Application of the symbolic regression method for solving the inverse problem of control of the 3-RPR platform

A I Diveev¹, L A Rybak², D I Malyshev² and M A Averbukh²

¹Federal Research Center “Computer Science and Control” of Russian Academy of Sciences, 44, Vavilova str., 119333, Moscow, Russia
²Belgorod State Technological University named after V. G. Shukhov, 46, Kostyukova str., 308012, Belgorod, Russia

E-mail: aidiveev@mail.ru, rl_bgtu@intbel.ru

Abstract. The paper is devoted to finding the functional dependence of the position of the 3-RPR platform on the control. In this construction, the controls are the lengths of the brackets attached to the corners of the platform. To solve the problem, one of the methods of symbolic regression the network operator method is used. Initially we got a training sample. The coordinates of the corners of the platform are found for the set of control by a numerical evolutionary PSO algorithm. Then, at the second stage, approximation of the obtained data was the network operator method. As a result, we obtain six functional equations that describe depending on the coordinates of the platform angles on the plane, on the values of the lengths of the control leashes. The solution of the inverse problem was performed for each coordinate of the platform angle separately. The feature of this task is that for the same value of control the platform has several provisions. To account for this property, an additional parameter is introduced into the 3-RPR platform control model, which takes an integer value. Each value of the additional parameter corresponds to one position.

1. Introduction

Robots of a parallel structure are mechanisms that quickly enough, due to parallel control, move an object to a given point in space [1–4]. The object of control for such robots is a certain working platform on which working tools or other mechanisms are mounted. Depending on the constructive implementation of the robot, control mechanisms can take various forms. One of the problems arising in the management of such robots is the determination of the position of the object or the working platform from the values of the control bodies. This paper discusses the mechanism that controls the triangular working platform. To move it, the design has two hinged joints and one reciprocating mechanism on each leg (Figure 1). The type of such construction is denoted 3-RPR. This means that the mechanism has 3 connections consisting of a hinge, which corresponds to the designation "R", a reciprocating mechanism, which corresponds to the designation "P" and again a hinge "R".

One of the problems of controlling such a mechanism is to determine the position of the working platform in space by the lengths of the reciprocating joint. The problem is further complicated by the fact that with the same lengths of reciprocating joint, the working platform can occupy several different positions in space. The situation with the control of this object can be significantly improved if we obtain mathematical expressions describing the relationship between the value of governing bodies, in this case, the lengths of the reciprocating connection, and the position of the working platform, i.e.
coordinates of the corners of the site. Analytically, this mathematical expression is rather difficult to
derive, and most likely it cannot be deduced for sure. To solve this problem, we use the symbolic regression numerical method, the network operator method [5]. Any method of symbolic regression allows you to find in coded form a mathematical expression using an evolutionary algorithm. A distinctive feature of the network operator method from other symbolic regression methods is that it encodes a mathematical expression in the form of a directed graph and uses the principle of small variations of the basic solution to find a solution.

In the task at hand, we initially numerically find the set of different positions for the set of controls. Here there are some positions for one value of control. To solve this problem, we introduce an additional abstract control parameter, and when searching, in the case of falling into the same position, we penalize the goal function if this position was found earlier. When constructing a set of controls and positions for the numerical solution of the problem, we use an evolutionary algorithm of a particle swarm optimization. At the second stage, for approximation of the obtained set the method of symbolic regression is used. We find analytic relations describing the relationship between the values of the controls and the position of the working platform.

2. Statement of the problem
We suppose that in the construction, shown in Figure 1, the coordinates of points A, B, C are fixed. It is necessary to find the coordinates of points a, b, c, depending on the lengths of the legs $l_1$, $l_2$, $l_3$, provided that the leg $l_1$ connects points a and A, the leg $l_2$ connects points b and B, and the leg $l_3$ connects points c and C. Points a, b, c are vertices of an equilateral triangle with side $l^*$, whose magnitude does not change.

The task is to find an analytic relationship between the lengths of the legs and the coordinates of the points a, b, c on the plane:

$$x_a = f_{2i-1}(l_1, l_2, l_3), \quad y_a = f_{2i}(l_1, l_2, l_3)$$

where $a = a, b, c, i = 1, 2, 3$.

![Figure 1. 3-RPR platform](image)

The basic idea of solving this problem is to initially obtain numerically a set of solutions, i.e. for the set of lengths of the $l_1$, $l_2$, $l_3$ lengths, find the coordinates $x_a, y_a, x_b, y_b, x_c, y_c$.

