Hindawi
Complexity
Volume 2020, Article ID 8564140, 10 pages
https://doi.org/10.1155/2020/8564140

Research Article

Applications of Artificial Intelligence Techniques to Enhance Sustainability of Industry 4.0: Design of an Artificial Neural Network Model as Dynamic Behavior Optimizer of Robotic Arms

Aydin Azizi

School of Engineering, Computing and Mathematics, Oxford Brookes University, Wheatley Campus, Oxford OX33 1HX, UK

Correspondence should be addressed to Aydin Azizi; aydin.azizi@brookes.ac.uk

Received 16 April 2019; Revised 6 November 2019; Accepted 19 December 2019; Published 21 March 2020

Guest Editor: Robert Hakl

Industrial robots have a great impact on increasing the productivity and reducing the time of the manufacturing process. To serve this purpose, in the past decade, many researchers have concentrated to optimize robotic models utilizing artificial intelligence (AI) techniques. Gimbal joints because of their adjustable mechanical advantages have been investigated as a replacement for traditional revolute joints, especially when they are supposed to have tiny motions. In this research, the genetic algorithm (GA), a well-known evolutionary technique, has been adopted to find optimal parameters of the gimbal joints. Since adopting the GA is a time-consuming process, an artificial neural network (ANN) architecture has been proposed to model the behavior of the GA. The result shows that the proposed ANN model can be used instead of the complex and time-consuming GA in the process of finding the optimal parameters of the gimbal joint.

1. Introduction

In the definition of the sustainable economic model, natural resources such as energy and material are considered as limited resources. Industrial robots can be utilized as a potential solution for the issue by increasing the productivity and reducing the wastes which results in the reduction in emission and contributing to sustainability [1]. In the past decade, designing more precise and accurate robots has been a point of research interest of many scientists. In particular, many research studies have been concentrated to utilize artificial intelligence (AI) techniques to contribute to economic and environmental sustainability [2–4].

A variety of novel techniques such as ring probabilistic logic neural networks (RPLNNs) [5], genetic algorithms (GA), particle swarm (PS), hybrid algorithms, and agent-based algorithms have been presented by researchers to provide an approximate solution of the synthesis of mechanisms [5, 6]. In 2017, Soldberg [7] investigated the possibility of object detection in agricultural robots via deep neural networks. Jin et al. [8] presented the special form of neural networks to optimize the redundant manipulators. Fuzzy wavelet ANN approach has been utilized by Yen et al. [9] in 2017 to control the industrial robot manipulator using fuzzy wavelet neural networks. Also, in 2018, they adopted recurrent fuzzy wavelet neural network approach to control the manipulator [10].

In this paper, an ANN architecture has been adopted to model the behavior of the GA optimizer in the process of finding the optimal parameters of robotic arms equipped with a gimbal driver. The research presents a novel approach in the training process of ANN by adopting data of genetic algorithms (GAs) to train the proposed ANN, so at the end of the training process, the proposed ANN acts as a function optimizer instead of GA. In this research, MATLAB software has been utilized to generate the ANN model and perform the GA optimization process. It is important to highlight the point that other effective optimization techniques such as PSO, bee colony, and RPLNN can be adopted to train the proposed ANN, and GA has been utilized just as an example of an optimization technique.

2. Methodology

The ultimate aim of this research is to develop an ANN model as a function approximator which is able to model the dynamic behavior of the GA optimizer in the process of finding the optimal parameters of gimbal drive in robotic arms.

Different from previous works, the designed ANN has not been trained by the general extracted data from the mathematical model of the robotic arm, but its reference model is the GA as the parameter optimizer of the gimbal equipped robotic arm model. It means that the proposed ANN should act as a function optimizer and will eliminate the need of utilizing optimization techniques.

To reach this goal at the first step, the behavior of the gimbal drive should be modeled mathematically, and the next step is utilizing the introduced gimbal drive in 3 different well-known robotic arms and introducing the related mathematical models. The third step is to find the optimum design parameters utilizing GA which is one of the well-known AI evolutionary techniques. The last step which is the novelty part of this research is to design and train an ANN to act instead of the optimization algorithm, GA. The proposed approach has been illustrated as a flowchart in Figure 1.

To fulfill the task, the rest of the manuscript has been prepared in the following sequence: a brief introduction of gimbal mechanisms is presented in the next section. Also, the resistivity ellipsoid is formulated to measure the manipulability of robotic arms. Next, to be able to apply maximum force in a desired direction, genetic algorithm (GA) as one of the well-known evolutionary optimization techniques has been employed to optimize the highly nonlinear fitness function. As GA is very time-consuming, an artificial neural network (ANN) as the novelty of this research has been utilized to learn the relation between the inputs and outputs of GA. The ANN is trained using the optimization results obtained from GA in several randomly generated configurations. Thus, the trained ANN could replace the whole optimization process and act as a function optimizer. This way enables utilizing the trained ANN model to obtain the optimized results for new configurations instead of the GA optimizer. To have a better perception of the effectiveness of this method, three simple robotic arms are selected to investigate the task.

2.1. *The Gimbal Drive.*

In Figure 2, a regular one-degree-of-freedom gimbal drive is shown. Following is the governing equation on gimbal mechanism [11]:

$$\theta_{\text{out}} = \theta_{\text{of\_out}} + \tan^{-1}\left(\tan\theta_{\text{c}}\cos\theta_{\text{in}}\right), \tag{1}$$

where θ_{in} is the input angle (angle of rotation of vertical shaft), θ_{out} is the output angle (angle of rotation of the horizontal shaft/frame), θ_{c} is the gradient of the truncated cylinder, and $\theta_{\text{of\_out}}$ is the offset angle at the output. It can be deduced that the gradient of the truncated cylinder, θ_{c}, is a design parameter because it sets the range of output angle.

