CAMES, **28**(2): 119–138, 2021, doi: 10.24423/cames.322

Designing a Cross-Trainer Using an Artificial Neural Network

Siddhartha PATRA

Department of Mechanical Engineering
Jadavpur University
Kolkata, India
e-mail: siddharthapatra16@gmail.com

Cross-trainers are machines that use link mechanisms to mimic walking or running as part of workout sessions or rehabilitation systems. The simplest cross-trainer incorporates a crank-rocker or a crank-slider mechanism and provides a nearly elliptical path for foot motion. However, the natural human foot trajectories are far from being elliptical. Therefore, existing designs require modifications. Artificial neural networks are used for this purpose. Instead of trying to match the foot trajectory directly, here we tried to match different geometric properties of the area enclosed by the foot trajectory. Neural networks are trained to predict these geometric properties as outputs with the dimensions of the linkage as inputs. With the help of the same trained network, the "best-fit dimensions" were predicted for the desired trajectories.

Keywords: cross-trainer, artificial neural network, foot trajectory, optimization.

1. Introduction

1.1. Gait cycle of human foot motion

Rehabilitation systems are used for patients who lost locomotor abilities due to injuries or diseases. To enhance such systems, the accurate study of the gait cycle of human locomotion is important. Such studies were extensively conducted over the past two decades. Many of these studies focused on planar motion, while some focused on motion over the staircase. The shapes of the foot trajectories turn out to be quite different in these two cases.

Schmidt *et al.* [1] built a robotic device to facilitate the movement of the patients' feet in programmable trajectories. Grasso *et al.* [2] studied the improvement in the gait cycles in rehabilitation training of the patients suffering from spinal cord injuries. Chang and Troje [3] studied the local inversion effects

in the motion perception of the human foot trajectory by dividing it into multiple segments. Shirota *et al.* [4] studied recovery strategies of the transfemoral amputees through the gait cycles of their sound and prosthesis sides. Mendoza-Crespo *et al.* [5] through studies on human subjects, developed a human-like gait pattern generator. Figure 1a represents the ankle trajectory as reported by Mendoza-Crespo *et al.* [5] and Fig. 1b represents the same in the centroidal principal coordinate system of the enclosed area.

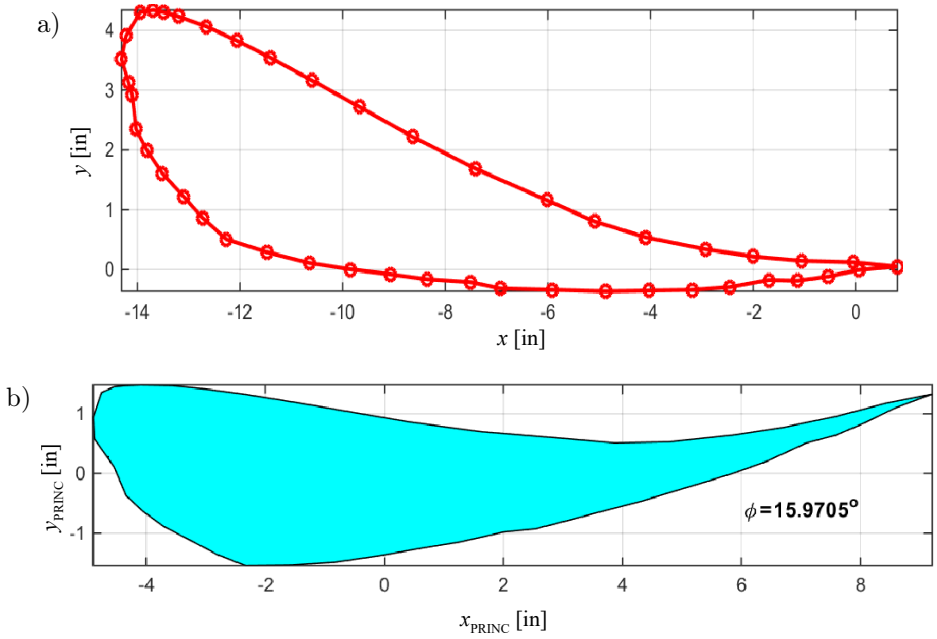

FIG. 1. Human ankle trajectory on a plane walk (a) as reported by Mendoza-Crespo *et al.* [5]; b) the same represented in the centroidal principal coordinate system.

Figure 2a shows the ankle trajectory reported by Park *et al.* [6], while Fig. 2b shows the reoriented configuration in the centroidal principal coordinate system of the enclosed area. While most commercial robotic locomotion rehabilitation systems are designed for a plane walk, Park *et al.* [6] studied the stair-gate patterns and applied them to a robotic rehabilitation system to facilitate vertical motion. Unlike plane motion, here the vertical span of the trajectory is noticeably high and almost matches the horizontal span. The angle ϕ in Figs 1b and 2b represents the amount of rotation required to orient the shapes in their respective centroidal principal coordinate systems.

Cross-trainers are machines that use link mechanisms to mimic walking or running as part of workout sessions or as rehabilitation systems. The simplest cross-trainers are four-bar crank-rocker mechanisms with one degree of freedom

