

Article

Estimation of the Kinematics and Workspace of a Robot Using Artificial Neural Networks

Cătălin Boanta and Cornel Brişan *

Department of Mechatronics and Machine Dynamics, Faculty of Automotive, Mechatronics and Mechanical Engineering, Technical University of Cluj-Napoca, 400114 Cluj-Napoca, Romania

* Correspondence: cornel.brisan@mdm.utcluj.ro

Abstract: At present, in specific and complex industrial operations, robots have to respect certain requirements and criteria as high kinematic or dynamic performance, specific dimensions of the workspace, or limitation of the dimensions of the mobile elements of the robot. In order to respect these criteria, a proper design of the robots has to be achieved, which requires years of practice and a proper knowledge and experience of a human designer. In order to assist the human designer in the process of designing the robots, several methods (including optimization methods) have been developed. The scientific problem addressed in this paper is the development of an artificial intelligence method to estimate the size of the workspace and the kinematics of a robot using a feedforward neural network. The method is applied on a parallel robot composed of a base platform, a mobile platform and six kinematic rotational-universal-spherical open loops. The numerical results show that, with proper training and topology, a feedforward neural network is able to estimate properly values of the volume of the workspace and the values of the generalized coordinates based on the pose of the end effector.

Keywords: machine learning; neural network; feedforward; robot; kinematic analysis; workspace; artificial intelligence



Citation: Boanta, C.; Brişan, C. Estimation of the Kinematics and Workspace of a Robot Using Artificial Neural Networks. *Sensors* **2022**, *22*, 8356. <https://doi.org/10.3390/s22218356>

Academic Editor: Carina Soledad González

Received: 22 September 2022

Accepted: 25 October 2022

Published: 31 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

At present, industrial robots are almost a prerequisite in industrial applications such as milling, drilling, painting, machining or pick-and-place applications, mostly due to the fact that industrial processes have become more and more complex. This is why, with the increase in complexity of the industrial operations, in order to meet the specific criteria of the several industrial processes, robots are being continuously redesigned in what regards the architecture or the control system.

The action of redesigning a robot is usually realized by human designers, who have developed several mathematical methods to improve the design and reduce the required amount of time. Most of these mathematical methods are based on optimization algorithms (for numerical optimization), but, in past years, several artificial intelligence-based methods were developed. These newly developed methods are efficient and may decrease the required computational time of the redesign process. Undoubtedly, there are some disadvantages, such as the limited capability of online implementation and the capability of neural network to identify/establish a relation between the input and the output is dependent by the quality and the size of the training data.

When it comes to the process of designing or redesigning, one important aspect is the kinematics of the robot. In addition, in recent years, several scientific papers have proposed, for the design of robots, the use of neural networks for kinematic analysis. Although these papers focus on the estimation of the generalized coordinates based on the poses of the tool center point of the robot, little work has been conducted considering the workspace as a whole as the output of a neural network based on the generalized coordinates.

Moreover, the process of redesigning a robot is required by the applications in which the robot will be used and by the specific criteria, either imposed by the application or a human designer. In order to meet the criteria, several complex mathematical models have to be developed, which are based on highly nonlinear equations. Considering this aspect, the design of robots requires a large amount of experience from a human designer, an aspect that may be bypassed or improved with the mean of optimization methods [1–7]. Moreover, recently, a high amount of work was performed for the mathematical modeling of industrial robots using machine learning methods, as presented in [8–10].

In the following, several examples are presented using neural networks in order to estimate the kinematics of robots. The authors of [11] proposed the use of neural networks for the offline estimation of the kinematic analysis of a robot and an online neural network in order to control the trajectory. A multilayer feedforward neural network was trained in [12] in order to solve the kinematic analysis of a planar parallel mechanism. The proposed solution overcomes some limitations of the Newton–Raphson numerical algorithm in what regards obtaining multiple solutions for the generalized coordinates.

Another example is [13], which proposed a supervised learning neural network in order to solve the inverse kinematic problem of serial robots, considering the assembly errors in the joints. In [14], the authors proposed adversarial neural network in order to solve the inverse kinematic and dynamic problems.

A more complex analysis was presented in [15], in which the inverse kinematic problem of a parallel robot was approached both in analytic form and using neural networks. Three types of neural networks were used: Multilayer Perceptron (MLP), Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU). The networks were trained and tested using a point cloud that represents the possible poses of the final effector and the corresponding generalized coordinates. The type of neural network that achieved the best accuracy was the GRU, which presented also the lowest demanded computational time.

A similar analysis was performed in [16], where the direct kinematic problem of a three-degrees-of-freedom redundant parallel robot was resolved based on different type of machine learning methods: an MLP neural network, a Radial Basis Function (RBF) neural network and Support Vector Machines (SVM). A backpropagation neural network was used to identify a parameter of the architecture of a parallel robot based on several performance indices as the workspace, the global conditioning index and the global rigidity index. The paper presented, also, some of the disadvantages of using the neural network for the kinematic analysis of robots, such as the neural networks may not be used in an online application (and the control of robots requires real time processing), the capabilities of the neural network to identify a mathematical function between the input and output are completely dependent on the way the system was trained and the precision of the discretization (for obtaining highly accurate results, a large amount of data is required), and a slow convergence ratio.

Furthermore, ref. [17] proposed the estimation of the kinematic analysis of a parallel robot that contains six prismatic–sphere–sphere open loops by implementing several machine learning methods, such as Multiple Linear Regression, Multi-Variate Polynomial Regression, Support Vector Regression, Decision Tree Regression and Random Forest Regression.

The direct kinematic problem of an industrial robot solved using a feedforward Multilayer Perceptron was presented in [18]. A cloud of points that describes the poses of the end effector of the robot were generated using Denavit–Hartenberg matrices. A part of the points was used for training the network and the other part for testing and verification. A similar work was presented in [19], where the authors used a Deep Neural Network in order to solve the kinematics of a serial robot with five degrees of freedom.

As presented in [20], the workspace, e.g., the geometrical locus of all the possible coordinates of the tool center point, is one of the main criteria in the design of robots. The importance of the workspace is provided by: the capability of execution of a certain task (the poses of the tool center for a certain task must lay inside the workspace), safety criteria (if a worker occupies the workspace of a robot that performs a task he or she might be injured)

or factory floor or enclosure dimensioning and nesting (the environment of the robot has to allow the free movement of the tool center point). In what regards workspace estimation using neural networks, in the scientific literature, there are few published papers that treat this subject. For example, ref. [21] proposed the use of a deep neural network to estimate the workspace by the mean of subspace learning. The paper states that the computational time is lower than classical methods of workspace generation. The paper [22] presented a development of the work from [21], i.e., the generation of the full workspace of serial robots using a deep-learning framework for a given pose.

The work presented in this paper is an extended version of [23] and presents an artificial neural network-based method (a feedforward neural network) that is used to estimate the kinematics and the volume of the workspace of a robot, considering as inputs the parameters that describe the architecture of the robot and the pose of the end effector. The proposed framework is applied on a six-degrees-of-freedom parallel manipulator with rotational active joints. The novelty of the paper rests mostly on two aspects. The first aspect is the fact that the neural network estimates the values of the generalized coordinates from each active joint in particular (for each generalized coordinate, there is an output from the network), an aspect that has the advantage of and offers the possibility to apply the method for manipulators with a different number of active joints. The second one is the fact that, in the process of designing or redesigning a robot, a human designer can estimate the workspace directly, each time a parameter of the robot is changed, without the requirement of computing it with the classical workspace generation methods.

2. Proposed Method

The work presented in this paper is summarized by Figure 1. In the following subsections, each step from the problem formulation is described in detail.

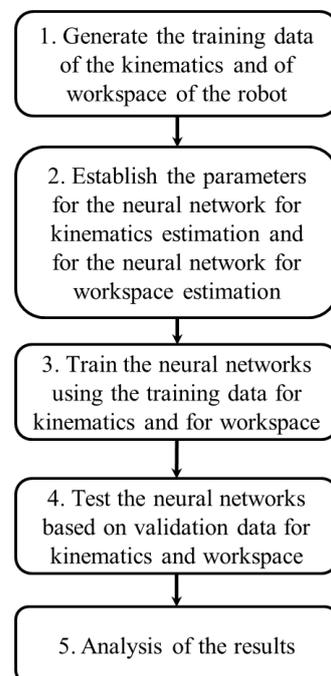


Figure 1. Representation of the problem formulation.

2.1. Generate the Training Data and of the Workspace of the Robot

In the first step, the kinematic model of a robot was developed. Therefore, for a given pose of the end effector, the mathematical equations that map the values of the generalized coordinates from the active joints were developed, as in Equation (1):

$$q = f(p, \varphi) \quad (1)$$

where q is a linear vector with the generalized coordinates, f represents the mathematical functions, and p and φ represent the pose of the end effector (p is the vector of the coordinates on the x , y and z axes, and φ is the vector of Euler's angles around the x , y and z axes).

By performing the kinematics, a dataset of values for the pose of the end effector and the corresponding values for the generalized coordinates was generated. Part of this data represent the training data and the rest was kept for validation.

Next, the generation of the workspace data was based on the kinematic analysis. There are several methods proposed in the scientific literature for workspace evaluation, with the previous work of the authors on this subject being presented in [24,25]. Firstly, a 3D space (a parallelepiped) is discretized in uniformly distributed points p_i . For each p_i , the kinematic analysis is performed and, if there is a valid solution for the generalized coordinates, the point p_i is included in the workspace. Finally, the workspace is generated as a cloud of points from all the points that generated a valid solution.

In order to evaluate the volume of the workspace, Equation (2) is used:

$$V_{WS} = v_{p_i} \cdot m \quad (2)$$

where V_{WS} is the volume of the workspace, v_{p_i} is the volume correspondent to a single point of the workspace and m is the total number of points.

The next step was to perform multiple evaluations of the workspace using the kinematic equations, using several parameters that describe the architecture of a robot. In this way, a second dataset was created.

For the two generated datasets (one for the kinematic analysis and one for the workspace), 80% of the data were used for training the neural networks and 20% of the data were used for validation and testing.

