ISSN: 0975-766X
CODEN: IJPTFI
Available Online through *Research Article*
www.ijptonline.com

ON NEURAL NETWORK MODEL DEVELOPMENT TO SOLVE PARALLEL ROBOTS KINEMATICS AND CONTROL PROBLEMS

Sergey Yuriyevich Khalapyan, Larisa Alexandrovna Rybak, Anton Igorevich Glushchenko, Yuri Aleksandrovych Mamaev
Stary Oskol Technological Institute (branch) n.a. A.A. Ugarov
NUST MISiS, Russia, 309516, Stary Oskol, mkr. Makarenko, 42.
Belgorod State Technological University n.a. V.G. Shukhov
Russia, 308012, Belgorod, Kostyukova str., 46.
Stary Oskol Technological Institute (branch) n.a. A.A. Ugarov
NUST MISiS, Russia, 309516, Stary Oskol, mkr. Makarenko, 42.
Belgorod State Technological University n.a. V.G. Shukhov
Russia, 308012, Belgorod, Kostyukova str., 46.

Received on 14-08-2016 *Accepted on 20-09-2016*

Abstract.

A problem of a parallel structure mechanism control implementation is considered. Such mechanisms could be controlled by a classical feedback based automatic control systems. But this feedback signal needs to be calculated in dependence with mechanism drive links movement values. So, the direct kinematic problem must be solved. Having made the research by the example of a robot-machine, we believe that this task is rather complicated for a class of machines, for which different movable platform positions may relate to the one drive links state. However, a neighborhood where a solution should be searched for could be determined, because the data about the previous output drive link position are known. It is shown that if a neural network is applied to solve direct kinematic problem, such information used as its inputs could improve solution accuracy significantly. The output link position error reduction leads both to its positioning accuracy and speed improvement. The obtained neural network based tripod model could be used both to solve direct kinematic and control problems for the mechanism under consideration. The proposed principles of the model development could be used to solve a problem of other parallel structure mechanisms automation.

Keywords: parallel structure mechanism, robot-machine, tripod, automatic control system, direct kinematic problem, neural network based model, solution neighborhood.

Introduction

In these days, parallel robotics machine tools [1, 2] are becoming more and more widely spread in industry. This fact could be explained, first of all, by their significant advantage over more often used today serial robots as far as positional precision, powerfulness and allowable load are concerned.

Such mechanisms could be controlled by automatic control systems based on classical principles [3]. Error term control implementation needs to know plant output value in real time, i.e. the mechanism output link coordinates and spatial orientation. In order to achieve that a position sensor might be applied, but its usage for a mobile platform is a really difficult problem. So, a feedback signal calculation is to be done on the basis of information about drive links movement, i.e. direct kinematics analysis problem is to be solved.

The mentioned above problem complexity could be explained by the necessity of nonlinear equations system solving. A lot of different ways to overcome this problem have been developed as a result of scientific research. Methods based on neural networks [4-6] should be distinguished among them. Such networks have an ability to find complex dependencies between input and output data and approximate them. These advantages allow neural networks to calculate the accurate result on the basis of input data, which are absent in a training set, incomplete or noisy.

Neural networks usage gives an opportunity to calculate mechanism output link coordinates in real time mode. This is especially important for a control problem solving. Both a sensor response time and its instrumental error decrease often allow to increase control quality. So, neural network performance and accuracy improvement is an actual problem.

It should be mentioned that direct kinematic problem solving ambiguity is typical for a range of parallel structure mechanisms [7]. That means that different output link coordinates may correspond with one certain drive links state. So, current coordinates identification of movable platform using such mechanisms becomes even more difficult, because the “right” solution of the direct kinematic problem is needed.

Research

Let’s consider neural network usage to solve mentioned above problem via example of tripod [3] – a machine with three bars of adjustable length with three degrees of freedom (fig.1).

Firstly, we have tried to solve this problem as is commonly done when neural networks are used. The network was trained to calculate mechanism output coordinates on the basis of input coordinates only. The known equations of tripod inverse kinematic problem (1-3) [3] were used to do that.

