

Algorithms for building products using 3D printing with cable-suspended robot

Larisa Aleksandrovna Rybak

Belgorod State Technological University named after
V.G. Shoukhov
Belgorod, Russia
308012, Kostukov St., 46
rl_bgtu@intbel.ru

Dmitry Ivanovich Malyshev

Belgorod State Technological University named after
V.G. Shoukhov
Belgorod, Russia
308012, Kostukov St., 46
dimon-belgorod@yandex.ru

Elena Vladimirovna Gaponenko

Belgorod State Technological University named after V.G. Shoukhov
Belgorod, Russia
308012, Kostukov St., 46
rl_bgtu@intbel.ru

Abstract – The article presents algorithms for the control of a robotic complex for building products using 3D printing, including a module for the main movements and a module for precise movements based on a robot with a cable drive. The description of software modules for the implementation of algorithms and the technology of printing with the use of a printing module and a control unit for the supply of printing material are described. The basic principles of the operation of the program modules and the program interface are considered. There is a detailed selection of 3D print settings in the developed software module such as layer height, type of filling of the model with printing material, printing of supports, etc. Recommendations are given on the optimal choice of these settings to achieve the required quality of the obtained product models. The results of modeling are presented.

Keywords – cable robot, control algorithm, 3D printing, software module.

I. INTRODUCTION

Cable robots are the one of the most promising areas for the development of modern robotics. They have a number of desirable attributes that make them well suited for various applications. Since motors can reel a large amount of cable, it can have very large working spaces. [1]. Since motors should not be installed near a moving platform, they are suitable for operation in explosive environments [2]. In addition, their load capacity is relatively comparable even with construction cranes [3]. Their high ratio of payload to weight makes them attractive for high-speed manipulation tasks [4]. Compared to parallel hard-coupled handhelds, parallel cable-operated manipulators (CDPMs) have some promising advantages in many practical applications. The control inputs for the winches are usually calculated using a simplified kinematic model of the robot that takes into account the geometry of the platform and the frame, assuming that the attachment points on the platform and the contact points on the winch are time invariants.

II. DESCRIPTION OF COMPLEX CONTROL ALGORITHMS

The use of a cable robot for 3D printing is discussed in Barnett and C. Gosselin work [5]. It allowed significantly expanding the working area for 3D printing, but it can be further increased. In the article, the robotized complex, assuming usage of cable and portal technologies, is considered in fig. 1. The complex includes the module of the main movements of the platform in three coordinates and the module of precise movements, fixed on the platform by means of cables, changing their length. In addition, the portal platform moves along railroad tracks, thereby significantly expanding the working area for solving the problem of printing of building structures and structures of considerable size.

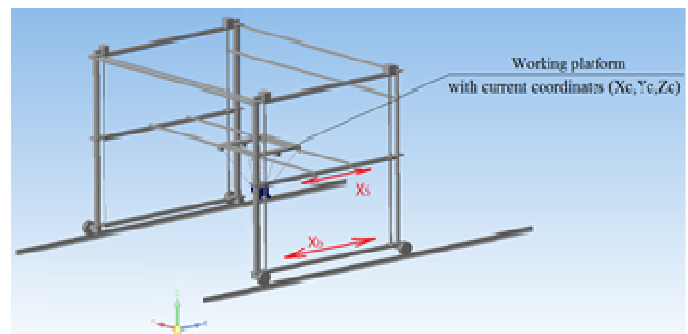


Fig. 1. Scheme of robotic complex.

The module of the main movements represents the associated mechanical and hardware-software parts that perform a smooth and precise movement of the working element in a given working space with the positioning on the created map based on the CAD file. As the input information, restrictions on the positioning accuracy and the speed of

movement of the working element, the original CAD file for further processing, and general safety data are used [6]. The algorithm realizes the transformation of the original models of the printed product into a control map of the movements of the robotic complex, carrying out this operation. At the same time, in conjunction with the transformation, the order of the product's printing, the possible trajectories of the working body's passage are calculated. The algorithm operates on the basis of a model of the technological control process in the form of a Petri net, Fig. 2.