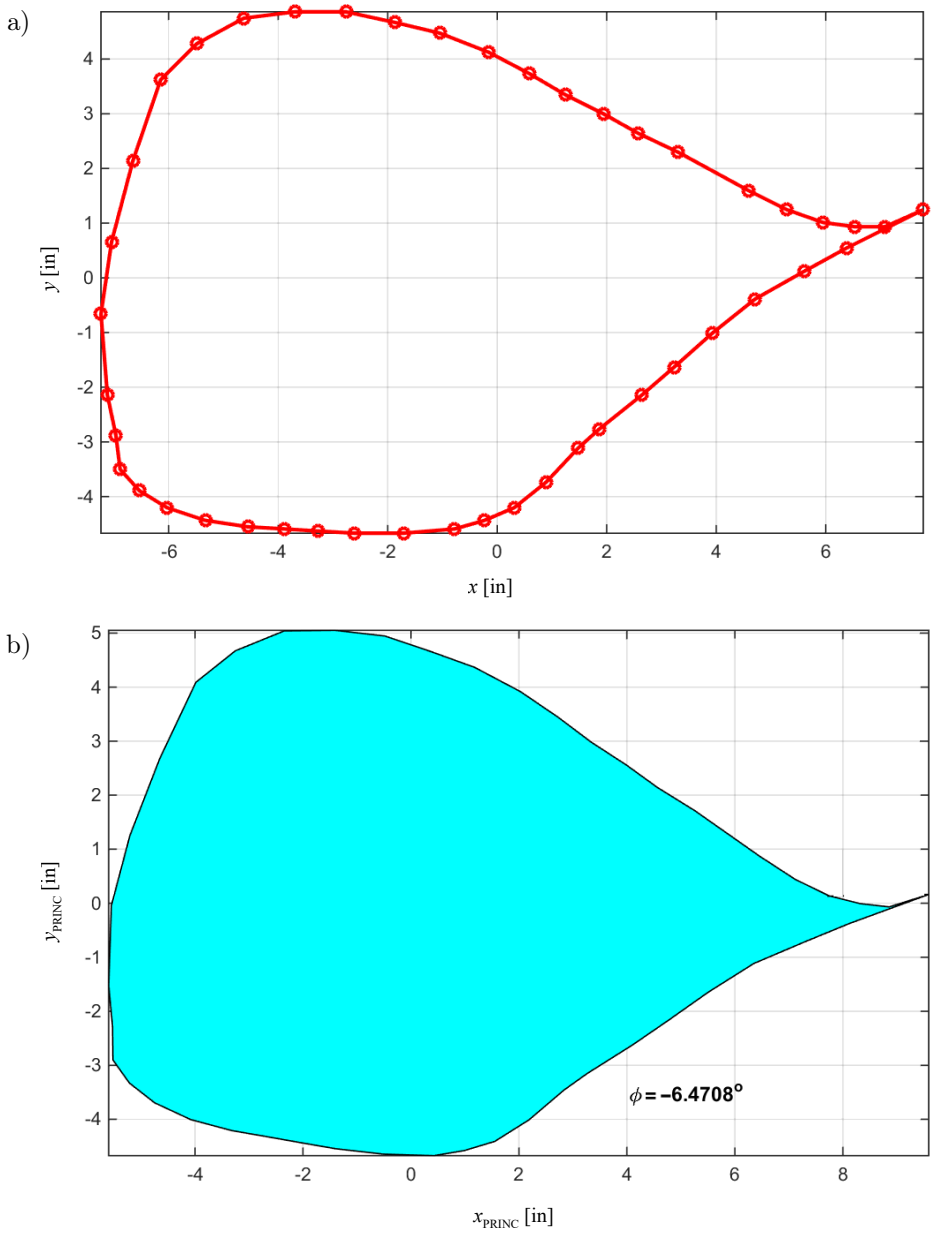

FIG. 2. Human ankle trajectory on stair walk (a) as reported by Park *et al.* [6], b) the same represented in the centroidal principal coordinate system.

(DOF). One such mechanism (Fig. 3a) is considered here. Table 1 describes the different components of the mechanism along with the dimensions.

The sample shown in Fig. 3a is considered as the existing design. The footrest for the existing design lies on the coupler, therefore making $r_2 = 0$. Figure 3b

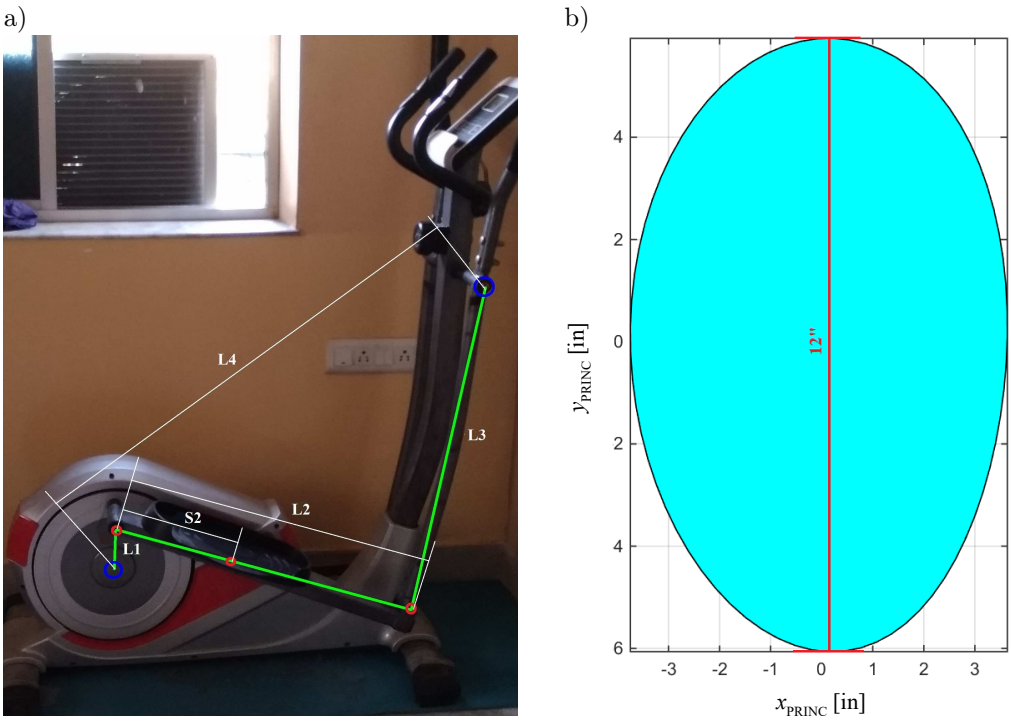

FIG. 3. Existing cross-trainer design: a) identification of different dimensions; b) foot trajectory represented in a centroidal principal coordinate system.

TABLE 1. Different components of the mechanism and their dimensions.

Components	Crank	Coupler	Ratios defining foot position		Rocker	Fixed length
			Parallel to L_2	Normal to L_2		
Symbols [units]	L_1 [in]	L_2 [in]	$r = \frac{S_2}{L_2}$ (NA)	r_2 (NA)	L_3 [in]	L_4 [in]
Dimensions	6	25	0.3846	0	30	39.60

shows the ankle trajectory represented in its centroidal principal coordinate system. All the dimensions in this analysis are taken in inches.

These two trajectories represent the two extremes of the wide spectrum of foot-trajectories. In the current study, these two trajectories are considered as targets.

The objective of the present work is to develop a two-way correlation between the dimensions of the components of the cross-trainer and the resulting foot trajectory. Artificial neural networks (ANN) will be trained for this purpose. Therefore, once trained, the network should be able to predict the foot trajectory given the dimensions of the components as inputs. On the other hand, the same

network should be able to predict the "best fit" dimensions if presented with the desired trajectory.

2. Design of cross-trainers

Numerous designs of cross-trainers are available in the commercial market. However, academic studies are also conducted to improve their designs further to make them more suitable for gait rehabilitation. Lee [7] used the creative mechanism design method to present five alternative designs of elliptical trainers with two DOFs. Nelson *et al.* [8] used a crank-rocker mechanism with a curved-contoured secondary coupler participating in a small-displacement slider-based sublinkage. The sublinkage helped in fine- tuning the motion path. Nelson and Burnfield [9] proposed a new design by replacing the crank with a modified Cardan gear for a flatter foot trajectory. Chen *et al.* [10] proposed a quick return mechanism to mimic the timing of foot trajectory for jogging. Burnfield *et al.* [11] designed a modified elliptical trainer for children with physical disabilities through a series of seat adjustments, pedal height, step length, etc. Chen *et al.* [12] proposed an innovative design for an elliptical trainer to effectively mimic the foot-trajectory of jogging. The design was achieved by incorporating a timing adjustment wheel with a protruded slider, which slides through the slotted flywheel which doubles as the crank of the main linkage. Hummer *et al.* [13] designed an elliptical trainer to achieve a converging footpath and a reduced inter pedal distance to reduce overuse injuries.

Most of these advanced designs are more complicated than a crank-rocker or a crank-slider mechanism and sometimes have more than one DOF. However, in the present work, only one DOF crank-rocker mechanism is considered.