$$l_1 = R(\cos\vartheta - 1)e_1 + R\sin\varphi\cos\vartheta\, e_2 + (L + z - R\cos\varphi\sin\vartheta)e_3 \qquad (1)$$

$$l_2 = \frac{1}{2}R(1-\cos\vartheta)e_1 + \frac{1}{2}R\left(\sqrt{3}\cos\varphi - \sin\varphi\sin\vartheta - \sqrt{3}\right)e_2 + \\ + \left(L + z + \frac{1}{2}R\cos\varphi\sin\vartheta + \frac{\sqrt{3}}{2}R\sin\varphi\right)e_3 \qquad (2)$$

$$l_3 = \frac{1}{2}R(1-\cos\vartheta)e_1 + \frac{1}{2}R\left(-\sqrt{3}\cos\varphi - \sin\varphi\sin\vartheta + \sqrt{3}\right)e_2 +$$

$$+\left(L + z + \frac{1}{2}R\cos\varphi\sin\vartheta - \frac{\sqrt{3}}{2}R\sin\varphi\right)e_3 \qquad (3)$$

where R – radius of a circle circumscribed around worktable, L – distance between platforms in their initial state, z – worktable movement along a vertical axis, [phi] and [theta] – movable platform rotation angle along mutually perpendicular axes x and y, e_1, e_2, e_3 – basis vectors, l_1, l_2, l_3 – bars length vectors.

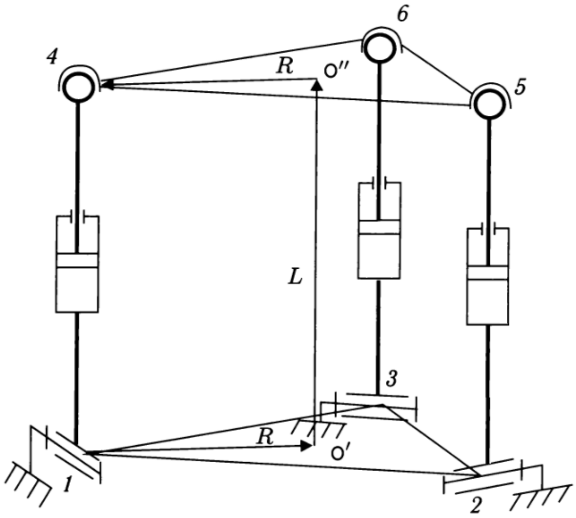


Fig. 1 Tripod diagram

Scalar bars lengths are equal as basis vectors are mutually perpendicular to each other:

$$|l_1| = \sqrt{\left(R(1-\cos\vartheta)\right)^2 + (R\sin\varphi\cos\vartheta)^2 + (L + z - R\cos\varphi\sin\vartheta)^2} \qquad (4)$$

$$|l_2| = \frac{1}{2}\Big(\left(R(1-\cos\vartheta)\right)^2 + \left(R\left(\sqrt{3}(\cos\varphi - 1) - \sin\varphi\sin\vartheta\right)\right)^2 +$$

$$+\left(2(L+z) + R\left(\cos\varphi\sin\vartheta + \sqrt{3}\sin\varphi\right)\right)^2\Big)^{\frac{1}{2}} \qquad (5)$$

$$|l_3| = \frac{1}{2}\Big(\left(R(1-\cos\vartheta)\right)^2 + \left(R\left(\sqrt{3}(-\cos\varphi + 1) - \sin\varphi\sin\vartheta\right)\right)^2 +$$

$$+\left(2(L+z) + R\left(\cos\varphi\sin\vartheta - \sqrt{3}\sin\varphi\right)\right)^2\Big)^{\frac{1}{2}} \qquad (6)$$

Let L = 100 cm, R = 30 cm and z could be changed from -20 to 20 cm by step of 2 cm, [phi] and [theta] – from $-\pi/2$ to $\pi/2$ by step of $\pi/20$. Then a training set of 21^3 = 9 261 examples was obtained with the help of equations (4-6). Testing set of needed volume was formed by repetitive pseudorandom selection of z, [phi] and [theta] values from the same intervals.

A feedforward multilayer neural network structure was used to develop neural network tripod model (fig. 2). l_1, l_2 and

l_3 were used as the model inputs, whereas z, [phi] and [theta] – as the model outputs.