Verity is that, for any input value (θ_{in}), there exist two possible output configurations (θ_{out}), which is another advantage of the gimbal mechanism. The effect of the reduction ratio should be considered too. We also should notice that the gradient of θ_{c} is a design parameter because it will set the range of the output angle.

To study manipulability modality, the classical criterion of resistivity ellipsoid is formulated for manipulators with traditional revolute joints. It is compared to the same modality of the same arm in which some joints are substituted with gimbal transmissions.

To have the maximum achievable force at the manipulator tip point, the classical hypothesis is to assume that the Euclidian norm of the joint torques remains unity [11]:

$$\tau^{T}.\tau = 1. \tag{2}$$

The relation of the task-space force F and the vector of joint torques τ is given by [11]

$$\tau = J^{T}F, \tag{3}$$

where J is the Jacobian of the whole manipulator.

By substituting (3) in (2), one obtains the following equation:

$$F^{T}JJ^{T}F = 1. \tag{4}$$

Relation (4) changes the hypersphere of the joint forces (2) into a hyperellipsoid that is called the resistivity ellipsoid. This method is largely employed to evaluate the manipulability qualities of robotic arms.

Here, achieving greater force in a specified direction by using the gimbal mechanism is the main goal. The task-space force vector is expressed as [11]

$$F = f\begin{bmatrix} \mathrm{d}x \\ \mathrm{d}y \\ \mathrm{d}z \end{bmatrix} = f\widehat{F}, \tag{5}$$

where f is the magnitude of F and $\begin{bmatrix} \mathrm{d}x & \mathrm{d}y & \mathrm{d}z \end{bmatrix}^{T}$ are the unit vectors on the direction of f along X, Y, and Z axes, respectively, and to maintain conciseness, substitute by $\widehat{F}$. Substituting (5) in (4) yields the following equation [11]:

$$f^{2}\left(\widehat{F}^{T}JJ^{T}\widehat{F}\right) = 1. \tag{6}$$

Thus, the magnitude of the applicable force in a desired direction at a specific point of manipulator's workspace should be determined. Comparing the magnitudes of the applicable force, f, for different types of manipulators, would lead to recognizing the efficient robot design and whether using gimbal transmission instead of the traditional revolute joints improves robot's manipulability or not.

2.1.1. *Case Studies*

(1) RRR Arm. To get numerical examples, three case studies have been performed. First, a spatial robot with three revolute joints (RRR or 3R) shown in Figure 3 is investigated to analyze the advantages of gimbal drive in robot joint transmission. For convenience,

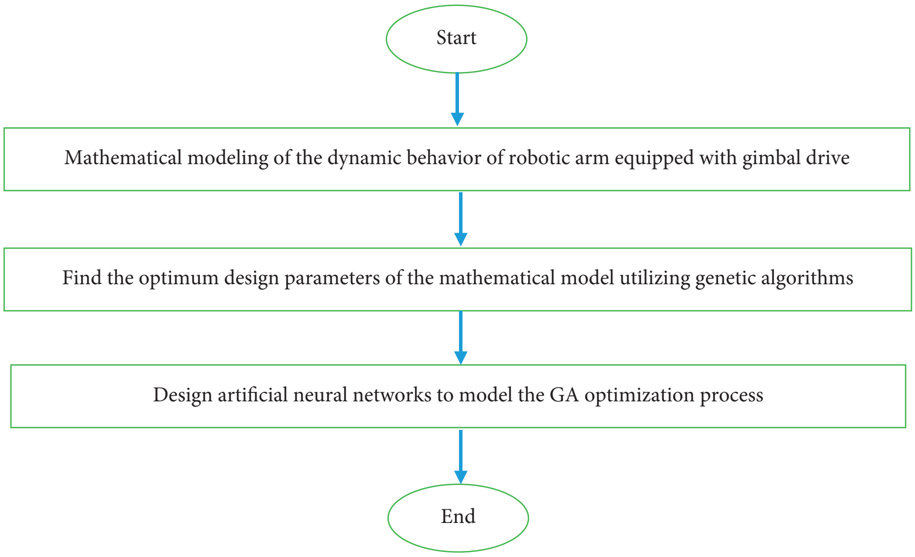
Start
Mathematical modeling of the dynamic behavior of robotic arm equipped with gimbal drive
Find the optimum design parameters of the mathematical model utilizing genetic algorithms
Design artificial neural networks to model the GA optimization process
End

Figure 1: Flowchart of the proposed methodology.

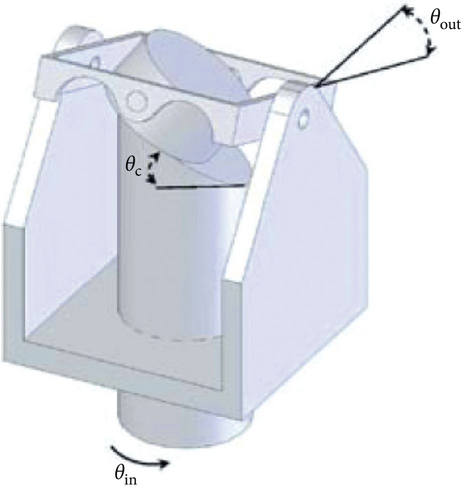

Figure 2: The gimbal drive [11].

