sensors

MDPI

Article

Introducing a Novel Model-Free Multivariable Adaptive Neural Network Controller for Square MIMO Systems

Arash Mehrafrooz [1], Fangpo He [2] and Ali Lalbakhsh [3,4,*]

[1] Macquarie University College, Macquarie University, Sydney, NSW 2113, Australia; arash.mehrafrooz@mq.edu.au
[2] Advanced Control Systems Research Group, College of Science and Engineering, Flinders University, Adelaide, SA 5042, Australia; fangpo.he@flinders.edu.au
[3] School of Engineering, Macquarie University, Ryde, NSW 2109, Australia
[4] School of Electrical & Data Engineering, University of Technology Sydney, Sydney, NSW 2007, Australia
[*] Correspondence: ali.lalbakhsh@mq.edu.au

Abstract: In this study, a novel Multivariable Adaptive Neural Network Controller (MANNC) is developed for coupled model-free n-input n-output systems. The learning algorithm of the proposed controller does not rely on the model of a system and uses only the history of the system inputs and outputs. The system is considered as a 'black box' with no pre-knowledge of its internal structure. By online monitoring and possessing the system inputs and outputs, the parameters of the controller are adjusted. Using the accumulated gradient of the system error along with the Lyapunov stability analysis, the weights' adjustment convergence of the controller can be observed, and an optimal training number of the controller can be selected. The Lyapunov stability of the system is checked during the entire weight training process to enable the controller to handle any possible nonlinearities of the system. The effectiveness of the MANNC in controlling nonlinear square multiple-input multiple-output (MIMO) systems is demonstrated via three simulation studies covering the cases of a time-invariant nonlinear MIMO system, a time-variant nonlinear MIMO system, and a hybrid MIMO system, respectively. In each case, the performance of the MANNC is compared with that of a properly selected existing counterpart. Simulation results demonstrate that the proposed MANNC is capable of controlling various types of square MIMO systems with much improved performance over its existing counterpart. The unique properties of the MANNC will make it a suitable candidate for many industrial applications.

Keywords: adaptive neural networks; model-free control; auto-tuning; error back-propagation; accumulated gradient; nonlinear systems; closed-loop stability

check for updates

Citation: Mehrafrooz, A.; He, F.; Lalbakhsh, A. Introducing a Novel Model-Free Multivariable Adaptive Neural Network Controller for Square MIMO Systems. *Sensors* **2022**, *22*, 2089. https://doi.org/10.3390/s22062089

Academic Editor: Chunguo Li

Received: 20 January 2022
Accepted: 7 March 2022
Published: 8 March 2022

1. Introduction

Over the past few years, there has been a significant improvement in controlling Multiple-Input Multiple-Output (MIMO) systems using adaptive control methods [1]. Many proposed adaptive controllers rely on model-based approaches [2,3] where mathematical models of the respective dynamic systems must be identified either directly or indirectly in advance. For most industry applications in practice, however, there are significant challenges with MIMO systems' model identifications. For instance, a predicted industrial plant model can be dynamically different from the true plant itself, due largely to the plant's structural uncertainties, unmodeled nonlinearities, and time-varying natures [4–6]. In some cases, if a constraint of an actual system for any reason changes, in order to achieve the desired outcomes, the system's model may need to be re-identified resulting in a redesigning of the corresponding controller [7]. Even at the circumstances where an exact model of a MIMO system could be identified, the controller designed for the predicted model of the system may still be subjected to conditional variations both internal and external to the system [8]. Due to these practical problems associated with model-based approaches, many existing adaptive control schemes are seen to be impractical or limited in controlling real industrial MIMO plants. In contrast to model-based approaches, model-free approaches [9–12] can entirely omit the modeling phase of a system, thus significantly reducing the time required for the design and tuning of the system's real-time controller. This will result in a faster and more precise control outcome [13–15]. Due to this consideration, model-free controllers are becoming more preferable than their corresponding model-based counterparts, especially for industrial MIMO control applications, where the modeling phase of a true plant can be time consuming and inaccurate. Among the recently introduced model-free control methods [13,15–19], the neural network technique stands out as a powerful and practical tool for controlling MIMO systems due mainly to its excellent capabilities in dealing with large volumes of data, estimating ambiguous relationships between a system's inputs and outputs, and predicting future behaviors of a system. The application of neural networks is not limited to the MIMO systems and has been widely used in a range of engineering solutions, including substance characterizing [20–23], industrial engineering [24,25], microwave engineering [26,27], radiation accuracy [28,29], and measurement technologies [30–33]. Other nature-based algorithms, such as particle swarm [34–36], grey wolf [37–39], genetic algorithm [40], and ant colony [41], can be used in conjunction with the neural networks for further performance improvement.

Recently, by adding adaptive features to neural network schemes, adaptive neural networks have been seen in controlling MIMO systems successfully [42–48]. Many adaptive control methods based on neural networks have been introduced to Single-Input Single-Output (SISO) systems and further developed for Uncoupled Multiple-Input Multiple-Output (U-MIMO) systems [49–52]. In [45], a neural network controller and its associated learning rules are proposed, which can be successfully applied to Single-Input Multi-Output (SIMO) plants. This controller is a combination of several SISO controllers cascaded together and cannot be further developed for coupled MIMO systems. Since in general control problems are more challenging if cross-couplings among the various inputs and outputs of a MIMO system exist, model-free control of coupled MIMO systems has become an active area of research with a growing number of publications [53–59]. Despite some improvements, however, neural network-based controllers have not been extensively used in industrial model-free control systems due to the following apparent deficiencies [60]:

- During the weight training process of the neural networks, the controlled systems can become unstable;
- It is not always clear when to stop the weight training process;
- A long training time for the weights can be unsatisfactory for the speed of the control systems;
- The traditional activation functions employed in the neural networks may not be suitable for control purposes;
- The common error back-propagation learning algorithm uses only the last two consecutive samples of the outputs in discrete derivative functions and does not comply with the requirement of a proper model-free approach in which a full history of inputs and outputs must be used in order to generate an effective control action.

In order to overcome the above-listed deficiencies, in this study, a novel model-free Multivariable Adaptive Neural Network Controller (MANNC) is proposed for controlling coupled MIMO systems in which the following are true.

By using the constraint generated from the Lyapunov stability conditions at each step of the online weight training process, the overall control system stability can be guaranteed at all time; this eliminates the risk of the system been falling into its local minimum [61,62] and prevents the loss of the control speed due to conservative learning rate selections [15,18,47,62,63]:

- By constantly observing the accumulated errors and comparing them with their desired values, the controller can decide to stop the learning algorithm and lock the neural network weights at an optimal point; this ensures the convergence of the controller weight adjustments and provides a clear optimal number for the weight training steps;

- By choosing proper initial learning rates and dynamically changing them during the learning process according to the system stability criteria, the weight training speed can be significantly increased; this forms a clear comparison with and improvement over the traditional static learning rates [15,18,47,62,63];
- By designing specific activation functions that utilize typical proportional, integral, and derivative operations in the neural network structure of the controller, the proposed controller is simple and straightforward in its configuration; this makes the controller a potential candidate suitable for replacing classical PID controllers in industrial applications;
- By applying accumulated gradients in the error back-propagation algorithm and using new partial derivative estimations, the proposed method fully uses the history of the system outputs together with the current weights to produce the outputs of the controller (the inputs of the system) for the next step. This new learning method significantly reduces the overshoot and settling time of the system by minimizing the summation of errors of the system outputs in each step rather than using only the last two consecutive samples of the system outputs as its traditional counterparts do [61,64–67], and allows for the closed-loop system to achieve its best control performance with a minimum number of weight training steps.

It is anticipated that, being truly model-free, the proposed MANNC can generate adequate control actions over a wide range of operating conditions. Additionally, by using a new cross-coupling network structure, the proposed controller is expected to be able to control strongly-coupled MIMO systems effectively. It should be mentioned that, according to the design to be presented in this paper, the proposed MANNC will only be applicable for square (n × n) MIMO systems. If an industrial MIMO system has different numbers of inputs and outputs, by squaring up (adding) or squaring down (removing) the inputs (i.e., the manipulated variables) or the outputs (i.e., the controlled variables), the given non-square MIMO system can be rewritten as a corresponding square MIMO system [68] to which the proposed MANNC will be applicable.

To reveal the proposed MANNC, the rest of this paper is organized as follows. In Section 2, the structure and matrix representation of the new neural network-based adaptive controller are introduced. In Section 3, the new learning method of the neural network based on the error back-propagation algorithm is described. The closed-loop system stability is analyzed in Section 4. In Section 5, the controlled system and the introduced method are specified for SISO systems. Section 6 demonstrates the validation results via simulation studies where the proposed method is seen to control three chosen nonlinear MIMO systems of a drum-boiler, a time-variant nonlinear MIMO system, and a hybrid system without the use of their respective models. Conclusions in relation to the design are drawn in Section 7.

2. Multivariable Adaptive Neural Network Controller

2.1. Closed-Loop Structure of MANNC

As previously proposed in [18] by the authors, the outputs of three types of neurons: P-type, I-type, and D-type, in the discrete form can be respectively expressed as:

$$o_P(k) = net_P(k) \tag{1}$$

$$o_I(k) = o_I(k-1) + net_I(k) \tag{2}$$

$$o_D(k) = net_D(k) - net_D(k-1) \tag{3}$$

where $o_X(k)$ and $net_X(k)$ represent the X-type neuron's output and input at the k^{th} sample time, respectively. In this study, for stability concerns, further constraints to the neurons' activation functions of Equations (1)–(3) are applied as follows:

$$o_P(k) = \begin{cases} 1 \; net_P(k) > 1 \\ net_P(k) - 1 \leq net_P(k) \leq 1 \\ -1 \; net_P(k) < -1 \end{cases} \tag{4}$$

$$o_I(k) = \begin{cases} 1 \; net_I(k) > 1 \\ o_I(k-1) + net_I(k) - 1 \leq net_I(k) \leq 1 \\ -1 \; net_I(k) < -1 \end{cases} \tag{5}$$

$$o_D(k) = \begin{cases} 1 \; net_D(k) > 1 \\ net_D(k) - net_D(k-1) - 1 \leq net_D(k) \leq 1 \\ -1 \; net_D(k) < -1 \end{cases} \tag{6}$$

In [18], a new Adaptive Neural Network Controller (ANNC) structure was proposed. The ANNC contained six neurons and three layers. In the input layer, there are two P-type neurons, which perform the distribution of the inputs in the constructed neural network. In the hidden layer, there are three neurons each being a P-type, an I-type, and a D-type, respectively; the P-neuron compares the desired output with the actual output; the I-neuron provides the necessary action to eliminate the steady-state error; the D-neuron predicts the future behavior of the error. In the output layer, there is a P-type neuron, which performs the summation of the PID functionalities of the neuron. In this paper, a new Multivariable Adaptive Neural Network Controller (MANNC) is proposed in Figure 1a as a closed-loop MIMO controller applicable to coupled square ($n \times n$) multivariable systems.

