Renewable Energy, Environment, and Power Quality Journal (REE&PQJ)
ISSN:3020-531X
Volume No.2, July 2024

Neural Network Based Configuring of Linear Delta Robots with Reduced Energy Consumption

V. Vodovozov[1], Z. Raud[1] and E. Petlenkov[2]

[1] Department of Electrical Power Engineering and Mechatronics
[2] Department of Computer Systems
Tallinn University of Technology
Ehitajate tee 5, 19086, Tallinn (Estonia)
Phone number: +372 53235223

Abstract. In this paper, a neural network based methodology of joints configuring is proposed for linear Delta robots, optimal in terms of minimizing the consumed energy. Since the techniques developed for serial robots are severely limited in parallel robotics, where many parameters affecting robot behaviour are either unstable or absent, a new purely data-driven approach has been developed and studied in the conditions of incomplete prior knowledge of robot geometric and mass properties. In the current research, the forward and inverse kinematics solvers are designed, the workspace of the robot is estimated, forward and inverse statics problems are resolved, and the most economical joint configurations are found with the help of the neural network robot model validated experimentally on the linear Delta robot EXPT-45-E1. The method offered in this study can be used in many other robotics applications, especially for the pick-and-place operations, where energy saving comes out on top.

Key words. Energy consumption, linear Delta robots neural network based model, pick-and-place robotics.

1. Introduction

At present, millions of robots of the serial architecture are widely used in various branches of industry across the globe [1], [2]. However, mechanisms with parallel topology have many advantages over them [3]. A parallel robot contains a fixed base and a mobile platform attached to it by identical legs, with a tool fixed on this platform. The legs are usually actuated by the motor-driven revolute or prismatic joints placed on the base. Most of parallel robots are well known for bigger payload capacity compared to their serial counterparts. In addition, they have fewer positioning errors, greater rigidity, less inertia and better dynamic characteristics suitable for rapid manipulation of processed objects, along with relatively large workspace [4].

Among the parallel robots, several studies point out the advantages of linear actuated Delta manipulators. In particular, this concerns linear versions of Delta Press, Delta Keops, and Delta Ibis robots presented in [5]. Their benefits are especially important for pick-and-place operations where these models move processed objects at extremely fast speeds and perfectly withstand significant inertial load when changing speeds. Their mechanics provide high accelerations that reach 100g in some applications [6]. Since the energy consumed by robots depends on their kinematic, static, and dynamic properties, these features of the Delta manipulators are even more important issues for energy optimisation than in the case of serial machines with the same payload [3], [7].

Forward kinematics (FK) and inverse kinematics (IK) models are used to explore kinematics in robotics. The FK models map such parameters as joint location, velocity, and acceleration from the joint space to Cartesian space while the full information about joints motion is known and the goal is to find the data of the tool movement. In the process of the opposed mapping from Cartesian space to joint space, called IK, the motion data of the tool are given and the goal is to find the movement information for all joints.

Kinematics analysis of Delta robots tends to be more difficult than the analysis of the manipulators with serial architectures [8]. This complexity comes mainly from the fact that the kinematic model is originated from geometrical considerations of robot arms connected within the close-loop chain. Aside from the complexity of parameters estimation, another issue is to compose a model of the full robot whose architecture is highly nonlinear, with many unknown factors.

Research of statics in robotics is also divided into two parts, namely forward statics and inverse statics analyses. Mapping the static parameters, such as torque, force, and power, from the joint space to Cartesian space is called forward statics, at which all joint static data are known and the goal is to find the static parameters of the tool. In the case of opposed mapping from Cartesian space to joint space called inverse statics, the static parameters of the tool are known and the goal is to find static data for all joints.

Estimation of the static behaviour of Delta robots is one of the most complex problems in robotics, since it is necessary to take into account all body masses and dimensions of the non-linear multi-leg close-loop chains along with their velocities and accelerations.

Additionally, dynamic analysis of the parallel architecture may also be required to design and evaluate robot paths, to size its actuators, to estimate the robot mechanical structure, and to handle robot operations.

