Comparative analysis of the speed of concurrent algorithms in solving the forward kinematics of a hexapod robot

D Malyshev\textsuperscript{1}, A Pisarenko\textsuperscript{2} and A Tereshchenko\textsuperscript{3}

\textsuperscript{1}BSTU named after V.G. Shukhov, Belgorod, Russia

E-mail: malyshev.d.i@ya.ru

Abstract. The paper discusses the use of parallel computing on a central processing unit (CPU) to solve the forward kinematics of a parallel hexapod robot. The paper proposes two algorithms to solve the forward kinematics. The first is based on the direct use of the system of equations of kinematic relations of the mechanism. The approximation of a set of system solutions represents a problem of a nine-dimensional grade. The second approach is based on the approximation of a set of solutions to the system of inequalities of a three-dimensional grade. A software package for implementing the proposed algorithms was developed. The comparative analysis of the effect of parallel computing on the speed of the presented algorithms was carried out.

1. Introduction
From year to year the problem of speeding plays an increasingly important role in solving technical problems. This is caused by continuous availability of computing problems of various complexity and limited computing capacity. One of the possible solutions is the use of multithread data processing (concurrent execution of parallel regions of the program). One of the key applied problems to which parallel computing can be applied is the design and control of parallel robots. Their peculiarity is that the end-effector is connected to the base by several kinematic chains. Parallel mechanisms have a number of advantages compared to serial structures [1-3]. Forward kinematics [4-7] determines the position of an end-effector at specified parameters (lengths) of links. Let us consider the solution of forward kinematics using the example of a spatial parallel robot.

2. Numerical algorithm for forward kinematics
The Gough-Stewart platform [1-2], also known as the hexapod, has 6 degrees of freedom and represents a mechanism consisting of 2 platforms. The movable platform is connected to the fixed one through 6 telescopic links ($l_1, l_2, l_3, l_4, l_5, l_6$) having spherical hinges at both ends (Fig. 1). The hinges on the fixed platform are arranged in pairs symmetrically over the entire circumference with the angle of rotation – 120°, angle $\alpha$ – between three pairs of adjacent points $A_i$ of link points. The centers of hinges of a movable platform form an equilateral triangle with length of side $c$. The desired position and orientation of the movable platform is achieved by changing the lengths of the drive links.
The forward kinematics is solved by determining the position and orientation of the end-effector – point O. Its coordinates and the normal vector of the plane of the movable platform are uniquely determined using the coordinates of the movable platform vertices representing a regular triangle

\[
x_O = \frac{\sum x_{Bi}}{3}, y_O = \frac{\sum y_{Bi}}{3}, z_O = \frac{\sum z_{Bi}}{3}
\]

The system of kinematic equations of the hexapod relations is written as:

\[
\begin{align*}
(x_{B2} - y_{B1})(z_{B3} - z_{B1}) - (y_{B3} - y_{B1})(z_{B2} - z_{B1}) & = 0 \\
(x_{B3} - y_{B1})(z_{B2} - z_{B1}) - (x_{B2} - x_{B1})(z_{B3} - z_{B1}) & = 0 \\
(x_{B2} - y_{B1})(y_{B3} - y_{B1}) - (x_{B3} - y_{B1})(y_{B2} - y_{B1}) & = 0 \\
\end{align*}
\]

(1)