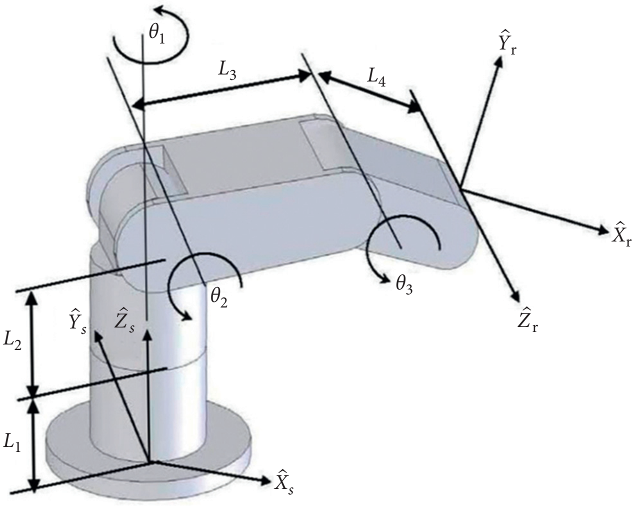

Figure 3: RRR spatial robot arm [11].

$$c_{m\ldots n} = \cos\left(\theta_m + \cdots + \theta_n\right),$$
$$s_{m\ldots n} = \sin\left(\theta_m + \cdots + \theta_n\right), \tag{7}$$

are used instead, respectively. Assuming $L_1 = L_2 = L_4 = 0.5$ m and $L_3 = 1$ m, the Jacobian matrix of the 3R spatial robot is given by [11]

$$J = \begin{bmatrix} -\frac{1}{2}s_1c_{23} - s_1c_2 & -\frac{1}{2}c_1s_{23} - c_1s_2 & -\frac{1}{2}c_1s_{23} \\ \frac{1}{2}c_1c_{23} + c_1c_2 & -\frac{1}{2}s_1s_{23} - s_1s_2 & -\frac{1}{2}s_1s_{23} \\ 0 & \frac{1}{2}c_{23} + c_2 & \frac{1}{2}c_{23} \end{bmatrix}. \tag{8}$$

Now it is desired to use gimbal transmissions at joints 2 and 3. To obtain a new Jacobian matrix, one shall substitute for θ_2 and θ_3 of the 3R spatial robot. Assuming $\theta_{\text{of\_out}} = 0$ and $\theta_c = 45^\circ$, the Jacobian matrix for the manipulator with gimbal transmissions at joints 2 and 3 becomes [11]

$$J_G = \begin{bmatrix} R_1 & R_2 & R_3 \\ R_4 & R_5 & R_6 \\ R_7 & R_8 & R_9 \end{bmatrix}, \tag{9}$$

where J_G is the Jacobian of the manipulator when a gimbal transmission is used at joints 2 and 3. Using manipulability ellipsoid, f in the simple and gimbal equipped 3R spatial robots has been compared by Mohammadi et al. [11].

(2) Stanford Arm. A simplified version of Stanford arm shown in Figure 4 has been investigated here. The Jacobian matrix for this arm, assuming $L_1 = 1$ m and $L_2 = 0.5$ m, is as follows [11]:

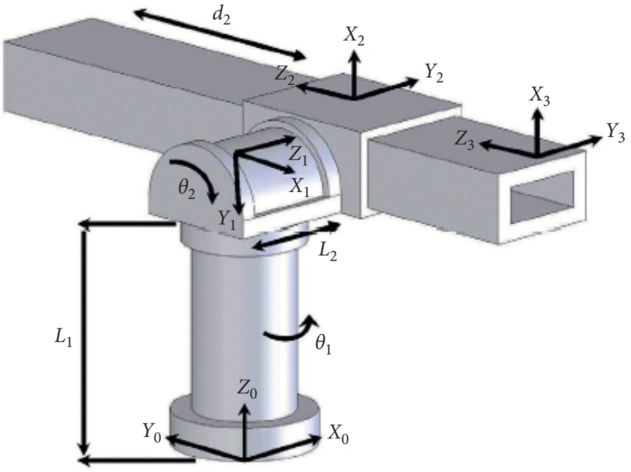

Figure 4: Stanford arm [11].

$$J = \begin{bmatrix} d_3c_1c_2 - \frac{1}{2}s_1 & -d_3s_2s_1 & s_1c_2 \\ -d_3c_2c_1 - \frac{1}{2}c_1 & -d_3c_1s_2 & c_1c_2 \\ 0 & -d_3c_2 & s_2 \end{bmatrix}. \tag{10}$$

Using gimbal drive at joint 2, the new Jacobian matrix is obtained by substituting from (1) for θ_2 in (11). Assuming $\theta_{\text{of\_out}} = 0$ and $\theta_c = 45^\circ$, the following Jacobian matrix is derived [11]:

$$J_G = \begin{bmatrix} R_1 & R_2 & R_3 \\ R_4 & R_5 & R_6 \\ R_7 & R_8 & R_9 \end{bmatrix}. \tag{11}$$

$$J_G = \begin{bmatrix} \frac{-s_1c_2d_3}{\left(1+c_2^2\right)^{(1/2)}} - \frac{1}{2}c_1 & \frac{-c_1s_2d_3}{\left(1+c_2^2\right)^{(1/2)}} + \frac{c_1c_2^2d_3s_2}{\left(1+c_2^2\right)^{(3/2)}} & \frac{c_1c_2}{\left(1+c_2^2\right)^{(1/2)}} \\ \frac{c_1c_2d_3}{\left(1+c_2^2\right)^{(1/2)}} - \frac{1}{2}s_1 & \frac{-s_1s_2d_3}{\left(1+c_2^2\right)^{(1/2)}} + \frac{s_1c_2^2d_3s_2}{\left(1+c_2^2\right)^{(3/2)}} & \frac{s_1c_2}{\left(1+c_2^2\right)^{(1/2)}} \\ 0 & \frac{c_2s_2d_3}{\left(1+c_2^2\right)^{(3/2)}} & \frac{1}{\left(1+c_2^2\right)^{(1/2)}} \end{bmatrix}. \tag{12}$$

where J_G is the Jacobian of the manipulator where a gimbal transmission is used at joint 2. Using manipulability ellipsoid, we compare f in the simple and gimbal equipped Stanford arm with traditional revolute joints and the arm equipped with gimbal transmissions at revolute joint 2. To this end, d_3 (the joint variable of the arm's mere prismatic joint) is set to 1 m. The direction of the desired force vector is assumed to be $dx = dy = dz = 1$ [11].