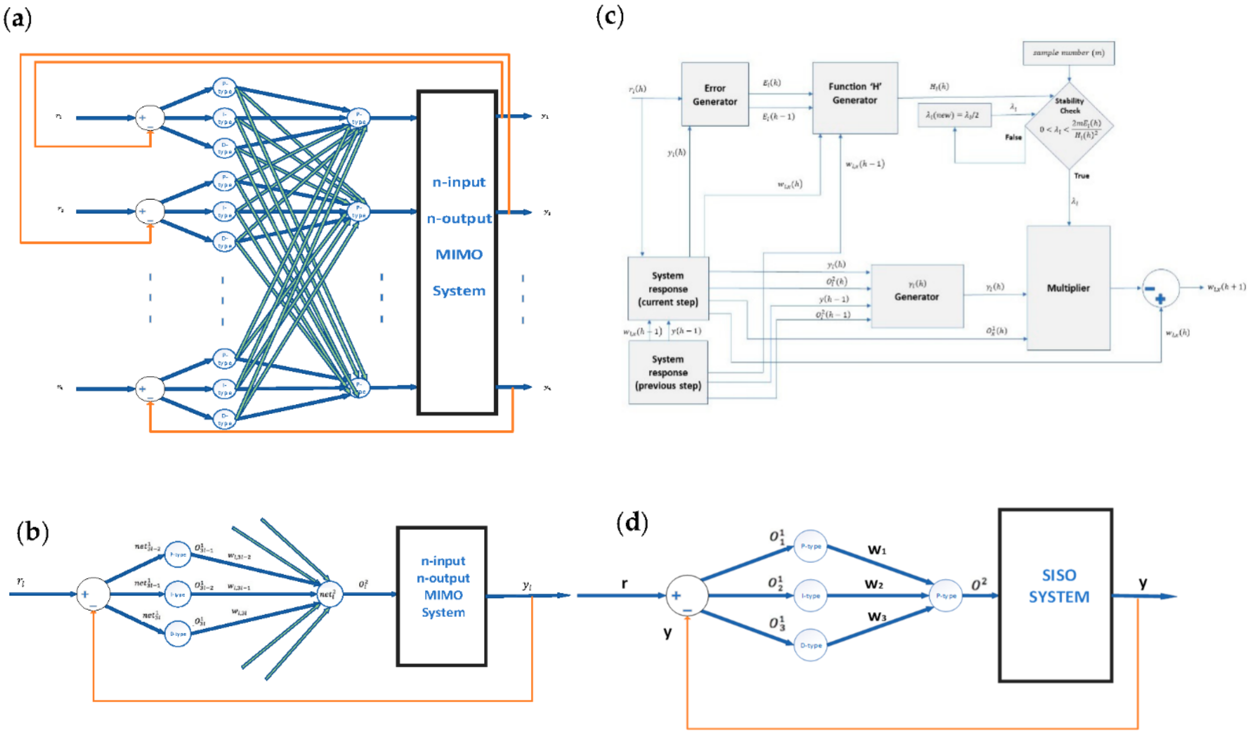

Figure 1. (**a**) Structure of the proposed MANN, (**b**) S-MANNC structure, (**c**) flowchart of real-time simultaneous stability criteria check and weight adjustment algorithm for MANNC, and (**d**) applying MANNC to a SISO system.

Assuming that there are strong cross-coupling between the n inputs and n outputs of a MIMO system of concern, the proposed MANNC is designed using a $n - 3n - n$ neural network structure. In this structure, the error for each output (i.e., the difference between each desired output (r_i) and its corresponding actual output (y_i) $(i = 1, 2, \ldots, n)$) is generated, and the generated errors propagate to the two layers, the hidden layer and the output layer of the MANNC network. In the hidden layer, there are $3n$ neurons including clusters of P-type, I-type, and D-type neurons that are repeated consecutively. In the output layer, there are n P-type neurons that form the outputs of the MANNC, i.e., the inputs of the $n \times n$ MIMO system. There are $3n^2$ weights in the output layer that are associated with the hidden-layer neurons and decide the impact of each neuron of the hidden layer on the generation of the inputs applied to the MIMO system.

2.2. Structure of Sub-MANNC (S-MANNC)

To be able to deal with the cross-couplings of an $n \times n$ MIMO system of concern, the proposed MANNC structure in Figure 1a is further decomposed into n-parallel sub-controllers each named as a Sub-MANNC (S-MANNC) and illustrated in Figure 1b. The neural network associated with the l^{th} S-MANNC with input r_l and output y_l $(l = 1, 2, \ldots, n)$ is designed to have two layers (hidden layer and output layer) and four neurons; each neuron is able to be connected with the neurons of the other S-MANNCs in order to account for the cross-coupling effects of the underlying MIMO system. In the hidden layer of the l^{th} S-MANNC, there are three neurons each being a P-type, an I-type, and a D-type, respectively. The P-neuron amplifies the difference between the desired output and the actual output, the I-neuron provides the necessary action to eliminate the steady-state error, and the D-neuron predicts the future behavior of the error. In the output layer of the l^{th} S-MANNC, there is a P-type neuron, which performs the summation of the PID functionalities of the hidden-layer neurons. This neuron accumulates the outputs of the hidden layer and forms the control command applied to the l^{th} output of the $n \times n$ MIMO system. Due to the fact the MANNC will be used for controlling coupled multivariable systems, the output neuron of each S-MANNC has 'n' number of inputs to be able to produce each control command by considering all the desired outputs and actual outputs. The inputs of the P-type, I-type, and D-type neurons (net^1_{3l-2}, net^1_{3l-1}, and net^1_{3l}) and the outputs of these neurons (O^1_{3l-2}, O^1_{3l-1}, and O^1_{3l}) are related together by the activation functions of the neurons represented in Equations (4)–(6). There are three weights ($w_{1,3l-2}$, $w_{1,3l-1}$, and $w_{1,3l}$) in the output layer of each S-MANNC, relating to the P-type, I-type, and D-type neurons in the hidden layer.

2.3. Matrix Representation

The matrix representation of a closed-loop square MIMO system using the proposed MANNC Figure 1a and S-MANNC (Figure 1b) is derived as follows. Let O^2_l and y_l be, respectively, the l^{th} input and the l^{th} output of the system, where $1 \leq \ l \ \leq n$, and let G_{ij} be the transfer function relating the j^{th}-component of the i^{th} output (y_i) to the j^{th} input (O^2_j), where $1 \leq \ i \ \leq n$ and $1 \leq \ j \ \leq n$. The vectors and matrices associated with Figure 1b are named in Table 1 and are defined as follows.

$$Y = Y_{n\times1} = {\begin{bmatrix} y_1 & y_2 & \cdots & y_l & \cdots & y_n \end{bmatrix}_{1\times n}}^T \tag{7}$$

$$O^2 = O^2_{n\times1} = {\begin{bmatrix} O^2_1 & O^2_2 & \cdots & O^2_l & \cdots & O^2_n n \end{bmatrix}_{1\times n}}^T \tag{8}$$

$$G = G_{n\times n} = \begin{bmatrix} G_{11} & G_{12} & \cdots & G_{1l} & \cdots & G_{1n} \\ G_{21} & G_{22} & \cdots & G_{2l} & \cdots & G_{2n} \\ \vdots & \vdots & & & & \vdots \\ G_{l1} & G_{l2} & \cdots & G_{ll} & \cdots & G_{ln} \\ \vdots & \vdots & & \vdots & & \vdots \\ G_{n1} & G_{n2} & \cdots & G_{nl} & \cdots & G_{nn} \end{bmatrix}_{n\times n} \tag{9}$$

$$net^2 = net^2_{n\times1} = {\begin{bmatrix} net^2_1 & net^2_2 & \cdots & net^2_l & \cdots & net^2_n \end{bmatrix}_{1\times n}}^T \tag{10}$$

$$W = W_{n\times 3n} = \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & \cdots & w_{1,3l-2} & w_{1,3l-1} & w_{1,3l} & \cdots & w_{1,3n-2} & w_{1,3n-1} & w_{1,3n} \\ w_{2,1} & w_{2,2} & w_{2,3} & \cdots & w_{2,3l-2} & w_{2,3l-1} & w_{2,3l} & \cdots & w_{2,3n-2} & w_{2,3n-1} & w_{2,3n} \\ \vdots & \vdots & & & & \vdots & \vdots & & & & \vdots \\ w_{l,1} & w_{l,2} & w_{l,3} & \cdots & w_{l,3l-2} & w_{l,3l-1} & w_{l,3l} & \cdots & w_{l,3n-2} & w_{l,3n-1} & w_{l,3n} \\ \vdots & \vdots & & & & \vdots & \vdots & & \vdots & \vdots & \vdots \\ w_{n,1} & w_{n,2} & w_{n,3} & \cdots & w_{n,3l-2} & w_{n,3l-1} & w_{n,3l} & \cdots & w_{n,3n-2} & w_{n,3n-1} & w_{n,3n} \end{bmatrix}_{n\times 3n} \tag{11}$$

$$O^1 = O^1_{3n\times 1} = \begin{bmatrix} O^1_1 & O^1_2 & O^1_3 & \cdots & O^1_{3n} \end{bmatrix}_{1\times 3n}{}^{T} \tag{12}$$

$$P = P_{3n\times 3n} = \begin{bmatrix} 1 & 0 & 0 & & & & \\ 0 & D^{-1} & 0 & \cdots & & \mathbf{0} & \\ 0 & 0 & D & & & & \\ & \vdots & & \ddots & & \vdots & \\ & & & & 1 & 0 & 0 \\ & \mathbf{0} & & \cdots & 0 & D^{-1} & 0 \\ & & & & 0 & 0 & D \end{bmatrix}_{3n\times 3n} \tag{13}$$

$$net^1 = net^1_{3n\times 1} = \begin{bmatrix} net^1_1 & net^1_2 & net^1_3 & \cdots & net^1_{3n} \end{bmatrix}_{3n\times 1}{}^{T} \tag{14}$$

$$R^{(3)} = R^{(3)}_{3n\times 1} = \begin{bmatrix} r_1 & r_1 & r_1 & r_2 & r_2 & r_2 & \cdots & y_n & y_n & y_n \end{bmatrix}_{1\times 3n}{}^{T} \tag{15}$$

$$Y^{(3)} = Y^{(3)}_{3n\times 1} = \begin{bmatrix} y_1 & y_1 & y_1 & y_2 & y_2 & y_2 & \cdots & y_n & y_n & y_n \end{bmatrix}_{1\times 3n}{}^{T} \tag{16}$$

$$I^{(3)} = I^{(3)}_{3n\times n} = \begin{bmatrix} 1 & & \\ 1 & \cdots & 0 \\ 1 & & \\ \vdots & \ddots & \vdots \\ & & 1 \\ 0 & \cdots & 1 \\ & & 1 \end{bmatrix}_{3n\times n} \tag{17}$$

$$R = R_{n\times 1} = \begin{bmatrix} r_1 & r_2 & r_3 & \cdots & r_n \end{bmatrix}_{1\times n}{}^{T} \tag{18}$$

Table 1. Matrices defined for MANNC.

