

Development of modular control software for construction 3D-printer

A Bazhanov, D Yudin, V Porkhalo

Belgorod State Technological University n.a. V G Shukhov, 46, Kostukova St.,
Belgorod, 308012, Russia

E-mail: all_exe@mail.ru

Abstract. This article discusses the approach to developing modular software for real-time control of an industrial construction 3D printer. The proposed structure of a two-level software solution is implemented for a robotic system that moves in a Cartesian coordinate system with multi-axis interpolation. An algorithm for the formation and analysis of a path is considered to enable the most effective control of printing through dynamic programming.

1. Introduction

In today's world, the issues of new construction technologies are most acute when addressing efficiency and resource-saving aspects. Additive printing of building structures and products is very urgent and is actively explored by the world engineering science. At present, there are a number of prototypes of construction 3D printers capable of printing entire buildings [1, 2], but the reliability of such robotic systems, the complexity of design solutions and the simplicity of obtaining the final product from the project to the printed design are still a long way from meeting the requirements [3].

Modern robots, on the one hand, demonstrate quite acceptable performance indicators for building structures based on additive printing technologies and justify the reduced cost of such structures, but on the other hand, they show the need for improving the technology [4]. Some studies involve the use of data mining when controlling construction printers online to improve the accuracy of their positioning [5]. Others are devoted to the development of new methods and algorithms to control a printer and build a path using high-performance computing tools, such as FPGA [6]. Still others are associated with experiments aimed to select suitable building materials [7, 8].

Since most of construction printing solutions are effective only when relied on a human being, an important task is to create a full-fledged software package that enables one to analyze inputs in the form of graphic project files, calculate the most effective paths and evaluate the possibility of reproducing such project in real conditions.

This article presents software solutions that enable one to perform the entire preprocessing cycle and conduct efficient construction, relying on industrial reliability and capabilities of automation components.

2. Task statement

The article mainly considers the development of software:

- to analyze and design an effective path of the extruder of a construction 3D printer under its multi-axis interpolation;



Content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](https://creativecommons.org/licenses/by/3.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

- to ensure simultaneous control of raw mixture preparation and feeding;
- to keep track of work member contamination and its timely clean-ups;
- to control the quality of printed structure.

The construction 3D printer as such is a four-axle portal structure [9], where translations occur along the orthogonal axes and two of the four axes are co-directional, but different in scale (Fig. 1). Let us denote the principal axis of translation along the rails as variable X_L and its co-directional axis of small translations as X_S .

The X coordinate of the platform with an extruder is connected with the X_L and X_S coordinates by the following relation:

$$X = X_L + X_S. \quad (1)$$

When moving the platform with an extruder, the total translation along the X_L axis should be as small as possible, since it involves the use of large displacement drives, which, due to greater load and inertia, are less precise and lead to large errors.

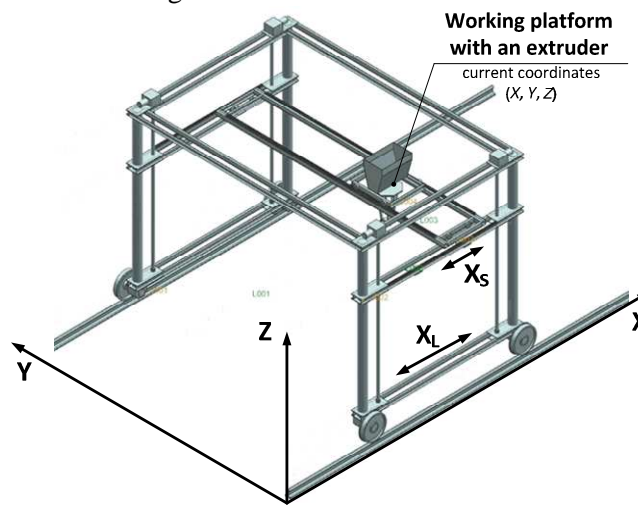


Figure 1. CAD model and motion axes location of main movements unit for construction 3D-printer

Mathematically, this requirement can be described as the problem of minimizing the functional:

$$Q = \sum_{i=1}^{n-1} |\Delta X_{Li}|,$$

where ΔX_{Li} is the X_L axis increment when passing from the i - point of the toolpath specified by the G-code to the $i+1$ point, n is the total number of toolpath points consisting of linear or circular interpolation sections.

