

## Improved Set-point Tracking Control of an Unmanned Aerodynamic MIMO System Using Hybrid Neural Networks

David Mohammed Ezekiel<sup>1</sup>, Ravi Samikannu<sup>2</sup> and Oduetse Matsebe<sup>3</sup>

<sup>\*</sup>Department of Electrical, Computer and Telecommunications Engineering, Botswana International University of Science and Technology (BIUST), Private Bag 16, Palapye, Botswana, <sup>β</sup>Department of Mechanical, Energy and Industrial Engineering, Botswana International University of Science and Technology (BIUST), Private Bag 16, Palapye, Botswana.

**ABSTRACT** Artificial neural networks (ANN), an Artificial Intelligence (AI) technique, are both bio-inspired and nature-inspired models that mimic the operations of the human brain and the central nervous system that is capable of learning. This paper is based on a system that optimizes the performance of an uncertain unmanned nonlinear Multi-Input Multi-Output (MIMO) aerodynamic plant called Twin Rotor MIMO System (TRMS). The pitch and yaw angles which are challenging to control and optimize in practice, are being used as the input to the Nonlinear Auto-Regressive with eXogenous (NARX) model, and eventually trained. The training features use the Matlab Deep Learning Toolbox. The NARX structure has its core in the neural networks' architecture. Data is collected from the TRMS testbed which is used to train the network. ANN as a Hybrid intelligent control strategy of ANN in combination with Pattern Search and Genetic Algorithm, is then utilized to optimize the parameters of the neural networks. At the end it was validated, tested and the optimized system run in simulation and compared with other intelligent and conventional controllers, with the proposed controller outperforming them, giving a very fast-tracking control, stable and optimal performance that satisfactorily met all our design requirements.

### KEYWORDS

Artificial neural network  
Nonlinear auto-regressive with eXogenous  
Twin rotor MIMO system  
Multi-input multi-output  
Aerodynamic  
Unmanned helicopter model

### INTRODUCTION

The modelling, optimization and control of rigid bodies and flexible structures/systems (Ahmad *et al.* 2000a,b; Moness and Daa-Eldeen 2017) (such as plates, shells, beams, frames, etc.) are increasingly gaining a considerable attention from researchers globally (Tavakolpour *et al.* 2010; Nasir and Tokhi 2014; TRahman *et al.* 2019). These bodies and structures are highly essential manufacturing elements in electro-mechanical, civil, marine and aerospace engineering. In this paper, the application of Feedforward Neural Networks (NN) is applied to the beam of a nonlinear uncertain system called, the TRMS. It is a highly nonlinear, high-order, complex system (Moness and Daa-Eldeen 2017; Toha and Tokhi 2009;

Alam *et al.* 2004; Toha and Tokhi 2010; Ahmad *et al.* 2016) the nonlinearities and complexities emanate from the cross-couplings between the twin-rotors. These pose as a serious challenge to effectively model, control and optimize. The modelling, control and optimization of the TRMS can be carried out in either the model-based/model-driven or data-driven approaches. The data-driven (i.e. black-box modelling) approaches for which this paper is based, requires some input/output dataset (Ljung and Gunnarsson 1990)[obtained through system simulation, offline and/or online. With this dataset, the system is identified through System Identification (SI) techniques.

The drawback of SI is that it has demonstrated a computational inadequacy with nonlinear systems, but much less uncomplicated with linear systems (Ahmad *et al.* 2000a). In spite of this, it is still indispensable and a powerful design strategy, especially if the system can be linearized about some equilibrium points. To use SI methodologies, require training of the network used in the design process, which can be parametric or non-parametric.

**Manuscript received:** 14 November 2023,

**Revised:** 16 December 2023,

**Accepted:** 21 December 2023.

<sup>1</sup>ed18100190@studentmail.biust.ac.bw

<sup>2</sup>ravis@biust.ac.bw (Corresponding author).

<sup>3</sup>matsebe@biust.ac.bw

The non-parametric SI (which is of interest here) involves the use of Artificial Intelligence, such as ANN (Sjöberg *et al.* 1994; Chu *et al.* 1990) or an Adaptive Neuro-Fuzzy Inference System (ANFIS) (Castillo *et al.* 2006). For brevity, ANN is simply referred to as NN. However, they suffer from being caught in a local minimum and a very slow convergence resulting from system complexities of nonlinear systems. To solve these problems metaheuristic methods are employed for faster convergence optimization. With this, the solution being trapped in a local minimum or local minima is prevented, thus guaranteeing an accurate solution (TRahman *et al.* 2019).

A number of these metaheuristics' approaches have been successfully used in the training of ANNs in engineering and scientific applications. Some of these methods include Symbiotic Organisms Search (SOS) scheme employed to train a feed forward NN to solve a classification problem (Wu *et al.* 2016), Genetic Algorithm (GA) (Sivadasan and Shiney 2023) Harmony Search, Simulated Annealing and Differential Evolution (DE) (Rere *et al.* 2016), a hybrid algorithm composed of Particle Swarm Optimization (PSO) used for optimization of a Convolutional NN to also solve a classification problem (Yaghini *et al.* 2013), ANN models trained for stock market price predictions/forecasts (Ghasemiyeh *et al.* 2017), the newly developed Stochastic Fractal Search Algorithm SFS by Salimi (Salimi 2015) and used to train ANNs (Mosbah and El-Hawary 2017; Khishe *et al.* 2018). Also, the successful applications of ANNs in estimating the nonlinear dynamics of dynamical systems have been reported for kinematics in (Xia and Wang 2001; Yoo *et al.* 2006; Abbas and Liu 2022) for dynamics in (Lin and Goldenberg 2001; El-Fakdi and Carreras 2013) and for control in (Xia and Wang 2001; Wai 2003; Palepogu and Mahapatra 2023)

Due to the extreme and profoundly serious (i.e., massive) nonlinearities the control of Unmanned Aerial Vehicles (UAVs), of which class the TRMS falls, is a challenging one (Agand *et al.* 2017). Rahideh *et al.* proposed a Model Inversion Control law to control a 1-DOF pitch model of the TRMS using ANN (Rahideh *et al.* 2012a). The ANN was used adaptively to tune the system model. The obtained control law was consequently used to achieve control and tracking. The scheme used an adaptive nonlinear iterative learning control (Patan and Patan 2023; Bensidhoum *et al.* 2023) for compensation of the errors due to modelling, thereby identifying the system. The use of the NARX neural networks based on a Back Propagation (BP) algorithm for network training was proposed by Tijani *et al.* in (Tijani *et al.* 2014) to solve a multi-objective optimization problem. The algorithm used a multi (or many)-objective DE algorithms to identify and control the nonlinear TRMS using real-time data from experiments. The motivation of this work stems from the fact that unlike linear systems and processes which a tremendous depth of knowledge exists on the control of such systems and processes, for nonlinear control systems are quite very challenging. Since most or nearly all control systems are nonlinear attention has shifted by researchers and control engineers globally on development of control techniques, methodologies and strategies to address these systems. Nowadays the research direction has shifted focus on Artificial Intelligence (AI) and Computational Intelligence (CI) which are at the cutting edge. From studies on the use of ANN, an AI-based technique, developing a controller using NN structure is quite very difficult, because of the dynamic nature of such systems, where the states are also dynamic in nature and constantly changing. This pose as a serious challenge to control such a system.