2.2. Establish the Parameters That Describe the Neural Networks Used for Kinematics and Workspace Estimation

The process of defining the neural network is a complex and important task and, for these applications, a proper method was not identified in the scientific literature. Both for the kinematics and workspace estimations, a multilayer, fully connected neural network was used. The network was implemented in the MATLAB environment. Figure 2 presents the general topology of the neural network.

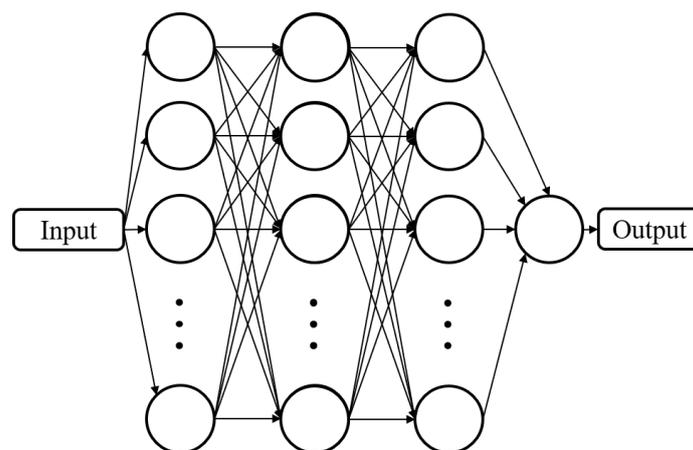


Figure 2. General topology of a multilayer fully connected neural network (reprinted with permission from ref. [23]. 2022, IEEE Proceedings).

The parameters that describe the neural network are the number of layers, lambda λ (the regularization rate) and the type of activation function, which may be the Rectified

Linear Unit (ReLU), the hyperbolic tangent \tanh , the sigmoid or identity function, functions that are presented, respectively, in the following Equations (3)–(6):

$$\phi(v) = \begin{cases} v, & v \geq 0 \\ 0, & v < 0 \end{cases} \quad (3)$$

$$\phi(v) = \tanh(v), \quad (4)$$

$$\phi(v) = \frac{1}{1 + e^v}, \quad (5)$$

$$\phi(v) = v. \quad (6)$$

In order to define these parameters, an optimization was performed in MATLAB using the Random Search Algorithm for both the networks used for the kinematics and for the workspace estimation (numerical examples of the optimization are presented in Sections 4.2 and 5.2).

2.3. Training the Neural Networks

After the establishment of the parameters that describe the networks, the neural network used for the kinematics estimation and the neural network used for the workspace estimation were trained based on the two datasets created in the first step. As presented, a fraction of the two datasets was kept for validation purposes.

In this step, another important aspect is the convergence criterion. This may be the maximal number of function evaluations, a specific minimal value of the mean squared error (MSE) or the confirmation of the performance of the trained network based on the data reserved for validation. This criterion is imposed by the user that defines the network but may be dependent on the application the neural network is used for.

2.4. Testing the Trained Neural Network Us

After the training of the neural networks from the third step, the performance of the networks was evaluated using the validation datasets (using the 20% fraction reserved in the first step from each dataset).

The outputs of the two neural networks, i.e., the estimated values of the generalized coordinates and the volume of the workspaces, were compared with the numerically evaluated ones. In order to quantify the error, two mean squared errors were computed, one for the kinematics and one for the volume of the workspace, as presented in the following equations:

$$MSE_{Kinematics} = \frac{1}{t} \sum_{i=1}^t (q_i - \hat{q}_i)^2 \quad (7)$$

$$MSE_{Workspace} = \frac{1}{p} \sum_{i=1}^p (V_{WS_i} - \hat{V}_{WS_i})^2, \quad (8)$$

where q_i is the i th vector with generalized coordinates evaluated numerically in the first step, \hat{q}_i is the vector with estimated values of the generalized coordinates corresponding to the same input as for the numerically evaluated one, t is the total number of test cases for kinematics, V_{WS_i} represents the numerically evaluated volume of the i th workspace from the first step, \hat{V}_{WS_i} is the volume of the workspace that was estimated by the neural network, corresponding to the same parameters of the robot, and p is the total number of test cases for the workspace.

2.5. Analysis of the Performance of the Neural Networks

The last step corresponds to an in-depth analysis of the performance of the neural networks. The estimated values (both for the kinematics and for the volume of the workspace) were compared in terms of relative difference with the numerically evaluated ones (from the validation dataset). Moreover, the convergence of the neural networks, the training history and the error histograms were presented for each neural network.

3. Prerequisite—Kinematics and Workspace Analysis of a Parallel Robot

In order to validate the proposed method and to generate the training and test data for the neural network, the kinematics and the workspace of a robot were developed in analytical form. In this paper, a six-degrees-of-freedom parallel robot was used, composed of a fixed platform, a mobile platform and six identical kinematic open loops that interconnect the platforms. These open loops are composed of two mobile elements, a rotational joint (R), a universal joint (U) and a spherical joint (S).

On the one hand, the reason for applying the methodology on a parallel robot is that, in the case of parallel robots, one disadvantage is the reduced size of the workspace in comparison with the dimensions of the elements of the robots. Moreover, in the case of the direct kinematics of a parallel robot, there is a closed-form analytical solution for computing the poses of the tool center point from the values of the generalized coordinates.

On the other hand, the methodology presented in Section 2 was developed regardless of the type of the robot to which it is applied, so it may be applied to other types of robots (e.g., serial or parallel robots).

The architecture of the robot is presented in Figure 3 and the top view of the platforms is presented in Figure 4.

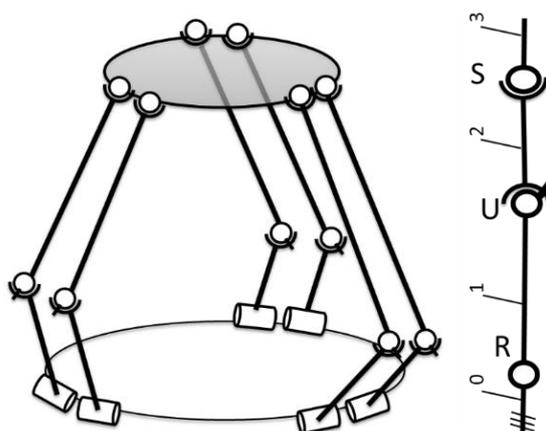


Figure 3. Parallel robot composed of 6 identical RUS open loops (adapted with permission from ref. [23]. 2022, IEEE Proceedings).

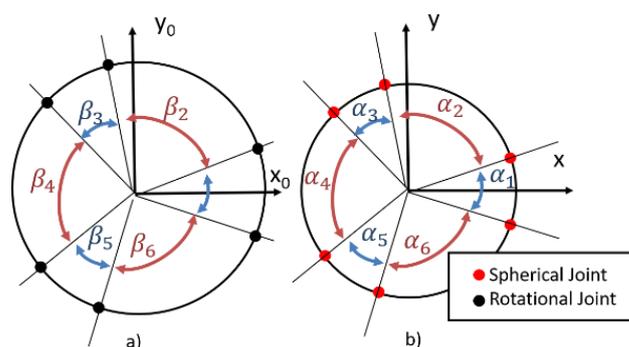


Figure 4. The fixed (a) and mobile (b) platforms of the parallel robot (adapted with permission from ref. [23]. 2022, IEEE Proceedings).

The parameters in Figures 3 and 4 represent:

- l_1 and l_2 are the lengths of the elements of the RUS kinematic open loops that interconnect the platforms;
- R and r are the radii of the two platforms (fixed and mobile);
- β_i and α_i are the angles that describe the positioning of the rotational joints on the fixed platform and of the spherical joints on the mobile platform, respectively (where $i = 1 \dots 6$).

3.1. Kinematic Analysis

In order to perform the kinematic analysis, Figure 5 is considered.

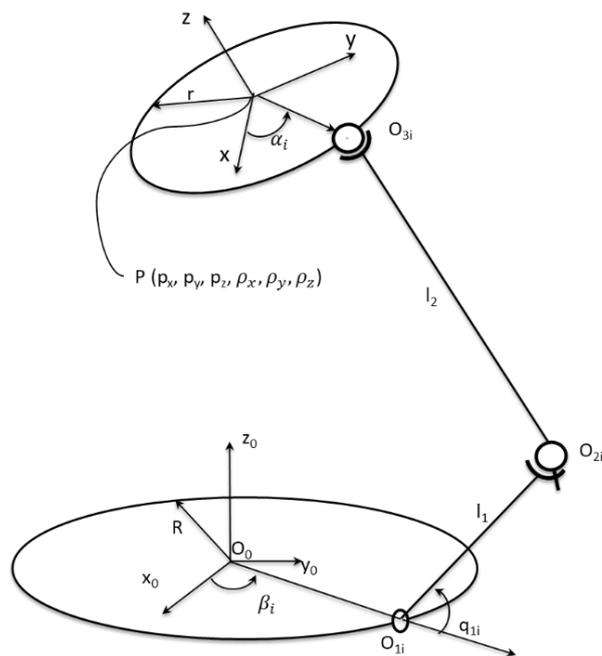


Figure 5. Parallel robot (only one RUS open loop is represented).