In this statement, it is necessary to achieve a minimum value of the functional Q :

$$Q_{\min} = \min_{\Delta X_{Li}} \left(\sum_{i=1}^{n-1} |\Delta X_{Li}| \right). \quad (2)$$

3. Modular control software

To solve this problem, it is proposed to apply a modular two-level structure. Its consolidated scheme is shown in Fig. 2.

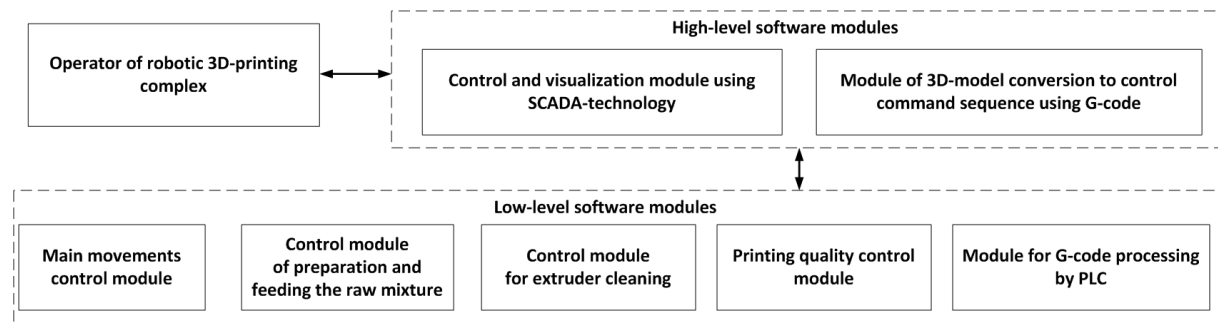


Figure 2. Modular two-level structure of construction 3D-printer control software

Low level software modules are powered by the internal real-time OS of the industrial Schneider Electric LMC058 logic motion controller (PLC). Inputs for G-code processing module is the trajectory of the extruder, its speed and consumption of building materials in the form of G-codes, which can be supplemented by back-up commands requesting for control of the printed layer, or specifying the actions of the codes. Outputs from this module are transmitted to the other two: the main movements control module and the control module of preparation and feeding of the raw mixture. The former is mainly responsible for generating corrective and controlling impacts on the drives of the construction printer based on the inputs and real-time data from the sensors, as well as providing security control of the working area. The control module of preparation and feeding of the raw mixture also interacts with the control module for extruder cleaning to ensure continuous uninterrupted printing. The printing quality control module works independently and enables to generate signals into high-level software modules thus informing on the current state of the printed object or emergencies.

The upper level of the software package includes a control and visualization module that uses SCADA technology. It implements the operator's position human-machine interface. HMI screen layout of generation module for control and corrective actions on the robotic system actuators is shown in Fig. 3. The displayed G-codes are preliminarily processed and checked in the software module for converting the 3D design project into a sequence of control commands. This module calls for a modern approach to the construction of automated building systems based on BIM-models [10]. The functioning algorithm of this module is described in detail in Section 4 of this paper.

In addition, the control and visualization module enables real-time monitoring and control of the low-level module for preparation and feeding of the raw mixture using the window shown in Fig. 4. The proposed software implements a three-stage procedure that includes the selection of raw mixture composition and parameters using a ceramic binder, mixing and feeding of the raw mixture using a hydraulic binder and signal pickups on their state for on-line monitoring and control via computer interface.