The system includes 9 variables corresponding to the coordinates of the \(B_i\) vertices of the movable platform. To approximate a set of solutions of this system, an algorithm based on the method of non-uniform coverings [8-9] previously discussed in work [5] can be used to solve forward kinematics of a flat robot with three degrees of freedom. To accelerate its implementation, let us modify the algorithm using parallel computing. Its flowchart is shown in Fig. 2. The left parts of the system of equations (1) represent coordinate functions written as \(g_i, i \in 1, \ldots, 9\). The intervals of the variables corresponding to the coordinates of the \(B_i\) points are given as \(n\)-dimensional boxes. The initial box \(Q_0\), which includes the entire set of \(X\) solutions, is determined by the constraints of the intervals \(a_i \leq x_i \leq b_i, i = 1, \ldots, n\). It is added to the \(\mathbb{P}_1\) list. Let us extract an arbitrary box \(Q\). Let \(m(Q) = \max_{i=1,m} \min_{x \in \mathbb{X}} g_i(x)\) and \(Q = \min_{i=1,m} \max_{x \in \mathbb{X}} g_i(x)\). If \(m(Q) > 0\) or \(M(Q) < 0\), then \(Q\) does not contain solutions for the system (1). Let us exclude such boxes. If the box cannot be excluded, it is divided into two smaller boxes provided its
diameter is not less than the specified accuracy \( \delta \), and is added to the \( \mathbb{P}_2 \) list. When the \( \mathbb{P}_1 \) list is empty, the validation procedure is performed for boxes from the \( \mathbb{P}_2 \) list. Those that cannot be excluded are divided into 2 smaller boxes and added to the \( \mathbb{P}_1 \) list. Boxes in lists are checked using parallel computing. These cycles are highlighted in Fig. 2 by dashed lines. The covering represents a set of boxes \( \mathbb{P}_A \) with a diameter less than or equal to a predetermined approximation accuracy \( \delta \).

**Fig. 2.** Algorithm for forward kinematics.

The resulting algorithm can be transformed in order to reduce dimension and increase speed taking into account the connectivity of the coordinates of the movable platform vertices.

3. Transformation of a system of nonlinear equations

Let us consider the method of transforming a system of nonlinear equations, which allows reducing the system (1) of a nine-dimensional grade to the system of inequalities of a three-dimensional grade. The set of positions of each of the \( B_i \) points, given only the restriction imposed by the connection with the \( A_j \) point, represents a sphere (Fig. 3a). By adding a constraint that takes into account the connection of two rods at one \( B_i \) point, the set of positions of the \( B_i \) points will decrease to the intersection of two spheres, i.e. to the circle \( U \) with the center \( O_i \), which is shown in Fig. 3 by a dashed line.
Fig. 3. Possible positions of $B_i$: a) intersections of work areas of two rods b) discrete determination of coordinates $B_i$.

Let us consider the circle of the set of possible positions of the $B_i$ point. Let us write down the coordinates of the center of circle

$$x_{O1} = x_{A2} + \frac{a_1(x_{A3} - x_{A2})}{b_1}, \quad y_{O1} = y_{A2} + \frac{a_1(y_{A3} - y_{A2})}{b_1}$$

where $x_{A2}, y_{A2}, x_{A3}, y_{A3}$ – coordinates of the centers of spheres, $b$ – center-to-center distance of spheres, $a$ – distance between the center of the sphere and the center of the circle calculated by the formula $a_1 = \frac{l_2^2 - l_3^2 + b^2}{2b_1}$, where $l_2$ and $l_3$ – lengths of the second and third rods, respectively. The radius $r_i$ of the circle is defined as $r_i = \sqrt{l_2^2 - a^2}$.

The planes in which the desired circles of the intersection of spheres are turned relative to the XOZ plane by angles, which are defined by the formula

$$y_1 = \frac{a_1(120^o - a^o)}{2} = \frac{120^o}{2} = 60^o; \quad y_2 = y_1 + 120^o, \quad y_3 = y_1 + 240^o$$

The coordinates of the center and the radius of circles for points $B_2$ and $B_3$ will be defined in the same way as for point $B_3$ by substituting the corresponding lengths of the links $l_j$ and the coordinates of the points $A_j$.

Let us define the coordinates of the vertices $B_i$

$$x_{Bi} = x_{Oi} + r_i \cos(\beta) \cos(y_i); \quad y_{Bi} = y_{Oi} + r_i \cos(\beta) \sin(y_i); \quad (2)$$

$$z_{Bi} = z_{Oi} + r_i \sin(\beta); \quad (3)$$

where $\beta$ – angle of rotation on the circle.