The notations from Figure 5 (other than ones presented in the previous two figures) represent (for each notation, $i = 1 \dots 6$):

- $O_0x_0y_0z_0$ is the absolute coordinate system attached to the fixed platform;
- P_{xyz} is the relative coordinate system attached to the mobile platform. The coordinates of the position vector p of the point P with respect to the absolute frame $O_0x_0y_0z_0$ are (p_x, p_y, p_z) ;
- The angles ρ_x, ρ_y, ρ_z are the Euler's rotation angles of the system P_{xyz} around x_0, y_0 and z_0 of the absolute coordinate system;
- q_{1i} is the active angle in the rotational joint i ;
- O_{1i} is the center of the rotational joint. The coordinates of the position vector o_{1i} of the point O_{1i} expressed in the absolute coordinate system are $(o_{1ix}, o_{1iy}, o_{1iz})$;
- O_{2i} is the center of the universal joint. The coordinates of the position vector o_{2i} of the point O_{2i} expressed in the absolute coordinate system are $(o_{2ix}, o_{2iy}, o_{2iz})$;
- O_{3i} is the center of the universal joint. The coordinates of the position vector o_{3i} of the point O_{3i} expressed in the absolute coordinate system are $(o_{3ix}, o_{3iy}, o_{3iz})$;

The kinematic analysis was solved for one open loop, in a similar manner as presented in [26]. Applying the same methodology for each open loop, the inverse kinematic analysis was solved for entire robot. Firstly, one has to evaluate the analytic expression of the active angle q_{1i} . The geometrical parameters of the mobile platform that are known are: $R, r, l_1, d, l_2, \beta_i, \alpha_i$. Writing the equation of the length of the second element, l_2 the following equation was obtained (for each equation, the value of $i = 1 \dots 6$):

$$(o_{3ix} - o_{2ix})^2 + (o_{3iy} - o_{2iy})^2 + (o_{3iz} - o_{2iz})^2 = l_2^2. \tag{9}$$

The coordinates of the point O_{3i} were evaluated from the pose of P_{xyz} towards $O_0x_0y_0z_0$:

$$[o_{3ix}, o_{3iy}, o_{3iz}]^T = [p_x, p_y, p_z]^T + R_{zyx} \cdot r \cdot [\cos(\alpha_i), \sin(\alpha_i), 0]^T. \tag{10}$$

The term R_{zyx} is the rotation matrix of $Pxyz$ towards $O_0x_0y_0z_0$, evaluated with:

$$R_{zyx} = R_z(\rho_z) \cdot R_y(\rho_y) \cdot R_x(\rho_x). \quad (11)$$

The coordinates $(o_{2ix}, o_{2iy}, o_{2iz})$ of the points O_{2i} were evaluated with the equation:

$$[o_{2ix}, o_{2iy}, o_{2iz}]^T = [o_{1ix}, o_{1iy}, o_{1iz}]^T + l_{1i} \cdot [\cos \beta_i \cdot \cos q_{1i}, \sin \beta_i \cdot \cos q_{1i}, \sin q_{1i}]^T. \quad (12)$$

The coordinates $(o_{1ix}, o_{1iy}, o_{1iz})$ of the point O_{1i} are:

$$[o_{1ix}, o_{1iy}, o_{1iz}]^T = r \cdot [\cos \beta_i, \sin \beta_i, 0]^T. \quad (13)$$

By replacing Equations (10)–(13) into Equation (9), we obtained:

$$\cos q_{1i} \cdot D_i + \sin q_{1i} \cdot E_i = F_i, \quad (14)$$

where D_i , E_i and F_i are evaluated as follows:

$$D_i = \cos \beta_i (o_{3ix} - o_{1ix}) + \sin \beta_i (o_{3iy} - o_{1iy}), \quad (15)$$

$$E_i = o_{3iz} - o_{1iz} \quad (16)$$

$$F_i = \frac{[l_1^2 - l_2^2 + (o_{3ix} - o_{1ix})^2 + (o_{3iy} - o_{1iy})^2 + (o_{3iz} - o_{1iz})^2]}{(2 \cdot l_1)}. \quad (17)$$

The analytic expression of the active angle q_{1i} is expressed by:

$$\sin(q_{1i}) = \frac{(E_i \cdot F_i - K_i \cdot D_i \cdot \sqrt{(D_i^2 + E_i^2 - F_i^2)})}{(D_i^2 + E_i^2)}, \quad (18)$$

$$\cos(q_{1i}) = \frac{(D_i \cdot F_i + K_i \cdot E_i \cdot \sqrt{(D_i^2 + E_i^2 - F_i^2)})}{(D_i^2 + E_i^2)}. \quad (19)$$

The term K_i from Equations (18) and (19) has the value either 1 or -1 , depending on the joint configuration (branch index). Each angle q_{1i} has two valid solutions in the inverse kinematic problem, depending on the term K_i . Only one of them was considered in the following equations. The final solution for the active angles was evaluated with the *atan2* function:

$$q_{1i} = \text{atan2}(\sin(q_{1i}), \cos(q_{1i})). \quad (20)$$

3.2. Workspace Analysis

In general, the workspace of a robot is defined by the space occupied by the end effector considering all the possible positions and orientation of its end effector. In the following, the zero-orientation workspace was developed for the six-degrees-of-freedom parallel robot analyzed in this paper. There several constraints that affect the workspace, as presented in [20] as:

1. Kinematic/geometrical constraints represent the solutions of the active angles from the inverse kinematics from Section 3.2:

$$q_i = f(p_x, p_y, p_z, \varphi_x, \varphi_y, \varphi_z). \quad (21)$$

2. Mechanical constraints are the physical limitations in the joints:

$$q_i \min \leq q_i \leq q_i \max. \quad (22)$$

3. Limitations regarding the minimal distance between elements (e_k and e_l being two random elements):

$$\text{dist}(e_k, e_l) > \text{dist}_{\min} \quad (23)$$

Firstly, a cube is discretized in points for which the inverse kinematic problem is evaluated. If there is a geometric solution of the kinematics, it is evaluated if the point lies within the singular configurations and if other mechanical constraints are respected. If all the requirements are true, the analyzed point lies within the workspace. By combining all the valid points from the discretized cube, the workspace of the robot is generated. The workspace of the parallel robot is presented in Figure 6.

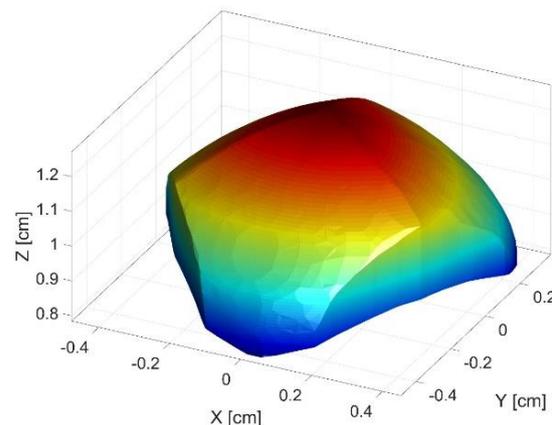


Figure 6. Workspace of the parallel robot.

4. Results—Estimation of the Kinematics Using a Neural Network

In this section, the methodology presented in Section 2 was applied for the inverse kinematics of the six-degrees-of-freedom parallel robot, i.e., a feedforward neural network was trained to estimate the values of the generalized coordinates based on the pose of the end effector.

4.1. Generation of the Training and Test Dataset

The desired output of the neural network is the generalized coordinates given the pose of the end effector. The training data have to be generated, i.e., a dataset of poses of the end effector are randomly generated (which is the input in the neural network) and the corresponding generalized coordinates are evaluated using the inverse kinematics algorithm presented in Section 3 (the generalized coordinates are the output of the neural network). For each generalized coordinate, the neural network was trained individually meaning that there were six models trained, for each generalized coordinate.

The pose of the end effector is given by the coordinates of the tool center point (p_x, p_y, p_z) on the x, y and z axes and the angles of rotations around each axis (ρ_x, ρ_y, ρ_z). In order to generate the training data, a number of 10,000 random values were considered for each coordinate and angle. The random values were generated between the following minimal and maximal limits for each parameter, as presented in Equation (24):

$$\begin{aligned} -0.3 < p_x < 0.3 [m], \\ -0.3 < p_y < 0.3 [m], \\ 0.8 < p_z < 1.2[m], \\ -\frac{\pi}{6} < \rho_x < \frac{\pi}{6}, \\ -\frac{\pi}{6} < \rho_y < \frac{\pi}{6}, \\ -\frac{\pi}{12} < \rho_z < \frac{\pi}{12}. \end{aligned} \quad (24)$$

The random values were combined in 10,000 input random combinations. For each combination, the correspondent generalized coordinates of the robot were evaluated using the inverse kinematics analysis. In this way, the dataset required for training the neural network was generated. A fraction of 80% of this dataset was reserved for the training of the neural network and the remaining 20% was reserved for validation.

4.2. Define the Parameters of the Neural Network for Kinematic Estimation

The type of network that was used to evaluate the volume of a workspace based on the input parameters was a feedforward, fully connected neural network implemented in the MATLAB environment. The parameters that define the topology of the neural network have a strong influence upon the numerical results. This is why, in order to increase the efficiency, the parameters of the network were established based on an optimization.

The optimization was run using a random search algorithm from the MATLAB environment. The optimization was run for 1000 iterations and, on each iteration, the neural network was trained based on the training data already prepared. The objective function of the optimization was the mean squared error after training. On each iteration, the neural network was trained up to the point in which the number of epochs for training reached 1000. The resulting parameters that provided the best neural network are presented in Equations (25)–(28):

$$No_{layers} = 2, \quad (25)$$

$$Size_{layers} = [18, 13] \quad (26)$$

$$\lambda = 7.943 \times 10^{-8}, \quad (27)$$

$$Act_{fcn} = \tanh. \quad (28)$$

In Equations (25)–(28), the parameters that were optimized are: No_{layers} —the number of intermediate layers (without the input and output layers), $Size_{layers}$ —a vector that contains the size (the number of neurons) of each intermediate layer, λ —the regularization rate and Act_{fcn} —the type activation function.

4.3. Numerical Results

The feedforward neural network was trained in the MATLAB environment using 80% of the training data. The equipment that was used was a PC with an Intel I7-4770k processor and 64 GB of RAM.

As presented before, six trained models were generated, using the same neural network parameters each time. Therefore, each model estimated one generalized coordinate. In each case, the condition of convergence of the neural network was reaching 1 million epochs or an MSE of 1×10^{-7} .

4.3.1. Results for the First Generalized Coordinate q_1

Figure 7 presents the convergence of the neural network on a logarithm scale. From the 1st epoch to the 1000th epoch, the mean squared error (MSE) reached from 3.88 to 1.8×10^{-3} . In the 1 millionth epoch, the MSE reached its lowest value, as in Equation (29).