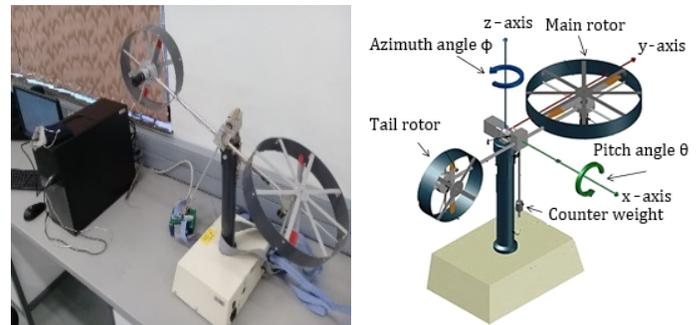
In this study, the use of Deep Neural Networks architecture using NARX Shallow NN for the ANN training is used to identify,

optimize and control the nonlinear TRMS lab-scaled helicopter. The NARX model is used here identifies/ capture the nonlinear dynamics of the nonlinear TRMS testbed. The NARX network is a feedforward neural network composed of 2 layers, with a sigmoid transfer or activation function in the hidden layer and a linear transfer function in the outer layer. Tapped delay lines are also used by the network to store previous values of the input and output sequences. Here, the outputs are fed back into the inputs through the delay lines, since if  $y(t)$  is the output, then  $y(t)$  is a function of

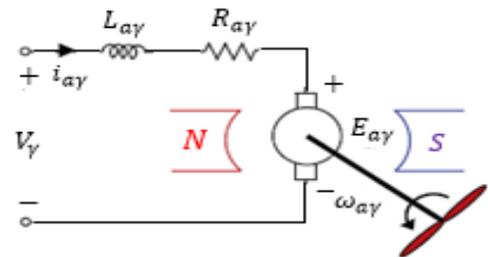
$$y(t-1), y(t-2), \dots, y(t-d). \quad (1)$$

The learning rules algorithms employed mostly are the Bayesian Regularization (trainbr), Levenberg-Marquardt (trainlm), and the Scaled Conjugate Gradient (trainscg). The first two algorithms are based on the Jacobian calculations while the last training method is based on the gradient calculations. In this paper, the 2 inputs (elevation and azimuth) and the outputs/target vectors (pitch and yaw) are composed of 181 datasets each, at random, roughly divided into 70% for the training phase, 15% for the validation phase and 15% for the test phase to generalize the network. 2 different set of numbers of hidden neurons of 10 and 1000 were used, with a tapped delay line of 4. The paper is organized as follows: Section 2 presents the experimental arrangements as well as the governing equations of motion; Section 3 presents the NN architecture and theoretical background; Section 4 gives the training results and final simulations.

## SETUP OF EXPERIMENT



**Figure 1** (a) The Real-world experimental setup at the Botswana International University of Science and Technology (BIUST), with the beam inclined at  $60^\circ$  to the horizontal at rest and showing I/O communication cables (b) Schematic graphic (Abdulwahhab and Abbas 2017; Ezekiel *et al.* 2020, 2021).



**Figure 2** The electrical circuit connection of the DC motor of the TRMS

The model of the DC motors is given in (Darus and Lokaman 2010; Rahideh et al. 2008) as:

$$\frac{di_{a\gamma}}{dt} = \frac{1}{L_{a\gamma}} (V_{\gamma} - E_{a\gamma} - R_{a\gamma}i_{a\gamma}) \quad (2)$$

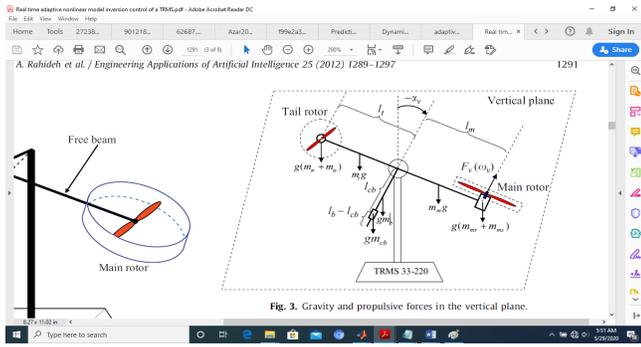
$$E_{a\gamma} = K_{a\gamma}\phi_{\gamma}\omega_{\gamma} \quad (3)$$

$$J_{\gamma r} = c_{\gamma aw} = -G_s \quad (4)$$

$$T_{e\gamma} = K_{t\gamma}i_{a\gamma} \quad (5)$$

$$T_{L\gamma} = K_{t\gamma}i_{o\gamma}|w_{\gamma}| \quad (6)$$

Where,  $V_{\gamma}$  is the control voltage input to either the vertical or the horizontal channel,  $E_{a\gamma}$ ,  $i_{a\gamma}$ ,  $R_{a\gamma}$ , and  $L_{a\gamma}$  are respectively the e.m.f, current, resistance, and inductance in the armature of the main/tail motor;  $k_{a\gamma}$  and  $k_{t\gamma}$  are constants;  $\phi_{\gamma}$  is the flux linkages;  $\omega_{\gamma}$  is the angular velocity of either the main or tail motor,  $T_{e\gamma}$ ,  $T_{L\gamma}$  are the magnetic torque and load torque respectively in the main/tail motor;  $J_{\gamma r}$ ,  $B_{\gamma r}$  are the moments of inertias and viscous friction damping coefficients of the rotors in the main/tail motors.



**Figure 3** Planar (vertical plane) representation of the TRMS, showing the gravity and propulsive forces (Rahideh et al. 2012b)

### Governing Equations of Motion of the TRMS

Being a dynamical system that has rotational motion, Newtonian mechanics for rotational dynamics or Lagrangian mechanics may be used to develop the dynamic equations of motion. Based on Newtonian mechanics for rotational dynamics, the dynamic equations of motion (using Newton's laws of motion for rotational dynamics) (Coelho et al. 2007a, 2008, 2007b) of the TRMS, representing the flight in the pitch (or vertical) plane and the yaw (or horizontal) plane are respectively given by:

$$\begin{aligned} \frac{dS_v}{dt} &= \frac{M_v}{J_v} = \frac{l_m F_v(\omega_m) - \Omega_v K_v + g[(A - B) \cos \alpha_v - C \sin \alpha_v]}{J_v} \\ &\quad - \frac{1}{2} \frac{\Omega_h^2 (A + B + C) \sin 2\alpha_v}{J_v} \\ &= \frac{l_m F_v(\omega_m) + g[(A - B) \cos \alpha_v - C \sin \alpha_v] - T_{\text{fric},v}}{J_v} \quad (8) \end{aligned}$$

$$\frac{dS_h}{dt} = \frac{M_h}{J_h} = \frac{l_t F_h(\omega_t) \cos \alpha_v - \Omega_h K_h}{D \sin^2 \alpha_v + E \cos^2 \alpha_v + F} = \frac{l_t F_h(\omega_t) \cos \alpha_v - \Omega_h K_h}{J_h} \quad (8)$$

where,  $\Omega_v$  and  $\Omega_h$  are the angular/rotational velocities of the rotors for the pitch and yaw orientations, respectively, given by:

$$\Omega_v = \frac{d\alpha_v}{dt} = S_v + \frac{J_{tr}\omega_t}{J_v} \quad (9)$$

$$\Omega_h = \frac{d\alpha_h}{dt} = S_h + \frac{J_{mr}\omega_m \cos \alpha_v}{J_h} = S_h + \frac{J_{mr}\omega_m \cos \alpha_v}{D \sin^2 \alpha_v + E \cos^2 \alpha_v + F} \quad (10)$$