(3) RPR Planar Arm. Figure 5 shows a revolute-prismatic-revolute (i.e., RPR) planar manipulator. The Jacobian matrix for the arm, assuming $L_1 = 1$ m and $L_2 = 0.5$ m, is as follows [11]:

$$J = \begin{bmatrix} -c_{13}L_2 - s_1L_1 + c_1d_2 & s_1 & -c_{13}L_2 \\ -s_{13}L_2 - c_1L_1 + s_1d_2 & -c_1 & -s_{13}L_2 \\ 0 & 0 & 0 \end{bmatrix}. \tag{13}$$

Gimbal transmission is used at joint 3 by replacing the traditional revolute joint. To obtain a new Jacobian matrix, one shall substitute from (1) for θ_3 of the robot. Assuming

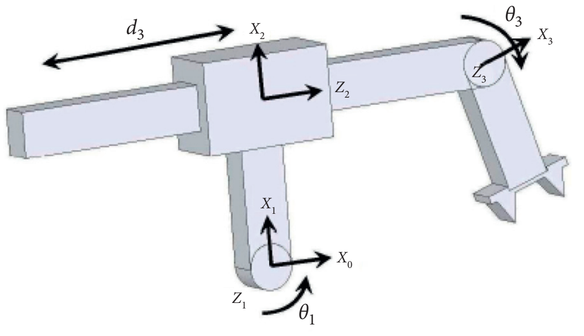

Figure 5: RPR arm [11].

$\theta_{\text{of\_out}} = 0$, the Jacobian matrix for the manipulator with gimbal transmission at joint 3 becomes [11]

$$J_G = \begin{bmatrix} R_1 & R_2 & R_3 \\ R_4 & R_5 & R_6 \\ R_7 & R_8 & R_9 \end{bmatrix}, \qquad (14)$$

where J_G is the Jacobian of the manipulator when a gimbal transmission is used at joint 3. To maintain conciseness, the elements of J_G are not shown here.

2.2. Genetic Algorithms.

2.2. Genetic Algorithms. Genetic algorithm is one of the well-known evolutionary optimization techniques, which has been adopted by many researchers to optimize complex problems [12–14]. Briefly, the optimization process by GA can be divided into 6 steps as follows [15]:

(1) Creating population of possible answers
(2) Evaluation of fitness function
(3) Creating the next generation of possible answers
(4) Applying crossover
(5) Applying mutation
(6) Repeat steps 2–5

In this research, genetic algorithm has been utilized to find optimal parameters of the gimbal joints (truncation angles). For this purpose, the population generation techniques as listed above have been explored to generate the new population of the weight matrices. This process continues until the selection of the last weight matrix or matrices has been performed. In this simulation, a population of weight matrices is produced randomly when the GA starts. In each generation, the matrices of this population have been modified through discrete crossovers and uniformly random mutations, and their fitness values have been evaluated. The cycle of reconstructing the new population with better individuals and restarting the search is repeated until a better solution is found.

The fitness function has been defined as maximizing the applicable force magnitude, *f*, by minimizing the parameters of equation (6) [16]. To this end, to generate the fitness function for each of the three robotic manipulators (RRR, RPR, and Stanford), it is needed to substitute Jacobian matrix of each of the case studies, equations (9), (12), and (14) in equation (6) [16]. Then, the optimization process has been started with a population of 20 individuals and has been run for 1000 times. As it is shown in Table 1, the probability of happening of crossover and mutation has been assumed as 0.4 and 0.01, respectively.

2.3. Neural Networks. Artificial neural networks have been utilized to different engineering and science fields such as control, data processing [17], robotics [18], function approximation [19], and pattern and speech recognition [20]. An ANN consists of interconnecting neurons which have been categorized into three layers which are, namely, input layer, hidden layer, and output layer [21]. There could be more than one hidden layer in an ANN making it more flexible and accurate to learn at the cost of learning time and effort.

Based on the type of connections between these neurons, artificial neural networks can be divided into two different groups: weighted artificial neural networks and weightless artificial neural networks [22]. This research focuses on utilizing weighted neural networks as a function optimizer.

As seen in Figure 6, parameters of the networks should be adjusted to enable the network to act as the plant (reference model). It means that by giving the same input to the ANN and the plant, the output of the network should be similar to the output of the plant. The first step in training a feed-forward network is to create the network object. It requires three arguments and returns the network object. The first argument is a matrix of sample R-element input vectors. The second argument is a matrix of sample S-element target vectors. The sample inputs and outputs are used to set up network input and output dimensions and parameters. The third argument is an array containing the sizes of each hidden layer (the output layer size is determined from the targets). In this research, the mentioned unknown function is the GA optimizer. The proposed ANN consists of a sigmoid hidden layer and a linear output layer.