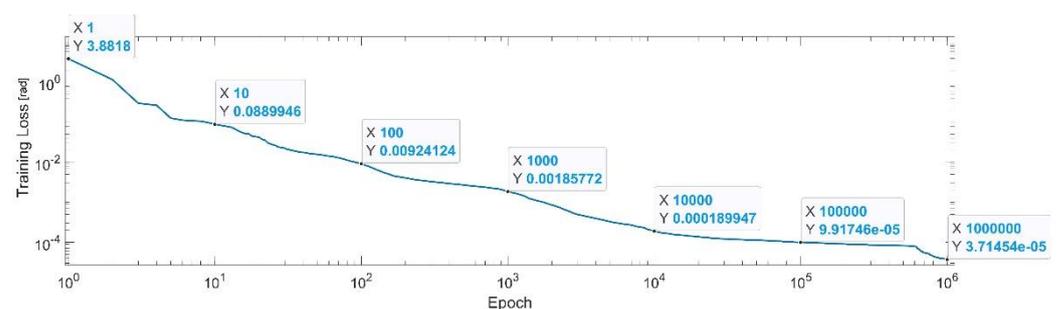


Figure 7. Convergence of the neural network for the first generalized coordinate q_1 (log scale).

$$MSE_Best_q_1 = 3.71 \times 10^{-5}. \quad (29)$$

Using the trained model, the values of the first generalized coordinate were estimated, using the fraction of 20% from the dataset reserved for validation. Figure 8 presents the real values of the first generalized coordinate in comparison with the estimated ones from the neural network.

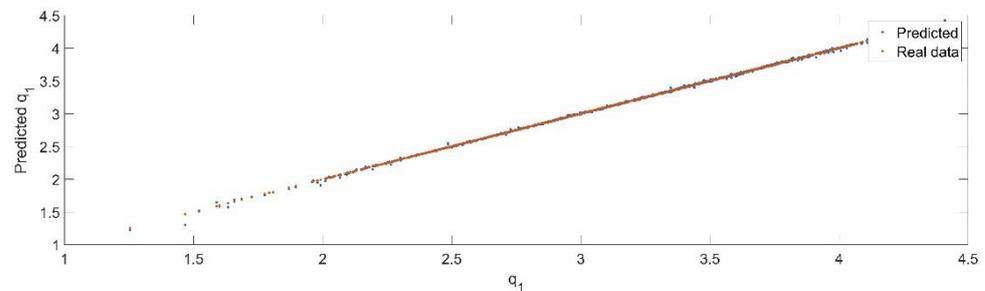


Figure 8. The comparison between the real data (the real value of q_1) and the predicted data (the predicted value of q_1).

Figure 9 presents the relative error between the real validation data and the predicted data and the model and Figure 10 presents the error histogram between the target and predicted values.

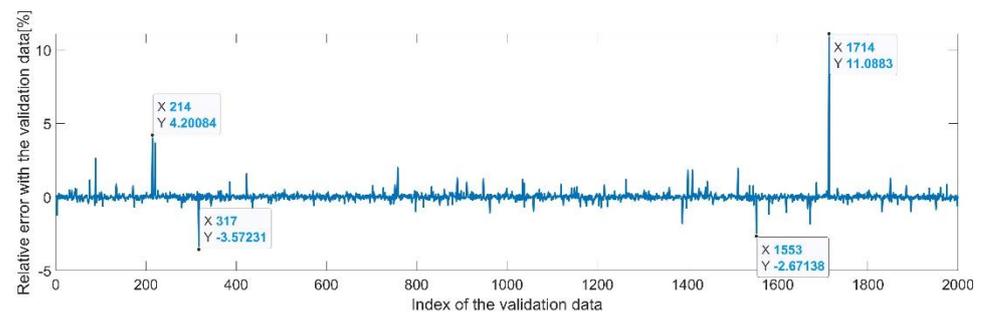


Figure 9. Relative errors between the target values and the predicted values for q_1 .

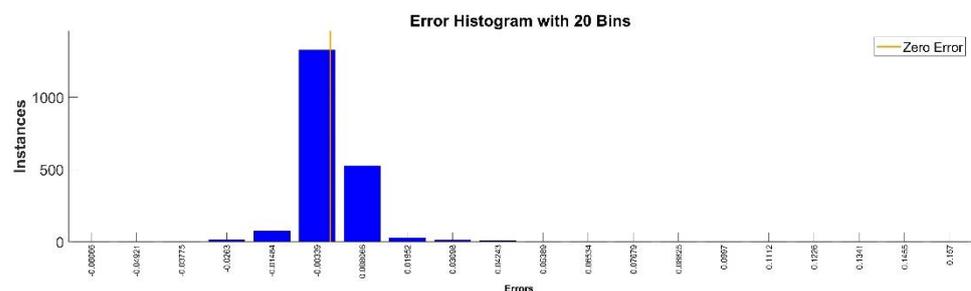


Figure 10. Error histogram between the target values and the estimated values for q_1 .

By analyzing Figures 9 and 10, the maximal error between the target and prediction was 11.08%. Nevertheless, among the 2000 target values, only for 17, the relative error was higher than 1%. The mean error between the target and prediction for the first generalized coordinate was 0.0103%.

4.3.2. Results for the First Generalized Coordinate q_2

Figure 11 presents the convergence of the neural network on a logarithm scale. From the 1st epoch to the 1000th epoch, the mean squared error reached from 7.08 to 1.3×10^{-3} . In the 1 millionth epoch, the MSE reached its lowest value, as in Equation (30).

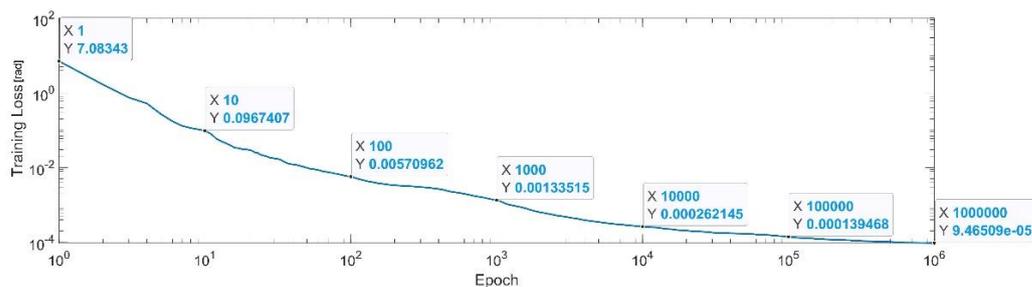


Figure 11. Convergence of the neural network for the first generalized coordinate q_2 (log scale).

$$MSE_Best_q_2 = 9.46 \times 10^{-5}. \tag{30}$$

Using the trained model, the values of the first generalized coordinate were estimated, using the fraction of 20% from the dataset reserved for validation. Figure 12 presents the real values of the first generalized coordinate in comparison with the estimated ones from the neural network.

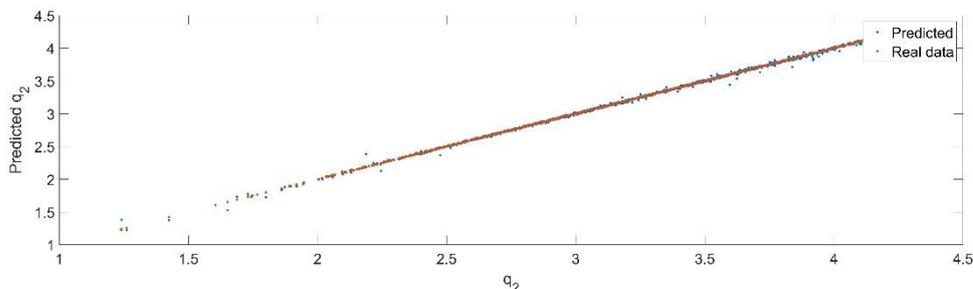


Figure 12. The comparison between the real data (the real value of q_2) and the predicted data (the predicted value of q_2).

Figure 13 presents the relative error between the real validation data and the predicted data and the model and Figure 14 presents the error histogram between the target and predicted values.

By analyzing Figures 13 and 14, the maximal error between the target and prediction was -11.48% . Nevertheless, among the 2000 target values, only for 38, the relative error was higher than 1%. The mean error between the target and prediction for the second generalized coordinate was 0.104%.

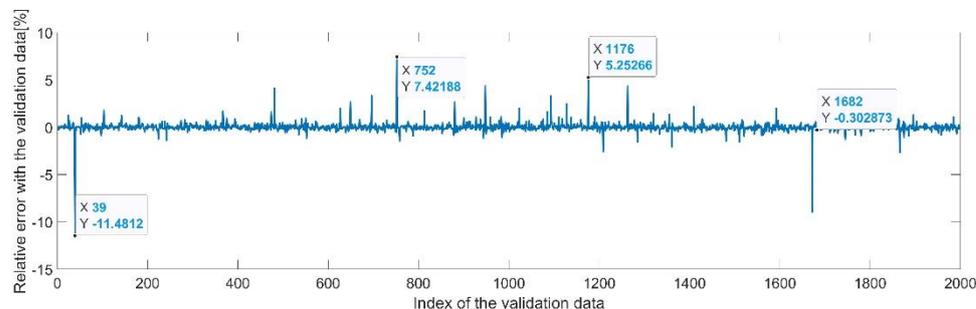


Figure 13. Relative errors between the target values and the predicted values for q_2 .

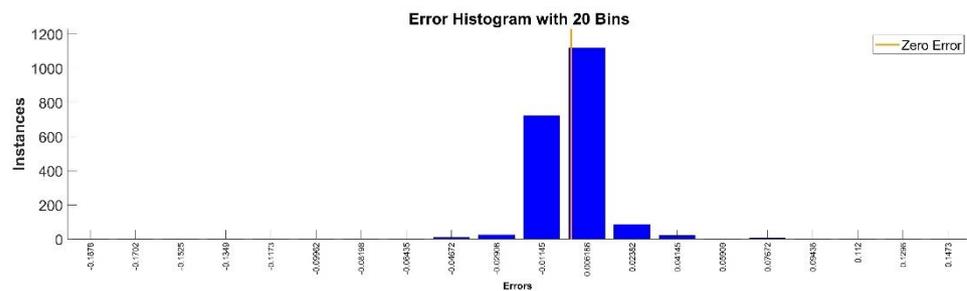


Figure 14. Error histogram between the target values and the estimated values for q_2 .

4.3.3. Results for the First Generalized Coordinate q_3

Figure 15 presents the convergence of the neural network on a logarithm scale. From the 1st epoch to the 1000th epoch, the mean squared error reached from 9.71 to 1.07×10^{-3} . In the 1 millionth epoch, the MSE reached its lowest value, as in Equation (31).