$\boldsymbol{Y_{n\times 1}}$	$\boldsymbol{R_{n\times 1}}$	$\boldsymbol{O^2_{n\times 1}}$	$\boldsymbol{G_{n\times n}}$
System's outputs	System's desired outputs	System's inputs	System's transfer matrix
$\boldsymbol{net^2_{n\times 1}}$	$\boldsymbol{W_{n\times 3n}}$	$\boldsymbol{O^1_{3n\times 1}}$	$\boldsymbol{P_{3n\times 3n}}$
Output layer's inputs	Neural Network Weights	Neurons' outputs	Activation functions
$\boldsymbol{net^1_{3n\times 1}}$	$\boldsymbol{R^{(3)}_{3n\times 1}}$	$\boldsymbol{Y^{(3)}_{3n\times 1}}$	$\boldsymbol{I^{(3)}_{3n\times n}}$
Hidden layer's inputs	Triple desired outputs	Triple system's outputs	Triple unit

Considering that the system at each calculation step can be identified as being linear or been linearized around an operation point, the relationship between the inputs and outputs of the system will be as follows:

$$Y_{n\times 1} = G_{n\times n} O^2_{n\times 1} \tag{19}$$

Since net^2_l is the l^{th} input of the P-type neuron in the output layer, one has the following:

$$O^2_{n\times 1} = net^2_{n\times 1} \tag{20}$$

where the relationship between the inputs and outputs in the output layer of the neural networks can be expressed as follows:

$$net^{2}_{n\times 1} = W^{2}_{n\times 3n} O^{1}_{3n\times 1} \tag{21}$$

As the hidden layer has clusters of P-type, I-type, and D-type neurons, the proportional, integral, and derivative operators (1, D^{-1}, and D) are considered, respectively, in the matrix form of the activation function. Thus, one has the following:

$$O^{1}_{3n\times 1} = P_{3n\times 3n} net^{1}_{3n\times 1} \tag{22}$$

where the inputs of the hidden layer are defined as the differences between the desired outputs and the actual outputs and are expressed as follows:

$$net^{1}_{3n\times 1} = R^{3}_{3n\times 1} - Y^{3}_{3n\times 1} \tag{23}$$

By having

$$R^{(3)}_{3n\times 1} = I^{(3)}_{3n\times n} \times R_{n\times 1} \tag{24}$$

and

$$Y^{(3)}_{3n\times 1} = I^{(3)}_{3n\times n} \times Y_{n\times 1} \tag{25}$$

and using Equations (19)–(25), one obtains

$$Y = GWP(R^{(3)} - Y^{(3)}) \;=\; GWP(I^{(3)}R - I^{(3)}Y) \tag{26}$$

Hence, the system's outputs can be derived as follows:

$$Y = GWPI^{(3)}R(I + GWPI^{(3)})^{-1} \tag{27}$$

where $\left|I + GWPI^{(3)}\right| \neq 0$.

Additionally, by having

$$Y = GWPI^{(3)}(R - Y) \tag{28}$$

one has

$$G^{-1}Y = WPI^{(3)}(R - Y) \tag{29}$$

where $|G| \neq 0$.

Since $PI^{(3)}(R - Y)$ is a non-square matrix and P is symmetrical, from

$$G^{-1}Y\{PI^{(3)}(R - Y)\}^{T} = WPI^{(3)}(R - Y)\{PI^{(3)}(R - Y)\}^{T} \tag{30}$$

one derives

$$W = G^{-1}Y(R - Y)^{T} I^{(3)T} P\left[PI^{(3)}(R - Y)(R - Y)^{T} I^{(3)T} P\right]^{-1} \tag{31}$$

where W is expressed as a function of the desired outputs and the actual outputs, and $\left|PI^{(3)}(R - Y)(R - Y)^{T} I^{(3)T} P\right| \neq 0$ and $|W| \neq 0$.

The system can then be identified as follows:

$$G = Y(R - Y)^{T} I^{(3)T} P\left[PI^{(3)}(R - Y)(R - Y)^{T} I^{(3)T} P\right]^{-1} W^{-1} \tag{32}$$

where $|W| \neq 0$.

It should be pointed out that, although Equations (19)–(32) are derived under the assumption that the system is linear or can be linearized around an operating point, they can potentially be used for nonlinear systems where the nonlinearities of the systems can be approximated by piece-wise linear systems whose time-varying nature can account for the nonlinearities of the systems satisfactorily. It should also be pointed out that if any of the non-singularity conditions for matrices $\left(I + GWPI^{(3)}\right)$, (G), $\left(PI^{(3)}(R-Y)(R-Y)^T I^{(3)T} P\right)$, and (W) could not be met, the selected weights (matrix W) would not be acceptable and would be re-updated until all these matrices become non-singular.

3. Learning Algorithm

To achieve a precise control effect for a square MIMO system, the neural network weights of the MANNC are adjusted using the principle of the multi-step error back-propagation algorithm described in [69]. Back-propagation is one of the standard methods of training artificial neural networks. This method efficiently computes the weights of one layer at a time using the delta rule. It helps calculate the gradient of a loss function with respect to all the weights in the network. The algorithm is to find the set of weights that minimizes the error by the steepest descent direction calculated for the loss function versus the present weights. The weights will be updated along the steepest descent direction, and the error is reduced in every step [70,71]. The choice of this algorithm is based on the considerations that the system is treated as a "black-box" and the controller will be designed based on a model-free approach. In this study, instead of using merely the current gradient of the system error as the literature does for SISO systems [47,62,63], an accumulated gradient of the system error for the last m samples is proposed to be used to achieve a more precise control performance. The proposed method minimizes the sum of the square accumulated gradient of the error for each system output in each learning step, where the error is taken as the difference between the desired output $r_l(k)$ (i.e., the system setpoint) and the actual output $y_l(k)$. The Euclidean norm E is defined for calculating the quadratic cost function of the system for the system error. The power of two in this expression makes the error of each output positive, so that larger errors will be more significant than the smaller errors. The cost function for the l^{th} S-MANNC in Figure 1b is then defined as follows:

$$E_l(h) = \frac{1}{2}\left(\sum_{k=1}^{m}(r_l[k] - y_l(h)[k])\right)^2 \tag{33}$$

where $E_l(h)$ is the error of the l^{th} output in the h^{th} step number of the learning algorithm and m is the required number of discrete samples in the system output and the set-point. By increasing m, the system output will be compared more accurately with the setpoint. However, a large value of m may decrease the adjustment speed of the controller and become undesirable when the speed of the control system (often a critical requirement for a real-time industrial system) is of concern. Therefore, a reasonable value of m must be used to make a necessary trade-off between the desired accuracy and the required speed of the control system. The total cost function of the system (J), which is the sum of n errors by considering all outputs, is written as follows:

$$J(h) = \sum_{l=1}^{n} E_l(h) = \frac{1}{2}\sum_{l=1}^{n}\left(\sum_{k=1}^{m}(r_l[k] - y_l(h)[k])\right)^2 \tag{34}$$

Using the accumulated gradient of the system error, a learning algorithm must be designed to minimize the defined cost function and to bring the system outputs as close as possible to the desired outputs. According to the principle of the error back-propagation learning algorithm, the output layer's weights must be adjusted so that in each step they move slightly in the opposite direction of the gradient of the cost function. This is to ensure that the cost function will be decreasing step by step. The weights of the output layer will therefore be adjusted based on the following learning rule:

$$w_{l,x}(h+1) = w_{l,x}(h) - \lambda_l \frac{\partial J(h)}{\partial w_{l,x}} \tag{35}$$

where $1 \leq x \leq 3l$ and $1 \leq l \leq n$, h is the step number of the learning algorithm, and $w_{l,x}(h)$ and $w_{l,x}(h+1)$ are the weights of the output layer in the current and the following steps, respectively; λ_l is the learning rate which decides how fast the cost is changing and, in particular, determines the weight adjustment speed. The gradient of the error subject to each weight is required to be calculated. Using partial derivatives, one has

$$\frac{\partial J}{\partial w_{l,x}} = \frac{\partial J}{\partial E_l} \frac{\partial E_l}{\partial y_l} \frac{\partial y_l}{\partial O_l^2} \frac{\partial O_l^2}{\partial net_l^2} \frac{\partial net_l^2}{\partial w_{l,x}} \tag{36}$$

where

$$\frac{\partial J}{\partial E_l} = 1 \tag{37}$$

$$\frac{\partial E_l}{\partial y_l} = \sum_{k=1}^{m} (y_l(h)[k] - r_l(h)[k]) \tag{38}$$

$$\frac{\partial y}{\partial O_l^2} \cong \frac{\sum_{k=1}^{m}(y(h)[k]) - \sum_{k=1}^{m}(y(h-1)[k])}{\sum_{k=1}^{m}\left(O_l^2(h)[k]\right) - \sum_{k=1}^{m}\left(O_l^2(h-1)[k]\right)} \tag{39}$$

$$\frac{\partial O_l^2}{\partial net_l^2} = 1 \tag{40}$$

Because the output neuron is a 'P-type' neuron, one writes

$$\frac{\partial net_l^2}{\partial w_{l,x}} = \sum_{k=1}^{m} \left(O_x^1(h)[k] \right) \tag{41}$$

Substituting Equations (37)–(41) into (36), one obtains

$$\frac{\partial J}{\partial w_{l,x}} \cong 1 \times \sum_{k=1}^{m} (y_l(h)[k] - r_l(h)[k]) \frac{\sum_{k=1}^{m}(y(h)[k]) - \sum_{k=1}^{m}(y(h-1)[k])}{\sum_{k=1}^{m}\left(O_l^2(h)[k]\right) - \sum_{k=1}^{m}\left(O_l^2(h-1)[k]\right)} O_x^1(h)[k] \tag{42}$$

Defining $\gamma_l(k)$ as

$$\gamma_l(h) = \sum_{k=1}^{m} (y_l(h)[k] - r_l(h)[k]) \frac{\sum_{k=1}^{m}(y(h)[k]) - \sum_{k=1}^{m}(y(h-1)[k])}{\sum_{k=1}^{m}\left(O_l^2(h)[k]\right) - \sum_{k=1}^{m}\left(O_l^2(h-1)[k]\right)} \tag{43}$$

the output layer weight adjustment rule can thus be derived as

$$w_{l,x}(h+1) = w_{l,x}(h) - \lambda_l [\gamma_l(h) \sum_{k=1}^{m} \left(O_x^1(h)[k] \right)] \tag{44}$$

The resulting weight adjustment algorithm is summarized in Table 2.

Table 2. Weights adjustment learning algorithm.

$1 \leq l \leq n$	$w_{l,x}(h+1) = w_{l,x}(h) - \lambda_l[\gamma_l(h) \sum_{k=1}^{m} \left(O_x^1(h)[k]\right)]$
$1 \leq x \leq 3n$	$\gamma_l(h) = \sum_{k=1}^{m} (y_l(h)[k] - r_l(h)[k]) \frac{\sum_{k=1}^{m}(y(h)[k]) - \sum_{k=1}^{m}(y(h-1)[k])}{\sum_{k=1}^{m}\left(O_l^2(h)[k]\right) - \sum_{k=1}^{m}\left(O_l^2(h-1)[k]\right)}$

Using the described learning method, all weights are simultaneously tuned, and their values at the current step, together with all the current inputs and all the current outputs, are used to create the outputs for the weights' training at the next step. This online dynamic learning feature of the proposed method (i.e., applying the accumulated gradient and the accumulated errors) makes it more suitable for MIMO systems than its existing counterparts that are mainly applicable for SISO systems [61,62,72]. The neural network structure presented for the MANNC is designed for a general n-input n-output system with cross-couplings among all its inputs and outputs. This structure can be simplified for systems with less cross-coupling to reduce the number of the associated weights and, consequently, to increase the learning speed.

4. Stability Analysis

It is well known that when a new control method is proposed, it is necessary to investigate the stability condition of the resultant closed-loop system in order to ensure the achievement of the desired control outcomes. For an unconstrained control system, the system stability can be defined using Bounded-Input Bounded-Output (BIBO) stability criteria. While the eigenvalue analysis concept based on the BIBO stability criteria can be used to investigate the stability condition of a linear system, it can not be applied to nonlinear systems. Instead, the Lyapunov stability analysis concept becomes a useful tool for nonlinear systems. As the proposed model-free MANNC method will inherently result in a nonlinear closed-loop control system, its closed-loop stability investigation will need to be carried out using the Lyapunov stability analysis concept. Although, unlike a linear system, the stability of a nonlinear system may not need to be global as the system can have multiple equilibrium points and limit cycles, and global asymptotic stability is sought in this study for the proposed MANNC to ensure its satisfactory closed-loop performance over a wide range of operating points. This will make the controller more suitable for use in industrial applications. According to the Lyapunov global asymptotic stability theorem, for a defined function $V(x)$, if:

(i) $V(0) = 0$
(ii) (For all $x \neq 0$, $V(x) > 0$ (i.e., V is positive definite)
(iii) For all $x \neq 0$, $\Delta V(x) < 0$

Then every trajectory of $\dot{X} = f(X)$ will converge to zero as $t \rightarrow \infty$ and the system will be globally asymptotically stable.