where A, B, C, D, E, F are constants, and are given by:

$$A = \left(\frac{m_t}{2} + m_{tr} + m_{ts}\right) l_t; \quad B = \left(\frac{m_m}{2} + m_{mr} + m_{ms}\right) l_m; \quad C = \frac{m_b}{2} l_b + m_{cb} l_b; \quad D = \frac{m_b}{3} l_b^2 + m_{cb} l_{cb}^2;$$

$$E = \left(\frac{m_m}{3} + m_{mr} + m_{ms}\right) l_m^2 + \left(\frac{m_t}{3} + m_{tr} + m_{ts}\right) l_t^2;$$

$$F = m_{ms} r_{ms}^2 + \frac{m_{ts}}{2} r_{ts}^2$$

The aerodynamic propulsive forces,  $F_v(\omega_m)$  and  $F_h(\omega_t)$ , are produced by the main/tail rotors in the vertical/horizontal planes, respectively, and are given by:

$$F_v(\omega_m) = \frac{J_v \hat{\Omega}_v + g[(A - B) \cos \alpha_v - C \sin \alpha_v] - T_{\text{fric},v}}{l_m} \quad (11)$$

$$\hat{\Omega}_v = \frac{d^2 \alpha_v}{dt^2} \quad (12)$$

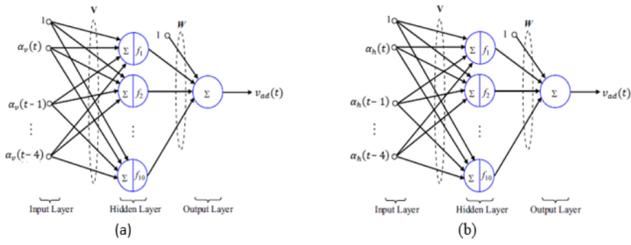
$$F_h(\omega_t) = \frac{J_h \hat{\Omega}_h - T_{\text{fric},h}}{l_t \cos \alpha_v} \quad (13)$$

$$\hat{\Omega}_h = \frac{d^2 \alpha_h}{dt^2} \quad (14)$$

The variables  $M_v$  and  $M_h$  represent the sum of moments in the pitch and yaw planes, respectively. Similarly,  $J_v$  and  $J_h$  denote the sum of moments of inertias in the vertical and horizontal planes.  $T_{\text{fric},v}$  stands for the frictional torque developed in the pitch plane. The masses  $m_{mr}$  and  $m_{tr}$  correspond to the composite mass of the Main/Tail rotor plus Main/Tail DC motor.  $m_m$  and  $m_t$  are the masses of the beam's Main/Tail portion,  $m_{cb}$  is the mass of the counterweight, and  $m_{ms}$  and  $m_{ts}$  represent the masses of the Main/Tail shield. The lengths  $l_b$ ,  $l_{cb}$ ,  $l_m$ , and  $l_t$  refer to the counterbalance beam length, the distance from the pivot joint to the counter-balance or counterweight, and the lengths of the beam's Main/Tail portion. The angles  $\alpha_v$  and  $\alpha_h$  represent the angles for the pitch and yaw, respectively.

### THE NEURAL NETWORK STRUCTURE

The NN structure presents the layers arrangements and the no. of neurons in each layer. Each of the two planes/axes of our plant in question (i.e., the pitch and yaw) are presented with a Feedforward NN structure.



**Figure 4** The NARX shallow NN for the (a) pitch (b) yaw angles (Rahideh et al. 2012a,b).

$$\mathbf{V} = \begin{bmatrix} b_{v,1} & \cdots & b_{v,10} \\ \vdots & \ddots & \vdots \\ v_{3,1} & \cdots & v_{3,10} \end{bmatrix} \quad \text{and} \quad \mathbf{V} = \begin{bmatrix} b_{v,1} & \cdots & b_{v,1000} \\ \vdots & \ddots & \vdots \\ v_{3,1} & \cdots & v_{3,1000} \end{bmatrix} \quad (19)$$

$$f_j(z_j) = \frac{1}{1 + e^{-\alpha_j z_j}}, \quad j = 1, 2, \dots, 10 \quad (20)$$

$$\mathbf{F} = \begin{bmatrix} 1 \\ f_1(z_1) \\ \vdots \\ f_{10}(z_{10}) \end{bmatrix} \quad (21)$$

where,  $\mathbf{X}$ = the input vector,  $\mathbf{W}$ = the network weights,  $\mathbf{V}$ = the biases matrices,  $\mathbf{F}$ = the activation function Widrow and Hoff (1960).

### Training the Network

The neural network must be trained in order for biases and weights adjustments so as to obtain the optimum system parameters. This training could be carried out in offline or online scenarios, but here the offline training is adopted. The network weights and biases are updated/adjusted with the main aim of minimizing the tracking error response of the plant (TRMS). This adjustment is done according to the following formulations (Widrow and Hoff 1960):

$$\dot{\mathbf{W}} = - \left[ (\mathbf{F} - \mathbf{FV}^T \mathbf{X}) \mathbf{r}^T + \beta \|e\| \mathbf{W} \right] \Lambda_{\mathbf{W}} \quad (22)$$

$$\dot{\mathbf{V}} = -\Lambda_{\mathbf{V}} \left[ \mathbf{X} \mathbf{r}^T \mathbf{W}^T \mathbf{F}^T + \beta \|e\| \mathbf{V} \right] \quad (23)$$

where  $\Lambda_{\mathbf{W}}, \Lambda_{\mathbf{V}}$  represent the network learning rates, and with  $\beta > 0$  ensures tracking the error of the system  $e$  and the neural networks weights are bounded uniformly.  $e$  is given by:

$$e = \begin{bmatrix} \alpha_{v,\text{ref}} - \alpha_v \\ \dot{\alpha}_{v,\text{ref}} - \dot{\alpha}_v \end{bmatrix} \quad (24)$$

**Figure 5** The actual NARX shallow (1-layered) NN controllers for the (a) pitch (b) yaw, each having 10 neurons with network weights attached as seen above (c) the configuration showing the single-input single-output network (d) the feedforward configuration for both pitch and yaw angles.

### The Feedforward Neural Network

For the first NARX shallow Neural Networks (Fig. 4a) having one hidden layer with 10 hidden neurons, since 4 input tapped delays are employed for this research work, then the pitch angle of the beam is the inputs to the NN at the current time, and delayed 1, 2, 3, and 4 samples, i.e.,  $\alpha_v(t), \alpha_v(t-1), \alpha_v(t-2), \alpha_v(t-3), \alpha_v(t-4)$ . In a similar vein, the second NARX NN model is for the yaw angle with inputs at the current time instant, and those delayed by 1, 2, and 3 samples, i.e.,  $\alpha_h(t), \alpha_h(t-1), \alpha_h(t-2), \alpha_h(t-3), \alpha_h(t-4)$ . For both models, the output  $v_{ad}(t)$  is expressed as:

$$v_{ad}(t) = b_w + \sum_{j=1}^n w_j f_j \left( b_{vj} + \sum_{i=1}^3 v_{ij} x_i \right) = \mathbf{W}^T \mathbf{F}(\mathbf{V}^T \mathbf{X}) \quad (15)$$

$$\mathbf{X} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (16)$$

$$x_i = \alpha_v(t-i+1) = \alpha_h(t-i+1), \quad i = 1, 2, 3 \quad (17)$$

$$\mathbf{W} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}, \quad n = 10, 1000 \quad (18)$$