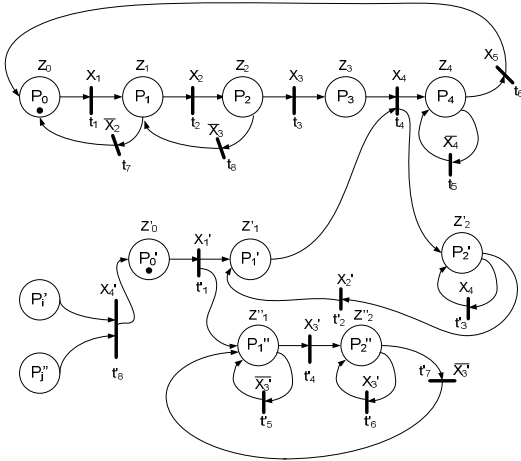


Fig. 2. Model of control process

The states (positions) Petri nets are labeled by symbols

$P_i^{(c)}$ and correspond to the start of subroutines $z_i^{(c)}$. [7]

Description of states $P_i^{(c)}$ and run the following subroutines $z_i^{(c)}$:

P_0 – selection of the 3D model source; P_1 – formation of a sequence of working body movements (WB); P_2 – check path WB; P_3 - initialization of communication with the PLC and the formation of a data packet for transmission; P_4 – the transfer of packet data in the PLC; P'_0 – initializing the PLC; P'_1 – installation connection to the workstation (PC); P'_2 – reception of a data packet from the PC; P''_0 – initializing the print mode; P''_1 – execution of the current data packet instructions; z_0 – loading a 3D model from a file; z_1 – launching a subroutine for forming a sequence of movements of the working unit (WB); z_2 – running the trajectory check subroutine WB; z_3 – launching a data transmission subroutine and generating a data packet for transmission; z_4 – the process of data transfer to the PLC; z'_0 – internal diagnostics of the PLC; z'_1 – startup of a subroutine for receiving data from a

PC; z'_2 – the process of receiving data; z''_1 – running the subroutine control of movement and operation of the WB; z''_2 – control of movements and work of the WB.

Each transition to Petri nets (denoted by symbols $t_i^{(c)}$) corresponds to conditions for its implementation. Starting positions on a workstation (PC) and programmable logic controller (PLC). If the PLC is restored after a power failure, the transition is made in the event of a trouble-free boot of the PLC (t'_8).

The resulting algorithm can be conditionally divided into two parallel branches according to the type of equipment on which calculations are performed. Subroutines Z_0, Z_1, Z_2, Z_3, Z_4 run on a workstation, and subroutines $Z'_0, Z'_1, Z'_2, Z''_1, Z''_2$ run on the programmable logic controller. Let us consider the first operations performed on the PC. The transition to the formation of a sequence of movements of the working body (P_1) is performed after downloading a file containing the 3D-model of the product (t_1). After forming the sequence of movements of the working element (PO), a transition to the verification of the formed trajectory occurs, P_2 . In the case of an error in the formation of the sequence (t_7), a transition is made from the P_1 to P_0 . In the case the generated trajectory is incorrect (t_8), a transition occurs from P_2 to P_1 . After checking, trajectory (t_3) initializes communication with the PLC and the formation of a data packet for transmission (P_3). The transition to data transfer (P_4) is performed after the packet buffer is checked for data (t_4) in it. Next, until the buffer is empty (t_5) the transfer is performed. In case the buffer is empty (t_6), the transition to P_0 .

PLC work begins with initial (P'_0). After the successful initialization (t'_1), the connection to the PC (P'_1) is enabled and the print mode is initialized (P'_1). If the packet buffer on the PC is not empty (t_4), the data are received (P'_2). If an error occurs during the reception of the packet (t'_3), the packet is re-transmitted. In the case when the entire trajectory is worked by the system and the buffer of the received PLC packets is empty (t'_2), there is a transition to P'_1 . After transmission, the data from the packets, received from the workstation, are converted into a sequence of coordinates of the WB trajectory and enter the coordinate buffer. After the print mode has been initialized (P''_1), it is checked if there are instructions for execution of extracted from the data packets, received from the PC. If the coordinate buffer is empty (t''_3),

then the PLC remains in P_1'' . If one needs to move the PO (t_4'), the PLC proceeds to the current instructions (P_2''). The PLC executes instructions from the coordinate buffer as long as it contains data (t_6'). As soon as the coordinate buffer is empty (t_7'), the PLC returns to P_1'' .