Consider a model-free MANNC control system whose Lyapunov function for each of its outputs is defined as follows:

$$V_l(h) = E_l(h) \tag{45}$$

where $E_l(h)$ is the cost function related to the h^{th} step of the learning algorithm. One can write

$$\Delta V_l(h) = (E_l(h) + \Delta E_l(h))^2 - (E_l(h))^2 = 2E_l(h)\Delta E_l(h) + (\Delta E_l(h))^2 \tag{46}$$

and:

$$\Delta E_l(h) \cong \Delta w_{l,x}(h) \sum_{k=1}^{m} \frac{\partial E_l(h)[k]}{\partial w_{l,x}} \tag{47}$$

From Equation (46), one has

$$\Delta w_{l,x}(h) = -\frac{\lambda_l}{m} \sum_{k=1}^{m} \frac{\partial J}{\partial w_{l,x}} \tag{48}$$

where

$$\frac{\partial J}{\partial w_{l,x}} = \frac{\partial J}{\partial E_l(h)} \frac{\partial E_l(h)}{\partial w_{l,x}} = \frac{\partial E_l(h)}{\partial w_{l,x}} \tag{49}$$

Substitute Equation (49) into Equation (47), respectively, and one derives

$$\Delta w_{l,x}(h) = -\frac{\lambda_l}{m}\sum_{k=1}^{m}\frac{\partial E_l(h)[k]}{\partial w_{l,x}} \tag{50}$$

and

$$\Delta E_l(h) \cong -\frac{\lambda_l}{m}\left(\sum_{k=1}^{m}\frac{\partial E_l(h)[k]}{\partial w_{l,x}}\right)^2 \tag{51}$$

Hence, Equation (46) can be expressed as

$$\Delta V(h) \cong -\frac{2\lambda_l E_l(h)}{m}\left(\sum_{k=1}^{m}\frac{\partial E_l(h)[k]}{\partial w_{l,x}}\right)^2 + \frac{{\lambda_l}^2}{m^2}\left(\sum_{k=1}^{m}\frac{\partial E_l(h)[k]}{\partial w_{l,x}}\right)^4 \tag{52}$$

Define

$$H_l(h) = \sum_{k=1}^{m}\frac{\partial E_l(h)[k]}{\partial w_{l,x}} \cong \sum_{k=1}^{m}\frac{\Delta E_l(h)[k]}{\Delta w_{l,x}(h)} \tag{53}$$

The condition for $\Delta V(h) < 0$ will yield the following constraint on the selection of the learning rate λ_l defined in Equation (44):

$$0 < \lambda_l < \frac{2mE_l(h)}{H_l(h)^2} \tag{54}$$

If the above constraint is satisfied at each training step, the system will be globally asymptotically stable during the entire training process. This indicates that Equation (54) must be checked in a real-time and simultaneous fashion at each training step during the operation of the weight adjustment algorithm. The flowchart of the whole real-time weight training process, including both the learning algorithm (Table 2) and the stability criteria check in Equation (54), is illustrated in Figure 1c.

5. Specifying MANNC to Control SISO Systems

The proposed MANNC can be used in a SISO environment where the number of inputs and outputs of the controller is chosen as one, i.e., $n = 1$. The resultant SISO closed-loop system is shown in Figure 1d. Considering Figure 1d and assuming a linear and time-invariant estimation for the system's SISO transfer function, one has

$$Y = G(s)O^2 = G(s)(w_1 O_1^1(s) + w_2 D^{-1} O_2^1(s) + w_3 D O_3^1(s)) \tag{55}$$

where

$$O_1^1(s) = O_2^1(s) = O_3^1(s) = R(s) - Y(s) \tag{56}$$

Then,

$$Y(s) = G(s)(w_1 + w_2 s^{-1} + w_3 s)(R(s) - Y(s)) \tag{57}$$

and

$$\frac{Y(s)}{R(s)} = \frac{G(s)(w_1 + w_2 s^{-1} + w_3 s)}{1 + G(s)(w_1 + w_2 s^{-1} + w_3 s)} \tag{58}$$

If w_1, w_2, and w_3 are taken as the coefficients of a classical proportional, integral, and derivative controller, i.e., K_P, K_I, and K_D, respectively, the resultant control system will perform similar to an auto-tune SISO PID control system whose closed-loop transfer function is expressed as follows:

$$T(s) = \frac{G(s)(K_P + K_I s^{-1} + K_D s)}{1 + G(s)(K_P + K_I s^{-1} + K_D s)} \tag{59}$$

The weight learning algorithm illustrated in Table 2 can be customized for SISO system applications and described in Table 3 where λ is the learning rate.

Table 3. Specified weight learning algorithm for auto-tune classical PID controller.

$K_P(h+1) = K_P(h) - \lambda\gamma(h) \sum_{k=1}^{m} (O_1^1(h)[k])]$
$K_I(h+1) = K_I(h) - \lambda\gamma(h) \sum_{k=1}^{m} (O_2^1(h)[k])]$
$K_D(h+1) = K_D(h) - \lambda\gamma(h) \sum_{k=1}^{m} (O_3^1(h)[k])]$
$\gamma(h) = \sum_{k=1}^{m} (y(h)[k] - r(h)[k]) \frac{\sum_{k=1}^{m}(y(h)[k]) - \sum_{k=1}^{m}(y(h-1)[k])}{\sum_{k=1}^{m}(O^2(h)[k]) - \sum_{k=1}^{m}(O^2(h-1)[k])}$

Following the Lyapunov stability analysis described in Equations (45)–(52) and redefining Equation (53) as follows:

$$H(h) = \sum_{k=1}^{m} \frac{\partial E(h)[k]}{\partial w_{l,x}} \cong \sum_{k=1}^{m} \frac{\Delta E(h)[k]}{\Delta w_x(h)} \tag{60}$$

the condition for $\Delta V(h) < 0$ yields

$$0 < \lambda < \frac{2mE(h)}{H(h)^2} \tag{61}$$

6. Simulation Results

Over the recent years, various computational approaches, such as finite element, finite difference time domain, finite-difference frequency-domain, and Lyapunov stability analysis have been successfully applied to a variety of control systems, including photonic crystals [73–78], high-frequency passive components [79–87], electromagnetic devices [88–94], fluid mechanical engineering [95,96], and MIMO control systems [97,98]. Simulation studies using MATLAB are carried out to evaluate the performances of the proposed MANNC in tracking setpoints, reducing unwanted overshoots or undershoots, and securing the global stability of a closed-loop system during the system's entire control process. The structure of the MANNC proposed in Section 2, the dynamic neural network algorithm developed in Section 3, and the stability criteria checking condition discussed in Section 4 are used in three simulation cases, each presenting a type of square MIMO systems. They are a time-invariant nonlinear system, a time-variant nonlinear system, and a hybrid system.

6.1. Case 1: Application of MANNC on a Time-Invariant Nonlinear Square MIMO System

In this case, a drum-boiler plant (Figure 2a), which is a generic nonlinear time-invariant coupled two-input two-output system with heat flow rate and mass flow rate as inputs and pressure and level as outputs, is chosen. The nonlinearities of the plant cause the system dynamic characteristics to vary with operating conditions. In addition, cross-couplings and parameter variations of the plant makes it a challenging case to control. For years, constructing a nonlinear controller directly from the original nonlinear model of the drum-boiler has been a general approach to use in order to improve the system performance in compensating the plant's nonlinearities. In this study, however, the proposed model-free MANNC method is used in which the nonlinear model of the drum-boiler is not required by the controller design process and only the inputs and outputs of the plant are used for the construction of the controller. The performances of the MANNC in improving the system setpoint tracking time and reducing undesirable overshoots are compared with those of the best and foremost existing neural network method reported in the literature.

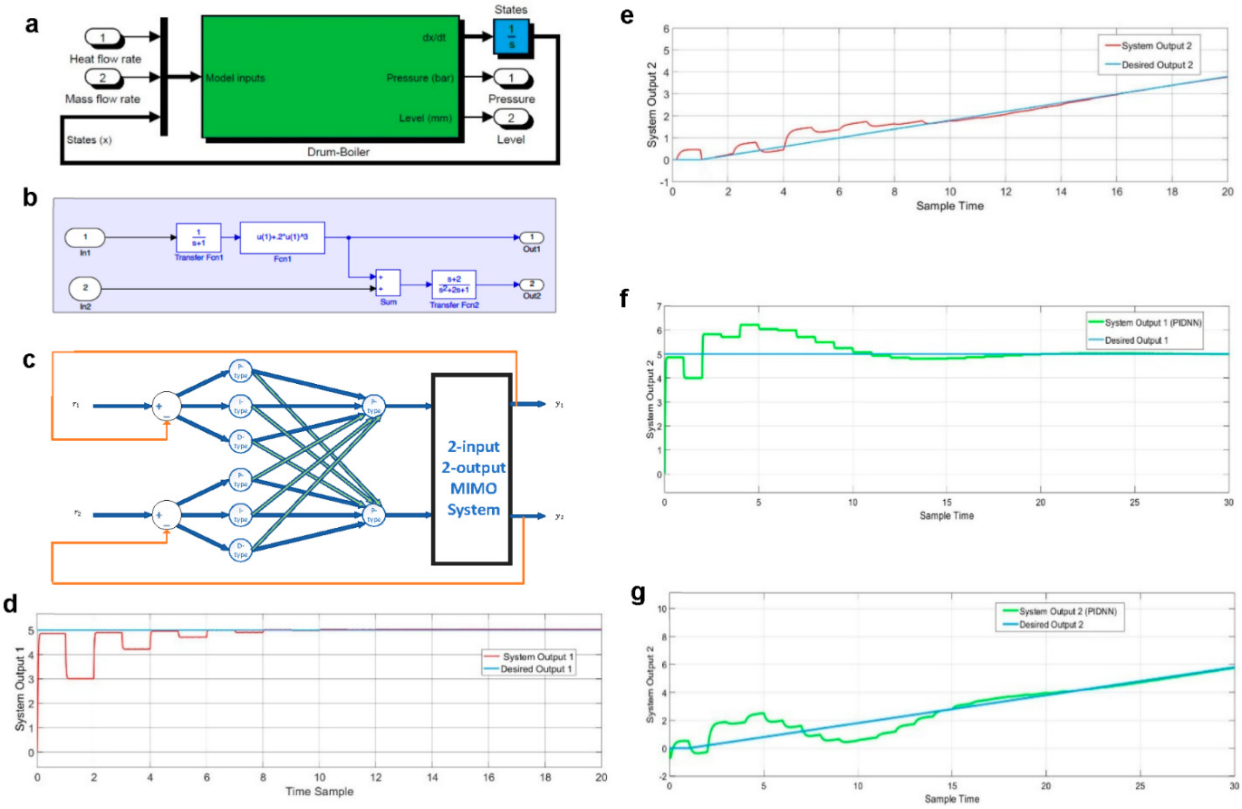

Figure 2. (**a**) Two-input two-output drum-boiler system, (**b**) two-input two-output nonlinear drum-boiler model, and (**c**) two-input two-output system controlled by MANNC. (**d**) Case 1: output one and desired output one (MANNC), (**e**) case 1: output two and desired output two (MANNC), (**f**) case 1: output one and setpoint one (PIDNN), and (**g**) case 1: output two and setpoint two (PIDNN).