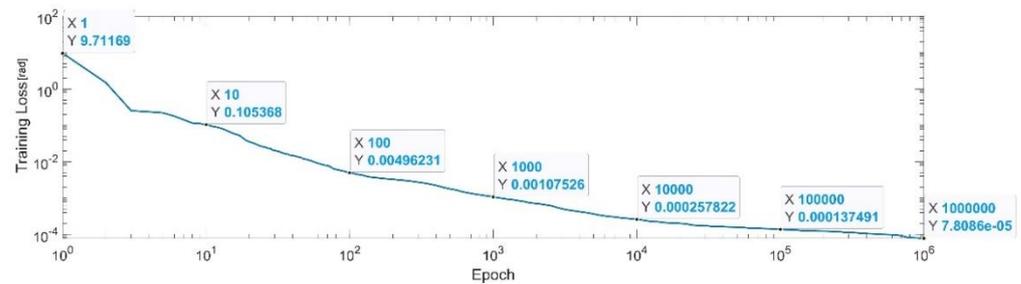


Figure 15. Convergence of the neural network for the first generalized coordinate q_3 (log scale).

$$MSE_{Best_q_3} = 7.8 \times 10^{-5}. \tag{31}$$

Using the trained model, the values of the first generalized coordinate were estimated, using the fraction of 20% from the dataset reserved for validation. Figure 16 presents the real values of the first generalized coordinate in comparison with the estimated ones from the neural network.

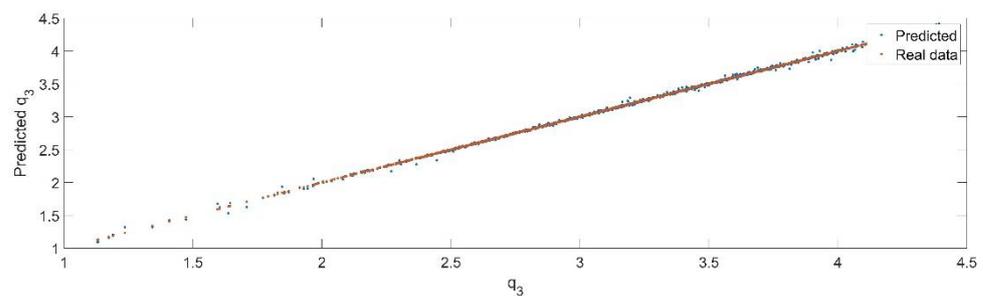


Figure 16. The comparison between the real data (the real value of q_3) and the predicted data (the predicted value of q_3).

Figure 17 presents the relative error between the real validation data and the predicted data and the model and Figure 18 presents the error histogram between the target and predicted values.

By analyzing Figures 16 and 18, the maximal error between the target and prediction was -6.76% . Nevertheless, among the 2000 target values, only for 29, the relative error was higher than 1%. The mean error between the target and prediction for the third generalized coordinate was 0.042%.

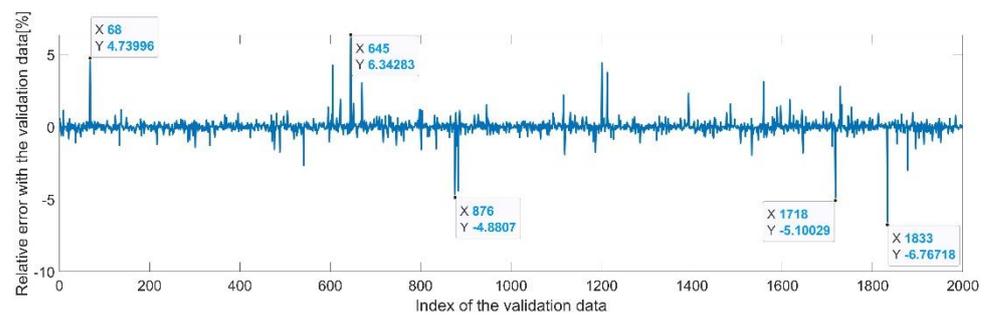


Figure 17. Relative errors between the target values and the predicted values for q_3 .

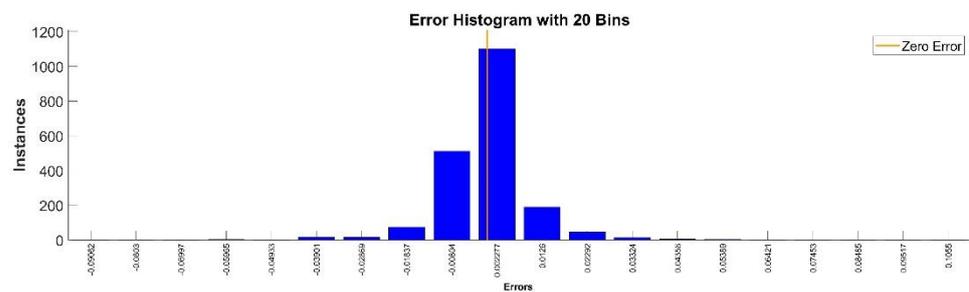


Figure 18. Error histogram between the target values and the estimated values for q_3 .

4.3.4. Results for the First Generalized Coordinate q_4

Figure 19 presents the convergence of the neural network on a logarithm scale. From the 1st epoch to the 1000th epoch, the mean squared error reached from 3.34 to 1.4×10^{-3} . In the 1 millionth epoch, the MSE reached its lowest value, as in Equation (32).

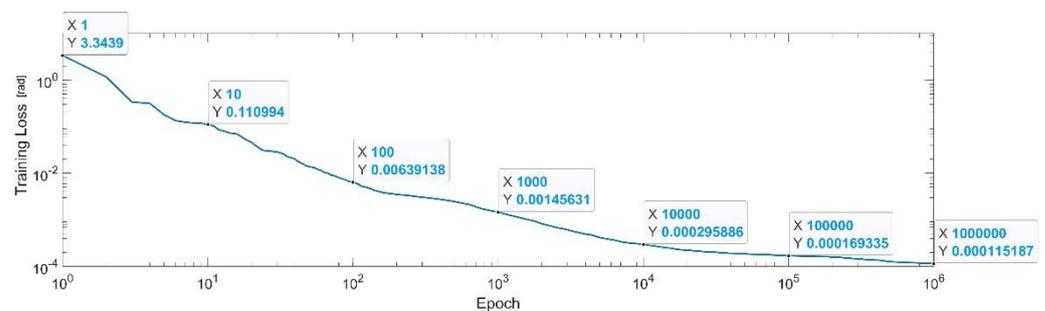


Figure 19. Convergence of the neural network for the first generalized coordinate q_4 (log scale).

$$MSE_{Best_{q_4}} = 1.14 \times 10^{-4}. \quad (32)$$

Using the trained model, the values of the first generalized coordinate were estimated, using the fraction of 20% from the dataset reserved for validation. Figure 20 presents the real values of the first generalized coordinate in comparison with the estimated ones from the neural network.

Figure 21 presents the relative error between the real validation data and the predicted data and the model and Figure 22 presents the error histogram between the target and predicted values.

By analyzing Figures 21 and 22, the maximal error between the target and prediction was around -9% . Nevertheless, among the 2000 target values, only for 33, the relative error was higher than 1%. The mean error between the target and prediction for the fourth generalized coordinate was 0.038%.

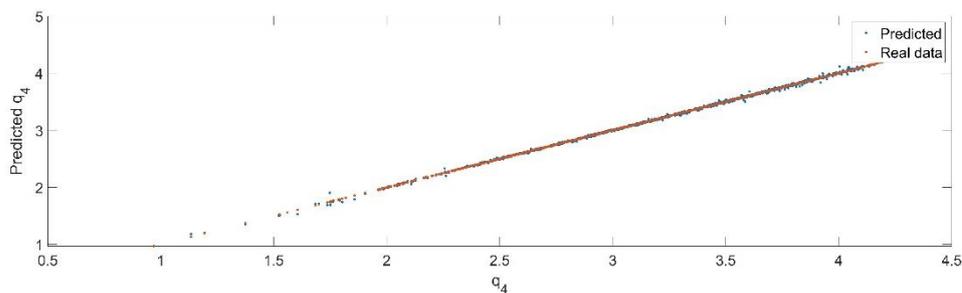


Figure 20. The comparison between the real data (the real value of q_4) and the predicted data (the predicted value of q_4).

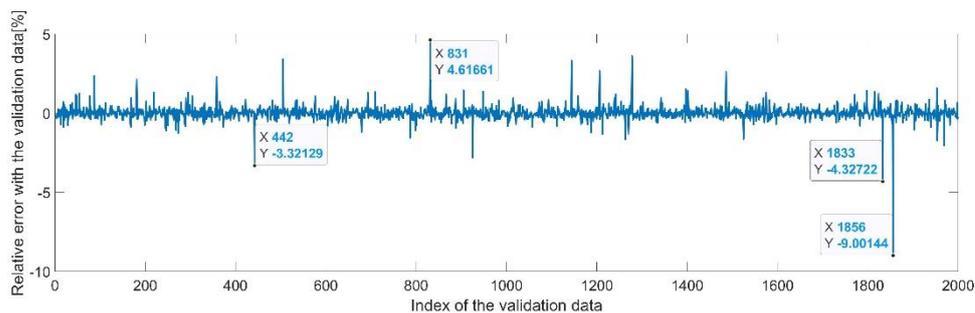


Figure 21. Relative errors between the target values and the predicted values for q_4 .

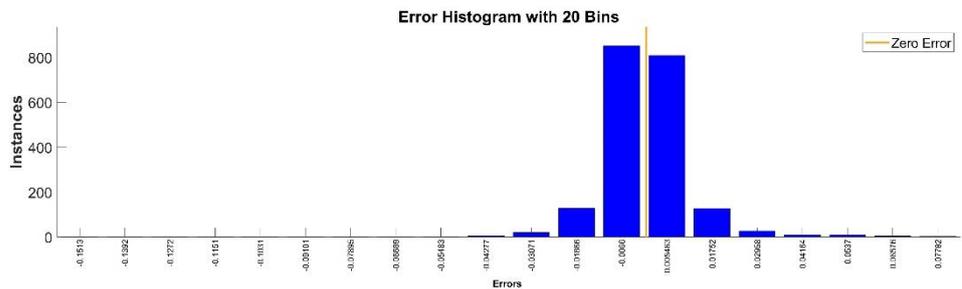


Figure 22. Error histogram between the target values and the estimated values for q_4 .