All inverse problems have widely known challenges [9]. The most popular issue concerns the lack of analytical methods that exist only if the number of constraints coincides with the degree of freedom (DOF) of the robot. In Delta robots, part of parameters relates to passive uncontrolled complex joints that cannot be directly handled. Under these conditions, the solution of inverse problems is sought in terms of only controllable active joints.

Another issue concerns an existence of multiple (or even infinitely many) solutions called joint configurations (JK). Based on the strong dependence on the robot architecture, some JKs are either very difficult or even fully impossible to obtain analytically and implement physically [10].

The main idea behind the current research is to create a method for optimal robot joints configuring in terms of minimizing the consumed energy. For this purpose, a renewed model of the linear Delta robot is designed, studied, and validated in this work.

In contrast to analytical solutions, typical for this sort of studies, the proposed model is based on artificial neural network (NN), an adaptive system containing a set of interconnected nodes (neurons) within a layered structure resembling a human brain. The main advantages of all NNs are their learning ability, fault tolerance, and real-time operation possibility resulting from the high computational speeds. They can be trained to classify data, recognize patterns, and detect sequential dependencies in the inputs [11]. To employ these benefits, various single- and multilayer, feedforward (forward propagation) and recurrent (backward propagation) NNs are developed, including deep NNs with many hidden layers, convolutional NNs suitable for image, audio, and text processing, and fitting applications to forecast future events [12].

The NN-based model of the linear Delta robot offered in this research is developed and studied in MATLAB/Simulink environment and validated experimentally on the linear Delta robot EXPT-45-E1. Herewith, the FK and IK solvers are designed, the workspace of the robot is analysed, forward and inverse statics problems are resolved, and the most economical JKs are proposed based on the solution found.

2. The Method

The described method is intended to configure a 3-DOF linear Delta robot (Fig. 1) using a NN that provides the best fit to a series of data points obtained from preliminary made measurements of a pick-and-place process. The approach is purely data-driven and does not require prior knowledge of all geometric and mass properties. If necessary, it can be adapted to the changes in the robot architecture.

To define a fitting problem, the components of three input vectors (predictors) A1, A2 and A3, three output vectors (responses) X, Y and Z that match the input vectors, and a vector P of the corresponding consumed powers are initially collected experimentally from the real robot by placing the tool in target positions in one or another order. Typically, a total of n=200 to 20000 observations are needed, depending on the robot workspace and the required accuracy of the pick-and-place operations.

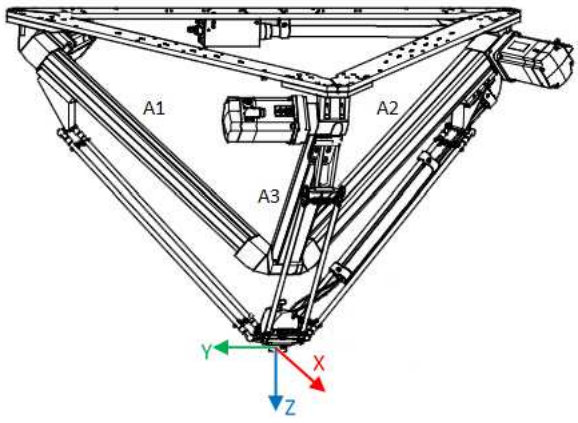

Fig. 1. 3-DOF linear Delta robot.

Next, the robot model is composed [13]. For that, all collected data are distributed between the training set for learning, validation set to verify that the NN is correctly generated and to stop learning before overfitting, and the test set to check independently the NN. As a start, a double-layer feedforward NN with few hidden neurons, a sigmoid transfer function in the hidden layer, and a linear transfer function in the output layer is tried, which is later complicated to improve the model quality.

The further step is the NN synthesis by training. Training with Levenberg-Marquardt backpropagation algorithm is usually suitable for most robotics applications [14]. However, for noisy or few-data problems, the Bayesian Regularization algorithm can result in a better solution, but at the cost of taking longer. For harder problems, the Scaled Conjugate Gradient algorithm might be recommended because it uses gradient calculations that are more memory efficient than the Jacobian calculations used in the previous two algorithms. During the model synthesis, the network structure is optimized by changing such NN parameters as the number of hidden layers, the number of neurons in each of them, and layer activation functions.