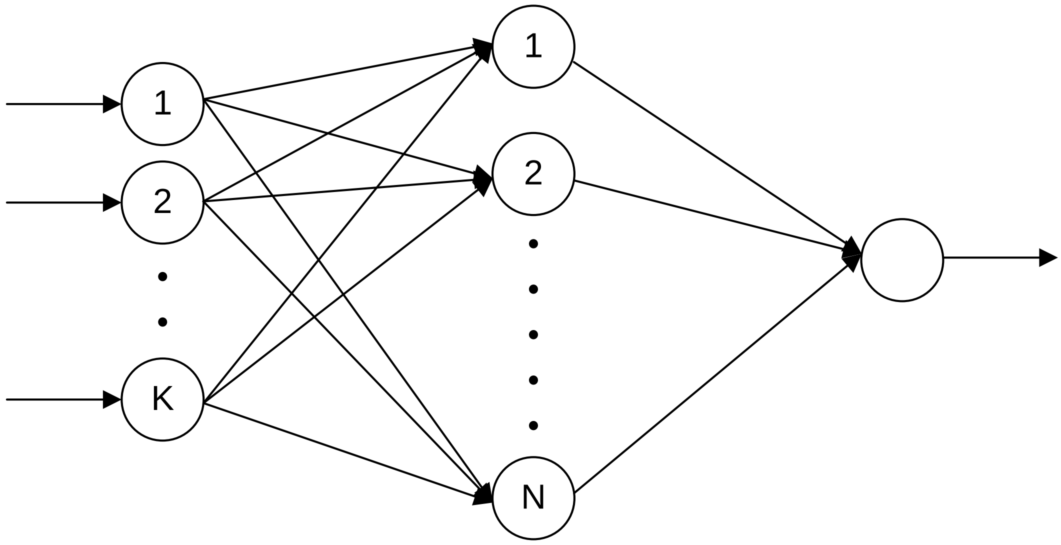


Fig. 2 Feedforward multilayer neural network structure

Both conducted experiments and existing experience of neural network based models [8-10] show that their performance and accuracy could be improved if such models were implemented as a set of neural networks, each of which has only one output neuron responsible for only one parameters estimation.

The problem of neural network structure selection was considered. Sigmoid function and hyperbolic tangent were used as an activation function of a hidden layer since they were believed to approximate nonlinear dependencies in the best way. Linear activation function was used in an output layer as output signals measured in cm and rad were within the limits [-20 cm; 20 cm] or [-π/2; π/2].

The model synthesis was made by the neural networks training. Their main parameters values are as follows:

- number of hidden layers – 1 or 2;
- number of neurons in a hidden layer, which were found as a result of brute force procedure from the range of [1:100], step size – 1;
- activation function of a hidden layer was chosen from mentioned above alternatives in order to take into consideration the plant nonlinearity.

The Leven berg-Marquardt algorithm [11] was used to train the neural network. This process was arranged as a repetitive (up to 1000 epochs) showing training samples to the network.

Mean square error was used as a training performance function. The error was the difference between the network output and values from the training and testing sets.

The tripod model was obtained as a result of the network training. As it had been expected, its approximation ability was rather low. The modeling error curves are shown in fig.3. The modeling error was the difference between the real z, [phi] and [theta] values and their estimations made by the model using the neural network, which showed the best result for the testing set.

Having analyzed the curves, the conclusion could be made that in some cases z, [phi] and [theta] values obtained from the model differ significantly from their real values. The root-mean-square deviation was calculated as follows:

- 2.1006% – for z value estimation;
- 5.5595% – for [phi] value estimation;