In the course of the data packet handling, subroutine reads the next command packet that contains the specified coordinates of the working platform (X_3, Y_3, Z_3). Taking into account the current position platform (X_T, Y_T, Z_T), three traffic control subroutines of the major motion module simultaneously run respectively along the axes X, Y and Z. As a result of their performance, deviation of the current position from the set does not exceed the value of $\epsilon x, \epsilon y$ and ϵz for the corresponding axis:

$$|\Delta x| = |X_3 - X_T| \leq \epsilon x, |\Delta y| = |Y_3 - Y_T| \leq \epsilon y, |\Delta z| = |Z_3 - Z_T| \leq \epsilon z.$$

Values $\epsilon x, \epsilon y$ and ϵz in (4) are selected in accordance with the terms of reference and should not exceed 10 mm. Algorithm regulation subroutine, for example, coordinates X of the work platform, provides a choice of actuators, on which there will be movement along the X axis as the robotic complex design provides the movement by the actuator main movement (X_B) and the drive precise movements (X_S).

Switching between the drives is carried out by checking the inequality $|\Delta x| > hx$, in which, if the module is the deviation of the current position of the set, $|\Delta x|$ exceeds threshold hx (selected on the basis of terms of reference, which is not less than 4500 mm), the platform is moving slightly to the right or left using a large movement drive. Otherwise the small movement drive functions takes place. The efficiency of the algorithm is verified using the model of the packet processing process for moving the main movements module based on the Petri net (Fig. 3).

After receiving the data, the packet is checked for deviation of the current position of the module main movements from a predetermined position, that is, checked condition $|\Delta x| \geq \epsilon \parallel |\Delta y| \geq \epsilon \parallel |\Delta z| \geq \epsilon$. In the case of the present conditions, the label goes directly into three positions (P_1, P_6, P_9) for working on the package.

Each of the three branches, corresponding to mentioned positions, is used for testing by moving the corresponding coordinate (x, y, z). From position P_1 , it moves to position four. The first two positions are for the "big" move to coordinate x, which means the movement of the main movements on the tracks module (forward and backward). Operations may be performed under the conditions: attribution move to the «big» ($|\Delta x| \geq hx$), deviation of the current position on the x axis must be more or equal to predetermined value ϵb ($|\Delta x| \geq \epsilon b$). It should also be pointed out that these movements and all others are carried out in the absence of emergency situation (\bar{X}_a).

