It should be noted that the corner from which the processing started is obtained without machining error. Smoothing algorithm is not applied here because a tool starts and finishes its motion at this point and thus is able to create an accurate sharp corner. All other trajectory corners are processed as fillets.

The machining error in the case of the proposed algorithm is present only on corners. Machining errors in the example are obtained as: $\epsilon_1 = 0.63$ mm, $\epsilon_1 = 0.82$ mm, $\epsilon_1 = 1.2$ mm. With the change of corner smoothing parameter

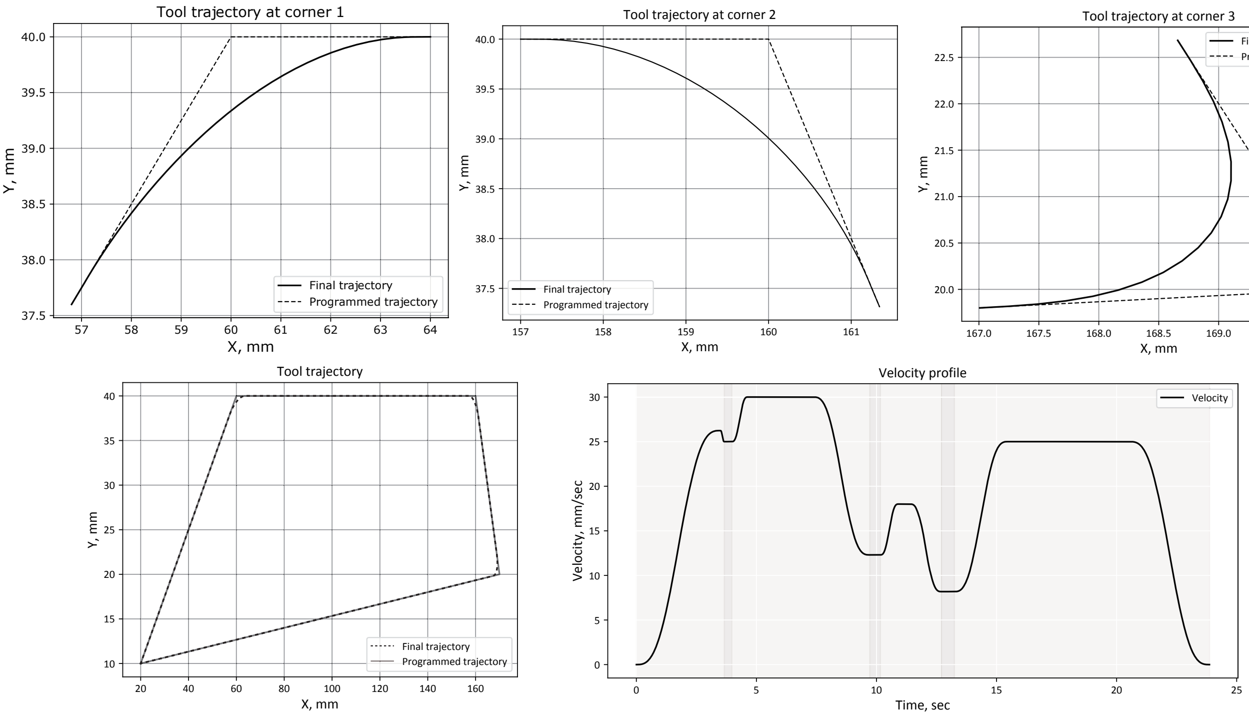


Fig. 6. Resulting trajectories and motion profile

it is possible to alter the resulting inaccuracy. For example, if the L_t values are set for the same tool path as $L_{t_1} = 0.5$, $L_{t_2} = 0.25$, $L_{t_3} = 0.25$, the following results will be provided: $\epsilon_1 = 0.08\,\text{mm}$, $\epsilon_1 = 0.07\,\text{mm}$, $\epsilon_1 = 0.10\,\text{mm}$. Thus users can adjust the machining error to satisfy the specific requirements.

The proposed motion profile allows processing of the example trajectory in $t_p = 23.9\,\text{s}$. As the two main criteria of the motion control algorithm are the processing time as well as the accuracy value, the obtained results need to be analyzed from the perspective of these two characteristics.