The nonlinear state equations describing the relationships between the inputs and outputs of the drum-boiler system of Figure 2a are written as follows:

$$\dot{x_1}(t) = -x_1(t) + u_1(t) \tag{62}$$

$$y_1(t) = x_1(t) + 2x_1^3(t) \tag{63}$$

$$x_2(t) = y_1(t) + u_2(t) \tag{64}$$

$$\ddot{y_2}(t) + 2\dot{y_2}(t) + y_2(t) = \dot{x_2}(t) + 2x_2(t) \tag{65}$$

where "dot" denotes time derivative, $u_1(t)$ and $u_2(t)$ are respectively the heat flow rate and mass flow rate as the system inputs, $y_1(t)$ and $y_2(t)$ are respectively the pressure and level as the system outputs, and $x_1(t)$ and $x_2(t)$ are the state variables. The simulation block diagram of the given nonlinear system is presented in Figure 2b, where box 'Fcn1' produces the highly nonlinear relationship between the inputs and outputs of the system. Due to the fact that Input 1 affects both Output 1 and Output 2, the system is also a cross-coupled MIMO system and, thus, will not be able to be controlled by multiple ANNC controllers introduced in [99], nor by any other SISO or non-coupled MIMO counterparts.

Figure 2c demonstrates the implementation of the proposed MANNC in the two-input two-output system. In this neural network, 12 weights in the hidden layer must be trained. The following desired outputs are selected for the drum-boiler system.

- $r_1(t) = 5.0u(t-1)$ where $u(t)$ is the standard unit step function.
- $r_2(t) = 0.2r(t-1)$ where $r(t)$ is the standard unit ramp function.

These desired outputs represent a scenario in which the pressure will tend to reach 5 bars and the level will increase following a smooth ramp function. The ramp function is chosen to be the setpoint for Output 2 in order to make the MANNC control more challenging. The objective is to find suitable values for the weights of the output layer so as to force the outputs, y_1 and y_2, to follow the desired setpoints, r_1 and r_2, respectively.

To apply the MANNC to the drum-boiler system, all initial weights are set to 1, the learning rates are set to 0.1 ($\lambda_1 = \lambda_2 = 0.1$), and the number of samples is set to 40 ($m = 40$). After repeating, simultaneously, the online learning algorithm (Table 2) and the Lyapunov stability criteria check (Figure 1c, 20 times), the final weight values of the output layer of the MANNC neural network are obtained and reported in Table 4.

Table 4. Final weight values of the output layer.

$w_{l,x}$	x=1	x=2	x=3	x=4	x=5	x=6
$l = 1$	10.23	0.33	6.93	−2.23	−2.13	−2.10
$l = 2$	−1.35	3.59	1.36	3.22	3.24	1.94

During the running of the weight training algorithm, the control system remains stable all the time as λ_1 and λ_2 are kept unchanged. If the control system becomes unstable at a training step, the learning rates will be reduced, and the learning algorithm will be continued with a lower training speed that is dictated by the new learning rates. Applying the final adjusted weights (Table 4) to the control system, Figure 2d shows that Output 1 tracks the desired step output properly with a zero overshoot. This zero-overshoot effect is most desirable for many industrial control systems, as unwanted overshoots during setpoint changes can bring devastated results to the industrial plants. For example, when filling a hazardous liquid tank in a water plant, overshoots can result in overflows if the setpoint is close to the tank's height [99]. Figure 2e demonstrates that Output 2 tracks the desired ramp output properly, and in less than nine samples the output can reach the desired level with a less than 5% error.

To demonstrate the anticipated highly improved performance of the MANNC over its existing adaptive counterparts, a most recent neural network controller (named PIDNN) introduced in [61] is chosen for the drum-boiler system. The PIDNN uses a learning algorithm that is based on two consecutive time-samples. The weights of the PIDNN are set after running the learning algorithm 20 times—the same condition upon which the weights of the MANNC are obtained. The comparison results are given in Figure 2f,g. It is seen that using the PIDNN method, Output 1 presents a 22% undesirable overshoot, and Output 2 exhibits a slower setpoint tracking with larger fluctuations.

The comparison results between the MANNC and the PIDNN control effects with an equal number of trainings are given in Table 5. It is evident that, compared to the PIDNN control system, the MANNC control system can significantly reduce the required training time (by 59%), achieve a zero overshoot (100% reduction), and reach the 5% and 2% error bands in much shorter times (47% and 50% faster for Output 1, and 47% and 36% faster for Output 2, respectively). These results demonstrate a superior performance of the MANNC over its existing counterpart. In particular, Output 1 under MANNC control exhibits a deadbeat-like response with minimum rise time, minimum settling time, no overshoot, and no steady-state error, comparable with an optimal closed-loop response. This remarkable result is attributed to the MANNC strategy that uses the accumulated error and the response of the system in consecutive learning steps rather than in consecutive sample times.

Table 5. Case 1: MANNC vs. PIDNN.

Controller	Number of Trainings	Time of Training	Output 1 Overshoot	Output 1 Maximum Error Less than 5%	Output 1 Maximum Error Less than 2%	Output 2 Maximum Error Less than 5%	Output 2 Maximum Error Less than 2%
MANNC	20	1.48 s	0%	8 s.t. *	10 s.t.	9 s.t.	14 s.t.
PIDNN	20	3.67 s	22%	15 s.t.	20 s.t.	17 s.t.	22 s.t.

* s.t. stands for sample times.

Figure 3a,b represent, respectively, the accumulated error versus the iteration number of the MANNC weight learning algorithm for Output 1 and Output 2. The errors shown in these figures are the differences between the actual outputs and the desired outputs. It is observed that, with the MANNC, the magnitudes of the errors generally decrease as the number of iterations in the weight learning algorithm increases. This is evidence for the convergence of the proposed learning algorithm in Section 3 and is viewed as a significant result that demonstrates the suitable performance of the proposed MANNC for the considered coupled two-input two-output nonlinear system. As shown in these figures, with 35 trainings, the errors are reasonably low, and the values of the weights can be locked in at this point. By continuing the weight adjustment algorithm up to 50 times, the steady-state errors for both outputs become nearly zero. In general, choosing the optimal training number in this method depends on the particular application in which the controller is employed, and implies a trade-off between the control speed and the control performance of the closed-loop system. It is possible to pre-define a desired accumulated error, so that when the actual error reaches to that value, the training process stops and the weights become locked until the next change in the system happens (e.g., a change in the model of the system and/or a change in any of the setpoints).

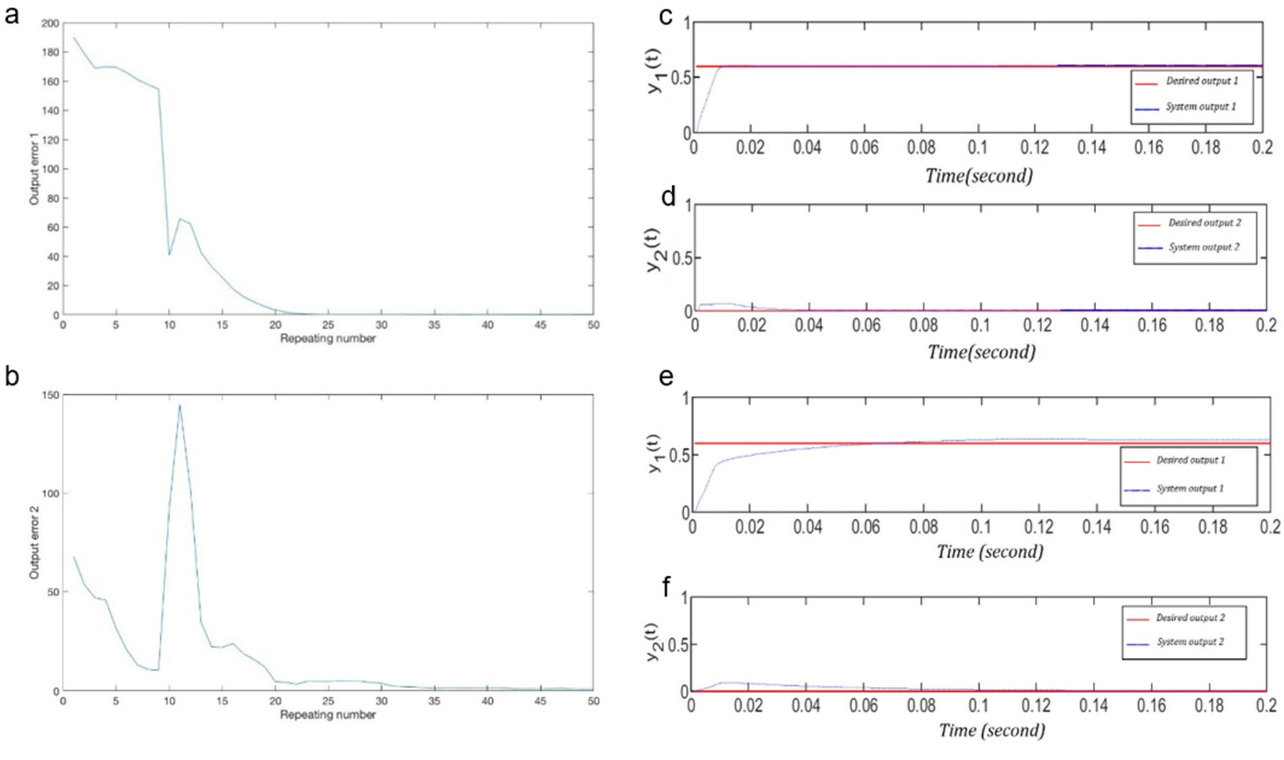

Figure 3. (**a**) Case 1: error value for Output 1 versus the repeating number and (**b**) case 1: error value for Output 2 versus the repeating number; (**c**) case 2: Output 1 and desired Output 1 by MANNC, (**d**) case 2: Output 2 and desired Output 2 by MANNC, (**e**) case 2: Output 1 and desired Output 1 by PIDNN, and (**f**) case 2: Output 2 and desired Output 2 by PIDNN.

6.2. Case 2: Application of MANNC on a Time-Variant Nonlinear MIMO System

In this case, a discrete time, highly nonlinear, time-variant two-input two-output coupled system is chosen to test the suitability and performance of the MANNC for time-variant MIMO systems. The chosen system is expressed as:

$$y_1(k+1) = \frac{1}{{y_1}^2(k)+1}(0.8y_1(k) + v_1(k-2) + 0.2v_2(k-3)) \tag{66}$$

$$y_2(k+1) = \frac{1}{{y_2}^2(k)+1}(0.9y_2(k) + 0.3v_1(k-3) + v_2(k-2)) \tag{67}$$

where $y_1(k)$ and $y_2(k)$ are the system outputs, and $v_1(k)$ and $v_2(k)$ are the system inputs. The desire outputs are selected as $r_1(k) = 0.6$ and $r_2(k) = 0$.

By applying, simultaneously, the online learning algorithm (Table 2) and the stability criteria check (Figure 1c) 50 times, the final weight values of the output layer of the MANNC neural network are obtained and reported in Table 6.

Table 6. Final weight values of the output layer.

$w_{l,x}$	*x*=1	*x*=2	*x*=3	*x*=4	*x*=5	*x*=6
$l = 1$	3.34	2.43	4.73	−5.12	−8.13	−2.19
$l = 2$	−11.30	−4.44	−7.74	6.32	3.55	3.82

The simulation results of the final adjusted weights of the MANNC are shown in Figure 3c,d. In order to compare the MANNC results with those of a properly selected existing counterpart, the PIDNN introduced in [61] is applied to the same system. The system outputs under the PIDNN control are shown in Figure 3e,f. It is demonstrated that both outputs of the MANNC control system track the desired outputs faster than those of the PIDNN control system. In addition, a zero overshoot in Output 1 using the MANNC is achieved, which is a significant result as this is not achievable by using the PIDNN. Decreasing the accumulated error by increasing the number of iterations, as shown in Figure 4, demonstrates the convergence of the proposed learning algorithm for the MANNC evidently. Table 7 presents the performance comparison between the MANNC controller and the PIDNN controller with an equal number of trainings. As can be seen, the MANNC control results outperform the PIDNN control results in all aspects.