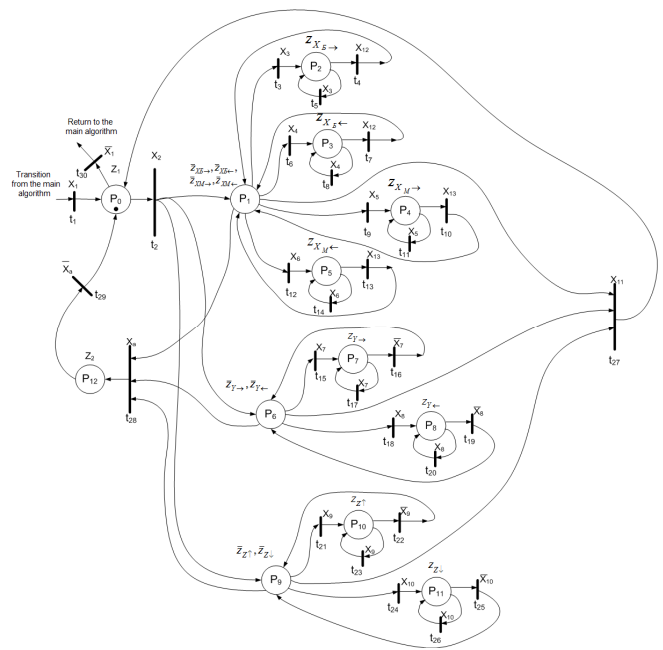


Fig. 3. Model of process of moving packet processing

The other two positions are for "small" move at the coordinate x, which means the movement of the module of the main displacements inside the working area of the complex (forward, backward). Operations may performed under the conditions: attribution move to the «small» ($|\Delta x| < hx$), deviation of the current position at x axis must be more or equal to a predetermined value ϵ ($\Delta x \geq \epsilon$). Going to a standstill on the movement of x is carried out under the condition that the current position in x corresponds to a given position or an accident signal. From the position P_6 can transition to the two positions for moving along the coordinate y (left, right). Operations may performed under the condition of the current position deviation on the y axis, must be more or equal to a predetermined value ϵ ($\Delta y \geq \epsilon$). Going to a standstill, the y-coordinate is carried out under the condition that the current position of the y corresponds to a given signal or accident. From the position, P_9 can move in two positions, designed to move in the z coordinate (up, down). Operations may performed under the condition of deviation of the current position on z, which must be more or equal to predetermined value ϵ ($\Delta z \geq \epsilon$). The movement to coordinate z resting state is carried out under the condition that the current situation with respect to z corresponds to a given signal or accident. After performing the movements in three dimensions (x, y, z) under condition X_{11} ($|\Delta x| < \epsilon \& |\Delta y| < \epsilon \& |\Delta z| < \epsilon$), three labels converge and make the transition to position P_0 , where a new transport package is received. In the absence of the packet (condition X_{12}), there is the transition to the basic algorithm.

III. DESCRIPTION OF THE ALGORITHM AND SOFTWARE MODULES FOR ADDITIVE PRINTING.

Let us consider the motion control algorithm of the robot and the material in the print head (Figure 4). The signal from Arduino MEGA 2560 controller 1, a platform of which includes 54 digital I / O, is fed to expansion card 2. Card Expansion Ramps 1.4 is fully compatible with the Arduino MEGA 2560 controller and allows one to build a full-fledged

controller of 3D-printers. The board also has feedback inputs from the limit switches that limit the movement along the axes and the input of the extruder thermometer. The board additionally has connectors for connecting a SD memory card and display. Engine driver Polulu A4988 3, which controls stepper motor 4 is installed on the expansion card. Engine models Nema 17 used as a stepper motor is the best option for power and cost as well as the stepping motor is widely used in the construction of 3D-printers. Also with the expansion board, 12V voltage is applied to heater 5, which produces heating of the printing material to the required temperature. Nozzle 6 is used as a control object and may have different diameters. Sensor T3650 7 (100kOhm) is used in the construction to measure the extruder temperature. The thermal sensor is connected to the controller and is a very important part of the construction since different types of material require certain temperatures and they need to be constantly monitored.

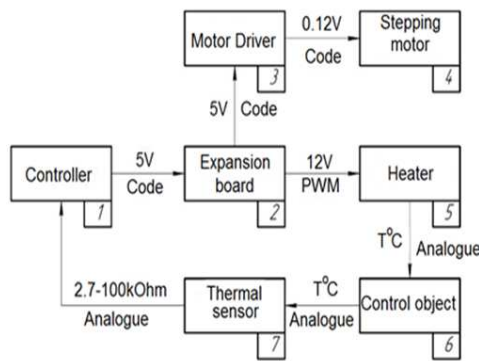


Fig. 4. Algorithm for controlling robot motion and material flow

When the print control unit interacts with the robot controller, the following signals occur:

1. RO 1 - exit from Robot 1 (entrance at block Control) - resolution on heat Printing material.
 - a) ON - heating is enabled b) OFF - heating is prohibited.
2. RO 2 - exit from Robot 2 (at block Control) - Rollback Printing Material.
 - a) ON - roll back the printing material b) OFF - roll back the print media forward.
3. RO, RO 4, RO 5 - exit from Robot 3, 4, 5 (log at block Control) - binary code Thickness layer Print. Thickness layer Depends on the Diameter nozzle.
4. RO, RO 7, RO 8 - exit from Robot 6, 7, 8 (entrance at block Control) - binary code Speeds Printing on A certain Thickness layer. Speed Printing is regulated at Bounds from 0 to 57 mm / sec.