3. Application of ANN in structural engineering

Application of ANN has entered into different structural and mechanical engineering fields, from modeling of constitutive relations to designing the manufacturing process. Modeling of material behavior has been a goal for many authors. For example, Shabani and Mazahery [14] used ANN for the finite element simulation of an aluminum-silicon alloy. Gupta *et al.* [15] used ANN to develop a constitutive model for 304 stainless steel. Desu *et al.* [16] used ANN to predict mechanical properties of stainless steel grades 304L and 316L. Chang *et al.* [17] used ANN for the application of machine tools for manufacturing purposes. Ciro De Filippis *et al.* [18] also used ANN for optimization of manufacturing purposes.

Artificial neural network also finds its application in structural design. Hong *et al.* [19] proposed an ANN-based method for obtaining preliminary designs

for cable-stayed bridges to assist inexperienced designers. Gomes and Beck [20] used ANN for structural designs with consequences of failure into account. Mote and Kumar [21] trained an ANN to design steel structure subject to design constraints. Liu [22] reviewed the application of ANN in mechanical design, optimization, structural analysis, and fault detection.

In order to use ANN in a given application, the architecture of the neural network must be identified. In addition, the methods for initializing and updating the weights must also be chosen carefully. In the next section. these aspects of ANN are explored.

3.1. Choice of number of hidden neurons

In order to implement a neural network, it is important to decide on the number of hidden neurons N_H. Panchal and Panchal [23] reviewed the methods for selecting the number of hidden neurons. They listed five different ways to obtain the value of N_H. Apart from the usual trial and error method, there is the rule of thumb method. In this method, the N_H is a function of the number of neurons in the input layer N_I, and the number of neurons in the output layer N_O. Three possible expressions (Eq. (1)) of N_H are found using this method

$$\begin{aligned} &N_H \in [\min(N_I, N_O) \max(N_I, N_O)] \\ &\text{or} \quad N_H = \frac{2}{3} N_I + N_O \quad \text{or} \quad N_H \leq 2N_I. \end{aligned} \tag{1}$$

None of the above expressions takes the number of data sets N_P into consideration. Apart from these two, there are the simple method, the two-phase method, which is a modification over the trial and error method, and the sequential orthogonal approach.

Kavzoglu and Reis [24] used an N_H value based on the number of N_P, N_I, N_O. The relationship is described in Eq. (2)

$$N_H = \frac{N_P}{\xi (N_I + N_O)} \quad \text{and} \quad \xi \in [2, 10]. \tag{2}$$

The parameter ξ depends on the problem's difficulty and typically varies between 5 and 10 but can be as low as 2.

3.2. Choice of activation function

In a simple feedforward neural network, a neuron in a given layer receives inputs from the neurons of the previous layer. The neuron is activated if the weighted sum of these inputs surpasses a given threshold. There are a wide

variety of activation functions described in the literature. Probably the most common activation function is the logistic or the sigmoid function. This is an S-shaped function given by Eq. (3)

$$\sigma = \frac{1}{1+\exp(-u)} \quad \text{and} \quad u = \sum_i w_i x_i + b, \tag{3}$$

where x_i's are the inputs from the previous layer, w_i's are the corresponding weights and b is the bias.

A variation of the sigmoid function is the hyperbolic tangent function, which offers a zero mean value and a faster learning rate [25]. Neural networks with non-smooth activation functions such as rectified linear unit (ReLU) (Eq. (4)) typically have softmax layers as their final layer [26]. A ReLU neuron with negative input encounters a zero value and a zero slope and becomes dead, i.e., does not update any further. A leaky ReLU (Eq. (5)) fixes the dying ReLU issue. The graphical representations of the activation functions are given in Fig. 4

$$f = \max(u, 0), \tag{4}$$

$$f = \max(u, 0.01u). \tag{5}$$

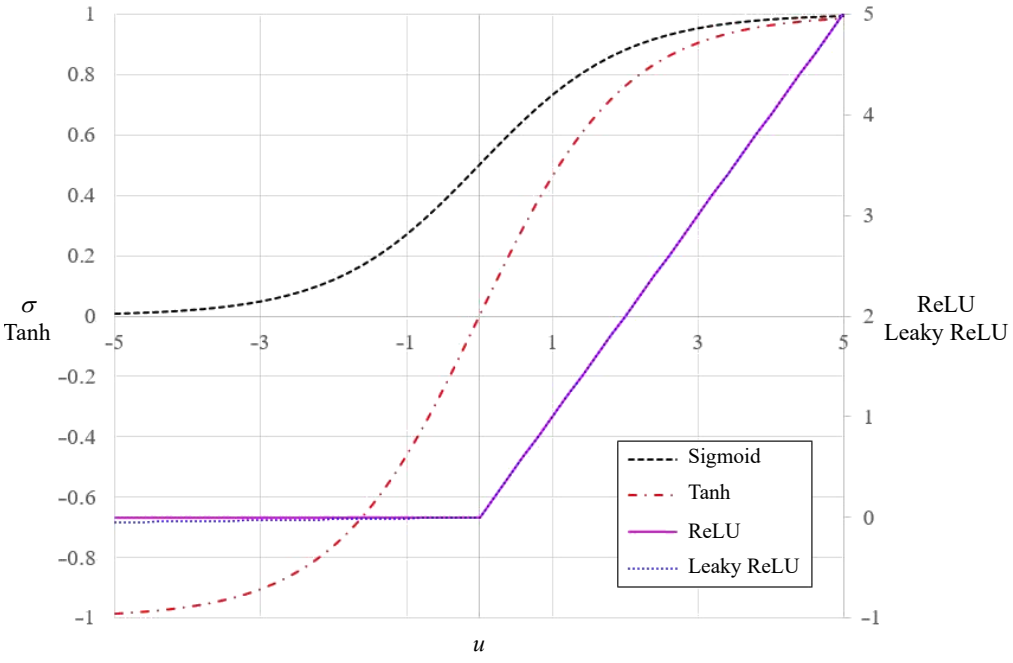

Fig. 4. Activation functions.

3.3. Initialization of the weights

Initialization of the weights and biases affect the convergence rate of the backpropagation (BP) algorithm. Different methods of randomly initializing the weights can be found in the literature. Li *et al.* [27] proposed a training strategy named Delta Pre-Training (DPT). In this method, each hidden layer is initially treated as a perceptron. Thus the weights and biases could be initialized as zeros. Next, the hidden variables are assigned distinct binary pattern, i.e., each hidden variable is either 1 or 0 for each training sample ($i \in [1, N_P]$). Then the weights and biases of the hidden layer are trained to yield the assigned distinct hidden pattern as closely as possible. Once the training converges, the method continues to the next hidden layer. The pre-trained weights become the initial weights in the BP training.

Glorot and Bengio [28] proposed a normalized initialization method, wherein the weights for $(j+1)$-th layer are drawn from a set of uniformly distributed random numbers within $\pm\frac{\sqrt{6}}{\sqrt{N_j+N_{j+1}}}$, where N_j is the number of neurons in the j-th layer.