Parameters of the ANN have been adjusted to enable the ANN to act as the GA optimizer (reference model) which optimizes the performance index of gimbal equipped robotic arm. It means that by giving the same input to the ANN and the GA, the output of the network should be like the output of the GA which is the evaluated fitness function. It means that the proposed ANN has a single input-single output (SISO) architecture. Levenberg–Marquardt algorithm has been adopted here to train the proposed ANN to model the dynamic behavior of the GA optimizer of the gimbal equipped robotic arms. The input of ANN is a Jacobian matrix with the dimension of $3 * 3$ and its output is the matrix of the calculated maximum force with the demotion of $1 * 1$.

Levenberg–Marquardt algorithm computes the approximate Hessian matrix, which has dimensions n-by-n. Gradient descent is the process of making changes to weights and biases, where the changes are proportional to the derivatives of network error with respect to those weights and biases. This is done to minimize network error. When the performance function has the form of a sum of squares (as is

Table 1: Parameters used for the genetic algorithm [16].

Population size	20
Generations	1000
Crossover probability	0.4
Mutation probability	0.01

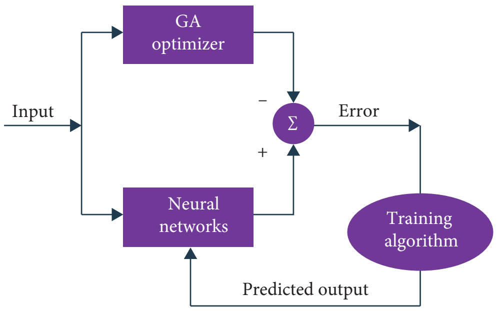

Figure 6: Proposed neural network model.

typical in training feed-forward networks), then the Hessian matrix can be approximated as

$$H = J^T J, \quad (15)$$

where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases and e is a vector of network errors. The gradient can be computed as

$$g = J^T e. \quad (16)$$

3. Results

As described previously to find the maximum value of f in each of the introduced case studies, GA optimization technique has been adopted in this research. The process has been started with a population of 20 individuals and has been run for 1000 times and the probability of happening of crossover and mutation has been assumed as 0.4 and 0.01, respectively. The results for normalized fitness function values of each of the robotic arms in each iteration of the GA optimization process have been shown in Figure 7. It can be observed that the optimization process for all the three cases works properly and normalized fitness values have been increased in each iteration.

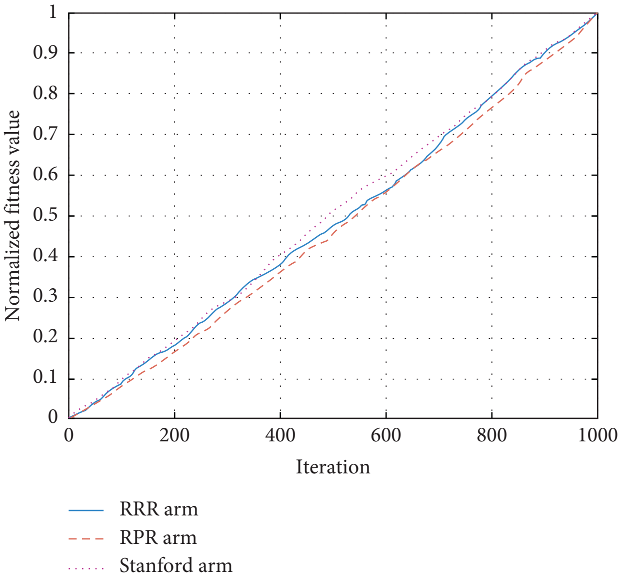

Figure 7: Normalized fitness function values in each step of GA optimization.

Table 2: GA optimization results for RRR arm [11].

Trial	1	2	3	4	Mean
θ_1	0	0	45	90	
θ_2	45	30	−15	15	
θ_3	−45	−30	30	30	
dx	0	1	1	1	
dy	0	1	1	1	
dz	1	1	0	1	
θc_2	85	85	10	85	**66.25**
θc_3	15.2	13.6	10	12.56	**12.84**
F for gimbal equipped arm	80.31	15.06	50.49	21.89	
F for simple arm	10.47	1.11	2.76	1.16	

Table 3: GA optimization results for RPR arm [11].

Trial	1	2	3	4	Mean
θ_1	0	30	45	90	
d_2	0.5	1	1	0.5	
θ_3	30	15	45	−30	
dx	1	0	1	1	
dy	1	1	1	1	
θ_c	49.10	61.37	35.8	85	**57.81**
F for gimbal equipped arm	1.09	0.81	1.97	1.41	
F for simple arm	0.97	0.76	1.41	1.29	

Table 2 shows the results achieved by running GA for four different joint values and directions based on the fitness function of RRR arm which is used to evaluate and optimize the maximum values for f, and also it compares the amount of maximum force in gimbal equipped RRR robot with the robot with revolute joints. It is shown that in these points, the amount of force delivered by the gimbal equipped robot is greater than the same quantity for the robot that uses revolute joints. Thus, it can be reasoned that gimbal drive has improved the manipulability of the robot. Here, the values of θ_{c2} and θ_{c3} are restricted to 10 to 85 degrees so that the design becomes feasible.

Table 3 shows the results achieved by running GA for four different joint values and directions based on the fitness function of RPR arm which is used to evaluate and optimize the maximum values for f. It can be inferred from results that implementation of gimbal drive in joint 2 has increased the maximum achievable force at the tip point of the manipulator. Here, the value of the truncation angle is bound to ensure design feasibility between 49 and 85 degrees.

Table 4 shows the results achieved by running GA for four different joint values and directions based on the fitness function of Stanford arm which is used to evaluate and optimize the maximum values for f. The amount of maximum force in gimbal equipped robot has been compared to

Table 4: GA optimization results for Stanford arm [11].