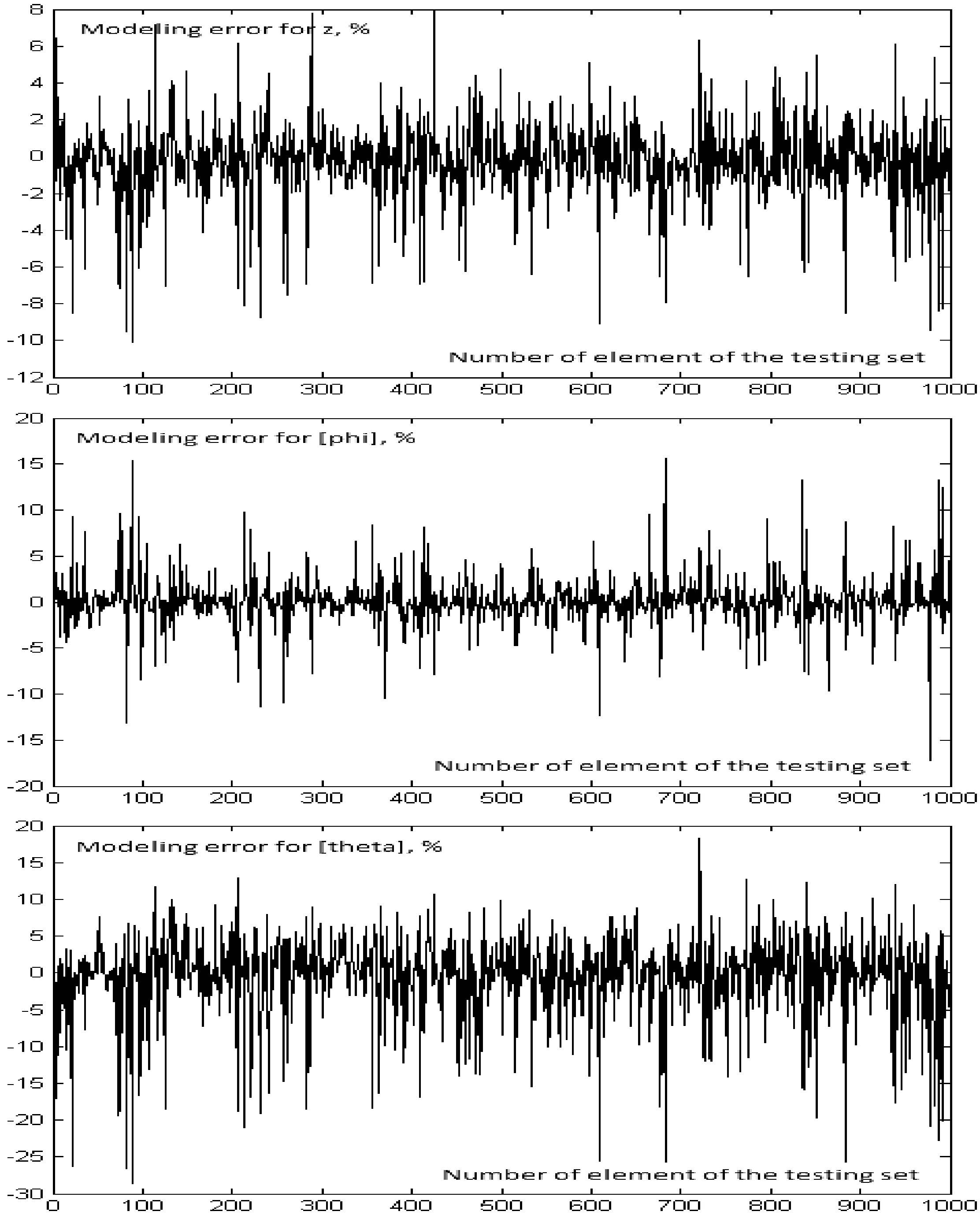


Fig. 3 Modeling error for z, [phi] and [theta] for the best network with 3 inputs

- 2.6167% – for [theta] value estimation.

This result was caused by the mentioned above fact that tripod is a mechanism which direct kinematic problem has ambiguous solution.

For an instance, let $z = -20$ cm, $\varphi = \theta = -\pi/2$. Using equations (4-6), the following result is obtained $l_1 = 90.6$ cm, $l_2 = 69.4$ cm, $l_3 = 107.6$ cm. But the same result could be obtained from another initial condition (output link state): $z = -11.113$ cm, $\varphi = -0.835$, $\theta = -0.082$. The training set contained only the first of the given examples. So, it was not worth waiting that the neural network would be able to react properly to the second one. Besides the training set could contain incompatible examples.

It should be mentioned that if the output link coordinates are defined regularly and with high frequency during the control process, then the "right" direct kinematic problem solution choice is not so difficult task. Little output coordinates change corresponds with little drive links movement. So, the solution needed is in the limited neighborhood of the movable platform previous position.

Therefore, the probability of the correct movable platform coordinates calculation will improve if additional neurons are included in the input layer. Their inputs are previous output link coordinates. There is also an idea to include neurons, which input signals are the bar length or drive links movements.

9 261 additional training samples were calculated on the base of (4-6) equations in order to check mentioned above hypothesis. These samples were chosen pseudorandomly from the neighborhood of each initial training set sample. The neighborhood width was determined according to the movable platform coordinates: each coordinate z, [phi] and [theta] change was in limits of ±1% from the allowance range of the respective parameter.

Each of the additional samples modeled the previous platform position, which is known according to the experiment conditions. Existence of such samples allowed to add the following input neurons to the neural network:

- z_0, $[phi]_0$ and $[theta]_0$ – previous mechanism output link position (coordinates);
- l_{10}, l_{20}, l_{30} – respective bar lengths (or $[delta]l_1$, $[delta]l_2$, $[delta]l_3$ – their increments).

So, the approximation ability of the neural networks with the following sets of the input neurons needs to be checked:

- l_1, l_2, l_3, z_0, $[phi]_0$, $[theta]_0$ (6 inputs);
- l_1, l_2, l_3, z_0, $[phi]_0$, $[theta]_0$, l_{10}, l_{20}, l_{30} or l_1, l_2, l_3, z_0, $[phi]_0$, $[theta]_0$, $[delta]l_1$, $[delta]l_2$, $[delta]l_3$ (9 inputs).

It should be mentioned that not only could z, [phi] and [theta] values be used as the network outputs, but also their increments [delta]z, [delta][phi] and [delta][theta] from the previous mechanism output link position usage was sensible. But in that case eventually cumulative measurement error may appear in the control system.