Martens and Sutskever [29] proposed the so-called sparse initialization technique, where each neuron in a given layer is connected to 15 neurons of the previous layer whose weights are drawn from unit Gaussian. In our study, we used the sparse initialization method with modifications suggested by Sutskever *et al.* [30].

3.4. Convergence rate of backpropagation method

The convergence rate of the BP method can be accelerated considerably towards local minima using the momentum method [31]. Nesterov's accelerated gradient [32, 33] can be thought of as an improvised momentum method and can achieve quadratic convergence towards global minima. In this study, Netsorov's method is used with the modifications suggested by Sutskever *et al.* [30].

3.5. Controlling overfitting

Overfitting is a common issue with ANN training wherein the weights and biases get excessively customized for the training data set. As a result, the predictions closely match the training set. However, the predictions are incorrect for input parameters outside the training set. There are multiple ways to control the overfitting of the weights.

One of the most common ways is to do an early stopping proposed by Prechelt [34] wherein a small fraction, typically 20%, of the total data set is used as a validation set and the remaining data set is used as a training set. The weights are trained using the training set, say through BP, while the error from

the validating set is continuously evaluated. The goal here is to stop the training iterations before the training process "overfits" the weights to the training set. Initially, as the fitting process progresses, the error from both the training and the validating sets decreases. In early stopping, the training process is terminated once the error from the validating set reaches a minimum and increases beyond that point. The weights corresponding to the minimum error from the validating set are used for the ANN.

Another way of controlling the overfitting issue is by incorporating L_1 or L_2 regularization as explained by Ng [35]. In this process, the loss function is augmented with a tiny fraction (λ) of the L_1 or L_2 norm as a penalty function. This prevents the weights from blowing up. In Eq. (6) the modified loss function LF^* is the sum of the loss function LF and the $\lambda \times L_2$ norm of the weights

$$LF^* = LF + \lambda \sum w_i^2. \tag{6}$$

The dropout technique was first introduced by Hinton *et al.* [26]. In this method, during the training procedure, a certain percentage, typically 50%, of the neurons from the hidden layers and at times also from the input layer is randomly ignored. The network is treated as if those neurons do not exist. The weights of the other neurons are scaled accordingly to maintain consistency. This prevents the output of the network from being too much dependent on the respective neuron and, in turn, helps preventing the overfitting issue.

The dropConnect method used by Lian *et al.* [36] and Wan *et al.* [37] is a variation of the dropout method wherein instead of individual neurons, the individual connections between any two neurons are randomly ignored.

4. Method

Here the goal is to achieve the foot-trajectories as close to the targets as possible. The lengths of the links of the four-bar mechanism (L_2, L_3, L_4) except for the crank length (L_1) and the foot position ($r = \frac{S_2}{L_2}$, r_2) are the design variables (Fig. 3a).

The analysis is done independently of the crank length to see if a span in a principal direction higher than twice the crank length can be achieved. For the existing design this span is exactly equal to the twice crank length (Fig. 3b). The shape of the foot trajectory is aligned in its centroidal principal coordinate system. This allows the comparison of the properties of the shape, such as moments of inertia across the varied set of shapes observed through the variation of the link lengths. The target paths for a plane walk and stair walk span are between 15–20% more than twice the crank length (Figs 1b and 2b).

In order to train the ANNs, 10 000 input data sets are generated at random. For each set, five input properties and five output properties are identi-

fied. The output variables are obtained in the centroidal principal coordinate system. The shapes are approximated as polygons and the geometric properties such as the area (A), the perimeter (p), coordinates of the centroid (x_c, y_c), moments of inertia (I_{xx}, I_{yy}, I_{xy}) are calculated using Green's formulae as described by Brlek *et al.* [38] and shown in Eqs (7)–(13).

$$A = \sum_{i=1}^{n} \left(x_i y_{i+1} - y_i x_{i+1} \right), \tag{7}$$

$$p = \sum_{i=1}^{n} \sqrt{\left(x_{i+1} - x_i \right)^2 + \left(y_{i+1} - y_i \right)^2}, \tag{8}$$

$$x_c = \frac{1}{6A} \sum_{i=1}^{n} \left(x_i y_{i+1} - y_i x_{i+1} \right) \left(x_i + x_{i+1} \right), \tag{9}$$

$$y_c = \frac{1}{6A} \sum_{i=1}^{n} \left(x_i y_{i+1} - y_i x_{i+1} \right) \left(y_i + y_{i+1} \right), \tag{10}$$

$$I_{xx} = \frac{1}{12} \sum_{i=1}^{n} \left(x_i y_{i+1} - y_i x_{i+1} \right) \left(y_i^2 + y_i y_{i+1} + y_{i+1}^2 \right), \tag{11}$$

$$I_{yy} = \frac{1}{12} \sum_{i=1}^{n} \left(x_i y_{i+1} - y_i x_{i+1} \right) \left(x_i^2 + x_i x_{i+1} + x_{i+1}^2 \right), \tag{12}$$

$$I_{xy} = \frac{1}{24} \sum_{i=1}^{n} \left(x_i y_{i+1} - y_i x_{i+1} \right) \left(x_i \left(2y_i + y_{i+1} \right) + x_{i+1} \left(2y_{i+1} + y_i \right) \right), \tag{13}$$

where (x_i, y_i) are the coordinates of the i-th vertex of a polygon with n sides and $i \in [1, n]$. Also, $n + 1 \equiv 1$. The eccentricity e is given by Eq. (14)

$$e = \frac{\sqrt{\min\left(|x_1|, |x_2| \right)^2 + \min\left(|y_1|, |y_2| \right)^2}}{\sqrt{\max\left(|x_1|, |x_2| \right)^2 + \max\left(|y_1|, |y_2| \right)^2}}, \tag{14}$$

where (x_1, y_1), (x_2, y_1), (x_2, y_2), (x_1, y_2) are the four corners of the bounding box of the shape in its centroidal principal coordinate system. The principal moments of inertia (I_1, I_2) along with the area, perimeter and eccentricity represent the output variables. The input and output variables are shown in Table 2.

Table 2. Input and output parameters.

Input parameters	Output parameters
L_2, r, r_2, L_3, L_4	A, I_1, I_2, p, e

The maximum variation of the parameters for the design analysis is taken to be ±25% of the dimensions in the existing design. The parameter r_2, which represents the height of the foot position normal to the coupler as a fraction of L_2, is taken to be within ±0.25.