To validate the developed NN robot model as a FK solver, the following algorithm is proposed. First, a PRED (m to 3) array of predictors is assembled, where m is a number of permissible A1, A2, and A3 positions. Using the NN robot model, this array is transformed to another RESPOWER (n to 4) array of responses and powers, where n is the number of X, Y, and Z tool positions with consumed powers P that correspond to the appropriate predictors.

Since the accuracy of the robot always exceeds the tolerance of operations that this robot performs, i.e. n is much less than m, this means that every predictor may produce a set of similar responses. Concerning that the consumed energy depends on the predictors rather than responses, different power may be consumed to reach the similar responses at different JKs. Therefore, selection of the best one is the key factor for energy optimisation.

In addition to the main FK problem solution, the obtained RESPOWER array can be also used to estimate the robot workspace as a set of all spatial positions that can be reached by the tool. Herewith, several restrictions are imposed aiming to avoid collisions and singularities. To identify the workspace, the FK problem is solved repeatedly while every predictor cyclically increases its value from minimal to maximal limits. At the end of each cycle, the found X, Y, and Z tool positions are saved. Lastly, the workspace diagram is plotted (Fig. 2).

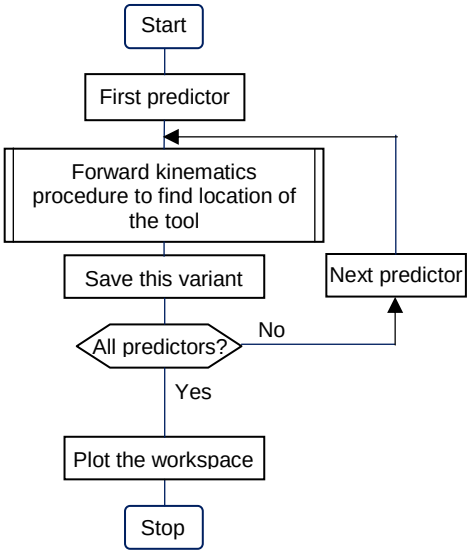

Fig. 2. Algorithm of robot workspace estimation.

The following step is devoted to the IK solver creation. In [15], different IK algorithms known in robotics to generate the affordable predictors are compared. Most of them represent iterative gradient-based optimization methods that start from an initial guess about the solution and seek to reduce a specific cost function. As soon as the cost of convergence becomes minimal, the IK problem is considered solved. Since most algorithms fail with some combinations of the initial assumption and target tool positions, some authors apply gradients of the cost function guesses and target tool positions to generate approximate second-derivative information that determines the next iteration. Although these algorithms are more reliable in finding solutions, not all requirements can be met with their help as well. This is especially true for linear Delta robots, in which the position of the tool can suddenly change, requiring the control system to make a prompt decision in real time.

Taking into account this factor, another iterative solution of the IK problem is applied in this work (Fig. 3), which represents the sequence of FK problem solutions very similar to that used for the robot workspace estimation. Herewith, the FK solver runs repeatedly while the joints cyclically increase their positions with a specified increment. At the end of each iteration, the condition is verifying if the tool is currently in its target position with an acceptable tolerance. If this condition is met, the revealed predictors are saved. The algorithm has adjustable parameters such as a maximal number of iterations, a maximal time during which the cycle is executed before the timeout expires, and an accuracy tolerance that specifies the error between the tool positions generated based on the solution and the target position, as well as a minimal step allowed by the solver. Due to its static nature and the absence of derivatives, the approach helps drastically reduce the number of iterations needed to estimate possible predictors. After all predictors are generated, at which the tool can achieve the target positions, they are stored and the identified JKs are considered found.

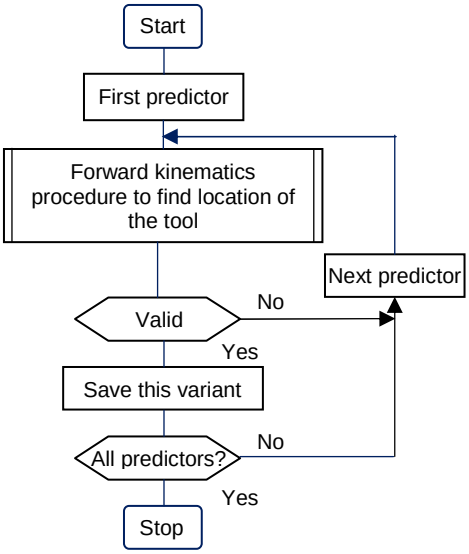

Fig. 3. IK algorithm developed for finding the JKs.