4.3.5. Results for the First Generalized Coordinate q_5

Figure 23 presents the convergence of the neural network on a logarithm scale. From the 1st epoch to the 1000th epoch, the mean squared error reached from 7.72 to 1.3×10^{-3} . In the 1 millionth epoch, the MSE reached its lowest value, as in Equation (33).

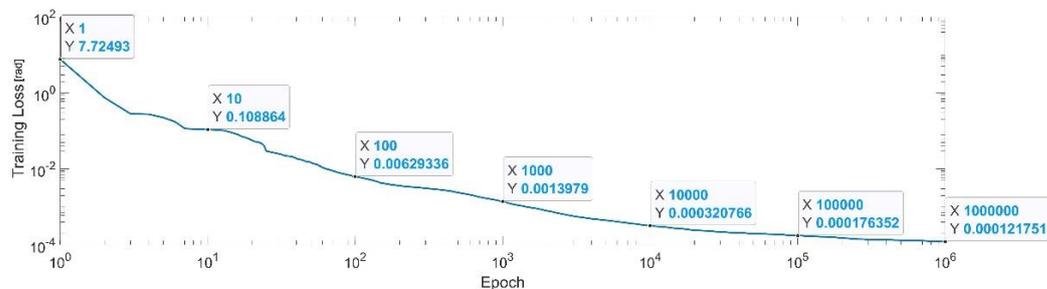


Figure 23. Convergence of the neural network for the first generalized coordinate q_5 (log scale).

$$MSE_{Best_q_5} = 1.2 \times 10^{-4}. \tag{33}$$

Using the trained model, the values of the first generalized coordinate were estimated, using the fraction of 20% from the dataset reserved for validation. Figure 24 presents the real values of the first generalized coordinate in comparison with the estimated ones from the neural network.

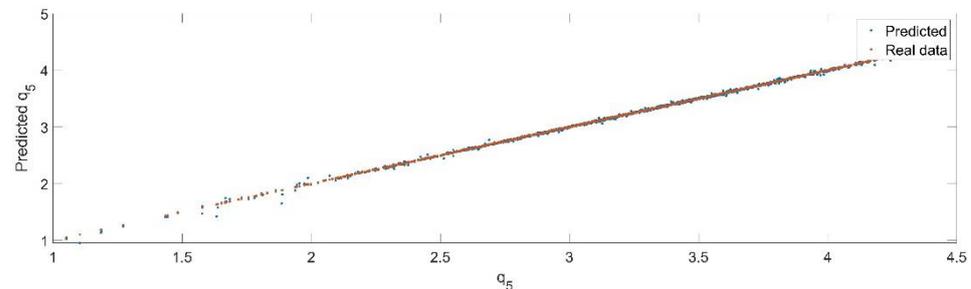


Figure 24. The comparison between the real data (the real value of q_5) and the predicted data (the predicted value of q_5).

Figure 25 presents the relative error between the real validation data and the predicted data and the model and Figure 26 presents the error histogram between the target and predicted values.

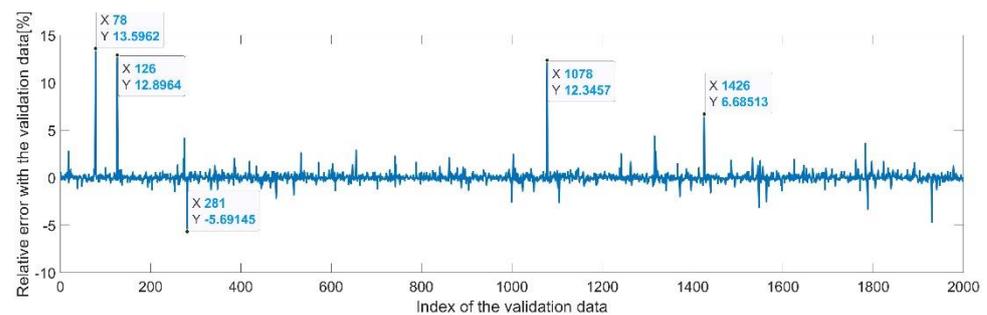


Figure 25. Relative errors between the target values and the predicted values for q_5 .

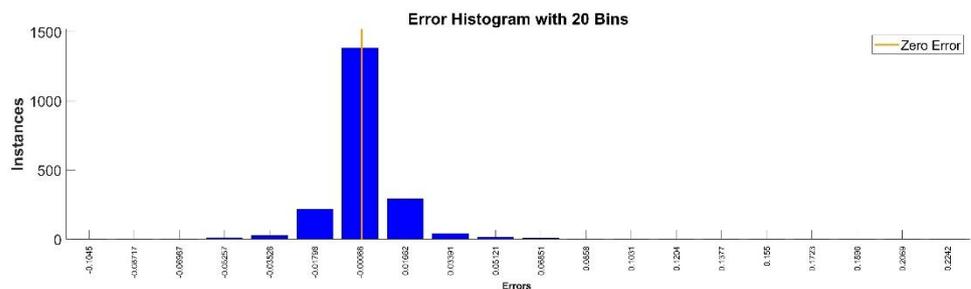


Figure 26. Error histogram between the target values and the estimated values for q_5 .

By analyzing Figures 25 and 26, the maximal error between the target and prediction was 13.59%. Nevertheless, among the 2000 target values, only for 38, the relative error was higher than 1%. The mean error between the target and prediction for the fifth generalized coordinate was 0.0309%.

4.3.6. Results for the First Generalized Coordinate q_6

Figure 27 presents the convergence of the neural network on a logarithm scale. From the 1st epoch to the 1000th epoch, the mean squared error reached from 0.72 to 1.2×10^{-3} . In the 1 millionth epoch, the MSE reached its lowest value, as in Equation (34).

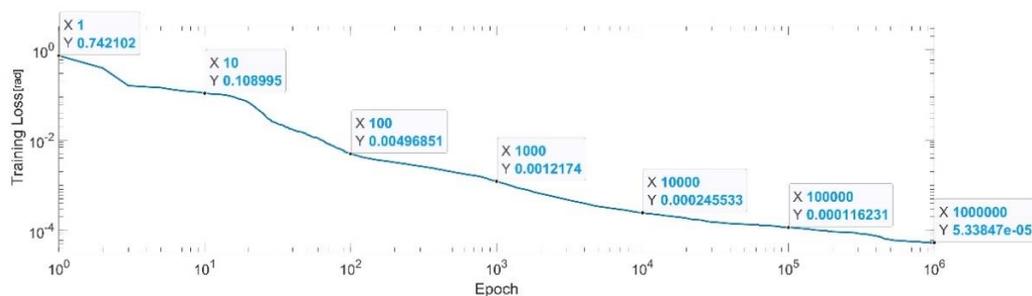


Figure 27. Convergence of the neural network for the first generalized coordinate q_6 (log scale).

$$MSE_Best_q_6 = 5.37 \times 10^{-5}. \tag{34}$$

Using the trained model, the values of the first generalized coordinate were estimated, using the fraction of 20% from the dataset reserved for validation. Figure 28 presents the real values of the first generalized coordinate in comparison with the estimated ones from the neural network.

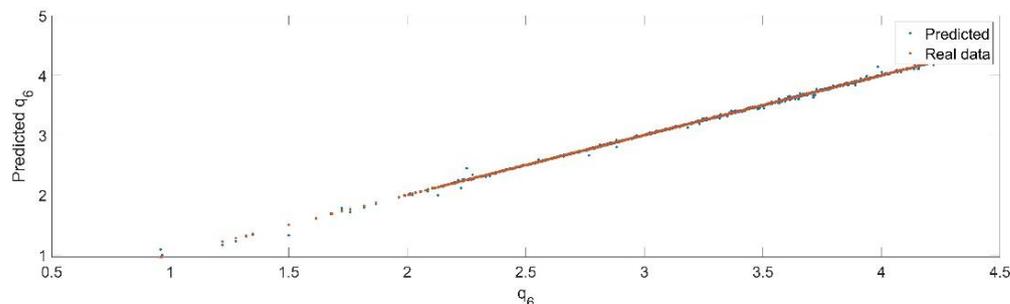


Figure 28. The comparison between the real data (the real value of q_6) and the predicted data (the predicted value of q_6).

Figure 29 presents the relative error between the real validation data and the predicted data and the model and Figure 30 presents the error histogram between the target and predicted values.

By analyzing Figures 29 and 30, the maximal error between the target and prediction was -13.3% . Nevertheless, among the 2000 target values, only for 27, the relative error was higher than 1% . The mean error between the target and prediction for the sixth generalized coordinate was 0.0058% .

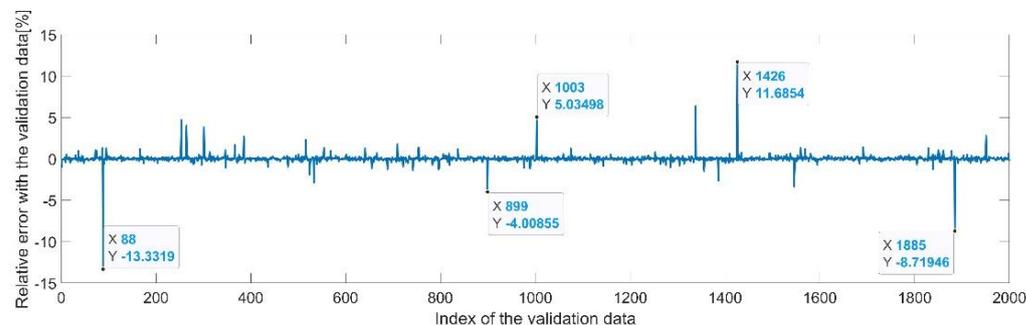


Figure 29. Relative errors between the target values and the predicted values for q_6 .