Trial	1	2	3	4	Mean
θ_1	0	90	90	90	
θ_2	0	0	90	90	
d_3	0.185	0.185	0.185	0.885	
dx	1	1	0	1	
dy	1	1	0	1	
dz	1	1	1	1	
θ_c	85	69	10	10	**43.5**
F for gimbal equipped arm	1.53	1.35	1	1.53	
F for simple arm	1.51	1.34	1	1.21	

the robot with traditional revolute joints and shows that in these points, the amount of force exerted by the gimbal equipped robot is greater than the same quantity for the robot that uses traditional revolute joints. Thus, it can be reasoned that gimbal drive has considerably improved the manipulability of the robot. Here, the value of the truncation angle is bound to ensure design feasibility between 10 and 85 degrees.

Now, by having the optimized results of the GA, the training of the proposed ANN can be performed. Basically, since the GA has been utilized to optimize three different case studies which have different Jacobian matrixes, three different ANNs have been trained for each of the robotic arms. It is important to know that since the number of the hidden layer's neurons should be chosen arbitrary, in this research the effect of increasing the hidden layer neurons has been investigated for 15, 20, and 25 neurons, and also the training mean square error limit has been defined as 0.9. Figure 8 shows the ANN training results for RRR arm with assuming 15, 20, and 25 neurons in the hidden layer.

From Figure 8, it can be observed that by increasing the number of neurons from 15 to 20, training of the ANN has been finalized in fewer steps, but the training mean square error has been increased and it makes the simulation result worse. On the other hand, by increasing the number of neurons from 20 to 25, training of the ANN has been finalized in fewer steps, and the training mean square error has been decreased even it is less than the case with 15 neurons and it makes the simulation result better.

Training results for RPR arm with assuming 15, 20, and 25 neurons in the hidden layer have been illustrated in Figure 9.

In Figure 9, although the results show that the trainings were successful, they do not follow the previous path as described for RRR. The difference is in the accuracy of the simulations results which indicate that by increasing the neurons in the hidden layer, training time and iterations will be reduced, but the mean square error of the ANN with 15 neurons is the lowest one, so the related ANN has a superior performance in comparison with the other two.

Training results for Stanford arm with assuming 15, 20, and 25 neurons in the hidden layer have been illustrated in Figure 10.

The results shown in Figure 10 are completely different from the results of RRR and RPR; also, the designed ANN shows different behavior. Like the previous two cases, it can be observed that by increasing the number of neurons in the hidden layer, the training iterations have been increased. Here, different from the results for RPR and RRR, the training mean square error has been decreased by increasing the number of neurons in the hidden layer. It means that the ANN with 25 neurons in the hidden layer has the best performance than the other two.

The summary of all these observations has been shown in Tables 5–7. The results indicate that the behavior of the proposed ANN cannot be predicted by the number of hidden layer's neurons, but one thing is obvious, and it is the fact of the proposed ANN working properly.

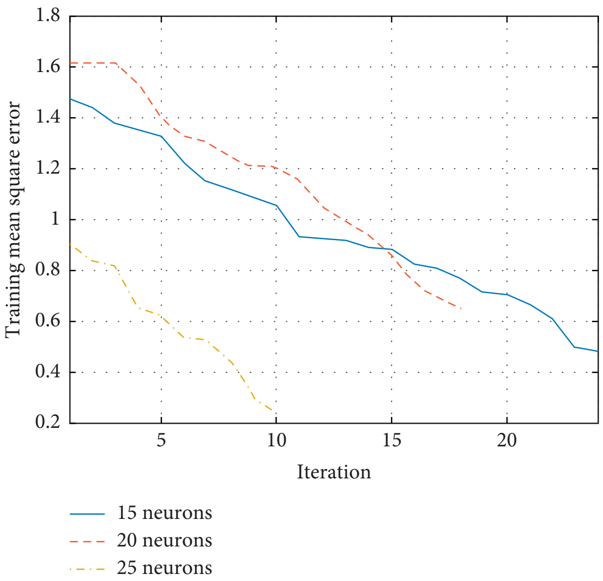

Figure 8: ANN training results for RRR arm with assuming 15, 20, and 25 neurons in the hidden layer.

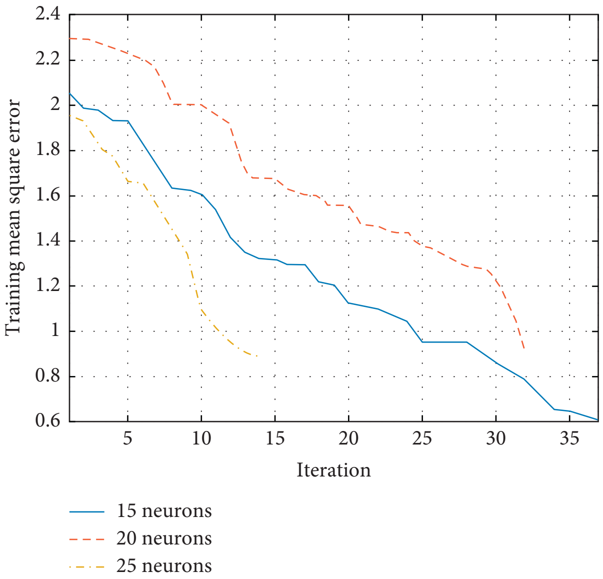

Figure 9: ANN training results for RPR arm with assuming 15, 20, and 25 neurons in the hidden layer.