Firstly, in order to correctly estimate the machining time obtained, it is advisable to compare the resulting speed profile with profile, created in the absence of the used look ahead and corner smoothing approach. The resulting speed chart is depicted in Fig. 7 and has the time of processing equaling 25.4 s. The algorithm on the other side gives the processing time tp of 23.9 sec. So the proposed approach is 5.9 % faster.

It should also be noted that velocities of three of the four blocks in the simulation were not reached. Maximal speeds, obtained are 19.2, 27.3, 12.9, and 25.0 mm/s. So in this case, the motor will almost every time work at lower power and high processing itself would be impossible.

To sum up, the resulting velocity profile allows high-speed processing with reduced jerk and vibrations. Consequently, it is possible to obtain high surface quality with the proposed algorithm.

VI. Discussion

The main results of the proposed algorithm should be discussed: the accomplished tasks, the aspects that need to be improved and plans for the future development.

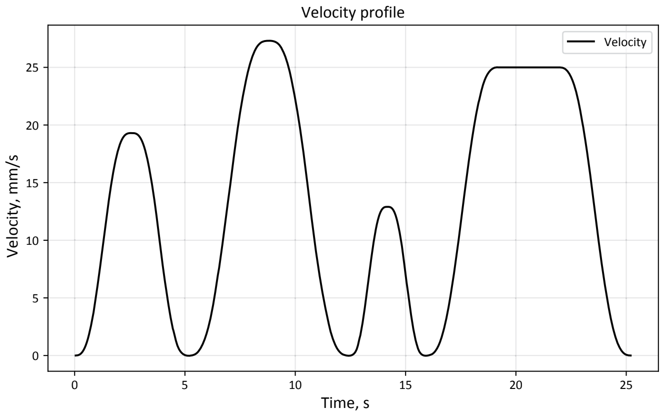


Fig. 7. Motion profile without corner smoothing and look ahead

In the course of project implementation the following tasks were accomplished. The efficient acc/dec control was presented for motion profile generation. A look-ahead algorithm was also applied which provides smooth transition of velocity. The simulation results, described in Section III, show processing time reduction as well as braking exclusion on the block edges. This algorithm allows high-speed machining with high precision that, in comparison to the individual execution of each block with stoppage at corners, provides approx. 6 % faster processing.

The corner smoothing technique that has been presented allows for the creation of a fillet with a predefined tolerance. Hence, it is possible to keep the tool velocity and still maintain the required accuracy. It has been shown that the machining error, received from a corner smoothing is alterable by changing L_t and n parameters value.

The velocity at corners is calculated based on the block length and the programmed feed rate. The corner velocity is generated by comparing allowable acceleration and the actual speed change at corner. This method prevents velocity transitions greater than what a motor can follow correctly.

However, there are some aspects in this research that need to be improved for better performance. One of the problems that have not been solved is corner velocity generation. The method of feed rate control used at corners is carried out independently from the shape of the curve.

Another problem is that even though the simulation results show the machining error appear only at corners and being predefined, in the real life tests, carried out on the CNC equipment, it is very plausible to receive the inaccuracy because of the minor vibrations. So experiments on CNC equipment are needed.

One of the criteria for the implementation of the algorithm is the ability to apply it on stepper motor driven machine as well as in servo driven systems to achieve universality. For that it is necessary to calculate the duration of the next step based on the previous move. For now the step calculation of the acceleration and deceleration motion stages has not been determined.

The algorithm has been implemented in the Python programming language (link to repository: `github.com/stwinter2014/Motion-profile-control-for-CNC-equipment`). The final version of CNC software will be presented in C language.

To sum up, the solution proposed in the present paper provides the motion control for high speed processing with a machining error being present only at corners. It is also made possible to adjust the said machining inaccuracy by creating a fillet of different size and shape.

However, certain improvements should be made such as evaluation of the effect of the curve shape and size on the corner speed as well as algorithm adaptation to stepper motor driven machines.

VII. Conclusion

An efficient trajectory optimization methodology is presented in this paper. The acc/dec control with look-ahead is designed to create a smooth motion profile that will provide for high-precision processing. A Bezier blend generation is made to reduce the unnecessary decelerations at corners and simultaneously allows users to control the machining error by adjusting the curves size and curvature.