This IK procedure always has at least one solution thanks to its iterative nature. However, it does not bypass another challenge of IK – existence of multiple solutions. Because of this happy circumstance, this algorithm is suitable for comparing different solutions in order to find the best ones in terms of energy saving.

In Delta robots, this situation is manifested in the presence of several JKs that match the same target position. To find the best JKs, the inverse statics problem is solved for each predictor needed to place the tool in the particular target position. Since not only the Cartesian coordinates are stored in the RESPOWER array, but also information about power, the optimal predictor may be selected in such a way as to ensure minimum energy consumption. An appropriate algorithm is also designed in this work.

3. Case Study

To verify the method, it was tested on the linear Delta robot EXPT-45-E1 shown in Fig. 1. This tripod robot contains a fixed base platform, three identical arms, and a mobile tool flange attached to them. Each arm consists of two parts, namely the upper motor-driven along A1, A2, or A3 axis linear rail with a toothed belt gear and the lower parallelogram rod connected at one end to the rail by passive joints and at the other end to the tool flange by other passive joints. These passive joints drag the tool flange by the Cartesian XY frame parallel to the base platform and by Z axis vertically. Robot dimensions are presented in Table I [16].

Table I. - Robot Dimensions

Parameter	Value
Workspace diameter	450 mm
Base platform shoulder, B1	0.549 m
Linear rail of upper sub-arm, B2	From 0 to 0.224 m
Parallelogram rod length, B4	0.738 m
Tool flange shoulder, B6	0.200 m
Maximal payload	5 kg
Maximal velocity	7 m/s
Maximal accuracy	0.0005 m
Maximal Z force	1300 N
Servomotor EMMS-AS-100S of every arm	360 V; 3.8 A; 1497 W; 3.76-4.71 Nm
Maximal power consumed by robot in dynamics	4900 W
Maximal power consumed by robot in statics	900 W
Minimal power consumed by robot	320 W

To collect the components of predictors, responses, and consumed power vectors, the workspace of the robot was provisionally divided into a set of 400 constituent cells. During systematic tool movement along these cells, a total of 1331 JKs were recorded, each represented by 7 features: 3 joint positions in joint frames A1, A2 and A3 that play a role of predictors, 3 tool positions in Cartesian frame X, Y and Z, and appropriate powers P consumed by motors that play a role of responses.

Next, a set of predictors PRED (400x3 double) and a set of responses RESPOWER (400x4 double) were imported in the Neural Net Fitting app (*nftool*) from MATLAB and distributed between training, validation, and test sets as follows: 70% for training, 15% for validation, and 15% for testing.

The NN was trained using a Levenberg–Marquardt gradient descent algorithm with random data division and mean squared error (MSE) performance. A synthesized double-layer feedforward NN architecture of 20 neurons with sigmoid activation functions in a hidden layer and four linear output neurons is shown in Fig. 4.

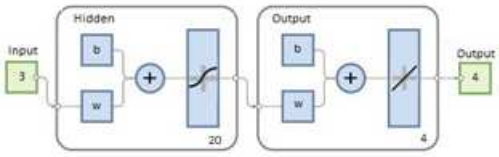

Fig. 4. Festo EXPT–45–E1 Tripod NN model.

The following results were obtained:

- in training: MSE=147 and regression R=0.9989;
- in validation: MSE=80 and R=0.9994;
- in testing: MSE=36 and R=0.9997.

Figure 5 displays the robot workspace plotted with the help of the designed FK solver. In Fig. 6, a cloud of consumed powers is shown, in which the blue area corresponds to the JKs with minimal power consumption.