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \frac{\partial f_1(z_1)}{\partial z_1} & 0 & \cdots & 0 \\ 0 & \frac{\partial f_2(z_2)}{\partial z_2} & \cdots & \vdots \\ 0 & \cdots & \ddots & \frac{\partial f_2(z_2)}{\partial z_2} \\ 0 & \cdots & 0 & 0 \end{bmatrix} \quad (25)$$

$$\mathbf{r}^T = (\mathbf{e}^T \mathbf{P} \mathbf{B})^T \quad (26)$$

where,  $\mathbf{P}$  is the Lyapunov candidate solution for the nonlinear equation:

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} + \mathbf{Q} = 0 \quad (27)$$

where  $Q$  must be a positive-definite matrix (i.e.,  $Q > 0$ ),  $A$  and  $B$  are matrices for the tracking error ( $e$ ), given by:

$$A = \begin{bmatrix} 0 & 1 \\ -k_p & -k_d \end{bmatrix} \quad (28)$$

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (29)$$

### Validation of the Model

The tools used for validating the nonlinear model of the TRMS include, One Step-Ahead (OSA) Prediction, Mean Squared Error (MSE), Correlations Tests (Autocorrelation and cross-correlation functions) and Normalization.

### OSA Prediction

The OSA prediction for the NARX network or model occurs when the feedback loop of the network is open. The NARX network/model thus predicts the next value of the output  $v_{\alpha d}(t)$  from the previous ones of  $v_{\alpha d}(t)$  and the input  $\alpha_\gamma(t)$ . For a Multi-Step-Ahead prediction, the feedback loop must necessarily be closed. The OSA is a measure of the accuracy in modeling. It is given by:

$$\hat{v}_{\alpha d}(t) = f[\alpha_\gamma(t), \alpha_\gamma(t-1), \dots, \alpha_\gamma(t-n_{\alpha_\gamma}), v_{\alpha d}(t-1), \dots, v_{\alpha d}(t-n_{v_{\alpha d}})] \quad (30)$$

where,  $f$  is a nonlinear function approximator,  $\alpha_\gamma, v_{\alpha d}$  represent the input/output respectively,  $\hat{v}_{\alpha d}(t)$  is the prediction value and  $\gamma$  represents pitch or yaw. The OSA is an extension of the NARX model, where the NARX model is given by:

$$v_{\alpha d}(t) = f[v_{\alpha d}(t-1), \dots, v_{\alpha d}(t-n_{v_{\alpha d}}), \alpha_\gamma(t-1), \dots, \alpha_\gamma(t-n_{\alpha_\gamma})] \quad (31)$$

$$e_{\text{res}} = v_{\alpha d}(t) - \hat{v}_{\alpha d}(t) \quad (32)$$

### Mean Squared Error (MSE)

The MSE (Mean Squared Error) is a validation test, providing the average of the sum of mean squares of the differences between the actual and predicted outputs ( $v_{\alpha d}(t), \hat{v}_{\alpha d}(t)$ ) of the TRMS system. The outputs are generated using the input and the optimized parameters of the network. MSE is given by:

$$\text{MSE} = f(e) = \frac{1}{n} \sum_{i=1}^n |e_{\text{res}}|^2 = \frac{1}{n} \sum_{i=1}^n |v_{\alpha d}(t) - \hat{v}_{\alpha d}(t)|^2 \quad (33)$$

where,  $n$  is the number of input/output samples.

The MSE (Mean Squared Error) algorithm helps in adjusting the network weights and biases, minimizing the MSE. Fortunately, the MSE performance indicator is a quadratic function, which will either have a global minimum, a weak minimum, or no minimum at all, determined by the nature of the input vectors. Hence, a unique solution may or may not exist. The MSE algorithm, or Widrow-Hoff learning algorithm (Demuth and Beale 2000), approximates MSE based on the steepest descent algorithm at each iteration.

Taking the partial derivatives of the MSE with respect to weights and biases at the  $k$ th iteration, we get:

$$\frac{\partial e_{\text{res}}^2(k)}{\partial w_{i,j}} = 2e_{\text{res}}(k) \frac{\partial e_{\text{res}}(k)}{\partial w_{i,j}}, \quad j = 1, 2, \dots, R \quad (34)$$

$$\frac{\partial e_{\text{res}}^2(k)}{\partial b} = 2e_{\text{res}}(k) \frac{\partial e_{\text{res}}(k)}{\partial b} \quad (35)$$

Taking the partial derivative w.r.t error

( $e_{\text{res}}$ )

$$\frac{\partial e_{\text{res}}(k)}{\partial w_{i,j}} = \frac{\partial}{\partial w_{i,j}} [t(k) - \alpha(k)] = \frac{\partial}{\partial w_{i,j}} [t(k) - (Wp(k) + b)] \quad (36)$$

or

$$\frac{\partial e_{\text{res}}(k)}{\partial w_{i,j}} = \frac{\partial}{\partial w_{i,j}} \left[ t(k) - \left( \sum_{i=1}^R w_{1,i} p_i(k) + b \right) \right] \quad (37)$$

where,  $p_i(k)$  is the  $i$ th element of the input vector at the  $k$ th iteration.

Further simplification yields:

$$\left. \begin{aligned} \frac{\partial e_{\text{res}}(k)}{\partial w_{i,j}} &= -p_j(k) \\ \frac{\partial e_{\text{res}}(k)}{\partial b} &= -1 \end{aligned} \right\} \quad (38)$$

### The Correlations Tests

These tests are statistical tests for bivariate dataset composed of the autocorrelation and cross-correlation functions (Darus and Lokaman 2010), given by:

$$\left. \begin{aligned} \phi_{\varepsilon\varepsilon}(\tau) &= E[\varepsilon(t-\tau)\varepsilon(t)] = \delta(t) \\ \phi_{x\varepsilon}(\tau) &= E[x(t-\tau)\varepsilon(t)] = 0 \quad \forall \tau \\ \phi_{x^2\varepsilon}(\tau) &= E[(x^2(t-\tau) - \bar{x}^2(t))\varepsilon(t)] = 0 \quad \forall \tau \\ \phi_{x^2\varepsilon^2}(\tau) &= E[(x^2(t-\tau) - \bar{x}^2(t))\varepsilon^2(t)] = 0 \quad \forall \tau \\ \phi_{\varepsilon(\varepsilon x)}(\tau) &= E[\varepsilon(t)\varepsilon(t-1-\tau)x(t-1-\tau)] = 0 \quad \tau \geq 0 \end{aligned} \right\} \quad (39)$$

where,  $\phi_{\varepsilon\varepsilon}(\tau)$  and  $\phi_{x\varepsilon}(\tau)$  are the autocorrelation and cross-correlation functions between  $x(t)$  and  $\varepsilon(t)$ , and  $\varepsilon(t)$  is the error of the prediction sequence.