To start printing products one needs a 3D model in the .stl format which must be converted to G-code [8]. In this case the Solidworks software complex was used to create 3D models in which 3D models of building products of various complexity were designed. The software that is responsible for converting the 3D model into G-code is called a "slice". It allows one to preview how the model is placed on the table before printing whether one needs support or chooses all the necessary

parameters including the printing speed, the thickness of the layer, the percentage of filling, etc. [9,10] Experimenting with the settings of the slicer allows one to achieve excellent printing quality for different models types. The model is cut (slides) in layers. Each layer consists of a perimeter and/or a fill. The model can have a different fill percentage and there may not be a fill (hollow model). On each layer, XY axes move with the application of the melt of the printing material. After printing one layer along the Z axis, the layer above the next layer is printed and so on.

Figures 5 and 6 show the field of the "Slicer" program working where a trial 3D model and the "Print Settings" panel are added.

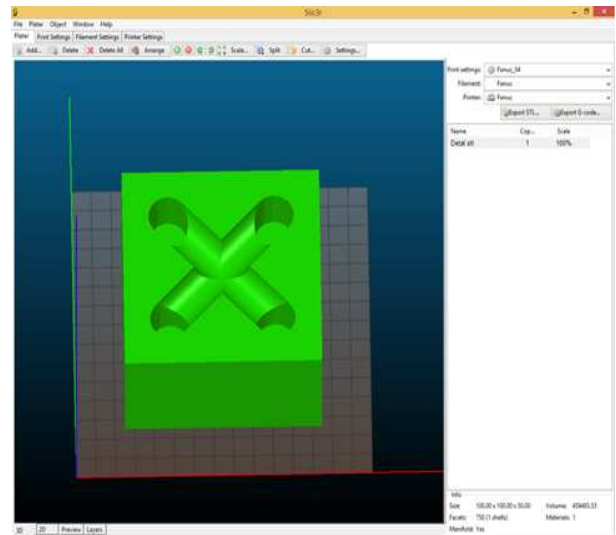


Fig. 5. Working field of program Slicer

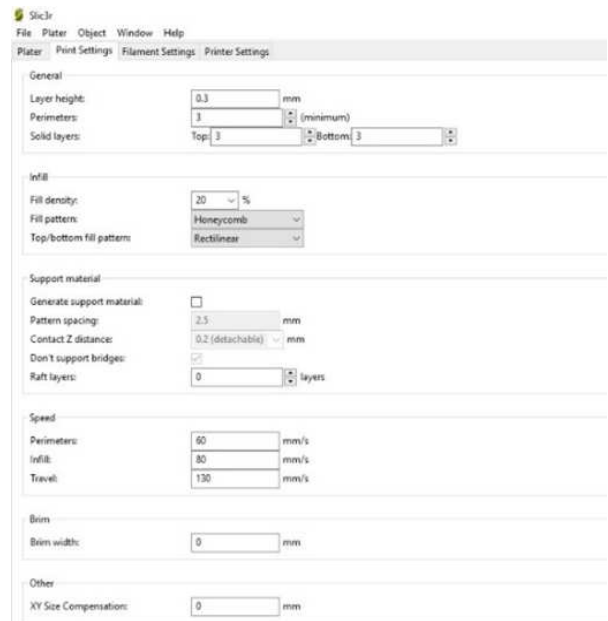


Fig. 6. Print settings panel in program Slicer

One of the most important parameters affecting the print quality is the layer height (Layer height). When setting the height of the layer, it is necessary to rely on the printed model

geometry and for how long it should be printed. With smaller parameters, printing will take much longer. The main rule when setting this parameter is it cannot exceed the thickness of the used nozzle. Also one must select the height of the first layer. It is usually given a little more option «Layer height» for securing a better model on the table. Perimeters' field sets the layers number of circuit patterns (or the number of layers of the wall model). The smaller the number, the smaller the wall thickness of the model and, hence, the more fragile the model will be. Minimal - 1, optimally - 3-4 pieces. In the Solid layers, it is necessary to add the number of layers with the upper (top) and lower (bottom) closing plane model (optimally 3-4).