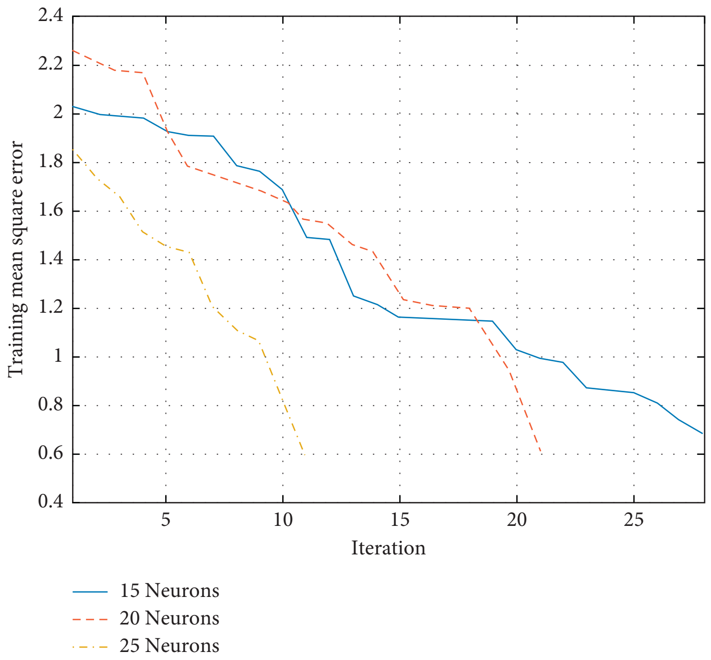

Figure 10: ANN training results for Stanford arm with assuming 15, 20, and 25 neurons in the hidden layer.

Table 5: Neural network training results for RRR arm.

Number of neurons	25	20	15
Number of iterations	10	18	24
Number of training samples	700	700	700
Number of validation samples	150	150	150
Number of testing samples	150	150	150
Training mean square error	0.24	0.65	0.48
Validation mean square error	0.069	0.24	0.24
Test mean square error	0.26	0.31	0.22
Total mean square error	*0.19*	*0.47*	*0.35*
Training regression	0.8173	0.702	0.4985
Validation regression	0.4143	0.6003	0.1689
Test regression	0.4905	0.4245	0.473
Total regression	*0.6648*	*0.6129*	*0.409*

Table 6: Neural network training results for RPR arm.

Number of neurons	25	20	15
Number of iterations	14	32	37
Number of training samples	700	700	700
Number of validation samples	150	150	150
Number of testing samples	150	150	150
Training mean square error	0.89	0.9	0.61
Validation mean square error	0.95	0.74	0.97
Test mean square error	0.59	0.81	0.51
Total mean square error	*0.82*	*0.84*	*0.62*
Training regression	0.9449	0.9503	0.7899
Validation regression	0.8929	0.9076	0.877
Test regression	0.7317	0.9149	0.7060
Total regression	*0.8832*	*0.9317*	*0.7762*

Generally, following points regarding the training procedure of the proposed ANN for all three robot arms with three different number of neurons in the hidden layer for each case study can be observed from Tables 5–7.

Table 7: Neural network training results for Stanford arm.

Number of neurons	25	20	15
Number of iterations	11	21	28
Number of training samples	700	700	700
Number of validation samples	150	150	150
Number of testing samples	150	150	150
Training mean square error	0.59	0.61	0.68
Validation mean square error	0.63	0.18	0.23
Test mean square error	0.39	0.55	0.25
Total mean square error	*0.54*	*0.44*	*0.46*
Training regression	0.7555	0.8012	0.8318
Validation regression	0.7178	0.1051	0.4116
Test regression	0.6074	0.5734	0.4426
Total regression	*0.7141*	*0.3640*	*0.6491*

1000 random points (inputs and outputs) have been generated as a feed to ANN. They were generated according to normal distribution in the permissible zone of the joint space of each one of the robots. Moreover, validation and test data samples are each set to 15 percent of the original samples (see Tables 5–7). So, 300 random samples are used for validation and test, and 700 samples are left for training. Although it is correct, by changing this arrangement or using larger data samples and retraining network over and over, different results are faced, but one should note that it would give some marginal improvements. To achieve reliable results, initial conditions are fixed for all three cases (RRR, Stanford, and RPR).

A set of training samples are presented to the network during training and the network is adjusted according to its error. In addition, a set of validation data are used to measure network generalization and to halt training when generalization stops improving. Furthermore, a set of testing samples have no effect on training and provide an independent measure of network performance during and after training.

Mean square error is the average squared difference between outputs and targets which lower values are better, and zero means there is no error (see Tables 5–7).

Regression R values measure the correlation between outputs and targets. An R value of 1 means a close relationship and 0 means a random relationship. As the data fed into the neural network might be of the scattered type, the discretion and correlation of data values are good measures of those data. This way the regression values help decide on the closeness of the data used for training and the way the ANN is predicting the unknown results.

It can be interpreted from Tables 5–7 that by increasing the number of neurons from 15 to 25, in this special case study, the number of iterations decreases which means that the mean square error decreases to zero. It is important to know this result cannot be used as a pattern for all cases, and there is no guarantee to get better results by increasing the number of neurons and there is a possibility to get worse result by increasing the number of neurons.

The results demonstrate that the increasing number of neurons lead to a decreasing number of iterations which means that the mean square error has been approaching to zero, and the regression data have become closer to one in the case study of this research. So, one may conclude that by increasing the number of neurons, the proposed neural network performance for this research has been increased.

4. Conclusion and Future Work

In this paper, synthesis of three robotic arms such as RRR, RPR, and Stanford with gimbal drive which can be used to achieving the maximum force have been analyzed. Genetic algorithm has been employed to find optimal parameters of the gimbal joints (truncation angles), and then it has been utilized as a model reference for the proposed neural networks. Thousands of randomly generated points have been used as the inputs to the gimbal dynamic model in the GA optimization process. These points have been fed to the neural network to find the relation between inputs and outputs to investigate how well the proposed ANN algorithm can act instead of the GA optimizer of the parameters of the mechanism equipped with gimbal drive.