Table II represents a solution of the IK problem for the particular pick-and-place process (37 JKs out of 1331), with appropriate consumed power data. This table demonstrates that, to reach each of the required targets in the Cartesian workspace X, Y, Z, three or four JKs A1, A2, A3 may be applied that require different consumed powers P. The JKs with minimal consumed power offered for robot pathing are highlighted. In the last column, the maximal amounts of the saved power ΔP are calculated for each tool position.

Table II. - Several Solutions of the IK Problem with Power Data

#	X	Y	Z	A1	A2	A3	P	ΔP
80	130	-29	112	**0**	**156**	**44**	**527**	52
181				22	112	89	549	
182				22	112	112	579	
202				22	156	67	572	
87	97	42	149	**0**	**156**	**201**	**684**	69
350				44	201	179	753	
469				67	201	134	727	
480				67	224	134	749	
880	-87	-57	142	156	44	224	751	22
890				**156**	**67**	**201**	**750**	
1001				179	44	224	772	
1111				201	22	224	772	
107	133	42	137	0	201	156	685	21
117				**0**	**224**	**134**	**684**	
238				22	224	134	705	
39	122	-42	102	**0**	**67**	**112**	**503**	21
40				0	67	134	524	
160				22	67	112	524	
71	135	-45	116	0	134	89	550	46
72				0	134	112	571	
91				**0**	**179**	**44**	**549**	
203				22	156	89	595	
700	-80	13	198	**112**	**179**	**134**	**750**	44
711				112	201	134	772	
722				112	224	134	794	
73	135	-37	121	**0**	**134**	**134**	**593**	48
94				0	179	112	618	
204				22	156	112	617	
234				22	224	44	616	
235				22	224	67	639	
876	-89	-99	137	**156**	**44**	**134**	**661**	71
979				179	0	224	728	
989				179	22	201	731	
999				179	44	179	732	
1142	-130	-7	153	**201**	**89**	**179**	**801**	73
1154				201	112	201	842	
1166				201	134	224	874	

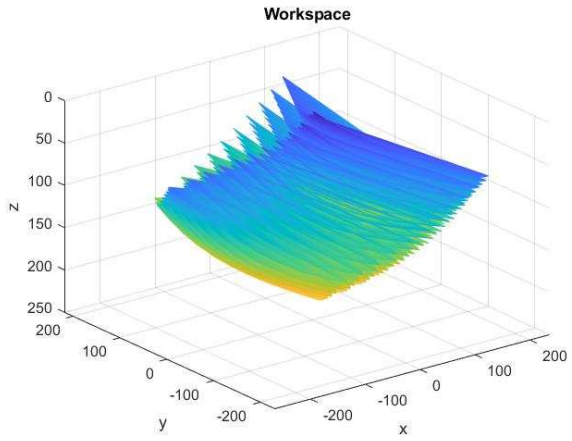

Fig. 5. Festo EXPT–45–E1 workspace.

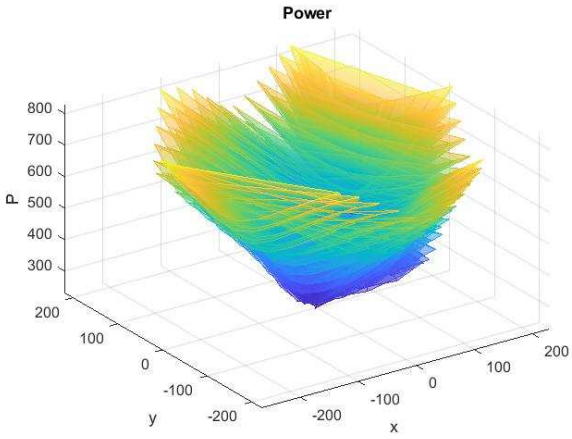

Fig. 6. Consumed powers of Festo EXPT–45–E1 Tripod.

4. Conclusion

This paper describes a neural network based configuring method optimal in terms of minimizing the consumed energy of linear Delta robots. It is shown that, since analytical modelling is not easy to derive when many parameters affected the robot behaviour are either instable or fully absent, the techniques applied for serial robots are not appropriate in parallel robotics. Therefore, special approaches need to be used.

A renewed model of the linear Delta robot is designed, studied, and validated. Herewith, the forward and inverse kinematics solvers are developed, the workspace of the robot is analysed, forward and inverse statics problems are resolved, and the joint configurations are proposed based on the solution found that ensure the most economical Delta robot behaviour.