Only 3000 of these data sets are used for training ($N_P = 3000$) the network and 10 000 data sets were used for validating. A smaller training sample, say 1000 or below, showed an overfitting tendency. For the BP, Nesterov's approach [33], as discussed in the previous section, was followed with the changes suggested by Sutskever *et al.* [30]. For the initiation of the weights, the sparse initiation technique is used with the modifications suggested by Sutskever *et al.* [30]. The hyperbolic tangent function was used as the activation function. The input and output variables of the data sets were normalized between −0.9 to 0.9 using the expression given in Eq. (15)

$$x_j^{IO} = -0.9 + 1.8 \times \frac{X_j^{IO} - X_{j\,\min}^{IO}}{X_{j\,\max}^{IO} - X_{j\,\min}^{IO}}, \tag{15}$$

where x_j^{IO} is the j-th normalized input or output variable and X_j^{IO} is the actual value of the j-th input or output variable. $X_{j\,\max}^{IO}$, $X_{j\,\min}^{IO}$ are respectively the corresponding maximum and the minimum values of the j-th input or output variable. The dimensional quantities can be obtained from the normalized quantities using the following expression (Eq. (16)):

$$X_j^{IO} = X_{j\,\min}^{IO} + \left(X_{j\,\max}^{IO} - X_{j\,\min}^{IO}\right) \times \frac{x_j^{IO} + 0.9}{1.8}. \tag{16}$$

The neural network is a simple feedforward network with five input variables ($N_I = 5$) and five output variables ($N_O = 5$) and a single hidden layer. The choice of the number of hidden neurons is made using Eq. (2). The minimum number of hidden neurons $N_H^{\min} = 30$. Here the goal is to minimize the total RMS error $E_{\mathrm{RMS}}^{\mathrm{TOT}}$. The expression for $E_{\mathrm{RMS}}^{\mathrm{TOT}}$ is given by Eq. (17)

$$E_{\mathrm{RMS}}^{\mathrm{TOT}} = \sqrt{\frac{1}{N_P N_O} \sum_{i=1}^{N_P} \sum_{j=1}^{N_O} {\Delta x_j^O}^2}. \tag{17}$$

5. Results and discussion

The objective of the first part was to train the ANN to accurately predict the output variables. The trained ANN was then used in the second part to address the inverse problem of identifying the best set of dimensions to achieve the target foot trajectories as closely as possible.

As a test run, $N_H = 5$ was taken to see the convergence behavior. The $E_{\mathrm{RMS}}^{\mathrm{TOT}}$ minimizes to 0.0351 as shown in Table 3. With $N_H = 30$, $E_{\mathrm{RMS}}^{\mathrm{TOT}} = 0.0266$ is achieved. It seems that the $E_{\mathrm{RMS}}^{\mathrm{TOT}}$ approaches a ceiling and only slightly improves by using several times higher N_H. Hence no further increment is done to the number of hidden neurons. Also, only one hidden layer is used. The use of a deeper network does not seem to improve the $E_{\mathrm{RMS}}^{\mathrm{TOT}}$ significantly for the same total number of weights and bias parameters. Figure 5 shows the variation of the RMS error with the number of epochs (N_{EPOCH}).

TABLE 3. RMS error over all normalized output variables over all data sets ($E_{\mathrm{RMS}}^{\mathrm{TOT}}$).

	N_H	N_P	$E_{\mathrm{RMS}}^{\mathrm{TOT}}$
Training	5	3000	0.0351
Training	30	3000	0.0266
Validating	30	10 000	0.0238
Optimizing plane walk	30	1	0.8518
Optimizing stare walk	30	1	0.7744

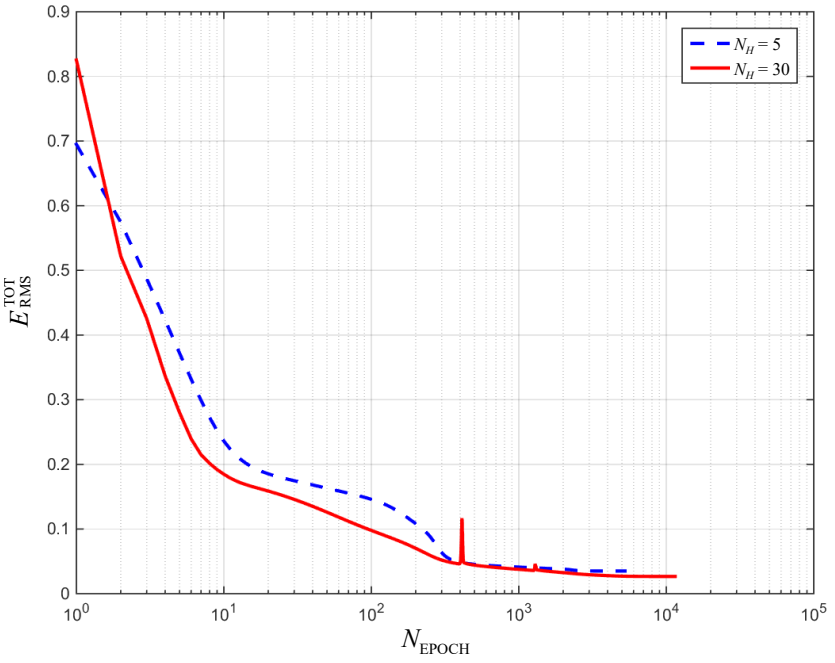

FIG. 5. Convergence with 5 and 30 hidden neurons during training.

Once the training was completed, an approximate functional relationship was established between the input and the output variables. Hence, for any arbitrary set of the input variables within limits, the values of the output variables can be

closely predicted. This is confirmed by the validating set of 10 000 data points, which yielded a $E_{\mathrm{RMS}}^{\mathrm{TOT}} = 0.0238$, as shown in Table 3.

In the second part of the analysis, the goal is to predict the set of input variables in Table 2 given the output variables. In this part of the analysis, the weights and biases remain fixed. Since only the variables in a given layer have to be optimized, a zero initialization, i.e., ($x_0^I = [0\ 0\ 0\ 0\ 0]$) was used. This corresponds to the dimensions of the existing design. Also, to ensure that the normalized input parameters stay within $[-1, 1]$, the gradient of the loss function was modified using Eq. (18)

$$\left(\nabla_{x^I}^*(LF)\right)_j = \left(\nabla_{x^I}(LF)\right)_j + K \times \left(\max\left(0, x_j^I - 1\right) + \min\left(0, x_j^I + 1\right)\right), \quad (18)$$

where K is very high penalty stiffness. In our analysis, $K = 1 \times 10^6$ along with $\gamma = 1 \times 10^{-6}$ seemed to get rapid convergence. Figure 6a shows the convergence behaviors for the plane walk and stair walk problems. Although the analysis was run to a maximum of 50 000 epoch, in both cases convergence occurred below 20 000 epoch and the analysis steps were really fast since only one data set ($N_P = 1$) was used in each case. The fitment error for each case is represented in Table 3.

The $E_{\mathrm{RMS}}^{\mathrm{TOT}}$ is higher than that in the training phase. However, they are significantly lower than the starting error, corresponding to the existing design. This shows that the existing design is far from either type of motion and significant improvement could be achieved through this analysis. Moreover, this is not the limitation of the fitting method but the limitation due to:

1) Use of exclusively four-bar mechanisms which can limit the possible shapes that can be generated through foot motion.
2) The amount of variation allowed over the original dimensions of the link lengths.