The second important parameter of the print settings is the «Infill» tab. Filling is the fill of empty space inside the model. The higher the percentage of filling, the stronger the model is obtained. But it prints much longer so the value of the field depends on the requirements for construction products. In the program, one can choose different types of filling the model with printing material (Figure 7)

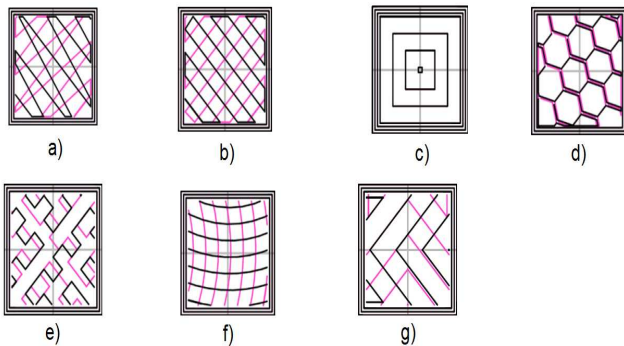


Fig. 7. Types of filling model with printing material: a) Line; b) Rectilinear; c) Concentric; d) Honeycomb; e) Hilbert Curve; f) Archimedean Chords; g) Octagram Spiral

For each type of filling by printing material, one can choose the percentage of filling in the «Fill density». The values entered in the field should be in the range from 0 (no padding) to 0.99 (entire). The most optimal type of the filling is printing material “deemed honeycomb”. 3D-model data obtained by filling are lightweight and have optimum strength. For the upper and lower layers of the printed model, one can choose one’s type of filling in the «Top/bottom fill pattern». Figure 8 shows the change of certain types of filling with increasing density (percent full).

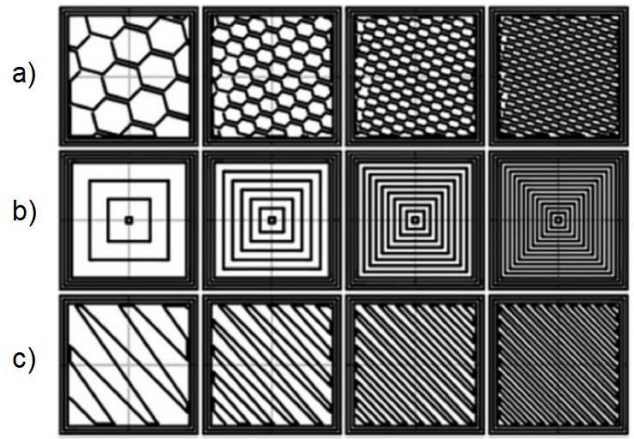


Fig. 8. Change of types of filling with increasing percentage of filling: a) Honeycomb; b) Concentric; c) Line

Another important parameter of setting is support (“Support material”). The supports for printing are the supports needed for models with strongly raised or hanging in the air elements. The supports for printing can be included in the field «Generate support material». The settings of printing can be selected based on how many layers will be printed for support as well as to set the angle of the overhang model. Together with the supports for printing, the printing of the raft can be selected in the field (Raft layers) from the first layer, mainly for the lifting model to a certain height and for fixing the recording material on the basis if the base has any mechanical defects.

The Printing Model of building structures using the robotic complex is shown in Fig. 9.