Training set forming is made in accordance with the algorithm shown in fig.4.

Modeling error curves for the neural networks with different numbers of input neurons and best results on testing set are shown in fig.5 and 6. The conclusion could be made that parameters estimation accuracy improved significantly in comparison with curves shown in fig.3. The same conclusion is appropriate according to root-mean square deviation, which has the following values for the networks with six and nine inputs respectively:

- for z value– 0.0994 and 0.0969%;
- for [phi] value – 0.1146 and 0.1000%;
- for [theta] value – 0.2735 and 0.2526%.

A comparison of the mentioned above values makes it possible to conclude that the neural network with nine inputs achieved a better result. At the same time usage of [delta]l_1, [delta]l_2, [delta]l_3 instead of l_{10}, l_{20}, l_{30} did not change obtained results. Another conclusion is that trained neural network performance with nine inputs decreased insignificantly in comparison with the network with six inputs. So, it is worth using network with nine inputs to identify the tripod model because of its higher accuracy.

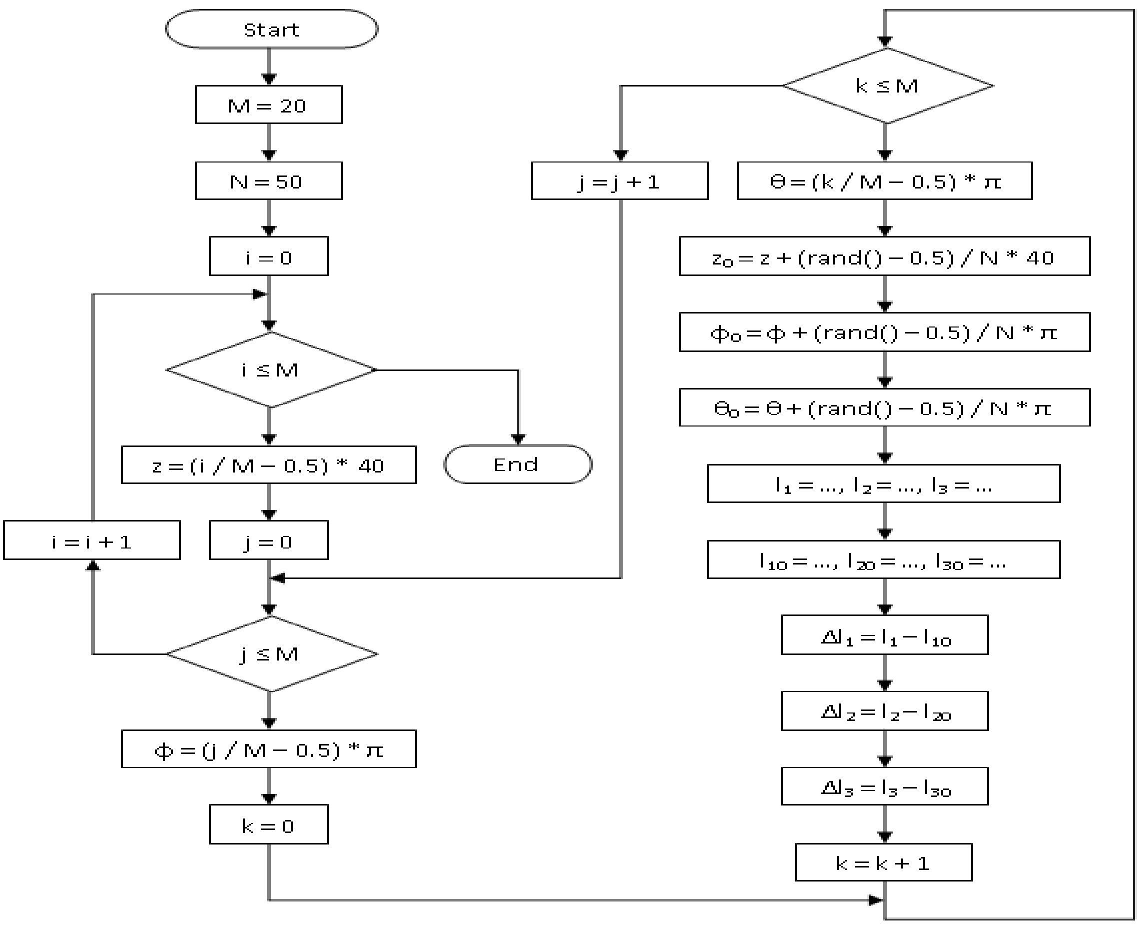
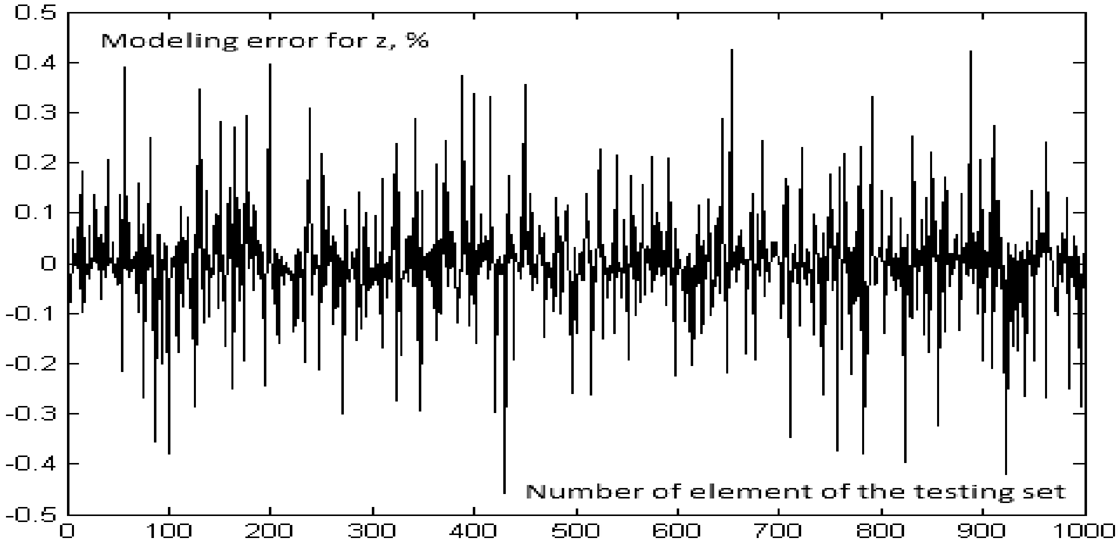
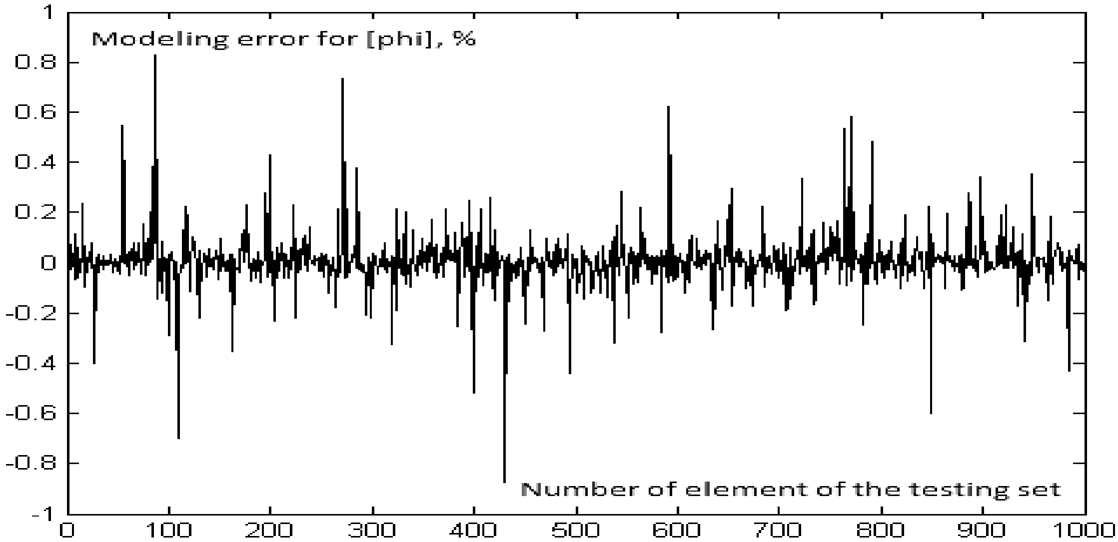


Fig. 4 Algorithm of forming of training set.