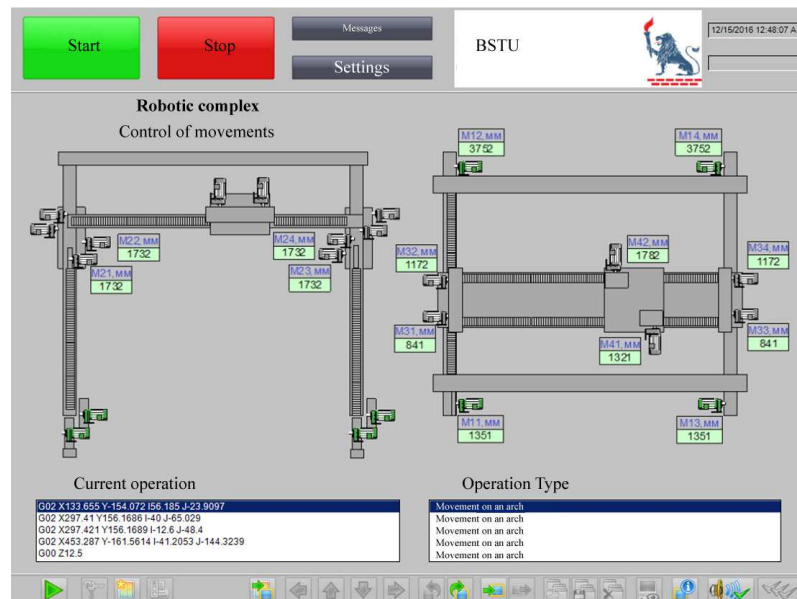


Figure 3. HMI screen layout of generation module for control and corrective actions on the robotic system actuators

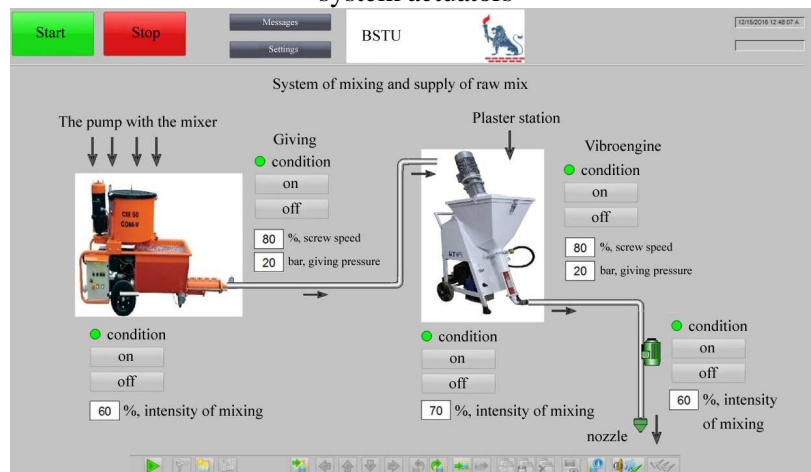


Figure 4. HMI screen layout of monitoring and control module for mixing and feeding of raw material mixture

4. Development of the G-code generation algorithm for construction printer control program

The proposed G-code generation algorithm for the construction printer control program comprises 3 stages:

1. Solid modeling of a building structure in one of the existing software packages, including SolidWorks [11], Autodesk Inventor [12], etc. followed by the reception of .stl file.

2. Import of the constructed model into printhead path generators for a standard 3D printer and generation of an intermediate G-code (.gcode file). Simplify3D [13], Slic3r [14], Ultimaker Cura [15], etc. can be used as such software package. Slic3r and Ultimaker Cura are freeware applications. The source code in Perl located on the GitHub platform [16] is available with the first package. Slic3r can be used both with a GUI application with a step-by-step printing visualization, and with an invoke in the command line, which is convenient for integration into third-party applications. Slic3r has a number of limitations related to configure filling of solid model boundaries. Ultimaker Cura does not have an open source code and works only as a GUI application, visualization possibilities of large 3D printers are also limited. Simplify3D is an inexpensive commercial application that has more extensive

functionality in terms of providing various print configurations, more fill types and settings. However, it does not provide an in-line mode, which lowers its possibilities to be embedded into third-party applications. At the same time, it has high possibilities for visualizing the printing process and a high speed of path generation, which is important for quick verification of the desired solid printed model.

Printhead path generation based on the intermediate G-code is visualized in Fig. 5.

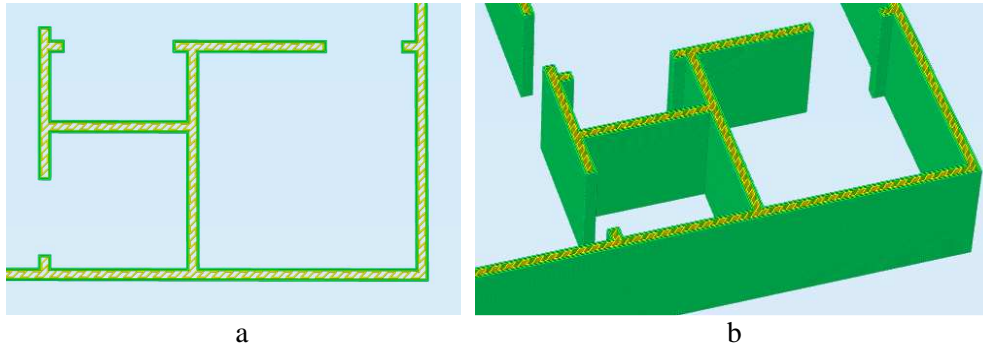


Figure 5. Printhead path of construction printer (green color means outer borders, orange means infill) generated by intermediate G-code: a – for single layer, b – for several layers

3. Conversion of the intermediate G code into the G code for the developed control system, where the X -axis movement is divided into two independent motions along the X_L coordinate (large displacement) and X_S (small platform displacement) according to the formula (1).