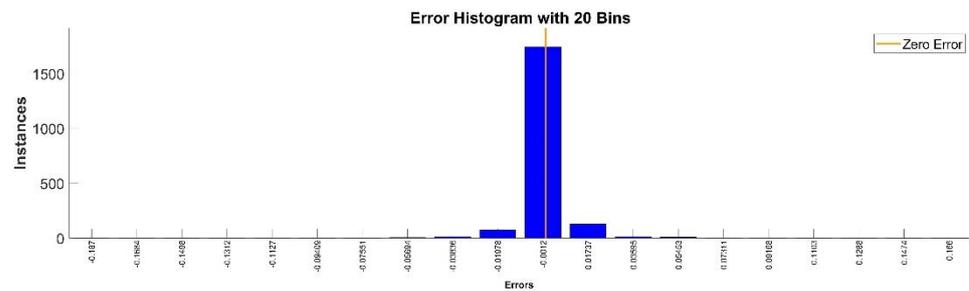


Figure 30. Error histogram between the target values and the estimated values for q_6 .

4.3.7. Comparison of the Results for the Generalized Coordinates

The numerical results for all generalized coordinates are similar in what regards the magnitude of each MSE after 1 million epochs, maximal errors and the number of values with errors higher than 1%.

In what regards the MSE, it reached similar degrees of magnitude for each generalized coordinate; the highest value was in the case of q_4 , where the MSE was 12.1×10^{-5} , and the lowest was in the case of q_1 , where the MSE was 3.7×10^{-5} .

The highest errors of estimation of the generalized coordinates were around 11–13% for q_1 , q_2 , q_5 and q_6 and around 6–9% for q_3 and q_4 , which may seem high in absolute value. Still, the maximal number of relative errors higher than 1% is 38, among 2000 validation cases for each generalized coordinate, which leads to a ratio of 1.9%. Moreover, considering the fact that the values of the mean relative errors for the estimation of each generalized coordinate were below 0.104% (correspondent for the case of q_2), it can be concluded that the estimation of the values of the generalized coordinate using the proposed methodology has a good level of accuracy.

5. Results—Estimation of the Workspace Using a Neural Network

In this section, we applied the methodology presented in Section 2 in order to estimate the volume of the workspace of the parallel robot presented in Section 3.

5.1. Generation of the Training and Test Dataset

The desired output of the neural network is the volume of the workspace given the parameters that describe the architecture of the robot. Therefore, in this step, the training data has to be generated, i.e., a dataset of workspaces is computed (which is the output of the neural network) based on the parameters that define the architecture of the robot (which are the input of the neural network).

By evaluating all the parameters that define the architecture of the robot, the ones that were considered as most influent upon the size of the workspace were: l_1 and l_2 , the lengths of the mobile elements of the RUS kinematic open loop, R and r , the radii of the fixed and mobile platforms, respectively, and β_i and α_i , the angles of positioning of the rotational joints on the fixed platforms and of the spherical joints on the mobile platform, respectively. Since the angles correspondent to an odd value of the index i are equal to each other and the angles correspondent to an even value of the index i are equal to each other, in order to reduce the number of parameters, the ratios between two consecutive angles from the fixed and from the mobile platforms were introduced, notated with $ratio_d$ and $ratio_u$, expressed in Equations (35) and (36) as:

$$ratio_d = \frac{\beta_1}{\beta_2} = \frac{\beta_3}{\beta_4} = \frac{\beta_5}{\beta_6}, \quad (35)$$

$$ratio_u = \frac{\alpha_1}{\alpha_2} = \frac{\alpha_3}{\alpha_4} = \frac{\alpha_5}{\alpha_6}. \quad (36)$$

Therefore, the input parameters for the neural network for estimation of the volume of the workspace were l_1 , l_2 , R , r , $ratio_d$ and $ratio_u$. In order to generate the training data, for

each of the parameters, a number of 10,000 random values were considered. The random values were generated between the minimal and maximal limits for each parameter:

$$\begin{aligned}
 0.1 < l_1 < 0.3 [m], \\
 0.2 < l_2 < 0.9 [m], \\
 0.1 < r < 0.3 [m], \\
 0.2 < R < 0.5 [m], \\
 0.5 < ratio_d < 2, \\
 0.5 < ratio_u < 2.
 \end{aligned} \tag{37}$$

With these random variables, a number of 10,000 of robot workspaces were evaluated, corresponding to 10,000 input parameters. A fraction of 80% of this dataset was reserved for training of the neural network and the rest of 20% was reserved for validation.

5.2. Define of the Parameters of the Neural Network

Similar to Section 4, the type of network that was used to evaluate the volume of a workspace based on the input parameters was a feedforward, fully connected neural network implemented in the MATLAB environment. In order to establish the parameters of the neural network, another optimization was run using a random search algorithm from the MATLAB environment. The optimization was run for 1000 iterations and, on each iteration, the neural network was trained based on the training data already prepared. On each iteration, the neural network was trained up to the point in which the number of epochs reached 1000. The following parameters were obtained for the neural network:

$$No_{layers} = 2, \tag{38}$$

$$Size_{layers} = [30, 11], \tag{39}$$

$$\lambda = 1.6331 \times 10^{-8}, \tag{40}$$

$$Act_{fcn} = tanh. \tag{41}$$

In Equations (38)–(41), the parameters represent the same as the ones from Equations (25)–(28).

5.3. Numerical Results for the the Workspace

Figure 31 presents the convergence of the neural network in log scale. The condition of convergence of the neural network was chosen as reaching 1 million epochs or reaching a mean squared error of 1×10^{-7} . From the 1st epoch to the 1000th epoch, the MSE reached from 0.03 up to 3.4×10^{-6} . From the 1000th epoch up to the 1 millionth epoch, the MSE was lowered with just one degree of magnitude. After reaching the first condition of convergence, 1 million epochs, the MSE for the last epoch is presented in Equation (42).

$$MSE_Best_WS = 5.57 \times 10^{-7} \tag{42}$$

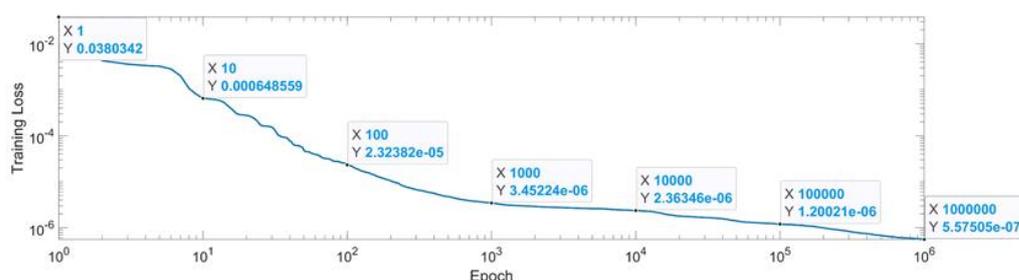


Figure 31. Convergence of the neural network for workspace (log scale).

Figure 32 presents a comparison between the real data, the actual values of the volume of the workspaces (validation data, evaluated numerically) and the predicted data, the value of the volume of the workspace predicted by the feedforward trained neural network. As seen in the figure, the trained model was able to predict the data with high precision.

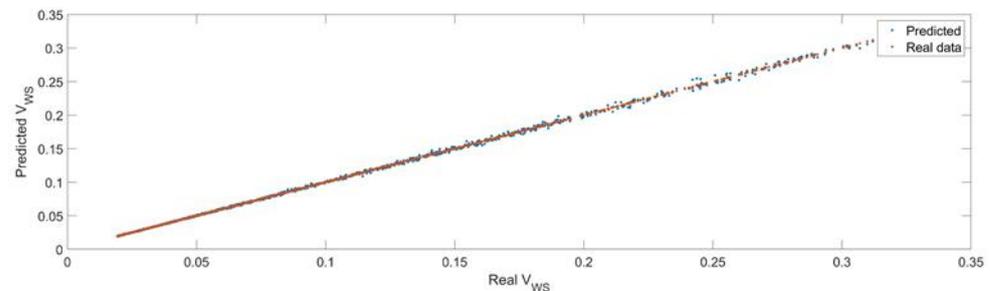


Figure 32. The comparison between the real data (the real volume of the workspaces) and the predicted data (the predicted values of the volume of the workspace).

Figure 33 presents the relative error between the real validation data and the predicted data and the model and Figure 34 presents the error histogram between the target and predicted values.

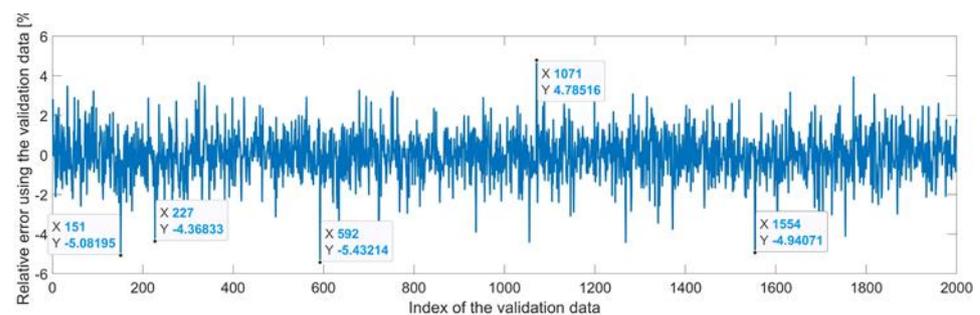


Figure 33. Relative errors between the target values and the predicted values for the workspace.

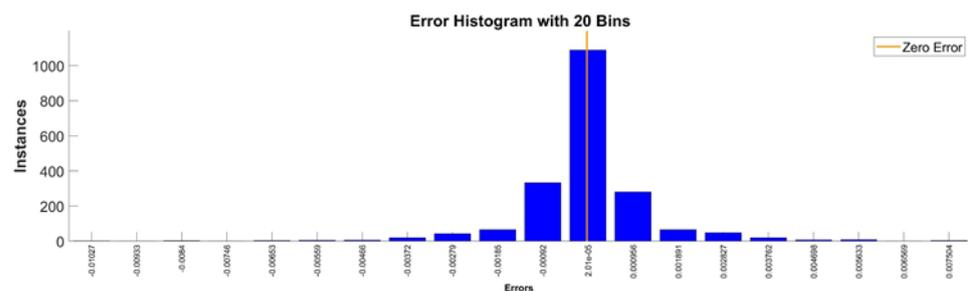


Figure 34. Error histogram between the target values and the estimated values for the workspace.