### Normalization

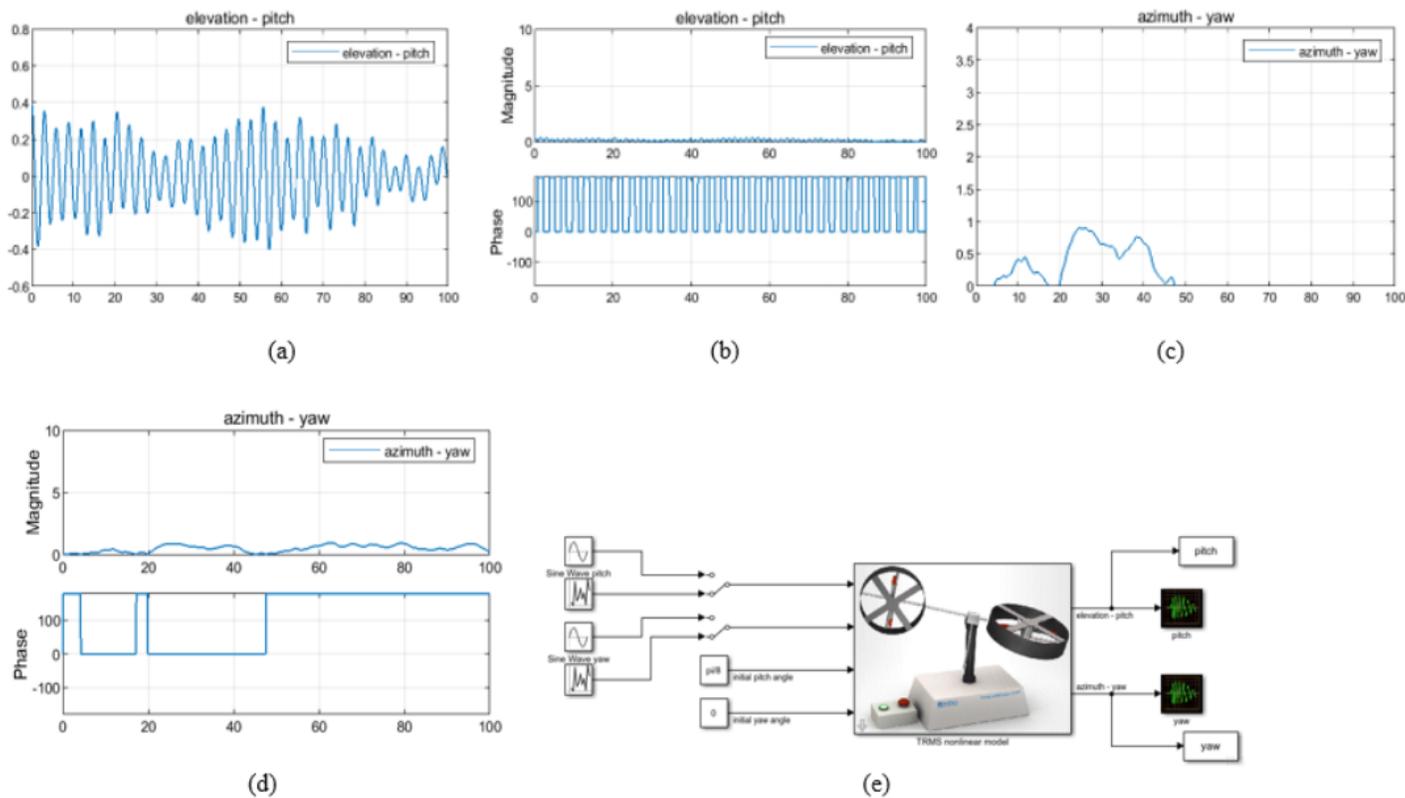
In practice, the correlations computed are normalized to ensure all the values fall within a given bandwidth and/or range. The normalized correlation function between two sequences  $\psi_1(t)$  and  $\psi_2(t)$  is given by:

$$\hat{\phi}_{\psi_1\psi_2}(\tau) = \frac{\sum_{i=1}^{N-\tau} \psi_1(t)\psi_2(t-\tau)}{\sqrt{\sum_{i=1}^N \psi_1^2(t) \sum_{i=1}^N \psi_2^2(t)}} \quad (40)$$

## RESULTS AND DISCUSSION

### The Nonlinear TRMS Modelling

The method employed in this research work involves the use of a time-domain closed-loop control approach. Here, the NARX shallow Neural Networks modelling is used as a compensator to train the network and provide the closed-loop control signal. The input signals to the TRMS and the NN are a uniform random signal, exciting both the pitch and yaw subsections of the plant/system. The Simulink model of the TRMS as well as the signals are given below.



**Figure 6** Simulation of the TRMS (a) Pitch Random signals for the input (b) Its Log/Magnitude (c) Yaw Random signals for the input (d) Its Log/Magnitude (e) Simulink model of the nonlinear TRMS

### Shallow NN Modelling

**Training with Levenberg-Marquardt (LM)** It is known that for every neural network structure designed/implemented as a solution to e.g. a control problem, the correction error functions must lie within an acceptable predefined region which is depicted in figs 7 (a - e), otherwise the control design objective will not be achievable. Also, the Best validation performance for our design must occur at an epoch where the best value falls the parameterized training, validation and testing performance scores. And as can be clearly seen, these were obtained at various epochs for individual runs of the TRMS plant, forming the available simulation data to the NN. 3 different runs each for 10 neurons (figs. 7 (g - i)) and for 1000 neurons (figs. 7 (j - l)) were used in order to show data integrity from the TRMS Simulink model obtained from first principles. For the training, testing and validation of the NN structure, Levenberg-Marquardt training algorithm was used throughout this research and the step-by-step procedure is giving above in fig. 7 (m) in Matlab.using the nntool command.

**Further Discussions** Figs 9 – 13 below show the results obtained from different controllers employed in this study. In Fig. 9, Classical PID control was used for simulation times of (a) 50 (b) 100 seconds, while in Fig. 10, ANN controllers were generated and deployed for simulation times of (a) 50 (b) 100 seconds. In Fig. 11, the developed ANN controllers were combined with meta-heuristic approaches of Pattern Search (PS) and Latin Hyperbole (LH) to improve on the neural controller. In Fig. 12, the ANN controllers were combined with PS and Genetic Algorithm (GA) this time to obtain highly improved tracking control performances for simulation times of (a) 50 (b) 100 seconds. Fig. 13 is merely