This set of solutions in terms of the apparatus of neural networks is called a training set. Then we approximate the numerical data by symbolic regression to obtain analytic dependencies. We can approximate the same data with a neural network, but this will not get analytic dependencies. We assume that if the training sample is sufficiently representative, or it is a big and it cover a wide range of variation of control parameters, as well as it has high accuracy of approximation, then the obtained analytic dependencies can be considered the problem for using in practical calculations.
3. Constructing a training set
To obtain a numerical solution of the problem, we use the evolution algorithm of particle swarm optimization (PSO-algorithm) [6]. This algorithm does not require the unimodality of the objective function and does not calculate the gradient of this function during the search.

The solution of the problem is the vector of parameters:

$$q = [q_1 \ldots q_n]^T$$

where $q_1 = x_o, q_2 = y_o, q_3 = x_p, q_4 = y_p, q_5 = x_r, q_6 = y_r$. 

The following goal function is used:

$$f_i(q) = |(q_i - x_o)^2 + (q_2 - y_o)^2 - l_1| + |(q_3 - x_p)^2 + (q_4 - y_p)^2| - l_2| + |(q_5 - q_1)^2 + (q_6 - q_2)^2 - l_3| +$$

$$+ |(q_5 - q_1)^2 + (q_6 - q_2)^2 - l_4| + |q_5 - q_3|^2 + |q_6 - q_4|^2 - l_5|$$

As can be seen from the form of writing the objective function, it cannot have a value less than zero and its zero value is possible only with the exact solution of the considered problem.

Further, in the course of a computational experiment, it was found that for the same values of the control parameters $l_1, l_2, l_3$, the objective function (3) can have more than one solution providing the zero value of the objective function. Therefore, a penalty function was added to the optimization problem (3), which increased the value of the objective function if the solution sought did not differ by more than from one of the previously found solutions. Thus, the problem was solved repeatedly, after finding a new solution for certain values of the control parameters $l_1, l_2, l_3$ it was added to the set of solutions $Q(l_1, l_2, l_3)$:

$$Q(l_1, l_2, l_3) \leftarrow Q(l_1, l_2, l_3) \cup \{q\}$$

which was originally empty $Q = \emptyset$.

As a result, the following goal function was used:

$$f_i(q) = f_i(q) + s(1 - \mathcal{H}(\varepsilon - \frac{\sum q_i}{n}))$$

where $s$ is a penalty, $\varepsilon$ is a small positive value, $q^*_i$ is a previously found solution, $q^*_i \in Q(l_1, l_2, l_3), \mathcal{H}(A)$ is Heaviside function:

$$\mathcal{H}(A) = \begin{cases} 1, & A > 0 \\ 0, & \text{otherwise} \end{cases}$$

4. PSO algorithm
The particle swarm optimization algorithm is one of the most popular evolutionary algorithms today. In the algorithm, the evolution of each possible solution is carried out on the basis of the values of the best possible solution found so far, the best among several randomly selected ones, and the information about the best solutions found earlier is also used.

The algorithm contains the following steps. Generation of a set of possible solutions, called population in evolutionary calculations:

$$q_j^* = \xi(q_j^+ - q_j^-) + q^*_j, j = 1, \ldots, H, i = 1, \ldots, p,$$

where $\xi$ is a random number in interval from 0 to 1, $q_j^+$, $q_j^-$ are maximal and minimal values of parameter $q_j$, $H$ is a cardinal number of the initial population, $p$ is a dimension of the vector of parameters.

Generation of the same set of historical vectors, which are also called velocity $s$. The initial values of the historical vectors are zero.

$$v_j^* = 0, j = 1, \ldots, H, i = 1, \ldots, p,$$

calculating the values of the objective function (5) for each possible solution from the population.
\[ f_j = \tilde{f}(q^j), j = 1, \ldots, H. \]  

Determining the best possible solution \( q^j \) at the moment:

\[ f(q^j) = \min\{f_j : j = 1, \ldots, H\}. \]  

Next, for each possible solution \( q^j \), we find the best possible solution \( q^{*(j)} \) among \( k \) possible solutions randomly selected from the population:

\[ f(q^{*(j)}) = \min\{f_{i1}, \ldots, f_{ik}\}. \]  

We change the historical vector \( \nu^j \):

\[ \nu^j \leftarrow \alpha \nu^j + \xi \beta (q^j - q^j) + \xi \gamma (q^{*(j)} - q^j), i = 1, \ldots, p, \]  

where \( \alpha, \beta, \gamma \) are parameters of the algorithm.