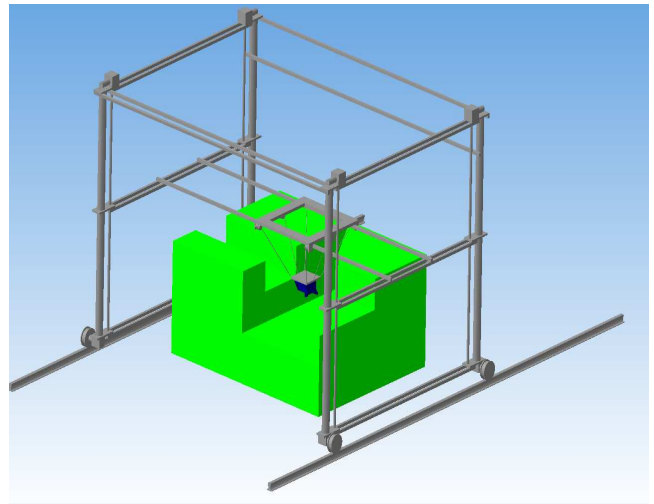


Fig. 9. Robotic complex model

IV. CONCLUSIONS

As a result of the problem discussed and solved, the implemented algorithms as a part of control programs, allowing one to universalize 3D-printing of products, have been developed. The control program on the basis of this

algorithm is more simple and versatile, allowing one to apply it to the entire series of robots with a similar controller. As a result, a robotic system for 3D-printing products of different sizes by means of the discharge nozzle that produces a spray of the working substance and provides basic settings for 3D printing with the best one, enabling the achievement of the desired printed models quality, is developed.

Acknowledgments

The work was carried out within the framework of the implementation of the federal target program "Research and development in priority areas for the development of the scientific and technological complex of Russia for 2014-2020", agreement No. 14.577.21.0193 on subsidies of October 27, 2015, subject: "Development of a robotic complex for implementation of full-scale additive technologies of innovative materials, composites, constructions and structures". The unique identifier is PNIER RFMEFI57715X0193.

References

- [1] Y. X. Su, B. Y. Duan, Nan, R. D. Nan, B. Peng, "Development of Large Parallel-Cable Manipulator for the Feed-Supporting System of a Next-Generation Large Radio Telescope," *Journal of Robotic Systems*, vol. 18(11), pp. 633–643, 2001.
- [2] J. Albus, R. Bostelman, N. Dagalakis, "The NIST Robocrane," *Journal of Robotic Systems*, vol.10, pp. 709–724, 1993.
- [3] R. Bostelman, J. Albus, N. Pagalakis, A. Jacoff, J.L. Gross, "Applications of the NIST ROBOCRANE," *Robotics and Manufacturing*, vol. 5, pp. 403–407, 1994.
- [4] S. Kawamura, H. Kino, C. Won, "High Speed Manipulation by Using Parallel Wire-Driven Robots," *Robotica*, vol. 18, pp. 13–21, 2000.
- [5] E. Barnett, C. Gosselin, "Large-scale 3d printing with a cable-suspended robot," *Additive Manufacturing*, vol. 7, pp. 27-44, 2015.
- [6] T.A. Duyun, A.V. Grinek, V.G. Rubanov, "Application of numerical simulation and neural network models for machining process control," *International Journal of Pharmacy & Technology*, vol. 8, Issue №4, pp. 22515-22524, December 2016.
- [7] A. Bazhanov, D. Yudin, V. Porkhalo, E. Karikov, "Control system of robotic complex for constructions and buildings printing," *Information and Digital Technologies*, pp. 22-31, July 2016, [International Conference on Information and Digital Technologies, p. 59, 2016].
- [8] Yu.A. Mamaev, A.A. Gunkin, A.S. Ananenko, "Mathematical modeling of dynamics parallel manipulator-tripod with six degrees of freedom," *World Applied Sciences Journal*, vol. 30, Issue №10, pp. 1298-1305, 2014
- [9] L.A. Rybak, Yu.A. Mamaev, D.I. Malyshev, L.G. Virabyan, "Software module for the implementation of the specified trajectory of motion of the output unit of the robot-hexapod for 3D printing of products," *The Bulletin of BSTU. named after V.G. Shukhov*, vol. 8, pp. 155-164, 2016
- [10] L.A. Rybak, Yu.A. Mamaev, A.A. Gunkin, "The software controls the robot with 6 degrees of freedom for the layered print," *International Journal of Pharmacy & Technology*, vol. 8, Issue №4, pp. 25036-25044, December 2016. URL:<http://www.ijptonline.com/wp-content/uploads/2017/02/25036-25044.pdf>