At the output, the converted G-code is stored to the corresponding .gcode file.

Solving for optimal motion of the construction printer along the X_L coordinate to minimize the functional Q will be carried out in accordance with the principle of dynamic programming [17].

In this case, the decision to choose ΔX_{Li} (the X_L increment) will be taken at each iteration as one move value ΔX_{Li} to be the minimum possible (see the example in Fig. 6).

1. To do this, let us first find the maximum possible displacement of the S_X platform with the extruder along the X_L coordinate in the path segment between the i and $(i + I)$ points:

$$S_X = \begin{cases} X_{\max} - X_{Li}, & \text{if } X_{i+1} \geq X_{Li}, \\ X_{\min} - X_{Li}, & \text{if } X_{i+1} < X_{Li}, \end{cases}$$

where X_{\max} and X_{\min} are respectively the maximum and minimum values of the X coordinate in the path segment between the i and $(i + I)$ points.

If the path segment is represented by linear interpolation, then:

$$X_{\max} = \max (X_i, X_{i+1}), X_{\min} = \min (X_i, X_{i+1})$$

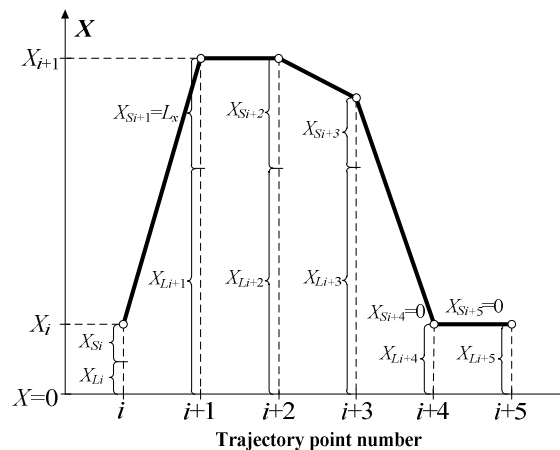


Figure 6. Example of transition from X coordinate of intermediate G-code to the X_L and X_S coordinates of construction 3D-printer in the case of linear interpolation.

If the path segment is described by circular interpolation [18], let us assume that it represents an arc connecting the i and $(i + 1)$ points, and the angle change does not exceed 180° (Fig. 7).

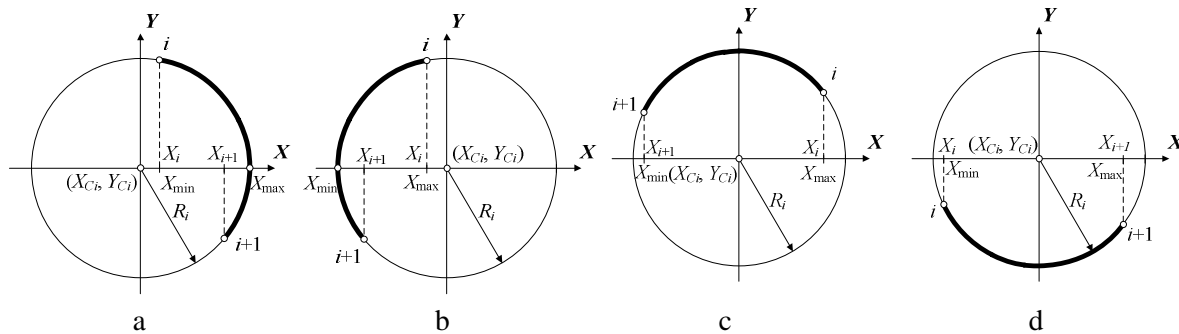


Figure 7. Various cases of determining the maximum and minimum X coordinates in the case of circular interpolation