For each possible solution \( q^j \), we determine its evolution \( q' \):

\[ q' = \begin{cases} q^j, & \text{if} q^j + \sigma v^j > q^j; \\ q^j, & \text{if} q^j + \sigma v^j < q^j; \\ q^j + \sigma v^j - \text{otherwise} & \end{cases} \]  

We calculate the value of the objective function for the evolution of a possible solution:

\[ \tilde{f}_j = f(\tilde{q}(q)) \]  

if the value of the objective function for the evolution of a possible solution is better than the value of the objective function for a possible solution, then we replace the possible solution with its evolution:

\[ q^j \leftarrow q' \text{ if } f_j < f_j. \]  

We perform the evolution (11) - (15) for each possible solution from the population and repeat this process together with the search for the best current possible solution (10) \( P \) times. After this, the best solution in the population is considered the solution to the problem.

In a computing experiment on creation of the training selection the following parameters of an algorithm were used: \( H = 200, P = 1800, p = 6, s = 10, \alpha = 0.729, \beta = 0.1, \gamma = 0.85, \sigma = 1. \)

Results of a computing experiment are given in table 1. As the additional managing director of parameter \( m \) which accepts two values 0 or 1, determining the position of the platform specified.

5. The network operator method

Further we approximate the data obtained as a result of a computing experiment analytic expressions:

\[ x_u = f_1(l_1, l_2, l_3, m), y_u = f_2(l_1, l_2, l_3, m) \]
\[ x_y = f_3(l_1, l_2, l_3, m), y_y = f_4(l_1, l_2, l_3, m). \]  

For this purpose, we use the network operator method [4, 5]. To code a mathematical expression by the network operator it is necessary to set some sets. A set of arguments of the mathematical expression:

\[ F_0 = (f_{01} = x_1, \ldots, f_{0u} = x_u, f_{0u+1} = q_1, \ldots, f_{0u+p} = q_p). \]  

A set of elementary function with one argument:

\[ F_1 = (f_{11}(z) = z, \ldots, f_{1w}(z)). \]  

This set surely includes the identity function \( f_{1,1}(z) \).

A set of functions with two arguments:

\[ F_2 = (f_{21}(z_1, z_2), \ldots, f_{2w}(z_1, z_2)). \]  

All functions with two arguments are \( f_{2,k}(z_1, z_2) = f_{2,k}(z_1, z_2) \), distributive \( f_{2,k}(f_{2,k}(z_1, z_2), z_3) = f_{2,k}(z_1, f_{2,k}(z_2, z_3)) \), and have unit element \( f_{2,k}(z, e_k) = z \), where \( e_k \) is a unit element for function \( f_{2,k}(z_1, z_2, k = 1, \ldots, V). \)
Firstly, we write mathematical expression in form of composition of elements of sets (17)–(19). To construct oriented computing graph by the record of the mathematical expression in the form correct composition of functions. At construction the graph we use the followings rules:

1) arguments of the mathematical expression are connected with source nodes of the graph;
2) functions with two arguments are connected with remain nodes of the graph;
3) functions with one argument are connected with arcs of the graph and an arc direct from the node that is connected with the argument of this function to a node that is connected with the function with two arguments one of that is the function connected with this arc. Let it set a mathematical expression:

\[ y = f(x_1, x_2, q_i) = x_1 + \cos(q_i x_2^2)). \]  

Mathematical sets of function (1)–(3) have to include the following elements:

\[ F = \{ f_0 = x_1, f_0 = x_2, f_0 = q_i, f_1 = (f_{1,1}(z) = z, f_{1,2}(z) = z_1 + z_2 f_{2,2}(z_1, z_2) = z_1 z_2) \}; \]

an image of a mathematical expression (19) as a composition of elements of the sets

\[ \psi = f_{2,1} \circ f_{0,1} \circ f_{1,2} \circ f_{2,2} \circ f_{0,3} \circ f_{1,3} \circ f_{0,2}. \]

We correct the composition according to rules 1–3

\[ y = f_{2,1} \circ f_{1,1} \circ f_{0,1} \circ f_{1,2} \circ f_{2,2} \circ f_{1,3} \circ f_{0,3} \circ f_{1,3} \circ f_{0,2}. \]

The computation graph of the mathematical expression is presented in Figure 2. Write numbers of functions instead of elements of the sets and arguments of the mathematical expression in the source-nodes. Then numerate nodes of the graph such that number of node where from arc goes out was less than number of node where arc comes in. In result we obtained a graph of network operator, that is presented in Figure 3.