In case of discrete determination of the coordinates of the point $B_i$ with a step $\Delta \beta$ (Fig. 3), the maximum error $\Delta c_i$ is determined as the chord length between the calculated point $B_i$ and the point located on the circle between the calculated point $B'_i$ and the next point $B''_i$

$$\Delta c_i = 2r_i \sin\left(\frac{\Delta \beta}{4}\right) \quad (4)$$

Given the formula (4), let us compose the condition for checking the mutual arrangement of $B_i$ points at the vertices of an equilateral triangle with a side length $c$

$$\begin{align*}
  c - \Delta c_1 - \Delta c_2 &\leq \sqrt{(x_{B1} - x_{B2})^2 + (y_{B1} - y_{B2})^2 + (z_{B1} - z_{B2})^2} \\
  c - \Delta c_2 - \Delta c_3 &\leq \sqrt{(x_{B2} - x_{B3})^2 + (y_{B2} - y_{B3})^2 + (z_{B2} - z_{B3})^2} \\
  c - \Delta c_1 - \Delta c_3 &\leq \sqrt{(x_{B1} - x_{B3})^2 + (y_{B1} - y_{B3})^2 + (z_{B1} - z_{B3})^2}
\end{align*} \quad (5)$$

The condition (5) allows checking the coordinates of $B_i$ points calculated by formulas (2)- (3) taking into account the error associated with the pitch discreteness of coordinate determination. Let us synthesize an algorithm that assumes a decrease in the dimension of the computing problem. Unlike the first
algorithm, boxes have three dimensions that correspond to intervals of angles $\beta_i$. For boxes let us perform iterative discrete selection of angles $\beta_i$ and for each of the combinations of angles let us calculate the coordinates $B_i$ according to formulas (2)-(3). For calculated values $B_i$ let us check the condition (5). The division is made by adding a box $Q_k$ with an interval $\beta_i \in \left[ \beta_i - \frac{\Delta \phi}{2}; \beta_i + \frac{\Delta \phi}{2} \right]$ to another list. The boxes in lists are checked similarly using parallel computing. Let us perform modeling using the developed algorithms.

4. Simulation results
A software package C++ was developed on the Ubuntu 20.04 operating system for a comparative analysis of the proposed algorithms. The compiler was GCC 9.3.0, which includes the OpenMP version 5.0 library. Calculations were performed on two available machine configurations (Table 1).

The use of parallel computing using this library to solve the task of defining a workspace was previously discussed by the authors in [10]. Simulation results are visualized using the developed module for exporting 3D STL models into a binary format. Simulation was performed for the following initial data: $l_1 = 1100 \text{ mm}$, $l_2 = 1200 \text{ mm}$, $l_3 = 900 \text{ mm}$, $l_4 = 1050 \text{ mm}$, $l_5 = 970 \text{ mm}$, $l_6 = 960 \text{ mm}$, $c = 400 \text{ mm}$, $\alpha = 30^\circ$ (Fig. 1), approximation accuracy $\delta = 0.01 \text{ mm}$. The number of solutions to the forward kinematics problem, which is eight, and the found positions (Table 2) for both algorithms coincided.

Fig. 7 shows the visualization of hexapod positions for given lengths. The time of parallel computing and without parallel computing for both proposed approaches is shown in Table 3. The running time was fixed at the end of all calculations in ticks per second and divided by an equivalent system value, thus obtaining the desired amount of time in seconds.

| Table 1. Computer configuration |
|--------------------------------|
| Configura- | Manufacturer | Number of processor cores / threads | CPU frequency, GHz | Ram frequency, MHz |
| tion No. | | | |
| 1 | Intel | 4/8 | 2400 | 1600 |
| 2 | AMD | 8/16 | 2200 | 2400 |