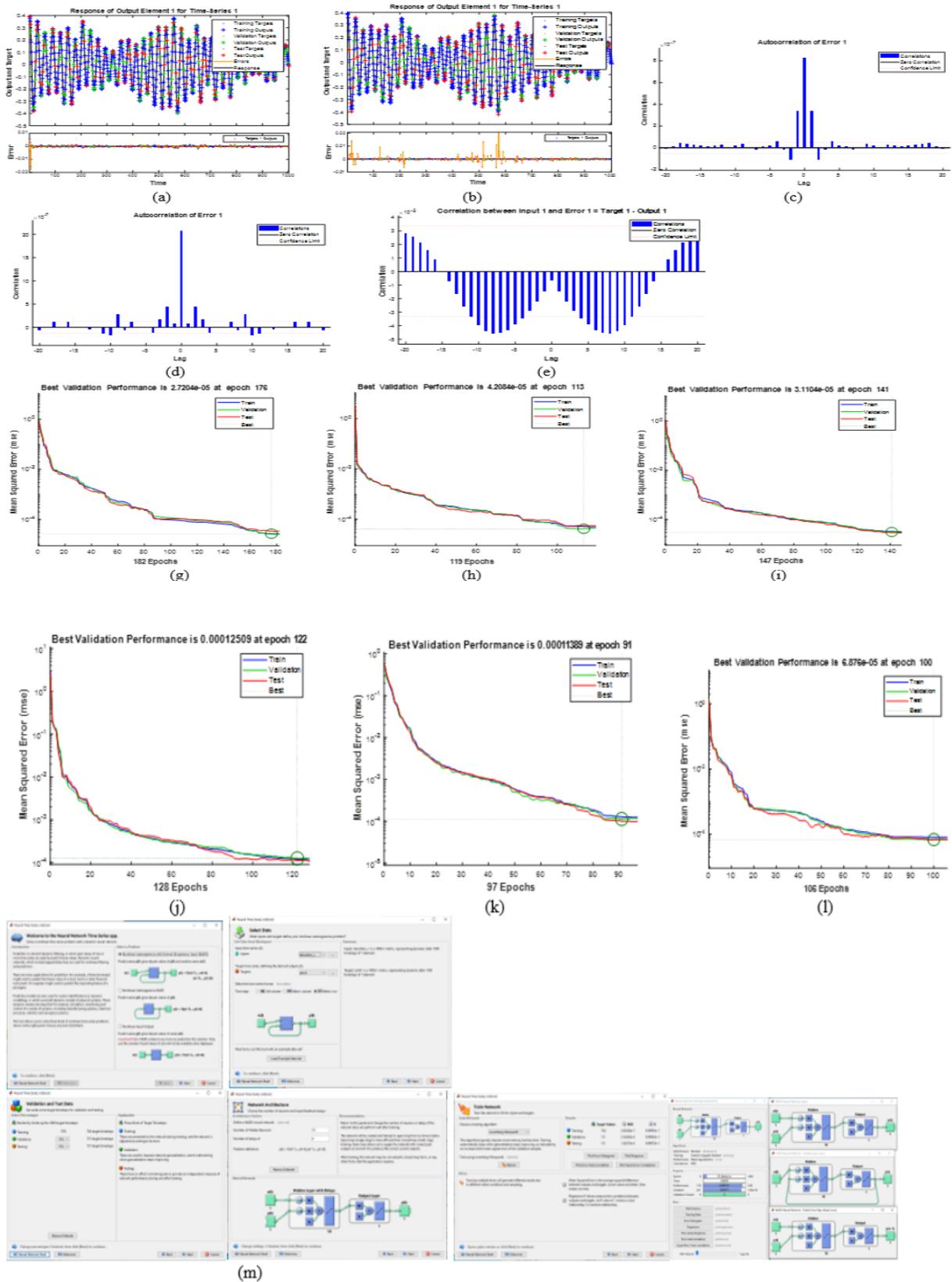
a comparison of these techniques above, combined, to depict the strength of the ANN + PS + GA strategy over the other methods in setpoint tracking of the commanded input to the TRMS prototype helicopter.

Since the performance measures can be given in terms of time-domain or frequency-domain specifications, here the performance indices are expressed in terms of the usual time-domain specification: rise time  $\tau_r$ , settling time  $\tau_s$ , and steady-state error  $e_{ss}$ . These results have been presented and tabulated in Tables 1 and 2 below. Note that the ISE and RMSE are functions of the squares of  $e_{ss}$  error coefficients, statistically designed as indices of performances of the control simulations. Since the results are presented in a composite fashion, the combined  $e_{ss}$  for the pitch and yaw angles for ANN + PS + GA is negligibly small compared to the other methods used, as seen in Fig. 12 and Table 1. This explains the best tracking performance and low control energy required.

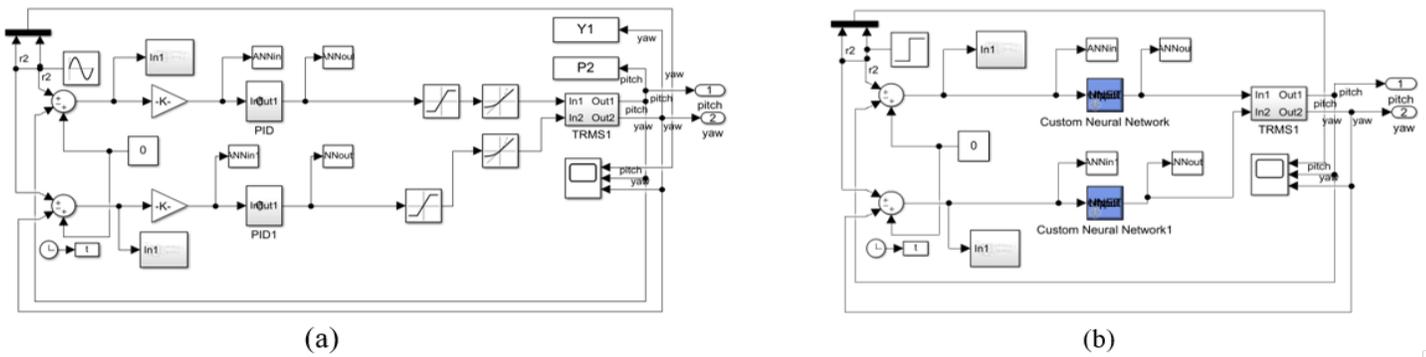
Note: PS = Pattern Search; LH = Latin Hyperbole;  $\tau_r$ =rise time;  $\tau_s$ =Settling time;  $e_{ss}$ =Steady-state error

Note: PS = Pattern Search; LH = Latin Hyperbole;  $\tau_r$ =rise time;  $\tau_s$ =Settling time;  $e_{ss}$ =Steady-state error; ISE = Integral Squared Error; RMSE = Root Mean Squared error

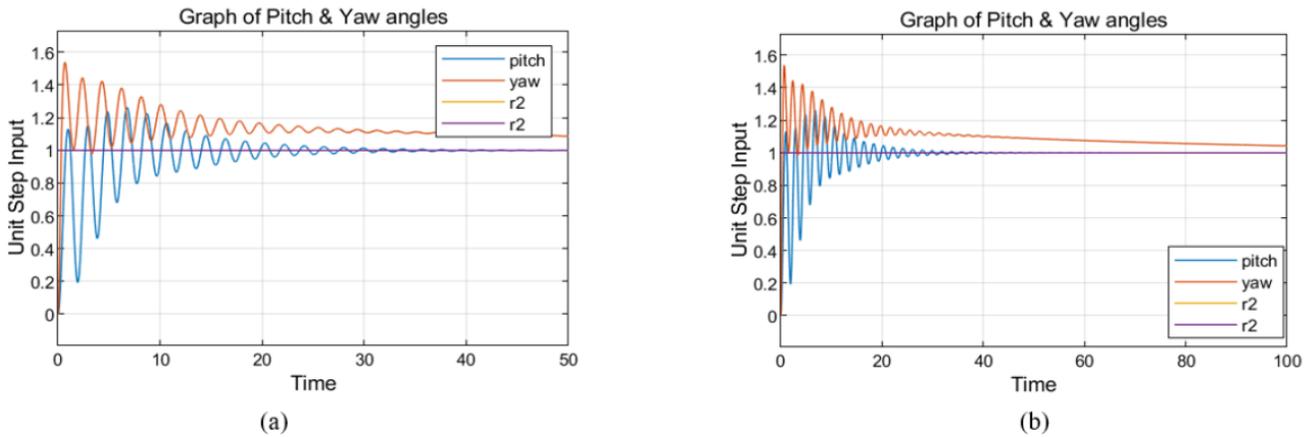
The TRMS plant is a highly uncertain and highly nonlinear plant with high-frequency oscillations, particularly with the pitch angle. This can pose as a serious control challenge in efforts to remove the rippling oscillations. This is shown in the scope of the designed control system using the conventional PID controllers (Fig. 9). For the real system, this evidently would affect the plant operation in the inability of the plant to settle within acceptable limits specified for effective control. The need, therefore, for im-