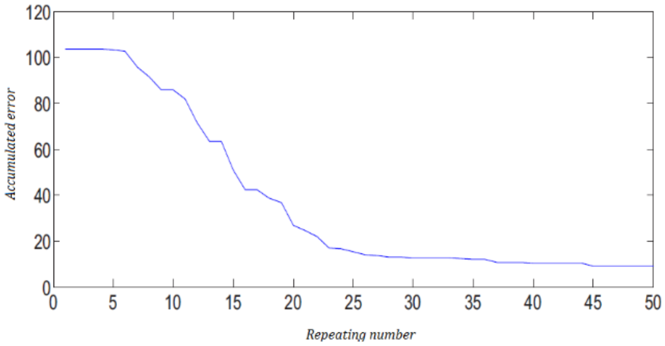

Figure 4. Case 2: accumulated error vs repeating number for MANNC.

Table 7. Case 2: MANNC vs. PIDNN.

Controller	Number of Trainings	Time of Training	Output 1 Overshoot	Output 1 Maximum Error Less than 5%	Output 2 Maximum Error Less than 5%
MANNC	50	2.39 s	0%	0.01 s	0.05 s
PIDNN	50	3.99 s	22%	0.03 s	0.08 s

6.3. Case 3: Application of MANNC on a Hybrid System

Since the MANNC has been designed as a universal controller for black-box square MIMO systems, in this case, the performance of this controller is tested on a hybrid system. By definition, a hybrid system is a dynamic system that switches between continuous states and, thus, involves both continuous and discrete behaviors. Due to sudden changes in system dynamics at the time of switching between two states, conventional control methods are usually unsuccessful for hybrid systems. In this study, the two-tank plant shown in Figure 5a was selected to test the control performance of the MANNC. The plant consists of two tanks, where tank T1 is filled by flow F_1 through a fully open valve V_1. The liquid is transferred from tank T1 into tank T2 via a connecting pipe. Valve V_2 is an on-off valve, which is either fully open or fully closed to adjust flow F_2 discretely. Similarly, flow F_3 is adjusted by another on-off valve V_3. When the behavior of the given plant is modeled, it must be considered that the liquid levels, H_1 and H_2, respectively, for both tanks will change separately when H_2 crosses level L. At $H_2 = H_1 + L$ the direction of flow through the inter-connecting pipe is reversed. Hence, for F_2, two dynamics must be modeled as:

$$F_2 = \begin{cases} k_1 \cdot V_2 \cdot \sqrt{(H_1 - H_2 + L)}\ if\ H_2 > L \\ k_1 \cdot V_2 \cdot \sqrt{(H_1)}\ if\ H_2 \le L \end{cases} \tag{68}$$

and

$$\dot{H}_1 = (F_1 - F_2)/k_3 \tag{69}$$

$$\dot{H}_2 = (F_2 - k_2 \cdot V_3 \cdot \sqrt{H_2})/k_4 \tag{70}$$

where k_1, k_2, k_3, and k_4 are constants depending on characteristic coefficients of the pipes and the cross-section areas of the tanks, and V_2 and V_3 represent the Boolean values in which “1” and “0” signify, respectively, a fully open valve and a fully closed valve.

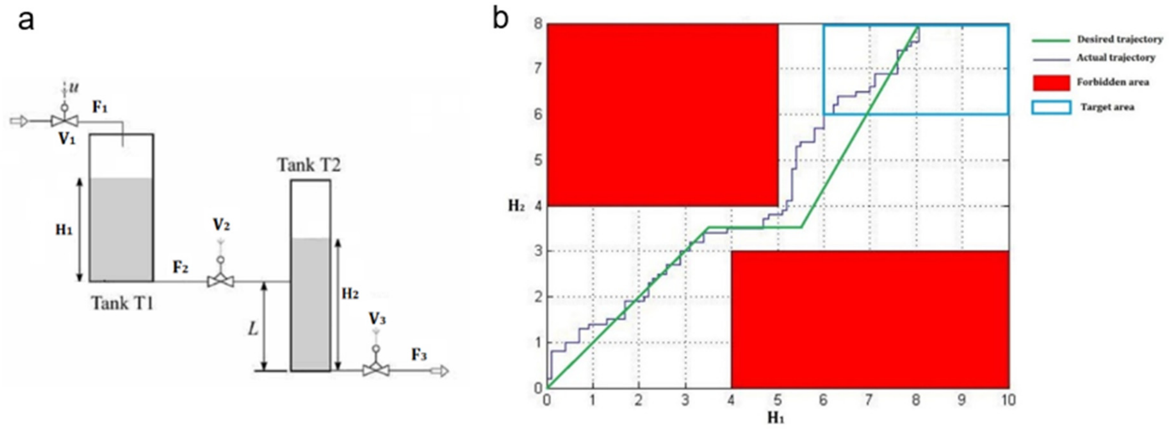

Figure 5. (**a**) Two-tank hybrid system, and (**b**) actual and desired state trajectories.

It is evident that the given plant is a nonlinear hybrid system with two inputs (V_2 and V_3) and two outputs (H_1 and H_2). The control problem is defined as: using the on-off control valves V_2 and V_3, the liquid heights (H_1 and H_2) in tanks should be derived from an initial state of $H_0 = (0.01, 0.01)$ to a target area of R_T with the following conditions:

$$H \in [0, 10] \times [0, 8] \tag{71}$$

$$H_f \in R_T = [6, 10] \times [6, 8] \tag{72}$$

where H_f represents the final desired heights located in the target area (R_T). In addition, the forbidden areas, R_{F1} and R_{F2}, are considered as:

$$H \notin R_{F1} = [0, 5] \times [4, 8] \tag{73}$$

$$H \notin R_{F2} = [4, 10] \times [0, 3] \tag{74}$$

The desired liquid heights that lead the actual liquid heights to the target area are defined as:

$$R_{H1} = 0.08r(t) \tag{75}$$

$$R_{H2} = \begin{cases} R_{H1} \, if \, R_{H1} < 3.5 \\ 3.5 \, if \, 3.5 \leq R_{H1} < 5.5 \\ 1.8R_{H1} - 6.4 \, if \, 5.5 \leq R_{H1} \leq 8 \end{cases} \tag{76}$$

where $r(t)$ is a standard ramp function. The final desired heights for both tanks in this simulation are selected as 8 meters i.e., $H_f = [8, 8]$. After repeating, simultaneously, the online learning algorithm (Table 2) and the stability criteria check 20 times, the state trajectories of the MANNC control system for the liquid heights were achieved and are shown in Figure 5b, where the forbidden areas and the desired track are also illustrated. It is clear that, by using the MANNC in this complex nonlinear hybrid control problem, the liquid height trajectory can pass the narrow zone between the forbidden areas successfully and eventually reach the target area. This is achieved due to the powerful and fast setpoint tracking property of the proposed method that can cope with changes in the system dynamics. In addition, the liquid heights of the two tanks versus time during this process are shown in Figure 6. The above results are achieved using the final weight values of the output layer of the MANNC neural network, shown in Table 8. Since the final desired liquid heights can be selected as any points in the target area, simulation studies are carried out for several points in this area and the accumulated errors are computed in a range of 285 to 310. Regardless of the location of the selected target point in the target area, the accumulated error is restricted, and the convergence is evidently achieved. This justifies the suitability and adequacy of the proposed MANNC in controlling the highly complex hybrid system.

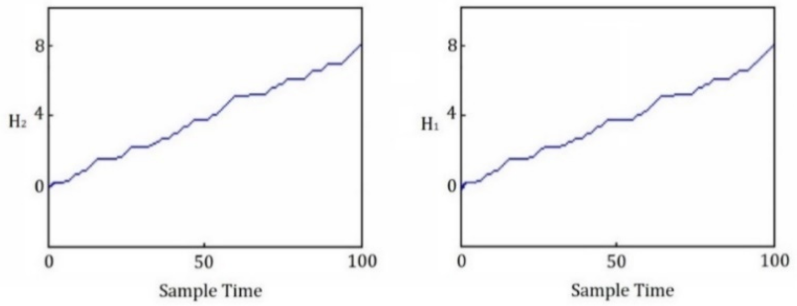

Figure 6. Height of liquid in tank T1 and tank T2.

Table 8. Final weight values of the output layer.

$w_{l,x}$	x=1	x=2	x=3	x=4	x=5	x=6
$l = 1$	3.1	1.21	2.43	−1.68	−4.34	−5.2
$l = 2$	−0.13	−1.58	4.28	0.88	2.06	3.41

7. Conclusions

A model-free MANNC capable of satisfactorily controlling nonlinear square MIMO systems with significant cross-couplings within a short period of time is presented in this paper. The MANNC uses a new auto-tune dynamic online learning algorithm with accumulated error back-propagation for the proposed neural network structure, and effectively tunes its weights to achieve the desirable control outcomes. The learning algorithm is integrated with the Lyapunov stability criteria while running and applied to the control system. The effectiveness of the proposed MANNC is validated via simulation studies on typical time-invariant and time-variant square MIMO systems. When compared with the best representative of existing counterparts (i.e., the PIDNN) for applications to both time-invariant and time-variant systems, the MANNC in the time domain is seen to provide less overshoot, less settling time, less accumulated errors, and faster setpoint tracking. The MANNC can be effectively used for several types of square MIMO control systems in industrial applications by selecting an appropriate number of samples, especially when overshoots in outputs are undesirable and fast setpoint tracking is critical. The simulation results for controlling a complex and challenging MIMO hybrid system demonstrate the superior performance of the MANNC when it deals with significant changes in the system dynamics. As a future study, adding more layers to the neural network structure of the MANNC can be considered in order to further improve the control performance, especially for highly nonlinear plants.

Author Contributions: Conceptualization, A.M.; Data curation, A.M.; Formal analysis, A.M.; Methodology, A.M. and F.H.; Project administration, A.L.; Resources, A.M. and A.L.; Supervision, F.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by Flinders University, and in part by Macquarie University.