| Table 2. Forward kinematics solutions. |
|-------------------------------------|
| n/n | $x_0$ | $y_0$ | $z_0$ | $N_x$ | $N_y$ | $N_z$ |
|-----|------|------|------|-------|-------|-------|
| 1   | -119,21 | 231,52 | 831,54 | 44533,9 | -33667,8 | -54389,3 |
| 2   | -268,64 | 77,06 | 809,01 | -46986,4 | 11684,7 | -61079,7 |
| 3   | 42,72 | 102,49 | 859,79 | -42927,2 | 9625,43 | -64339,9 |
| 4   | -209,96 | -12,42 | 849,81 | 18143,2 | 61272,9 | 44625,6 |
| 5   | 2,91 | -13,22 | 855,09 | 7443,58 | -49822,1 | -59475,6 |
| 6   | -36,58 | 230,34 | 829,88 | 31440,8 | 8882,85 | -70764,2 |
| 7   | -119,85 | 85,78 | 784,32 | -27538,3 | 2338,04 | 72877,8 |
| 8   | -116,7 | 83,90 | 911,76 | 2106,27 | 18225 | 75752,4 |
Comparing the proposed algorithms we can conclude that the numerical algorithm is calculated faster for both configurations of machines. The use of parallel computing reduced the operating time by more than a third for each experiment. The discrepancy in the computing time between the proposed algorithms on one particular platform is accompanied by the processing of a different number of possible solutions and the presence of critical sections. The algorithm for converting a system of nonlinear equations, despite the smaller dimension of the system of equations, processes a larger number of possible solutions. To improve its efficiency it is proposed to modify formulae (8) to (9) and eliminate critical sections in subsequent studies.

5. Conclusion
The results of the simulation confirmed the efficiency of parallel algorithms in solving the forward kinematics of the hexapod robot. The computing time of the first algorithm on the first computer configuration using parallel computing decreased by 62%, the second algorithm – by 67.9%, on the second computer configuration of the first algorithm it decreased by 79.5% and for the second – by 78.6%, which indicates a significant increase in performance. Future studies will consider the problems of choosing a solution to the forward kinematics from several solutions obtained, as well as the use of not only calculations on the CPU, but also the use of GPU resources. The obtained results can be used in selecting structural parameters of the hexapod robot, in solving problems of optimizing its trajectory, as well as in writing control programs.

Acknowledgements
This work was supported by the Ministry of Science and Higher Education of the Russian Federation, the agreement number 075-11-2021-060 of June 24, 2021 "Development of a high-tech production of robotic 6DOF mobility platforms for trainers and simulators for the aviation and rocket-space industry" (unique number 000000S407521QLB0002).

References
[1] Kong H, Gosselin C M 2007 Type Synthesis of Parallel Mechanisms (Springer)
[2] Merlet J-P 2007 Parallel Robots 2nd ed. (Springer)
[3] Rybak L, Khalapyan S, Gaponenko E 2018 Issues of planning trajectory of parallel robots taking into account zones of singularity *IOP Conference Series-Materials Science and Engineering* **327** 042092

[4] Laryushkin P A, Heilo S V, Chan K N, Glazunov V A 2011 Geometric interpretation of a direct problem on the positions of a parallel manipulator with three degrees of freedom *MIKMUS – 2011 XXIII International Innovative Conference of Young Scientists and Students* 114

[5] Rybak L, Gaponenko E, Malyshev D, Virabyan L 2019 The algorithm for planning the trajectory of the 3-RPR robot, taking into account the singularity zones based on the method of non-uniform covering *IOP Conf. Ser.: Mater. Sci. Eng.* **489** 012060

[6] Geng Z, Haynes L S 1992 Neural network solution for the forward kinematics problem of a Stewart platform *Robot Comput Integr Manuf.* **9**(6) 485–495

[7] Yee C S, Lim K B 1997 Forward kinematics solution of Stewart platform using neural networks *Neurocomputing* **16**(4) 333–349

[8] Evtushenko Y, Posypkin M, Rybak L, Turkin A 2018 Approximating a solution set of nonlinear inequalities. *J. Global Optimization* **7** 129–145

[9] Evtushenko Y, Posypkin M, Turkin A, Rybak L 2017 The non-uniform covering approach to manipulator workspace assessment. In: *Proceedings of the 2017 IEEE Russia Section Young Researchers in Electrical and Electronic Engineering Conference, ElConRus 2017* pp 386–389

[10] Malyshev D I, Posypkin M A, Gorchakov A Yu, Ignatov A D 2019 Parallel algorithm for approximating the work space of a robot. *International Journal of Open Information Technologies* **7**(1) 1–7