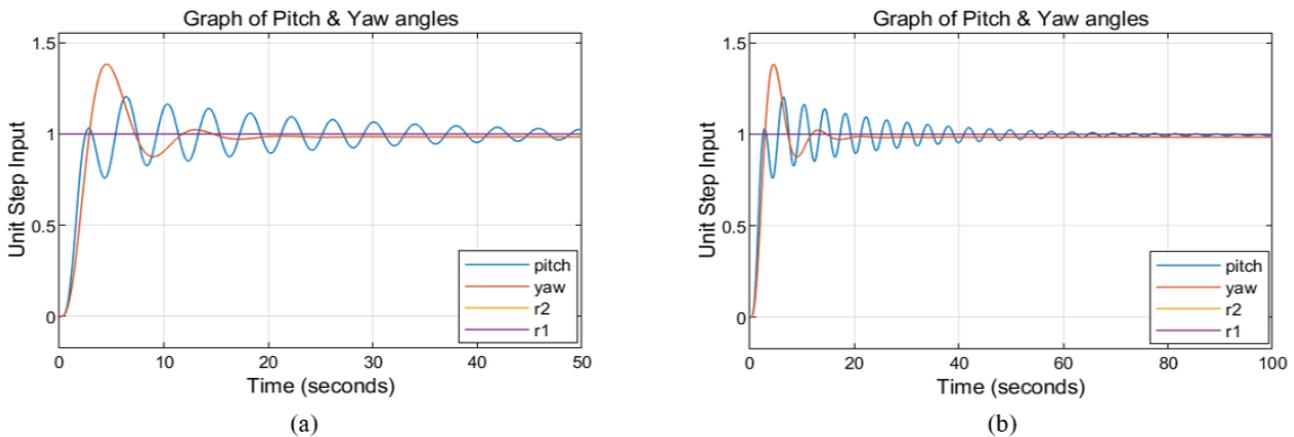
**Figure 7** Levenberg-Marquardt training with (a) 10 (b) 1000 hidden neurons; Autocorrelation Error for (c) 10 (d) 1000 hidden neurons; Input-Error Correlation for (e) 10 (f) 1000 hidden neurons; Best validation performance for (g)-(i) 10 (j) – (l) 1000 hidden neurons (m) Matlab nntool NARX NN GUI programming and execution.



**Figure 8** Control system designed to implement step input signal tracking control using (a) conventional PID controller (b) neural networks controllers generated using the 'gensim' command



**Figure 9** The tracking control for the elevation (pitch)-red and azimuth (yaw)-blue trackings of the TRMS using PID controllers for a simulation time of (a) 50 (b) 100 seconds

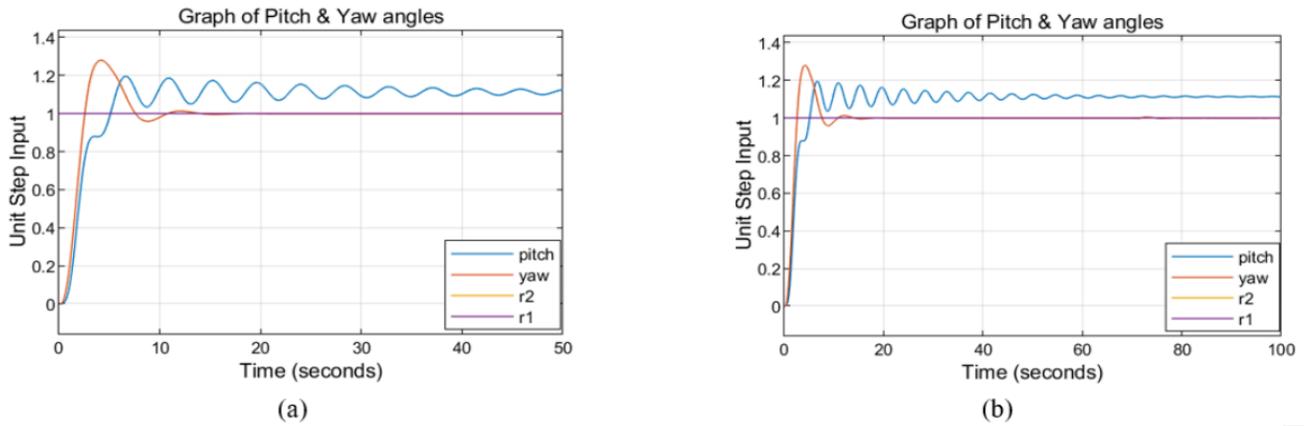


**Figure 10** The final actual output shown for the elevation (pitch)-red and azimuth (yaw)-blue trackings of the TRMS using ANN controllers realized using the "getsim" command for a simulation time of (a) 50 (b) 100 seconds

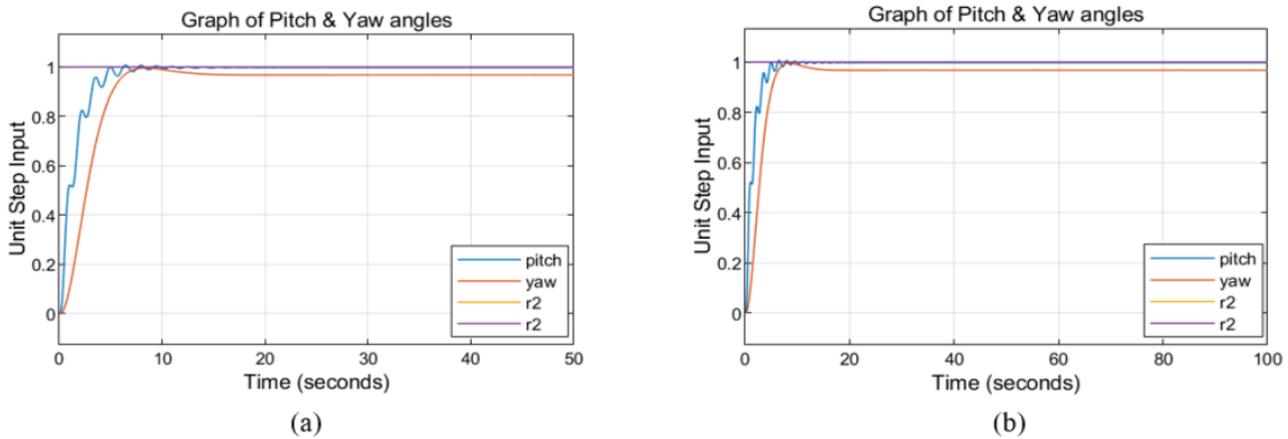
proved tracking becomes indispensable for such a safety-critical system.