Conclusion

Taking everything into consideration, the neural network based tripod model is developed as a result of the network structure optimization and the best structure choice. This model consists of three feedforward neural networks with one hidden layer, nine input neurons and one output neuron. Sigmoid activation function is used in the hidden layer and linear – in the output layer. The number of hidden layer neurons for all three networks is respectively equaled to:





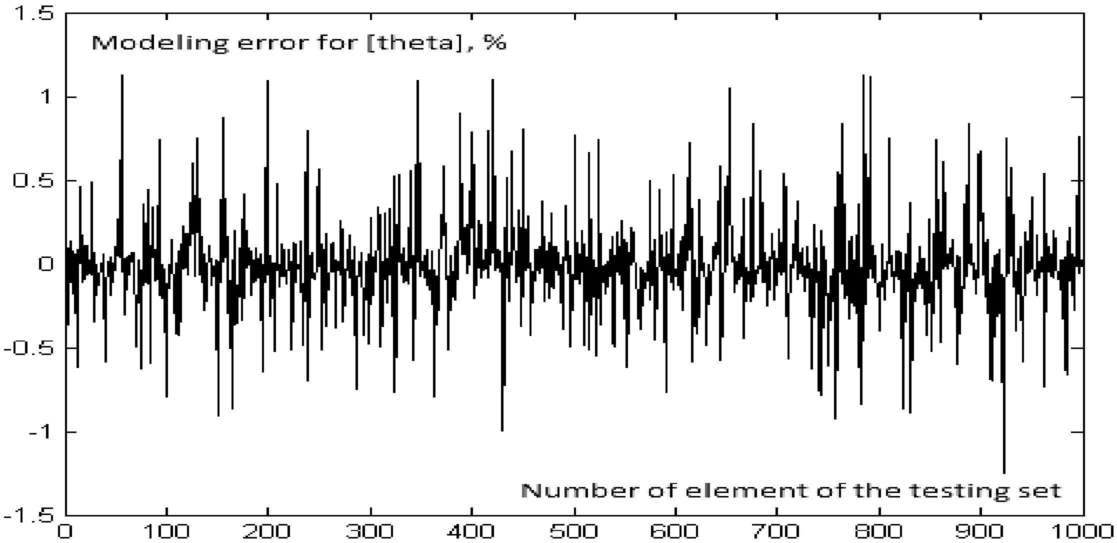


Fig. 5 Modeling error of z, [phi] and [theta] for the best network with 6 inputs.