The neural network approach offered in this study can be used in many other robotics applications where energy economy is in the spotlight.

Acknowledgement

This work was supported by the Estonian Research Council grant PRG658.

References

[1] C. Müller, World Robotics 2023 – Industrial Robots, IFR Statistical Department, VDMA Services GmbH, Frankfurt am Main, Germany (2023). Available: https://ifr.org/img/worldrobotics/Executive_Summary_WR_Industrial_Robots_2023.pdf.
[2] R. A. Kampa, What is a Serial Robot. Available: https://www.easytechjunkie.com/what-is-a-serial-robot.htm.
[3] Web Parallel Robots: The High-Speed Robotics Platform (2021). Available: https://howtorobot.com/expert-insight/parallel-robots.
[4] W. Khalil and E Dombre, "Introduction to geometric and kinematic modeling of parallel robots," In: Modeling, Identification and Control of Robots (2002), pp. 171 – 190. https://doi.org/10.1016/B978-190399666-9/50008-7.
[5] M. Bouri and R. Clavel, "The linear Delta: Developments and applications," 41st International Symposium on Robotics (ISR) and 6th German Conference on Robotics (ROBOTIK) (2010), pp. 1 – 8. Available: https://ieeexplore.ieee.org/document/5756938?arnumber=5756938.
[6] S. Ahangar, M. V. Mehrabani, A. P. Shorijeh and M. T. Masouleh, "Design a 3-DOF Delta parallel robot by one degree redundancy along the conveyor axis: A novel automation approach," 5th Conference on Knowledge Based Engineering and Innovation (KBEI), Tehran, Iran (2019), pp. 413 – 418. https://doi.org/10.1109/KBEI.2019.8734975.
[7] X. Yang, S. Wang, Y. Dong and H. Yang, "D2 Delta robot structural design and kinematics analysis," IOP Conf. Series: Materials Science and Engineering (2017), 274, pp. 1 – 12. https://doi,org/10.1088/1757-899X/274/1/012009.

[8] S. C. T. Filho and E. L. L. Cabra, “Dynamics and Jacobian analysis of a parallel architecture robot: The Hexa,” 18th International Congress of Mechanical Engineering, Ouro Preto, Brasilia (2005), pp. 1 – 8. Available: https://abcm.org.br/symposium-series/SSM_Vol2/Section_III_Robotics/SSM2_III_02.pdf.
[9] A. Csiszar, J. Eilers and A. Verl, “On solving the inverse kinematics problem using neural networks,” 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Auckland, New Zealand (2017), pp. 1 – 6. https://doi,org/10.1109/M2VIP.2017.8211457.
[10] C. Urrea, L. Valenzuela and J. Kern, “Design, simulation, and control of a hexapod robot in Simscape Multibody,” In: Applications from Engineering with MATLAB Concepts (2016). https://doi,org/10.5772/63388.
[11] C. C. Aggarwal, Neural Networks and Deep Learning, Springer, Yorktown Heights, NY, USA (2018), 510 p. https://doi.org/10.1007/978-3-319-94463-0.
[12] V. Vodovozov, A. Aksjonov, E. Petlenkov and Z. Raud, “Neural network-based model reference control of braking electric vehicles,” Energies (2021), 14, 2373. https://doi.org/10.3390/en14092373.
[13] Fit Data with a Shallow Neural Network (2023). Available: https://se.mathworks.com/help/deeplearning/gs/fit-data-with-a-neural-network.html#f9-43356.
[14] H. P. Gavin, “The Levenberg-Marquardt Algorithm for Nonlinear Least Squares Curve-Fitting Problems. Durham: NC, Duke University (2020), 19 p.
[15] V. Vodovozov, Z. Raud and E. Petlenkov, “Intelligent control of robots with minimal power consumption in pick-and-place operations,” Energies (2023), 16, 7418. https://doi.org/10.3390/en16217418.
[16] Electrical Tripod EXPT. 3D High Speed Pick & Place. Festo (2010) Available: https://www.festo.com/net/SupportPortal/Files/13547/EXPT-PSI-US.pdf.