![Figure 2. The computation graph of the mathematical expression](image)

![Figure 3. The computation graph of the mathematical expression](image)

To keep the network operator in memory of computer use the network operator matrix. Write the adjacency matrix for the graph of the network operator in Figure 3.

\[ A = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 
\end{bmatrix} \]

Change ones in the adjacency matrix onto numbers of corresponding functions with one argument and stand numbers of functions with two arguments on the diagonal of the matrix in the rows corresponding by nodes of the network operator graph. As result we obtain matrix of the network operator.

\[ \varphi = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 3 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 2 & 2 & 0 \\
0 & 0 & 0 & 0 & 1 
\end{bmatrix} \]

To calculate result of the mathematical expression by the network operator it is necessary to know how arguments of the mathematical expression are connected with nodes of the graph, or to know correspondence between numbers of rows of the matrix of network operator and elements of the set of arguments. Let all first numbers of the nodes correspond arguments of the mathematical expression.
6. A solution of inverse problem for 3RPR-platform

To find each function one column from the table was used. Formulas were rather bulky. Let’s give in quality an example of a formula for \( x_a \):

\[
x_a = p_{15}(E_1) + p_{14}(D_1) + p_{13}(x_a, x_b, (0.5A, p_b(r_z) r_z l_b, p_0(z_3)), p_0(z_1)).
\]

where

\[
E_1 = \rho_{26}(D_1) = \rho_{15}(E_1) + \chi_6(x_6(0.5A, p_b(r_z) r_z l_b, p_0(z_3)), p_0(z_1)) + \rho_7(\chi_6(0.5A, p_b(r_z) r_z l_b, p_0(z_3)), p_0(z_1)) + 2 \rho_7(\chi_6(0.5A, p_b(r_z) r_z l_b, p_0(z_3)), p_0(z_1)),
\]

\[
C_i = \chi_i(\rho_8(\chi_3(B_i) + \rho_{17}(\rho_4(r_z l_z, -r_z)), \rho_9(\chi_6(0.5A, p_b(r_z) r_z l_b, p_0(z_3)), p_0(z_1)))),
\]

\[
B_i = \chi_i(-A_i, \chi_6(A_i, \rho_{22}(r_z, r_y, l_z, r_z + r_y + l_z, \rho_{10}(\rho_4(r_z l_z, -r_z)), \rho_6, \rho_{13}(r_z \rho_3(z_1)) \rho_0(z_1)) + 2 \rho_7(\chi_6(0.5A, p_b(r_z) r_z l_b, p_0(z_3)), p_0(z_1)) + 2 \rho_7(\chi_6(0.5A, p_b(r_z) r_z l_b, p_0(z_3)), p_0(z_1)),
\]

\[
A_i = \chi_i(\rho_8(\chi_3(B_i) + \rho_{17}(\rho_4(r_z l_z, -r_z)), \rho_9(\chi_6(0.5A, p_b(r_z) r_z l_b, p_0(z_3)), p_0(z_1)) + 2 \rho_7(\chi_6(0.5A, p_b(r_z) r_z l_b, p_0(z_3)), p_0(z_1))),
\]

\[
\rho_2(z) = \begin{cases} 
1 & \text{if } z > 0 \\
0 & \text{otherwise} 
\end{cases}.
\]

Other formulas have a similar appearance and here is not brought because of economy of the place.

The network operator method codes formulas in the form of an integer matrix therefore for calculation of a formula, it should not be presented in the analytic form. For calculation on the computer it is enough to use the code of a formula in the form of a matrix.
Results of modeling of the received mathematical expression together with experimental data are given in figures 4–9. In the figures, red squares indicate the data taken from the table 1, black lines show the results of calculations on the mathematical expression obtained by the network operator method. As we can see from the obtained graphs, the accuracy of approximation is quite high. In all experiments, the calculations were stopped when the standard deviation error of 0.1 was reached.

7. Conclusion
The machine solution of the return task of management 3RPR robot is considered. Application of a method of symbolic regression allowed us to find the analytic expressions describing dependence of position of the platform on management. The received expressions have rather difficult appearance and demand simplification.

8. Acknowledgements
This work was supported by the Russian Science Foundation, agreement number 16-19-00148, using equipment of High Technology Center at BSTU named after V.G. Shukhov.

References
[1] Rybak L A, Mamaev Yu A, Malyshev D I and Virabyan L G 2016 Software module for realizing a given trajectory of motion of the output link of a robot-hexapod for 3D printing of products Bulletin of BSTU named after V. G. Shukhov 8 155–164
[2] Geng Z and Haynes L S 1992 Neural network solution for the forward kinematics problem of a Stewart platform Robot Comput. Integr. Manuf. 9(6) 485–495
[3] Yee C S and Lim K B 1997 Forward kinematics solution of Stewart platform using neural networks Neurocomputing 16(4) 333–349
[4] Chablat D, Wenger Ph and Bonev I A 2006 Self-motions of a special 3-RPR planar parallel robot Advances in Robot Kinematics 221–228 (Springer)
[5] Diveev A I 2012 A numerical method for network operator for synthesis of a control system with uncertain initial values J. of Computer and Systems Sciences Int. 51(2) 228–243
[6] Kennedy J and Eberhart R 1995 Particle swarm optimization Proc. of the IEEE Int. Conf. on Neural Networks IV (Perth) 1942–1948