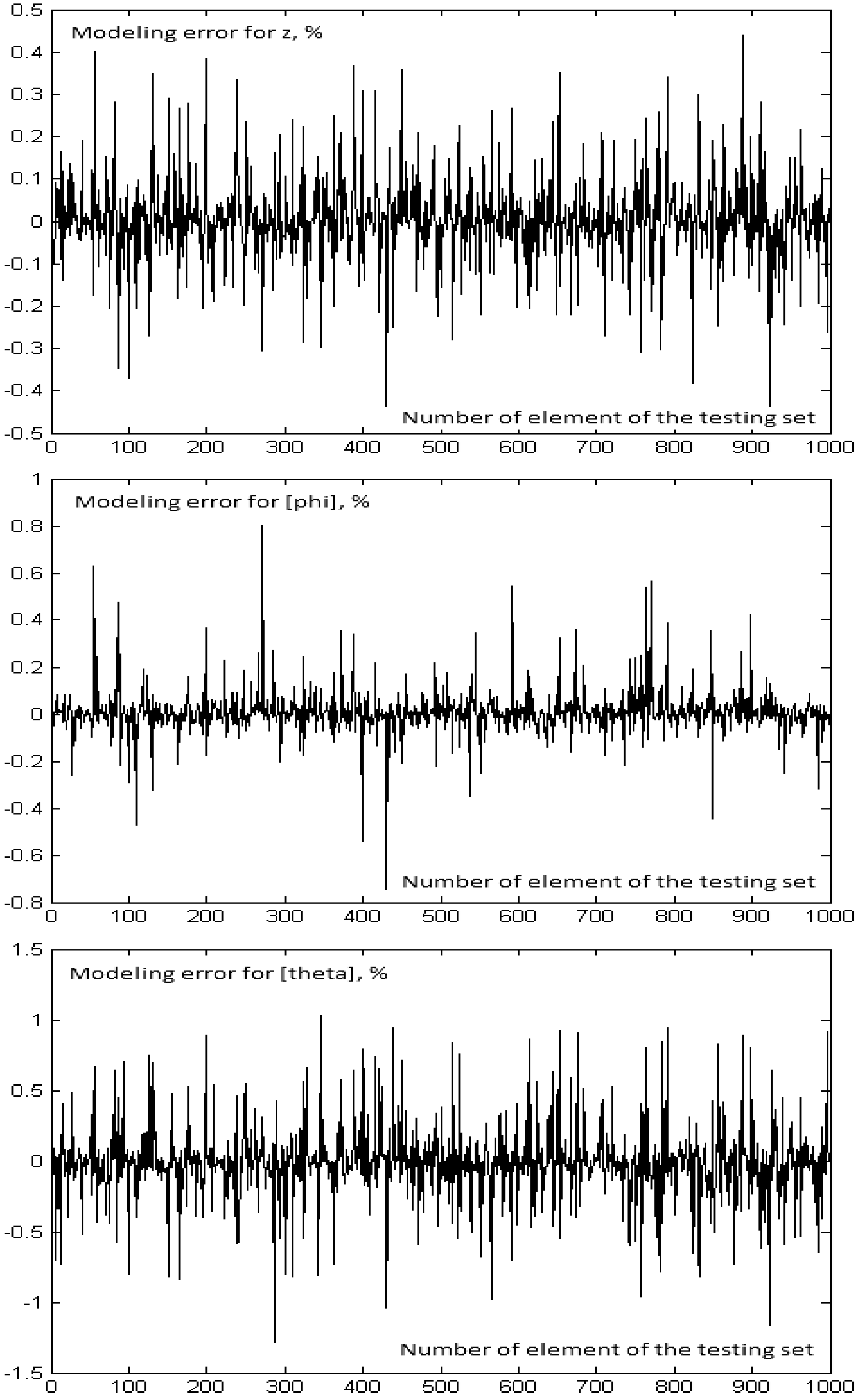


Fig. 6 Modeling error of z, [phi] and [theta] for the best network with 9 inputs.

- 88 – for the network calculating z;

- 38 – for the network calculating [phi];
- 93 – for the network calculating [theta].

Obtained model could be used to solve direct kinematic problem as a part of the tripod control problem. Proposed model synthesis principles could be applied to automatize other parallel structure mechanisms.

Summary

In this research a class of mechanisms, for which different output link positions relate to the one input drive links state, is considered. The direct kinematic analysis and control problem for such mechanisms is proposed to be solved by usage of the data about the previous mechanism position. It helps to determine the neighborhood, in which the solution of the considered problem is to be searched for.

The neural network based model is developed for the robot-machine with three degrees of freedom. It allows to solve the direct kinematic problem automatically. In its turn, this makes such mechanism automatic control possible.

The developed model is used to shown the effectiveness of the proposed movable platform position calculation method. Obtained mechanism output link position error reduction leads both to its positioning accuracy and speed improvement.

Acknowledgements

The work was performed with the financial support of the Russian foundation for basic research, the grant № 14-41-08058.

References

1. Merlet, J.-P., 2006. Parallel Robots, 2nd ed. Springer Dordrecht, The Netherlands, pp: 394.
2. Sakamoto, S., F. Hara, H. Hosokai, H. Kinoshita and Y. Abe, 2005. Parallel-Link Robot for Pipe Inspection. Shinryo Corp., Ibaraki, Japan, pp: 345-350.
3. Rybak, L.A., V.V. Ershukov and A.V. Chichvarin, 2011. Effective methods of kinematics and dynamics problem solving for robot-machine tool with parallel structure. Moscow, FIZMATLIT, pp: 148.
4. Geng, Z. and L.S.Haynes, 1992. Neural network solution for the forward kinematics problem of a Stewart platform. Robot Comput Integr Manuf, Vol. 9, Iss. 6, pp: 485-495.
5. Serrano, F.A., Caballero K. Yen and T. Brezina, 2007. Control of a Stewart platform with fuzzy logic and artificial neural network compensation. Florida International University, pp: 156-160.
6. Yee, C.S. and K.B. Lim, 1997. Forward kinematics solution of Stewart platform using neural networks. Neuro-

computing, Vol. 16, Iss. 4, pp: 333-349.

7. Rybak, L.A., Mamaev Y.A., Malyshev D.I. and L.G. Virabyan,2016. A software module for implementing a predetermined path of movement of the output link of the robot hexapod, for 3D printer products. Bulletin of Belgorod State Technological University named after V.G. Shukhov, (8):155-164.
8. Eremenko, Y.I., S.Y. Khalapyan and D.Y. Yarmurati, 2012. On possibility of neural network based automatic process control system operator training simulator development. Proceedings of 9th Russian school-conference of young scientists in Lipetsk, pp: 45-47.
9. Eremenko, Y.I., S.Y. Khalapyan, D.A. Poleschenko, D.Y. Yarmurati. Developing of Metallizing Shaft Furnace Model Based on Neural Network Technology. Mechatronics, automation, control, 4, pp: 56-60.
10. Eremenko, Y.I., S.Y. Khalapyan, D.Y. Yarmurati and A.A. Nevzorov, 2014. About creation of the neural network model of the behavior of operator that controls the thermal aggregates. Instruments and Systems: Monitoring, Control, and Diagnostics, 2, pp: 20-25.
11. Marquardt, D., 1963. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. SIAM Journal on Applied Mathematics, Vol. 11, 2: pp: 431-441.