Table 4 lists the values of the input variables fitted through the inverse functional relationship for both cases. It can be seen for both cases that each of the parameters r_2 and L_4 reaches one of their limiting values. Table 4 also lists the corresponding geometric properties of the shape produced by the foot trajectory for the best-fit plane walk and the best-fit stair walk. For the best-fit plane walk, the geometric properties are of the same orders of magnitude, and the difference in the values of the maximum and minimum principal moments of inertia I_1, I_2 of the target shape is closely captured. For the best-fit stair walk, the geometric properties are within 20% of the target, except for I_1. In both cases, the eccentricity e issignificantly higher than the target shapes. Apparently, this sort of e is difficult to achieve while closely approximating the other geometric parameters given the above- mentioned design conditions. However, as discussed before, the best-fit configurations seem to significantly improve the existing design.

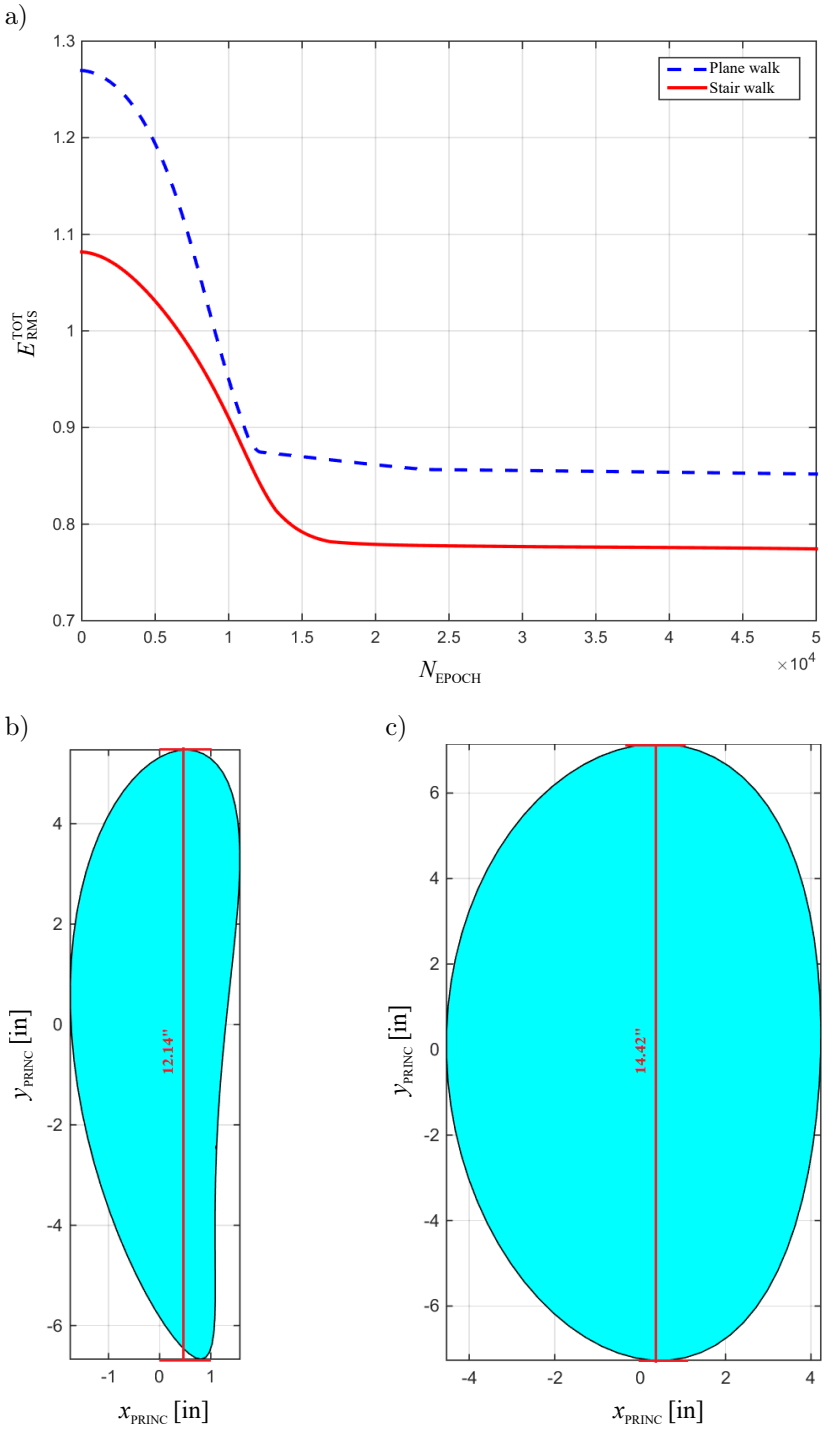

FIG. 6. Optimum solution: a) convergence curve; b) and c) designed motion paths for plane and stair walk represented in centroidal principal coordinate systems, respectively.

TABLE 4. Dimension and properties.

	L_2 [in]	r	r_2	L_3 [in]	L_4 [in]	A [in^2]	I_1 [in^4]	I_2 [in^4]	p [in]	e
Existing design	25.00	0.3846	0	30.00	39.60	69.51	625.14	236.83	30.89	0.9762
Target plane walk	–	–	–	–	–	21.02	197.26	9.75	30.33	0.5475
Best fit plane walk	23.04	0.4914	0.2777	38.31	28.59	28.63	260.18	18.41	26.45	0.8256
Target stair walk	–	–	–	–	–	87.99	862.68	484.36	39.45	0.6728
Best fit stair walk	22.90	0.2929	−0.2776	33.34	28.59	99.91	1308.10	484.91	37.16	0.9697

Figures 6b and 6c respectively show the generated shapes in their respective centroidal coordinate systems. It can be seen that in both cases, the maximum dimension is over twice the crank length, i.e., 12 inches. Also, the striking dissimilarity of the two shapes is evident from Figs 6b and 6c.

Figures 7a and 8a show the schematics of the four-bar linkages respectively for the best-fit plane walk and the best-fit stair walk in the sequences of 60° crank rotations, in the "correct" orientation. The correct orientation is obtained by first matching the target shapes with the best-fit shapes in their centroidal principal coordinate systems and then rotating further to match the original orientation of the target shape.

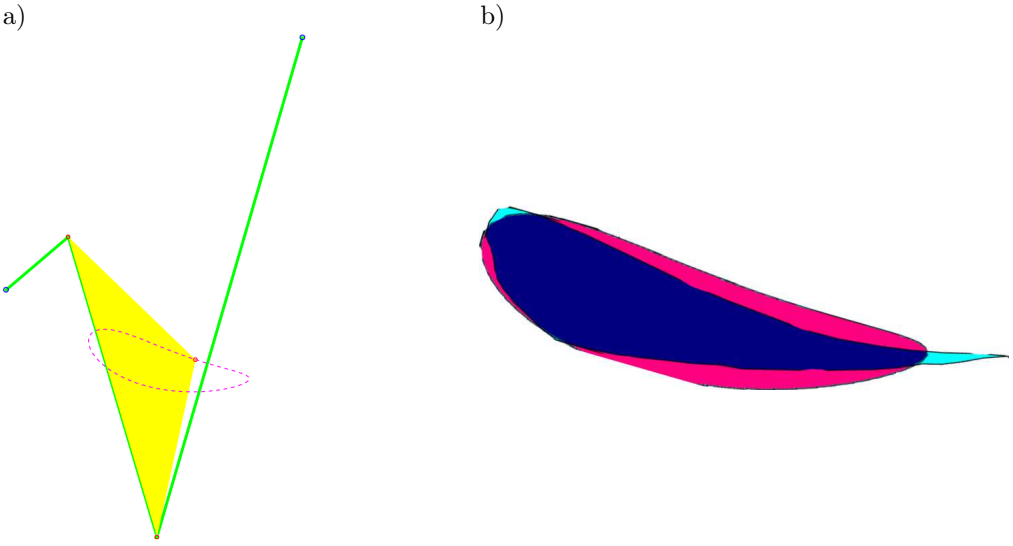

FIG. 7. Reoriented configuration plane walk: a) linkage mechanism; b) synthesized path and the existing path superimposed over the target.