By analyzing Figures 33 and 34, the maximal error between the target and prediction was around 5%. Nevertheless, among the 2000 target values, for 327 the relative error was higher than 1%. The mean error between the target and prediction values for the first generalized coordinate was -0.0029% .

6. Discussion

By analyzing the numerical results from Section 4, in the case of kinematics estimation, one relevant aspect is that each generalized coordinate is estimated separately with individual neural networks for each. Therefore, the presented method may be applied even for different numbers of DOFs or type of robots. The accuracy of the trained models is provided by the MSE and the relative mean error between the real validation and estimated

data, which was below 0.1%. The estimation of the kinematics using neural networks may be used in order to avoid carrying out classical kinematics algorithms for real-time implementation. Moreover, it may present advantages when applied to robots for which the kinematics problem may have more than one solution or to redundant robots. In the case of the estimation of kinematics using neural networks, the work presented may be further developed in what regards the estimation of first and second differentials of the generalized coordinates (i.e., the generalized velocities and accelerations).

In the case of estimation of the volume of the workspace, the main outcome of the proposed approach is for the process of designing or redesigning a robot. By applying this method, a human designer is able to estimate directly the volume of the workspace, without being required to numerically compute again the workspace (for each time the value of a parameters that defines the architecture of the robot has to be changed). The accuracy of the trained model is provided by the MSE and the relative mean error between the real validation and estimation data, which was below 0.1%. Still, in this case, the number of test cases that presented a relative error larger than 1% was higher than in the case of estimation of the kinematics.

7. Conclusions

This paper presented the development process of an artificial neural network, namely a feedforward fully connected neural network in order to estimate the inverse kinematics and the volume of the workspace of a robot. The values of the generalized coordinates of the robot were estimated by the neural network from the poses (position and orientation) of the end effector and the volume of the workspace was estimated based on the parameters that describe the architecture of the robot.

The datasets the neural networks were trained and validated were generated using classical inverse kinematics analysis and workspace generation for a six-DOF parallel robot composed by six identical RUS open loops that interconnect the fixed and mobile platforms.

The topology of the neural networks was established based on two optimizations implemented in MATLAB. The optimization had as objective function the MSE and the parameters that were optimized were the number of layers, the size of each layer, the type of the activation function and the regularization rate.

In what regards the future development of this work for the estimation of the workspace, in this paper, only the volume of the workspace was considered, without implementing the shape and size of the workspace as input or output. This is why the future outlook for the estimation of the workspace is to further develop the machine learning model to be able to estimate the workspace, including the shape and dimensions.

Author Contributions: Writing—original draft, C.B. (Cătălin Boanta); Writing—review & editing, C.B. (Cornel Brişan). The authors declare similar contributions in writing this article. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Project “Entrepreneurial competences and excellence research in doctoral and postdoctoral programs—ANTREDOC”, project co-funded by the European Social Fund financing agreement no. 56437/24.07.2019. Moreover, this paper received financial support by Romanian Ministry of Research and Innovation, project PN-IIIP2-2.1-PED-2019-0085 CONTRACT 447PED/2020 (Acronym POSEIDON).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Laribi, M.A.; Carbone, G.; Zeghloul, S. On the optimal design of cable driven parallel robot with a prescribed workspace for upper limb rehabilitation tasks. *J. Bionic Eng.* **2019**, *16*, 503–513. [\[CrossRef\]](#)
2. Yi, W.; Zheng, Y.; Wang, W.; Tang, X.; Liu, X.; Meng, F. Optimal design and force control of a nine-cable-driven parallel mechanism for lunar takeoff simulation. *Chin. J. Mech. Eng.* **2019**, *32*, 73. [\[CrossRef\]](#)
3. Gao, Z.; Zhang, D. Performance analysis, mapping, and multiobjective optimization of a hybrid robotic machine tool. *IEEE Trans. Ind. Electron.* **2014**, *62*, 423–433. [\[CrossRef\]](#)
4. El Hraiech, S.; Chebbi, A.H.; Affi, Z.; Romdhane, L. Genetic algorithm coupled with the Krawczyk method for multi-objective design parameters optimization of the 3-UPU manipulator. *Robotica* **2020**, *38*, 1138–1154. [\[CrossRef\]](#)
5. Cheng, H.; Li, W. Reducing the frame vibration of delta robot in pick and place application: An acceleration profile optimization approach. *Shock Vib.* **2018**, *2018*, 2945314. [\[CrossRef\]](#)
6. Diveev, A.; Dubrovin, G.; Malyshev, D.; Nozdracheva, A. Geometric parameters and workspace optimization of sitting-type lower limb rehabilitation robot. In *Latin American Symposium on Industrial and Robotic Systems*; Springer: Cham, Switzerland, 2019; pp. 279–289. [\[CrossRef\]](#)
7. Hamida, I.B.; Laribi, M.A.; Mlika, A.; Romdhane, L.; Zeghloul, S. Comparative study of design of a 3-DOF translational parallel manipulator with prescribed workspace. In *IFTToMM World Congress on Mechanism and Machine Science*; Springer: Cham, Switzerland, 2019; pp. 501–512. [\[CrossRef\]](#)
8. López, E.G.; Yu, W.; Li, X. Optimal design of a parallel robot using neural network and genetic algorithm. In Proceedings of the 2019 Tenth International Conference on Intelligent Control and Information Processing (ICICIP), Marrakesh, Morocco, 14–19 December 2019; pp. 64–69. [\[CrossRef\]](#)
9. Huang, G.; Zhang, D.; Zou, Q. Neural network and performance analysis for a novel reconfigurable parallel manipulator based on the spatial multiloop overconstrained mechanism. *Int. J. Aerosp. Eng.* **2020**, *2020*, 8878058. [\[CrossRef\]](#)
10. Chen, Q.; Yang, C. Hybrid algorithm for multi-objective optimization design of parallel manipulators. *Appl. Math. Model.* **2021**, *98*, 245–265. [\[CrossRef\]](#)
11. Gholami, A.; Homayouni, T.; Ehsani, R.; Sun, J.Q. Inverse Kinematic Control of a Delta Robot Using Neural Networks in Real-Time. *Robotics* **2021**, *10*, 115. [\[CrossRef\]](#)
12. López, E.J.; De La Mora-Pulido, D.S.; De La Mora-Pulido, R.S.; Ochoa-Estrella, F.J.; Flores, M.A.; Luna-Sandoval, G. Modeling in Two Configurations of a 5R 2-DoF Planar Parallel Mechanism and Solution to the Inverse Kinematic Modeling Using Artificial Neural Network. *IEEE Access* **2021**, *9*, 68583–68594. [\[CrossRef\]](#)
13. Csiszar, A.; Eilers, J.; Verl, A. On solving the inverse kinematics problem using neural networks. In Proceedings of the 2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Auckland, New Zealand, 21–23 November 2017; pp. 1–6. [\[CrossRef\]](#)
14. Ren, H.; Ben-Tzvi, P. Learning inverse kinematics and dynamics of a robotic manipulator using generative adversarial networks. *Robot. Auton. Syst.* **2020**, *124*, 103386. [\[CrossRef\]](#)
15. Toquica, J.S.; Oliveira, P.S.; Souza, W.S.; Motta, J.M.S.; Borges, D.L. An analytical and a Deep Learning model for solving the inverse kinematic problem of an industrial parallel robot. *Comput. Ind. Eng.* **2021**, *151*, 106682. [\[CrossRef\]](#)
16. Zhang, D.; Lei, J. Kinematic analysis of a novel 3-DOF actuation redundant parallel manipulator using artificial intelligence approach. *Robot. Comput. Integr. Manuf.* **2011**, *27*, 157–163. [\[CrossRef\]](#)
17. Sanjeev, M.M.; Thomas, M.J.; Kumar, T.S.; Sudheer, A.P.; Joy, M.L. Determination of inverse kinematic solutions for a 3 degree of freedom parallel manipulator using machine learning. In Proceedings of the 2020 IEEE Students Conference on Engineering & Systems (SCES), Prayagraj, India, 10–12 July 2020; pp. 1–6.
18. Šegota, S.B.; Anđelić, N.; Mrzljak, V.; Lorencin, I.; Kuric, I.; Car, Z. Utilization of multilayer perceptron for determining the inverse kinematics of an industrial robotic manipulator. *Int. J. Adv. Robot. Syst.* **2021**, *18*, 1729881420925283. [\[CrossRef\]](#)
19. Shah, S.K.; Mishra, R.; Ray, L.S. Solution and validation of inverse kinematics using Deep artificial neural network. *Mater. Today Proc.* **2020**, *26*, 1250–1254. [\[CrossRef\]](#)
20. Brisan, C.; Csiszar, A. Computation and analysis of the workspace of a reconfigurable parallel robotic system. *Mech. Mach. Theory* **2011**, *46*, 1647–1668. [\[CrossRef\]](#)
21. Liao, P. Deep neural network based subspace learning of robotic manipulator workspace mapping. In Proceedings of the 2018 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO), Prague, Czech Republic, 19–21 May 2018; pp. 109–120. [\[CrossRef\]](#)
22. Liao, P.; Mao, J. Full Workspace Generation of Serial-link Manipulators by Deep Learning based Jacobian Estimation. *arXiv* **2018**, arXiv:1809.05020. [\[CrossRef\]](#)
23. Boanta, C.; Brisan, C. Development of a Neural Network Method to Estimate the Workspace of a Parallel Robot. In Proceedings of the 2022 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, 19–21 May 2022; pp. 1–6.
24. Boanta, C.; Brisan, C. Topology based enhancement of the workspace of a reconfigurable parallel haptic interface. In Proceedings of the 2016 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, 19–21 May 2016; pp. 1–6.

-
25. Boanta, C.; Besoiu, S.; Brisan, C. Optimization of a Spatial 2 DOF Parallel Mechanism Used for Orientation. In *New Advances in Mechanism and Machine Science*; Springer: Cham, Switzerland, 2018; pp. 427–435. [[CrossRef](#)]
 26. Zhang, D. *Parallel Robotic Machine Tools*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009; ISBN 978-1-4419-1117-9. [[CrossRef](#)]