In the case of circular interpolation, not only the X , Y , Z coordinates of the platform are formed in the intermediate G-code, but also relative coordinates I and J of the center of the circle of radius R_i , the arc of which connects the i and $(i + 1)$ points of the trajectory. In Fig. 7, the coordinates of the center of circle (X_{Ci}, Y_{Ci}) are equal to:

$$X_{Ci} = X_i + I_i, \quad Y_{Ci} = Y_i + J_i.$$

The radius of the circle is equal to:

$$R_i = \sqrt{(X_i - X_{Ci})^2 + (Y_i - Y_{Ci})^2}.$$

The angles between the segments connecting the center of the circle and the i and $(i + 1)$ points of the trajectory and the X axis are equal, respectively:

$$\varphi_i = \arccos((X_i - X_{Ci}) / R_i) \cdot \text{sign}(Y_i - Y_{Ci}),$$

$$\varphi_{i+1} = \arccos((X_{i+1} - X_{Ci}) / R_i) \cdot \text{sign}(Y_{i+1} - Y_{Ci}).$$

If condition $(\varphi_i \cdot \varphi_{i+1}) < 0$ and $(|\varphi_i - \varphi_{i+1}| < \pi)$ is satisfied, i.e. the situation shown in Fig. 7a is observed, then segment $(X_i, Y_i; X_{i+1}, Y_{i+1})$ is divided into two:

$$(X_i, Y_i; X_{Ci} + R_i, Y_{Ci}) \text{ with } X_{\max} = X_{Ci} + R_i \text{ и } X_{\min} = X_i;$$

$$(X_{Ci} + R_i, Y_{Ci}; X_{i+1}, Y_{i+1}) \text{ with } X_{\max} = X_{Ci} + R_i \text{ и } X_{\min} = X_{i+1}.$$

If the condition $(\varphi_i \cdot \varphi_{i+1}) < 0$ and $(|\varphi_i - \varphi_{i+1}| > \pi)$, is satisfied, i.e. the situation shown in Fig. 7b is observed then the segment $(X_i, Y_i; X_{i+1}, Y_{i+1})$ is divided into two:

$$(X_i, Y_i; X_{Ci} - R_i, Y_{Ci}) \text{ with } X_{\max} = X_i \text{ и } X_{\min} = X_{Ci} - R_i;$$

$$(X_{Ci} - R_i, Y_{Ci}; X_{i+1}, Y_{i+1}) \text{ with } X_{\max} = X_{i+1} \text{ и } X_{\min} = X_{Ci} - R_i.$$

If the condition $(\varphi_i \cdot \varphi_{i+1}) > 0$, is satisfied, i.e. the situation shown in Fig. 7c and Fig. 7d is observed, then similar to linear interpolation, one has:

$$X_{\max} = \max(X_i, X_{i+1}), \quad X_{\min} = \min(X_i, X_{i+1}).$$

2. If value S_X is greater than threshold L_X - the maximum possible displacement of the platform along the X_S coordinate, i.e. $S_X > L_X$, then let us recalculate the X_L coordinate so that it changes to the minimum possible value:

$$\Delta X_{Li} = S_X - L_X,$$

If value S_X is negative, then:

$$\Delta X_{Li} = S_X.$$

Otherwise, X_L coordinate remains the same.

Thus, the new X_{Li+1} coordinate is calculated by the formula:

$$X_{Li+1} = \begin{cases} X_{Li} + S_x - L_x, & \text{if } S_x > L_x, \\ X_{Li} + S_x, & \text{if } S_x < 0, \\ X_{Li}, & \text{if otherwise.} \end{cases}$$

3. In accordance with (1), the X_L coordinate of the small displacement at the $i + 1$ point will be equal to:

$$X_{Si+1} = X_{i+1} - X_{Li+1}.$$

Repeating steps 1-3 for all other $n-1$ segments, let us eventually obtain the optimal solution the entire trajectory through with respect to minimizing the functional Q (2).

Table 1 shows an example of converting an intermediate G-code into a G-code to control the proposed 3D construction printer.