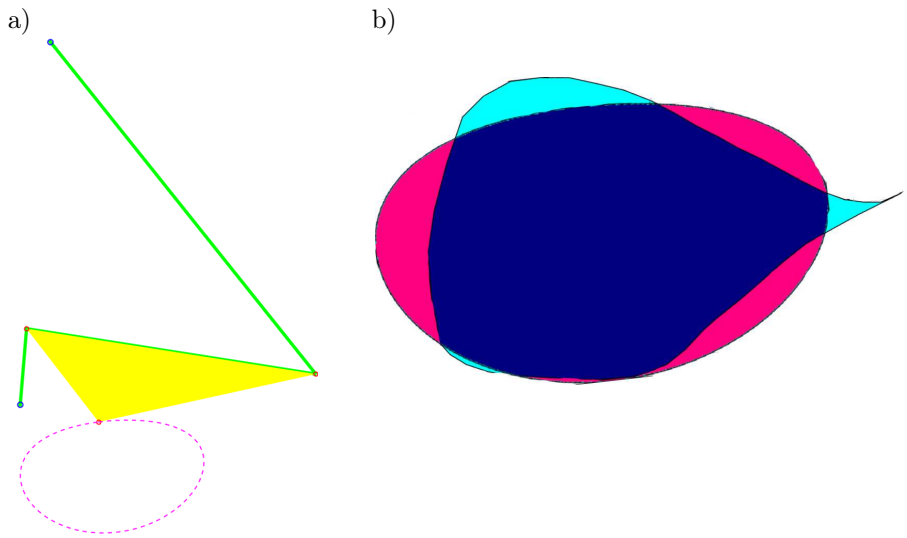

Fig. 8. Reoriented configuration stair walk: a) linkage mechanism; b) synthesised path superimposed over the target.

Figure 7b shows the superimposition of the shapes of the foot trajectories for the existing cross-trainer design and the new design (best fit) for the plane walk and the target for the plane walk. Although the new design lacks the narrow tail-like feature of the target shape, it is much closer to the target shape than the existing shape.

Figure 8b shows the superimposition of the shapes of the foot trajectories for the best-fit stair walk and the target stair walk. Although the shapes are different, they have significant overlap and as in the previous case, the narrow tail-like feature is left out. It seems that a narrow region is responsible for a smaller e.

6. Conclusion

In the first part of the analysis, through training of 3000 data sets, an approximate functional relationship was obtained between the input and output parameters so that for any arbitrary set of the input variables within limits, the values of the output variables could be closely predicted. This was also validated by a verification set of 10 000 data points. In the second part of the analysis, the inverse relationship was explored to identify the input variables for two qualitatively different sets of output variables. With a functional relationship established through the first part of the analysis, multiple such design operations could be performed quickly. The geometric properties of the foot trajectories with the predicted input parameters showed moderate to good agreement with the target geometric properties, with the maximum span of the shapes being above twice

the crank length. The only exception was eccentricity. Also, the shapes of the foot trajectories from the best fits moderately resembled the target shapes in most aspects except the narrow tail-like features. However, in each case, they provided a significant improvement over the existing design. The difference may be attributed to the constraint in the choice of mechanism and the allowed variations of the input variables. However, such differences can be reduced through fine-tuning by incorporating sub-mechanisms, similar to the approach of Nelson *et al.* [8].

Declarations

Funding

No funding was received.

Conflict of interest

The author declares that there is no conflict of interest.

Availability of data and material

The datasets used and/or analyzed during the current study are available from the corresponding author upon a reasonable request.

Authors' contribution

All the study reported in this manuscript is solely conducted by Siddhartha Patra.

Acknowledgement

The author declares that no acknowledgment is required.

References

1. H. Schmidt, D. Sorowka, S. Hesse, R. Bemhardt, Development aspects of a robotised gait trainer for neurological rehabilitation, *2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 2, pp. 1340–1343, 2001, doi: 10.1109/IEMBS.2001.1020444.
2. R. Grasso, Y.P. Ivanenko, M. Zago, M. Molinari, G. Scivoletto, V. Castellano, V. Macellari, F. Lacquaniti, Distributed plasticity of locomotor pattern generators in spinal cord injured patients, *Brain*, **127**(5): 1019–1034, 2004, doi: 10.1093/brain/awh115.

3. D.H.F. Chang, N.F. Troje, Acceleration carries the local inversion effect in biological motion perception, *Journal of Vision*, **9**(1): 1–17, 2009, doi: 10.1167/9.1.19.

4. C. Shirota, A.M. Simon, T.A. Kuiken, Transfemoral amputee recovery strategies following trips to their sound and prosthesis sides throughout swing phase, *Journal of NeuroEngineering and Rehabilitation*, **12**: 79 (12 pages), 2015, doi: 10.1186/s12984-015-0067-8.

5. R. Mendoza-Crespo, D. Torricelli, J.C. Huegel, J.L. Gordillo, J.L. Pons, R. Soto, An adaptable human-like gait pattern generator derived from a lower limb exoskeleton, *Frontiers in Robotics and AI*, **6** (36 pages), 2019, doi: 10.3389/frobt.2019.00036.

6. S.E. Park, Y.J. Ho, M.H. Chun, J. Choi, Y. Moon, Measurement and analysis of gait pattern during stair walk for improvement of robotic locomotion rehabilitation system, *Analysis of Human Behavior for Robot Design and Control*, **2019**: 1495289 (12 pages), 2019, doi: 10.1155/2019/1495289.

7. H.S. Lee, Creative design of an elliptical trainer with two degrees of freedom, *International Journal of Mechanical Engineering Education*, **36**(4): 284–293, 2008, doi: 10.7227/IJMEE.36.4.2.

8. C.A. Nelson, J.M. Burnfield, Y. Shu, T.W. Buster, A.P. Taylor, A. Graham, Modified elliptical machine motor-drive design for assistive gait rehabilitation, *Journal of Medical Devices*, **5**(2): 021001 (7 pages), 2011, doi: 10.1115/1.4003693.

9. C.A. Nelson, J.M. Burnfield, Improved Elliptical Trainer Biomechanics Using a Modified Cardan Gear, [in:] *Proceedings of the ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE*, Chicago, IL, USA, 2012, doi: 10.1115/DETC2012-70439.

10. F.-C. Chen, Y.-F. Tzeng, W.-R. Chen, On the mechanism design of an innovative elliptical exerciser with quick return effect, *International Journal of Engineering and Technology Innovation*, **8**(3): 228–239, 2018.