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. Zhou, Q.; Shi, P.; Tian, Y.; Wang, M. Approximation-based adaptive tracking control for MIMO nonlinear systems with input saturation. *IEEE Trans. Cybern.* **2015**, *45*, 2119–2128. [CrossRef] [PubMed]
2. Patino, H.; Liu, D. Neural network-based model reference adaptive control system. *IEEE Trans. Syst. Man Cybern. Part B* **2000**, *30*, 198–204. [CrossRef] [PubMed]
3. Liu, M. Delayed standard neural network models for control systems. *IEEE Trans. Neural Netw.* **2007**, *18*, 1376–1391. [CrossRef] [PubMed]
4. Meng, D.; Jia, Y.; Du, J.; Yu, F. Data-driven control for relative degree systems via iterative learning. *IEEE Trans. Neural Netw.* **2011**, *22*, 2213–2225. [CrossRef]
5. Oomen, T.; van der Maas, R.; Rojas, C.R.; Hjalmarsson, H. Iterative data-driven H∞ norm estimation of multivariable systems with application to robust active vibration isolation. *IEEE Trans. Contol Syst. Technol.* **2014**, *22*, 2247–2260. [CrossRef]
6. Zhang, M.; Gan, M.-G. Data-driven adaptive optimal control for linear systems with structured time-varying uncertainty. *IEEE Access* **2019**, *7*, 9215–9224. [CrossRef]
7. Hay, M.; Miklau, G.; Jensen, D.; Towsley, D.; Weis, P. Resisting structural re-identification in anonymized social networks. *Proc. VLDB Endow.* **2008**, *1*, 102–114. [CrossRef]
8. Donald, C.; Charles, K.A.; Ronald, K.J. *Control Systems*; McGraw-Hill Education: New York, NY, USA, 2005.
9. Fliess, M.; Join, C. Model-free control. *Int. J. Control* **2013**, *86*, 2228–2252. [CrossRef]
10. Lafont, F.; Balmat, J.-F.; Pessel, N.; Fliess, M. A model-free control strategy for an experimental greenhouse with an application to fault accommodation. *Comput. Electron. Agric.* **2015**, *110*, 139–149. [CrossRef]
11. Madadi, E.; Söffker, D. Model-free approaches applied to the control of nonlinear systems: A brief survey with special attention to intelligent PID iterative learning control. In Proceedings of the ASME 2015 Dynamic Systems and Control Conference, Columbus, OH, USA, 28–30 October 2015. [CrossRef]
12. Radac, M.-B.; Precup, R.-E.; Petriu, E.M. Model-free primitive-based iterative learning control approach to trajectory tracking of MIMO systems with experimental validation. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 2925–2938. [CrossRef]
13. Hou, Z.; Liu, S.; Tian, T. Lazy-learning-based data-driven model-free adaptive predictive control for a class of discrete-time nonlinear systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 1914–1928. [CrossRef] [PubMed]
14. Lu, W.; Zhu, P.; Ferrari, S. A hybrid-adaptive dynamic programming approach for the model-free control of nonlinear switched systems. *IEEE Trans. Autom. Control* **2015**, *61*, 3203–3208. [CrossRef]
15. Wang, Z.; Liu, L.; Zhang, H. Neural network-based model-free adaptive fault-tolerant control for discrete-time nonlinear systems with sensor fault. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 2351–2362. [CrossRef]
16. Gao, B.; Cao, R.; Hou, Z.; Zhou, H. Model-free adaptive MIMO control algorithm application in polishing robot. In Proceedings of the 6th Data Driven Control and Learning Systems Conference (DDCLS), Chongqing, China, 26–27 May 2017. [CrossRef]
17. Luo, B.; Liu, D.; Wu, H.-N.; Wang, D.; Lewis, F.L. Policy gradient adaptive dynamic programming for data-based optimal control. *IEEE Trans. Cybern.* **2016**, *47*, 3341–3354. [CrossRef]
18. Mehrafrooz, A.; He, F. Introducing a model-free adaptive neural network auto-tuned control method for nonlinear SISO systems. In Proceedings of the 2018 IEEE International Conference on Information and Automation, Wuyishan, China, 11–13 August 2018. [CrossRef]
19. Safaei, A.; Mahyuddin, M.N. Adaptive model-free control based on an ultra-local model with model-free parameter estimations for a generic SISO system. *IEEE Access* **2018**, *6*, 4266–4275. [CrossRef]
20. Roshani, G.H.; Nazemi, E.; Feghhi, S.A.H.; Setayeshi, S. Flow regime identification and void fraction prediction in two-phase flows based on gamma ray attenuation. *Measurement* **2015**, *62*, 25–32. [CrossRef]

21. Roshani, G.; Nazemi, E.; Roshani, M. Intelligent recognition of gas-oil-water three-phase flow regime and determination of volume fraction using radial basis function. *Flow Meas. Instrum.* **2017**, *54*, 39–45. [CrossRef]
22. Roshani, G.; Nazemi, E. Intelligent densitometry of petroleum products in stratified regime of two phase flows using gamma ray and neural network. *Flow Meas. Instrum.* **2017**, *58*, 6–11. [CrossRef]
23. Roshani, G.; Nazemi, E.; Feghhi, S. Investigation of using 60 Co source and one detector for determining the flow regime and void fraction in gas–liquid two-phase flows. *Flow Meas. Instrum.* **2016**, *50*, 73–79. [CrossRef]
24. Jamshidi, M.B.; Lalbakhsh, A.; Alibeigi, N.; Soheyli, M.R.; Oryani, B.; Rabbani, N. Socialization of industrial robots: An innovative solution to improve productivity. In Proceedings of the 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 1–3 November 2018. [CrossRef]
25. Jamshidi, M.B.; Alibeigi, N.; Lalbakhsh, A.; Roshani, S. An ANFIS approach to modeling a small satellite power source of NASA. In Proceedings of the 2019 IEEE 16th International Conference on Networking, Sensing and Control, Banff, AB, Canada, 9–11 May 2019. [CrossRef]
26. Jamshidi, M.B.; Lalbakhsh, A.; Mohamadzade, B.; Siahkamari, H.; Mousavi, S.M.H. A novel neural-based approach for design of microstrip filters. *AEU-Int. J. Electron. Commun.* **2019**, *110*, 152847. [CrossRef]
27. Jamshidi, M.; Lalbakhsh, A.; Lotfi, S.; Siahkamari, H.; Mohamadzade, B.; Jalilian, J. A neuro-based approach to designing a Wilkinson power divider. *Int. J. RF Microwav. Comput.-Aided Eng.* **2020**, *30*, e22091. [CrossRef]
28. Sattari, M.A.; Roshani, G.H.; Hanus, R.; Nazemi, E. Applicability of time-domain feature extraction methods and artificial intelligence in two-phase flow meters based on gamma-ray absorption technique. *Measurement* **2021**, *168*, 108474. [CrossRef]
29. Roshani, G.; Hanus, R.; Khazaei, A.; Zych, M.; Nazemi, E.; Mosorov, V. Density and velocity determination for single-phase flow based on radiotracer technique and neural networks. *Flow Meas. Instrum.* **2018**, *61*, 9–14. [CrossRef]
30. Roshani, M.; Sattari, M.A.; Ali, P.J.M.; Roshani, G.H.; Nazemi, B.; Corniani, E.; Nazemi, E. Application of GMDH neural network technique to improve measuring precision of a simplified photon attenuation based two-phase flowmeter. *Flow Meas. Instrum.* **2020**, *75*, 101804. [CrossRef]
31. Roshani, M.; Phan, G.; Roshani, G.H.; Hanus, R.; Nazemi, B.; Corniani, E.; Nazemi, E. Combination of X-ray tube and GMDH neural network as a nondestructive and potential technique for measuring characteristics of gas-oil–water three phase flows. *Measurement* **2021**, *168*, 108427. [CrossRef]
32. Roshani, G.; Feghhi, S.; Mahmoudi-Aznaveh, A.; Nazemi, E.; Adineh-Vand, A. Precise volume fraction prediction in oil–water–gas multiphase flows by means of gamma-ray attenuation and artificial neural networks using one detector. *Measurement* **2014**, *51*, 34–41. [CrossRef]
33. Roshani, M.; Phan, G.T.; Ali, P.J.M.; Roshani, G.H.; Hanus, R.; Duong, T.; Corniani, E.; Nazemi, E.; Kalmoun, E.M. Evaluation of flow pattern recognition and void fraction measurement in two phase flow independent of oil pipeline's scale layer thickness. *Alex. Eng. J.* **2021**, *60*, 1955–1966. [CrossRef]
34. Lalbakhsh, A.; Afzal, M.U.; Esselle, K.P. Multiobjective particle swarm optimization to design a time-delay equalizer metasurface for an electromagnetic band-gap resonator antenna. *IEEE Antennas Wirel. Propag. Lett.* **2016**, *16*, 912–915. [CrossRef]
35. Lalbakhsh, A.; Afzal, M.U.; Esselle, K.P.; Smith, S.L. A fast design procedure for quadrature reflection phase. In Proceedings of the 2017 IEEE Progress in Electromagnetics Research Symposium-Fall (PIERS-FALL), Singapore, 19–22 November 2017.
36. Lalbakhsh, A.; Afzal, M.U.; Esselle, K.P.; Smith, S. Design of an artificial magnetic conductor surface using an evolutionary algorithm. In Proceedings of the 2017 IEEE International Conference on Electromagnetics in Advanced Applications (ICEAA), Verona, Italy, 11–15 September 2017.
37. Roshani, G.; Nazemi, E.; Roshani, M. Usage of two transmitted detectors with optimized orientation in order to three phase flow metering. *Measurement* **2017**, *100*, 122–130. [CrossRef]
38. Nazemi, E.; Roshani, G.H.; Feghhi, S.A.H.; Setayeshi, S.; Zadeh, E.E.; Fatehi, A. Optimization of a method for identifying the flow regime and measuring void fraction in a broad beam gamma-ray attenuation technique. *Int. J. Hydrog. Energy* **2016**, *41*, 7438–7444. [CrossRef]
39. Karami, A.; Roshani, G.H.; Nazemi, E.; Roshani, S. Enhancing the performance of a dual-energy gamma ray based three-phase flow meter with the help of grey wolf optimization algorithm. *Flow Meas. Instrum.* **2018**, *64*, 164–172. [CrossRef]
40. Karambasti, B.M.; Ghodrat, M.; Ghorbani, G.; Lalbakhsh, A.; Behnia, M. Design methodology and multi-objective optimization of small-scale power-water production based on integration of Stirling engine and multi-effect evaporation desalination system. *Desalination* **2022**, *526*, 115542. [CrossRef]
41. Lalbakhsh, P.; Zaeri, B.; Lalbakhsh, A. An improved model of ant colony optimization using a novel pheromone update strategy. *IEICE Trans. Inf. Syst.* **2013**, *96*, 2309–2318. [CrossRef]
42. Wang, Y.; Zhang, Y.; Fu, Y.; Chai, T.; Fu, J. Adaptive decoupling switching control based on generalised predictive control. *IET Control Theory Appl.* **2012**, *6*, 1828–1841. [CrossRef]
43. Peng, K.; Fan, D.; Yang, F.; Gou, L.; Lv, W. A frequency domain decoupling method and multivariable controller design for turbofan engines. *IEEE Access* **2017**, *5*, 27757–27766. [CrossRef]
44. Nguyen, T.N.; Su, S.; Nguyen, H.T. Neural network based diagonal decoupling control of powered wheelchair systems. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2013**, *22*, 371–378. [CrossRef]
45. Cong, S.; Liang, Y. PID-like neural network nonlinear adaptive control for uncertain multivariable motion control systems. *IEEE Trans. Ind. Electron.* **2009**, *56*, 3872–3879. [CrossRef]