The results confirm that the proposed ANN optimizer of gimbal drive is a novel and effective technique which can be utilized as an effective optimizer for improving the performance of robots based on their applications. Since robots make a qualitative contribution to productivity, improvement of their performance contributes to sustainability by reducing energy consumption and waste in manufacturing processes.

One of the valuable topics of future works can be defined as measuring the power consumption and material waste reduction rates by utilizing the proposed ANN in real-life industrial implementation which can give a better view to understand how the proposed ANN contributes to sustainability. As another future work, other well-known optimization techniques such as hybrid optimization techniques, bee colony, and particle swarm can be utilized as a reference model to train the proposed neural networks. Also, different neural network models such as ring probabilistic logic neural networks (RPLNN) and RAM-based neural networks can be utilized instead of the proposed ANN.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Ogbemhe, K. Mpofu, and N. S. Tlale, "Achieving sustainability in manufacturing using robotic methodologies," *Procedia Manufacturing*, vol. 8, pp. 440–446, 2017.

[2] L. Wang, C. Wang, W. Du et al., "Parameter optimization of a four-legged robot to improve motion trajectory accuracy using signal-to-noise ratio theory," *Robotics and Computer-Integrated Manufacturing*, vol. 51, pp. 85–96, 2018.

[3] M. Koopialipoor, A. Fallah, D. J. Armaghani, A. Azizi, and E. T. Mohamad, "Three hybrid intelligent models in estimating flyrock distance resulting from blasting," *Engineering with Computers*, vol. 35, no. 1, pp. 243–256, 2018.

[4] A. Azizi, *Applications of Artificial Intelligence Techniques in Industry 4.0*, Springer, Berlin, Germany, 2019.

[5] A. Azizi, "Hybrid artificial intelligence optimization technique," in *Applications of Artificial Intelligence Techniques in Industry 4.0*, Springer, Berlin, Germany, 2019.

[6] A. Azizi, A. Vatankhah Barenji, and M. Hashmipour, "Optimizing radio frequency identification network planning through ring probabilistic logic neurons," *Advances in Mechanical Engineering*, vol. 8, no. 8, 2016.

[7] E. G. Solberg, *Deep Neural Networks for Object Detection in Agricultural Robotics*, Norwegian University of Life Sciences, Ås, Norway, 2017.

[8] L. Jin, S. Li, H. M. La, and X. Luo, "Manipulability optimization of redundant manipulators using dynamic neural networks," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 4710–4720, 2017.

[9] V. T. Yen, W. Y. Nan, P. Van Cuong, N. X. Quynh, and V. H. Thich, "Robust adaptive sliding mode control for industrial robot manipulator using fuzzy wavelet neural networks," *International Journal of Control, Automation and Systems*, vol. 15, no. 6, pp. 2930–2941, 2017.

[10] V. T. Yen, W. Y. Nan, and P. Van Cuong, "Recurrent fuzzy wavelet neural networks based on robust adaptive sliding mode control for industrial robot manipulators," *Neural Computing and Applications*, vol. 31, no. 11, pp. 6945–6958, 2018.

[11] F. Mohammadi, I. Hemmatian, and K. G. Osgouie, "Manipulability analysis for Gimbal driven robotic arms," in *Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Guilin, China, December 2009.

[12] A. Ashkzari and A. Azizi, "Introducing genetic algorithm as an intelligent optimization technique," *Applied Mechanics and Materials*, vol. 568–570, pp. 793–797, 2014.

[13] R. L. Kadri and F. F. Boctor, "An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: the single mode case," *European Journal of Operational Research*, vol. 265, no. 2, pp. 454–462, 2018.

[14] B. Yan, C. Yan, F. Long, and X.-C. Tan, "Multi-objective optimization of electronic product goods location assignment in stereoscopic warehouse based on adaptive genetic algorithm," *Journal of Intelligent Manufacturing*, vol. 29, no. 6, pp. 1273–1285, 2018.

[15] M. Mohammadi, M. Lakestani, and M. H. Mohamed, "Intelligent parameter optimization of savonius rotor using artificial neural network and genetic algorithm," *Energy*, vol. 143, pp. 56–68, 2018.

[16] F. Mohammadi, I. Hemmatian, and K. G. Osgouie, "Design optimization of gimbal robotic joints based on task space manipulability," in *Proceedings of the ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis*, vol. 3, Istanbul, Turkey, July 2010.

[17] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[18] L. Xiao, Z. Zhang, Z. Zhang, W. Li, and S. Li, "Design, verification and robotic application of a novel recurrent neural network for computing dynamic Sylvester equation," *Neural Networks*, vol. 105, pp. 185–196, 2018.

[19] A. Azizi, F. Entessari, K. G. Osgouie, and A. R. Rashnoodi, "Introducing neural networks as a computational intelligent technique," *Applied Mechanics and Materials*, vol. 464, pp. 369–374, 2014.

[20] W. Chan, J. Navdeep, L. Quoc, V. Oriol, and S. Noam, "Speech recognition with attention-based recurrent neural networks," US Patent 9,990,918, 2018.

[21] J. Liu, M. Gong, Q. Miao, X. Wang, and H. Li, "Structure learning for deep neural networks based on multiobjective optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2450–2463, 2018.

[22] A. Azizi, "Introducing a novel hybrid artificial intelligence algorithm to optimize network of industrial applications in modern manufacturing," *Complexity*, vol. 2017, Article ID 8728209, 18 pages, 2017.