11. J.M. Burnfield, T.W. Buster, C.M. Pfeifer, S.L. Irons, G.M. Cesar, C.A. Nelson, Adapted motor-assisted elliptical for rehabilitation of children with physical disabilities, *Journal of Medical Devices*, **13**(1): 011006 (9 pages), 2019, doi: 10.1115/1.4041588.

12. F.-C. Chen, Y.-F. Tzeng, M.-H. Hsu, Innovative design of an elliptical trainer with right timing of the foot trajectory, *Advances in Technology Innovation*, **5**(3): 190–201, 2020, doi: 10.46604/aiti.2020.5645.

13. E. Hummer, E. Murphy, D.N. Suprak, L. Brilla, J.G. San Juan, The effects of a standard elliptical vs. a modified elliptical with a converging footpath on lower limb kinematics and muscle activity, *Journal of Sports Sciences*, **38**(20): 2382–2389, 2020, doi: 10.1080/02640414.2020.1786241.

14. M.O. Shabani, A. Mazahery, The ANN application in FEM modeling of mechanical properties of Al-Si alloy, *Applied Mathematical Modelling*, **35**(12): 5707–5713, 2011, doi: 10.1016/j.apm.2011.05.008.

15. A.K. Gupta, H.N. Krishnamurthy, Y. Singh, K.M. Prasad, S.K. Singh, Development of constitutive models for dynamic strain aging regime in Austenitic stainless steel 304, *Materials and Design*, **45**: 616–627, 2013, doi: 10.1016/j.matdes.2012.09.041.

16. R.K. Desu, H.N. Krishnamurthy, A. Balu, A.K. Gupta, S.K. Singh, Mechanical properties of Austenitic Stainless Steel 304L and 316L at elevated temperatures, *Journal of Materials Research and Technology*, **5**(1): 13–20, 2015, doi: 10.1016/j.jmrt.2015.04.001.

17. C.-W. Chang, H.-W. Lee, C.-H. Liu, A review of artificial intelligence algorithms used for smart machine tools, *Inventions*, **3**: 41 (28 pages), 2018, doi: 10.3390/inventions3030041.

18. L.A. Ciro De Filippis, L.M. Serio, F. Facchini, G. Mummolo, ANN modelling to optimize manufacturing process, *Advanced Applications for Artificial Neural Networks*, A. El-Shahat [Ed.], vol. 11, pp. 201–225, 2018, doi: 10.5772/intechopen.71237.

19. N.K. Hong, S.-P. Chang, S.-C. Lee, Development of ANN-based preliminary structural design systems for cable-stayed bridges, *Advances in Engineering Software*, **33**(2): 85–96, 2002, doi: 10.1016/S0965-9978(01)00057-6.

20. W.J.S. Gomes, A.T. Beck, Global structural optimization considering expected consequences of failure and using ANN surrogates, *Computers and Structures*, **126**: 56–68, 2013, doi: 10.1016/j.compstruc.2012.10.013.

21. H. Mote, S.R.S. Kumar, Use of artificial neural network for initial design of steel, *2019 IOP Conference Series: Materials Science and Engineering*, **660**: 012064 (8 pages), 2019, doi: 10.1088/1757-899X/660/1/012064.

22. Y. Liu, The review of intelligent mechanical engineering based on artificial neural network, [in:] *Proceedings of the 2015 International Conference on Intelligent Systems Research and Mechatronics Engineering*, pp. 1969–1973, 2015, doi: 10.2991/isrme-15.2015.405.

23. F.S. Panchal, M. Panchal, Review on methods of selecting number of hidden nodes in artificial neural network, *International Journal of Computer Science and Mobile Computing (IJCSMC)*, **3**(11): 455–464, 2014.

24. T. Kavzoglu, S. Reis, Performance analysis of maximum likelihood and artificial neural network classifiers for training sets with mixed pixels, *GIScience & Remote Sensing*, **45**(3): 330–342, 2008, doi: 10.2747/1548-1603.45.3.330.

25. A. Ng, *One hidden layer. Neural Network. Computing a Neural Network's Output*, Standford University, [online], available: https://cs230.stanford.edu/files/C1M3.pdf.

26. G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, *arXiv:1207.0580*, 2012.

27. G. Li, H. Alnuweiri, Y. Wu, H. Li, Acceleration of backpropagations through initial weight pre-training with delta rule, *IEEE International Conference on Neural Networks*, vol. 1, pp. 580–585, 1993, doi: 10.1109/ICNN.1993.298622.

28. X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, [in:] *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, PMLR, 9, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 2010, http://proceedings.mlr.press/v9/glorot10a.html.

29. J. Martens, I. Sutskever, Training deep and recurrent networks with Hessian-free optimization, [in:] Montavon G., Orr G.B., Müller K.R. [Eds], *Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science*, vol. 7700, Springer, Berlin, Heidelberg, doi: 10.1007/978-3-642-35289-8_27.

30. I. Sutskever, J. Martens, G. Dahl, G. Hinton, On the importance of initialization and momentum in deep learning, [in:] *Proceedings of the 30th International Conference on Machine Learning*, PMLR, vol. 28(3), pp. 1139–1147, Atlanta, Georgia, USA, 2013, http://proceedings.mlr.press/v28/sutskever13.html.

31. B.T. Polyak, Some methods of speeding up the convergence of iteration methods, *USSR Computational Mathematics and Mathematical Physics*, **4**(5): 1–17, 1964.

32. Y. Nesterov, A method for solving the convex programming problem with convergence rate $O(1/k^2)$, *Soviet Mathematics Doklady*, **27**: 372–376, 1983.

33. Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, vol. 87, Kluwer Academic Publishers, 2004.

34. L. Prechelt, Early stopping – But when?, [in:] Orr G.B., Müller K.-R. [Eds], *Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science*, vol. 1524, pp. 55–69, Springer, Berlin, Heidelberg, 1998, doi: 10.1007/3-540-49430-8_3.

35. A.Y. Ng, Feature selection, L_1 vs. L_2 regularization, and rotational invariance, [in:] *ICML '04: Proceedings of the Twenty-First International Conference on Machine Learning*, 78 pages, Banff, Canada, July 2004, doi: 10.1145/1015330.1015435.

36. Z. Lian, X. Jing, X. Wang, H. Huang, Y. Tan, Y. Cui, DropConnect regularization method with sparsity constraint for neural networks, *Chinese Journal of Electronics*, **25**(1): 152–158, 2016, doi: 10.1049/cje.2016.01.023.

37. L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, R. Fergus, Regularization of neural networks using DropConnect, [in:] *Proceedings of the 30th International Conference on Machine Learning*, PMLR, vol. 28(3), pp. 1058–1066, Atlanta, Georgia, USA, 2013.

38. S. Brlek, G. Labelle, A. Lacasse, The discrete Green Theorem and some applications in discrete geometry, *Theoretical Computer Science*, **346**(2–3): 200–225, 2005, doi: 10.1016/j.tcs.2005.08.019.

Received February 2, 2021; revised version June 16, 2021.