The performed simulation of the algorithm implementation showed its compliance with the initial requirements. The machining error in computer simulation appears only at corners, where tool velocity is not equal to zero. The processing time received equals to 23.9 s, which has been shown to be 6 % faster than of the algorithm without corner smoothing and look ahead approach.

The performed simulation of the algorithm implementation showed its effectiveness and compliance with the initial requirements. It is planned to create an algorithm to carry out the acc/dec control of circular interpolation as well as generate the allowable velocity on a circular path. Future tests of the resultant algorithm will be conducted on CNC systems.

VIII. Acknowledgements

This work was carried out under project no. 617026 “Technologies of cyber-physical systems: management, computing, security” conducted at the Faculty of Control Systems and Robotics, ITMO University.

References

[1] S.-H. Suh, S.-K. Kang, D.-H. Chung, and I. Stroud, *Theory and Design of CNC Systems*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[2] P. Smid, *CNC Programming Handbook: A Comprehensive Guide to Practical CNC Programming*, ser. Industrial Press eBooks. Industrial Press, 2003.

[3] A. Hughes, *Electric Motors and Drives: Fundamentals, Types and Applications*. Elsevier Science, 2005.

[4] A. Overby, *CNC Machining Handbook: Building, Programming, and Implementation*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 2010.

[5] H. Li, Z. Gong, W. Lin, and T. Lippa, “Motion profile planning for reduced jerk and vibration residuals,” *SIMTech technical reports*, vol. 8, no. 1, pp. 32–37, 2007.

[6] J. Hu, L. Xiao, Y. Wang, and Z. Wu, “An optimal feed rate model and solution algorithm for a high-speed machine of small line blocks with look-ahead,” *The International Journal of Advanced Manufacturing Technology*, vol. 28, no. 9, pp. 930–935, Jul 2006.

[7] G. Zhao, Y. Zhao, and S. Wang, “The acceleration/deceleration control algorithm based on trapezoid-curve jerk in cnc machining,” *Research Journal of Applied Sciences, Engineering and Technology*, vol. 5, no. 5, pp. 1639–1645, 2013.

[8] T. Hu, L. Zhang, T. Wu, and F. Huang, “Study on an acc/dec method based on moving average technology for cnc machine tools,” *Applied Mechanics and Materials*, vol. 556-562, pp. 1413–1416, 2014.

[9] L. Rutkowski, A. Przybyl, and K. Cpalka, “Novel online speed profile generation for industrial machine tool based on flexible neuro-fuzzy approximation,” *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 1238–1247, Feb 2012.

[10] H. Ni, T. Hu, C. Zhang, S. Ji, and Q. Chen, “An optimized feed rate scheduling method for cnc machining with round-off error compensation,” *International Journal of Advanced Manufacturing Technology*, vol. 97, no. 5-8, pp. 2369–2381, 2018.

[11] X. Du, J. Huang, and L.-M. Zhu, “A complete s-shape feed rate scheduling approach for nurbs interpolator,” *Journal of Computational Design and Engineering*, vol. 2, no. 4, pp. 206–217, 2015.

[12] L. Wang and J. Cao, “A look-ahead and adaptive speed control algorithm for high-speed cnc equipment,” *International Journal of Advanced Manufacturing Technology*, vol. 63, no. 5-8, pp. 705–717, 2012.

[13] Q. Bi, Y. Wang, L. Zhu, and H. Ding, “A practical continuous-curvature bzier transition algorithm for high-speed machining of linear tool path,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7102 LNAI (PART 2), pp. 465–476, 2011.

[14] B. Sencer, K. Ishizaki, and E. Shamoto, “A curvature optimal sharp corner smoothing algorithm for high-speed feed motion generation of nc systems along linear tool paths,” *International Journal of Advanced Manufacturing Technology*, vol. 76, no. 9-12, pp. 1977–1992, 2015.

[15] Y. Zhang, P. Ye, J. Wu, and H. Zhang, “An optimal curvature-smooth transition algorithm with axis jerk limitations along linear segments,” *International Journal of Advanced Manufacturing Technology*, vol. 95, no. 1-4, pp. 875–888, 2018.

[16] B. Sencer, A. Dumanli, and Y. Yamada, “Spline interpolation with optimal frequency spectrum for vibration avoidance,” *CIRP Annals*, vol. 67, no. 1, pp. 377–380, 2018.