Table 1. G-code conversion

Intermediate G-code	Resulting G-code for construction printer control
G1 X400.000 Y400.000 E0.000	G1 X400.000 Y400.000 U0.000 E0.000
G1 X8000.000 Y400.000 E3.200	G1 X4000.000 Y400.000 U4000.000 E3.200
G2 X8000.000 Y4000.000 I-1120.000	G2 X4000.000 Y2200.000 U5000.000 I-1120.000
J1800.000 E5.350	J1800.000 E4.275
	G2 X3000.000 Y4000.000 U5000.000 I-2120.000
	J0.000 E5.350
G1 X400.000 Y4000.000 E8.550	G1 X0.000 Y4000.000 U400.000 E8.550
G1 X400.000 Y2800.000 E9.150	G1 X0.000 Y2800.000 U400.000 E9.150
G1 X400.000 Y1200.000 E9.150	G1 X0.000 Y1200.000 U400.000 E9.150
G1 X400.000 Y400.000 E9.750	G1 X0.000 Y400.000 U400.000 E9.750

G1 and G2 commands designate, respectively, linear and circular interpolation of the path segments [19]. The X coordinate in the resulting G-code corresponds to the X_S coordinate of the selected 3D printer in mm, the Y and Z coordinates correspond to the same coordinates of the printer in mm (see Fig. 1), I and J are the displacements in mm along the X and Y coordinates of the center of the circle for the path segment with circular interpolation, the additional U coordinate corresponds to the X_L coordinate of the construction printer in mm, E is the volume of the consumed building mixture in conventional units. In the example considered, L_x is assumed to be 4 m

5. Conclusion

The analysis of existing sources and developments allows us to conclude that the proposed software package of a modular two-level structure can solve the main tasks facing a modern 3D construction printer. The algorithm developed for the software module for 3D-model conversion into a sequence of control commands allows the generation of G-codes with linear and circular interpolation and provides the minimum possible transfer of the most energy-intensive degree of mobility of the robotic system under consideration based on the dynamic programming approach.

The software package allows one to make the 3D printer control system more stable, to avoid abnormal functioning due to a number of modules operating on the basis of industrial hardware and the analysis of additional information from the control sensors.

6. Acknowledgments

This article is written in pursuance of project № 14.577.21.0193 of the Federal Target Program "Research and development on the priority directions of scientific-technological complex of Russia for 2014 – 2020" on the theme: "Development of robotic complex for the implementation of full-scale additive technologies of innovative materials, composites, constructions and buildings". Unique identification number of Project is RFMEFI57715X0193.

References

- [1] Kietzmann J, Pitt L, Berthon P 2015 *Bus. Horiz.* **58** (2) 209–215
- [2] Liang F, Liang Y 2014 *Glob. J. Hum. Soc. Sci.* **14** (5) 7–10
- [3] Wu P Wang J Wang X 2016 *Automation in Construction* **68** 21–31
- [4] Lim S, Buswell R A , Le T T, Austin S A, Gibb A G F and Thorpe T 2012 *Automation in Construction* **21**(1) 262–268
- [5] Zavichi A and Behzadan A H 2012 *Construction Research Congress* 321–330
- [6] Musienko M P, Krainyk Ya M, Denisov A O and Bugaev V I 2014 *Universum: Engineering Sciences: electronic scientific journal.* **4**(5)
- [7] Cesaretti G, Dini E, Kestelier X, Colla V and Pambaguian L 2014 *Acta Astronautica* **93** 430–450
- [8] Goulas A and Friel R J 2016 *Rapid Prototyping Journal* **22**(5) 1–19
- [9] Bazhanov A G, Yudin D A, Porkhalo V A and Karikov E B 2016 *Proc. of The Intern. Conf. on IDT (IDT 2016) IEEE Xplore* 23 – 31
- [10] Ding L, Wei R and Che H 2014 Development of a BIM-based Automated Construction System *Procedia Engineering* **85** 123–131
- [11] SOLIDWORKS 3D CAD Packages <http://www.solidworks.com/sw/products/3d-cad/packages.htm>
- [12] Autodesk Inventor. 3D CAD for product design <https://www.autodesk.ru/products/inventor/overview>
- [13] Simplify3D <https://www.simplify3d.com/>
- [14] Slic3r. G-code generator for 3D printers <http://slic3r.org/>
- [15] Ultimaker Cura software <https://ultimaker.com/en/products/ultimaker-cura-software>
- [16] Open Source toolpath generator for 3D printers <https://github.com/alexrxj/Slic3r>
- [17] Cormen T, Leiserson C, Rivest R and Stein C 2009. Introduction to Algorithms (3rd ed.) *MIT Press and McGraw-Hill* 1296
- [18] Basic G-codes. Circular interpolation <http://planetacam.ru/college/learn/6-3/>
- [19] Smid P 2008 CNC Programming Handbook (3rd ed.), *Industrial Press*, 600