46. Hwang, C.-L.; Jan, C. Recurrent-neural-network-based multivariable adaptive control for a class of nonlinear dynamic systems with time-varying delay. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *27*, 388–401. [CrossRef]
47. Scott, G.; Shavlik, J.; Ray, W. Refining PID controllers using neural networks. *Adv. Neural Inf. Process. Syst.* **1991**, *4*, 555–565. [CrossRef]
48. Merabet, A.; Tanvir, A.A.; Beddek, K. Speed control of sensorless induction generator by artificial neural network in wind energy conversion system. *IET Renew. Power Gener.* **2016**, *10*, 1597–1606. [CrossRef]
49. Yang, B.-J.; Calise, A.J. Adaptive control of a class of nonaffine systems using neural networks. *IEEE Trans. Neural Netw.* **2007**, *18*, 1149–1159. [CrossRef] [PubMed]
50. Tee, K.P.; Ge, S.S.; Tay, F.E.H. Adaptive neural network control for helicopters in vertical flight. *IEEE Trans. Control Syst. Technol.* **2008**, *16*, 753–762. [CrossRef]
51. Park, J.-H.; Kim, S.-H.; Moon, C.-J. Adaptive neural control for strict-feedback nonlinear systems without backstepping. *IEEE Trans. Neural Netw.* **2009**, *20*, 1204–1209. [CrossRef] [PubMed]
52. Liu, Y.-J.; Tong, S.-C.; Wang, D.; Li, T.-S.; Chen, C.L.P. Adaptive neural output feedback controller design with reduced-order observer for a class of uncertain nonlinear SISO systems. *IEEE Trans. Neural Netw.* **2011**, *22*, 1328–1334. [CrossRef]
53. Zhang, J.; Ge, S.S.; Lee, T.H. Output feedback control of a class of discrete MIMO nonlinear systems with triangular form inputs. *IEEE Trans. Neural Netw.* **2005**, *16*, 1491–1503. [CrossRef] [PubMed]
54. Zhang, T.; Ge, S.S. Adaptive neural network tracking control of MIMO nonlinear systems with unknown dead zones and control directions. *IEEE Trans. Neural Netw.* **2009**, *20*, 483–497. [CrossRef]
55. Chen, M.; Ge, S.S.; How, B.V.E. Robust adaptive neural network control for a class of uncertain MIMO nonlinear systems with input nonlinearities. *IEEE Trans. Neural Netw.* **2010**, *21*, 796–812. [CrossRef]
56. Yang, Q.; Yang, Z.; Sun, Y. Universal neural network control of MIMO uncertain nonlinear systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 1163–1169. [CrossRef]
57. Chen, Z.; Ge, S.S.; Zhang, Y.; Li, Y. Adaptive neural control of MIMO nonlinear systems with a block-triangular pure-feedback control structure. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 2017–2029. [CrossRef]
58. Meng, W.; Yang, Q.; Sun, Y. Adaptive neural control of nonlinear MIMO systems with time-varying output constraints. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *26*, 1074–1085. [CrossRef]
59. Yang, Q.; Jagannathan, S.; Sun, Y. Robust integral of neural network and error sign control of MIMO nonlinear systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 3278–3286. [CrossRef] [PubMed]
60. Ronald, J. Neural network applications. In *Electronic Engine Control Technologies*; SAE: Warrendale, PA, USA, 2004; p. 661.
61. Yong, Z.; Hai-Bo, Z.; Tian-Qi, L. PIDNN decoupling control of boiler combustion system based on MCS. In Proceedings of the 2017 29th Chinese Control and Decision Conference (CCDC), Chongqing, China, 28–30 May 2017. [CrossRef]
62. Hernandez-Alvarado, R.; Garcia-Valdovinos, L.G.; Salgado-Jimenez, T.; Gomez-Espinosa, A.; Navarro, F.F. Self-tuned PID control based on backpropagation Neural Networks for underwater vehicles. In Proceedings of the OCEANS 2016 MTS/IEEE Monterey Conference, Monterey, CA, USA, 19–23 September 2016. [CrossRef]
63. Jing, X.; Cheng, L. An optimal PID control algorithm for training feedforward neural networks. *IEEE Trans. Ind. Electron.* **2012**, *60*, 2273–2283. [CrossRef]
64. Shu, H.; Xu, Y.-K. Application of additional momentum in PID neural network. In Proceedings of the 2014 4th IEEE International Conference on Information Science and Technology, Shenzhen, China, 26–28 April 2014.
65. Meng, L.; Zou, Z.-Y.; Wang, Z.-Z.; Gui, X.-J.; Yu, M. Design of an improved PID neural network controller based on particle swarm optimazation. In Proceedings of the 2015 IEEE Chinese Automation Congress (CAC), Wuhan, China, 27–29 November 2015. [CrossRef]
66. Teng, W.-F.; Pan, H.-P.; Ren, J. Neural network PID decoupling control based on chaos particle swarm optimization. In Proceedings of the 33rd IEEE Chinese Control Conference, Nanjing, China, 28–30 July 2014. [CrossRef]
67. Tian, Z.; Guo, H.; Ding, X.; He, X. A PID neural network control for position servo system with gear box at variable load. In Proceedings of the 2016 IEEE Vehicle Power and Propulsion Conference (VPPC), Hangzhou, China, 17–20 October 2016. [CrossRef]
68. Bahri, N.; Atig, A.; Abdennour, R.B.; Druaux, F.; Lefebvre, D. Multivariable adaptive neural control based on multimodel emulator for nonlinear square MIMO systems. In Proceedings of the 2014 IEEE 11th International Multi-Conference on Systems, Signals & Devices (SSD14), Barcelona, Spain, 11–14 February 2014.
69. Saerens, M.; Soquet, A. Neural controller based on back-propagation algorithm. *IEE Proc. F Radar Signal Process.* **1991**, *138*, 55–62. [CrossRef]
70. Goodfellow, I.; Bengio, Y.; Courville, A. 6.5 Back-propagation and other differentiation algorithms. In *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
71. Cilimkovic, M. *Neural Networks and Back Propagation Algorithm*; Institute of Technology Blanchardstown: Dublin, Ireland, 2015; Volume 15.
72. Merayo, N.; Juárez, D.; Aguado, J.C.; De Miguel, I.; Durán, R.J.; Fernández, P.; Lorenzo, R.M.; Abril, E.J. PID controller based on a self-adaptive neural network to ensure QoS bandwidth requirements in passive optical networks. *J. Opt. Commun. Netw.* **2017**, *9*, 433–445. [CrossRef]

73. Parandin, F. Ultra-compact terahertz all-optical logic comparator on GaAs photonic crystal platform. *Opt. Laser Technol.* **2021**, *144*, 107399. [CrossRef]
74. Parandin, F.; Heidari, F.; Rahimi, Z.; Olyaee, S. Two-dimensional photonic crystal Biosensors: A review. *Opt. Laser Technol.* **2021**, *144*, 107397. [CrossRef]
75. Abdollahi, M.; Parandin, F. A novel structure for realization of an all-optical, one-bit half-adder based on 2D photonic crystals. *J. Comput. Electron.* **2019**, *18*, 1416–1422. [CrossRef]
76. Saghaei, H.; Zahedi, A.; Karimzadeh, R.; Parandin, F. Line defects on As2Se3-chalcogenide photonic crystals for the design of all-optical power splitters and digital logic gates. *Superlattices Microstruct.* **2017**, *110*, 133–138. [CrossRef]
77. Karkhanehchi, M.M.; Parandin, F.; Zahedi, A. Design of an all optical half-adder based on 2D photonic crystals. *Photon-Netw. Commun.* **2017**, *33*, 159–165. [CrossRef]
78. Vahdati, A.; Parandin, F. Antenna patch design using a photonic crystal substrate at a frequency of 1.6 THz. *Wirel. Pers. Commun.* **2019**, *109*, 2213–2219. [CrossRef]
79. Dehghani, K.; Karimi, G.; Lalbakhsh, A.; Maki, S. Design of lowpass filter using novel stepped impedance resonator. *Electron. Lett.* **2014**, *50*, 37–39. [CrossRef]
80. Lalbakhsh, A.; Mohamadpour, G.; Roshani, S.; Ami, M.; Roshani, S.; Sayem, A.S.M.; Alibakhshikenari, M.; Koziel, S. Design of a compact planar transmission line for miniaturized rat-race coupler with harmonics suppression. *IEEE Access* **2021**, *9*, 129207–129217. [CrossRef]
81. Roshani, S. A compact microstrip low-pass filter with ultra wide stopband using compact microstrip resonant cells. *Int. J. Microw. Wirel. Technol.* **2017**, *9*, 1023–1027. [CrossRef]
82. Heshmati, H.; Roshani, S. A miniaturized lowpass bandpass diplexer with high isolation. *AEU-Int. J. Electron. Commun.* **2018**, *87*, 87–94. [CrossRef]
83. Pirasteh, A.; Roshani, S.; Roshani, S. A modified class-F power amplifier with miniaturized harmonic control circuit. *AEU-Int. J. Electron. Commun.* **2018**, *97*, 202–209. [CrossRef]
84. Roshani, S.; Dehghani, K.; Roshani, S. A Lowpass Filter Design Using Curved and Fountain Shaped Resonators. *Frequenz* **2019**, *73*, 267–272. [CrossRef]
85. Lotfi, S.; Roshani, S.; Roshani, S. Design of a miniaturized planar microstrip Wilkinson power divider with harmonic cancellation. *Turk. J. Electr. Eng. Comput. Sci.* **2020**, *28*, 3126–3136.
86. Bavandpour, S.K.; Roshani, S.; Pirasteh, A.; Roshani, S.; Seyedi, H. A compact lowpass-dual bandpass diplexer with high output ports isolation. *AEU-Int. J. Electron. Commun.* **2021**, *135*, 153748. [CrossRef]
87. Moloudian, G.; Bahrami, S.; Hashmi, R.M. A microstrip lowpass filter with wide tuning range and sharp roll-off response. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 2953–2957. [CrossRef]
88. Lalbakhsh, A.; Afzal, M.U.; Esselle, K.P.; Smith, S.L. All-metal wideband frequency-selective surface bandpass filter for TE and TM polarizations. *IEEE Trans. Antennas Propag.* **2021**, *70*, 1. [CrossRef]
89. Parandin, F.; Moayed, M. Designing and simulation of 3-input majority gate based on two-dimensional photonic crystals. *Optik* **2020**, *216*, 164930. [CrossRef]
90. Parandin, F.; Kamarian, R.; Jomour, M. A novel design of all optical half-subtractor using a square lattice photonic crystals. *Opt. Quantum Electron.* **2021**, *53*, 114. [CrossRef]
91. Paul, G.S.; Mandal, K.; Lalbakhsh, A. Single-layer ultra-wide stop-band frequency selective surface using interconnected square rings. *AEU-Int. J. Electron. Commun.* **2021**, *132*, 153630. [CrossRef]
92. Roshani, S.; Roshani, S. Design of a high efficiency class-F power amplifier with large signal and small signal measurements. *Measurement* **2020**, *149*, 106991. [CrossRef]
93. Parandin, F.; Mahtabi, N. Design of an ultra-compact and high-contrast ratio all-optical NOR gate. *Opt. Quantum Electron.* **2021**, *53*, 666. [CrossRef]
94. Bahrami, S.; Moloudian, G.; Miri-Rostami, S.R.; Bjorninen, T. Compact microstrip antennas with enhanced bandwidth for the implanted and external subsystems of a wireless retinal prosthesi. *IEEE Trans. Antennas Propag.* **2020**, *69*, 2969–2974. [CrossRef]
95. Roshani, G.H.; Roshani, S.; Nazemi, E.; Roshani, S. Online measuring density of oil products in annular regime of gas-liquid two phase flows. *Measurement* **2018**, *129*, 296–301. [CrossRef]
96. Roshani, M.; Phan, G.; Faraj, R.H.; Phan, N.-H.; Roshani, G.H.; Nazemi, B.; Corniani, E.; Nazemi, E. Proposing a gamma radiation based intelligent system for simultaneous analyzing and detecting type and amount of petroleum by-products. *Nucl. Eng. Technol.* **2021**, *53*, 1277–1283. [CrossRef]
97. Roman, R.-C.; Rădac, M.-B.; Precup, R.-E.; Stinean, A.-I. Two data-driven control algorithms for a MIMO aerodynamic system with experimental validation. In Proceedings of the 2015 19th International Conference on System Theory, Control and Computing (ICSTCC), Cheile Gradistei, Romania, 14–16 October 2015. [CrossRef]
98. Kalpana, D.; Thyagarajan, T.; Gokulraj, N. Modeling and control of non-square MIMO system using relay feedback. *ISA Trans.* **2015**, *59*, 408–417. [CrossRef]
99. Phillips, S.F.; Seborg, D.E. Conditions that guarantee no overshoot for linear systems. *Int. J. Control* **1988**, *47*, 1043–1059. [CrossRef]