This informed the use of the neural networks (NN) controllers (Fig. 10). The NN controllers were able to eliminate the undesirable oscillations or ripples in the final outputs for the yaw angle at first glance (Fig. 10), though with a large overshoot. When the network

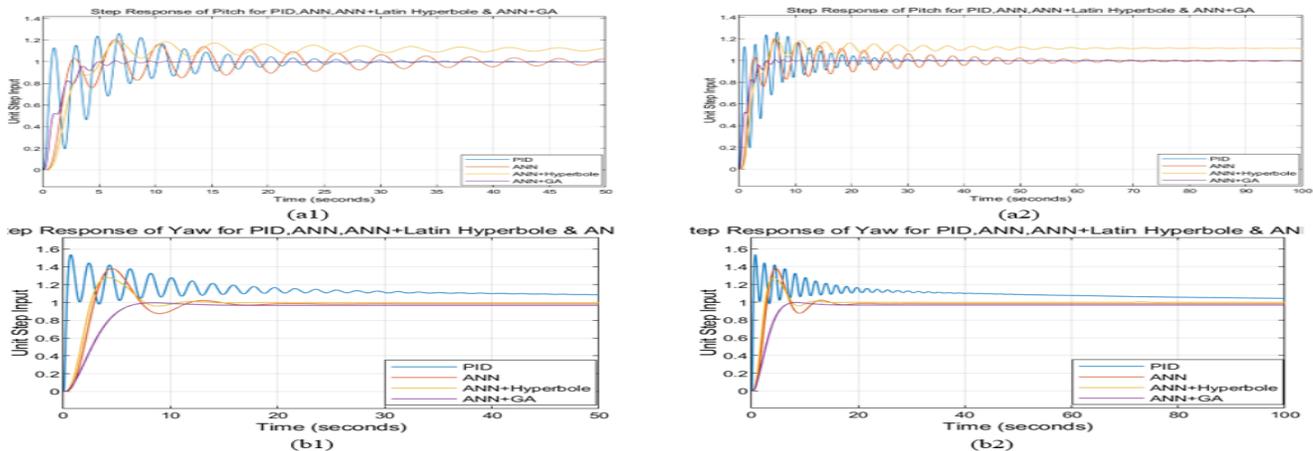
weights of the ANN controllers were optimized using intelligent schemes of Pattern Search with Latin Hyperbole (ANN + PS + LH) and Pattern Search with Genetic Algorithm (ANN + PS + GA), the GA-based ANN controllers completely eliminated the ripples for both angles and brought the system within acceptable bandwidths of control (Fig. 12) and fast tracking simulation time of 10 seconds.



**Figure 11** Improved step input tracking control for the pitch and yaw angles using neural networks controllers optimized using Pattern Search + Latin Hyperbole for a simulation time of (a) 10 (b) 100 seconds



**Figure 12** Final Improved step input tracking control for the pitch and yaw angles using neural networks controllers optimized using Pattern Search + GA for a simulation time of (a) 10 (b) 100 seconds



**Figure 13** Comparisons of the different controllers employed above for a step input tracking response for the (a) pitch and (b) yaw angles, for a simulation time of (a1 & b1) 50 (a2 & b2) 100 seconds

The ANN + PS + LH also performed well with a fast tracking response but with a very high overshoot for the yaw angle and a large steady-state error ( $e_{ss}$ ) (Fig. 11).

The best-performing algorithm as evidenced in Table 2 above is the proposed ANN + PS + GA, with the best settling time for the pitch angle ( $\tau_s = 5.14$ ) and the second-best settling time for the

■ **Table 1** Quantitative comparison of time-domain specifications between the proposed ANN + Pattern Search + GA controller, and the 3 other controllers design strategies for pitch & yaw angles

Controller Method	Horizontal plane ( $\phi$ angle )			Vertical plane ( $\theta$ angle )		
	$\tau_r$	$\tau_s$	$e_s$	$\tau_r$	$\tau_s$	$e_{ss}$
	PID	0.24	100	0.04	0.48	25.00
ANN	1.18	14.5	0.02	1.32	56	0.00
ANN + PS + LH	1.55	9.9	0.99	4	37	0.11
ANN + PS + GA	3.85	14.42	0.03	2.79	5.14	0.002

■ **Table 2** Quantitative comparison of performance indices between the proposed ANN + Pattern Search + GA controller, and 3 other controllers design strategies for pitch & yaw angles.

Controller Method	Horizontal plane ( $\phi$ angle )		Vertical plane ( $\theta$ angle )	
	ISE	RMSE	ISE	RMSE
	PID	1.402	0.087	1.135
ANN	1.862	0.016	1.556	0.223
ANN + PS + LH	1.835	0.002	2.243	0.123
ANN + PS + GA	2.034	0.032	72.36	0.758

yaw angle, as well as good rise times, i.e., very fast responses and good steady-state errors for both the pitch and yaw angles.

## CONCLUSION

Results for the NARX Feedforward NN methodology in the modeling of the nonlinear TRMS have been presented in this report. It has also been shown that different solutions are obtained for every NN training undergone. This is due to the differing initial weights conditions and biases as well as the arbitrary (i.e., random) division of the dataset into training, validation, and testing in the given ratios of 0.7, 0.15, 0.15. To ensure accuracy of the modeling results, retraining should be performed several times.

The final output results for the ANN and Pattern Search with Genetic Algorithm (ANN + PS + GA) show satisfactory control for the optimized performance of the nonlinear plant, outperforming three other controllers employed, as proven by the statistical and graphical results presented. The proposed controller met all our design requirements of within 5% of settling time, below 1% ( $\gg 1\%$ ) of overshoot, as well as excellent rise times, i.e., very fast (aggressive) responses for both the pitch and yaw angles, and no steady-state error for the pitch angle, and a negligible steady-state error for the yaw angle. Also of note was the minimum control energy used by the controller in achieving these objectives.

The neural controllers designed were based on SISO control architecture of the neural networks, each for the decoupled and independent pitch and yaw subsystems. For future work, a more robust and adaptive MIMO neural networks controller can be developed/ designed without going through the rigours of decoupling the TRMS helicopter model where some dynamics could be lost due to system approximations and simplification in modelling. The MIMO neural controller should automatically determine the network gains and biases for a neural networks structure with two-inputs two-outputs in a reasonable amount of time.

## Acknowledgments

Authors David Mohammed Ezekiel, Ravi Samikannu, and Matsebe Oduetse received initiation research grants from Botswana International University of Science and Technology (BIUST) to carry out this research work. Grant No: Ref/DVC/RDI/2/1/7 V (8) Account Project Code: S00170.

## Availability of data and material

Not applicable.

## Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

## LITERATURE CITED

- Abbas, N. and X. Liu, 2022 A mixed dynamic optimization with  $\mu$ -synthesis (dk iterations) via optimal gain for varying dynamics of decoupled twin-rotor mimo system based on the method of inequality (moi). *Journal of Control Engineering and Applied Informatics* **24**: 13–23.
- Abdulwahhab, O. W. and N. H. Abbas, 2017 A new method to tune a fractional-order pid controller for a twin rotor aerodynamic system. *Arabian Journal for Science and Engineering* **42**: 5179–5189.
- Agand, P., M. A. Shoorehdeli, and A. Khaki-Sedigh, 2017 Adaptive recurrent neural network with lyapunov stability learning rules for robot dynamic terms identification. *Engineering applications of artificial intelligence* **65**: 1–11.
- Ahmad, M., A. Ali, and M. A. Choudhry, 2016 Fixed-structure  $h_\infty$  controller design for two-rotor aerodynamical system (tras). *Arabian Journal for Science and Engineering* **41**: 3619–3630.
- Ahmad, S. M., A. J. Chipperfield, and M. O. Tokhi, 2000a Dynamic modelling and control of a 2-dof twin rotor multi-input multi-output system. In *2000 26th Annual Conference of the IEEE Industrial Electronics Society, IECON 2000. 2000 IEEE International Conference on Industrial Electronics, Control and Instrumentation. 21st Century Technologies*, volume 2, pp. 1451–1456, IEEE.
- Ahmad, S. M., A. J. Chipperfield, and O. Tokhi, 2000b Dynamic modeling and optimal control of a twin rotor mimo system. In *Proceedings of the IEEE 2000 National Aerospace and Electronics Conference. NAECON 2000. Engineering Tomorrow (Cat. No. 00CH37093)*, pp. 391–398, IEEE.
- Alam, M. S., F. M. Aldebrez, and M. O. Tokhi, 2004 Adaptive command shaping using genetic algorithms for vibration control. In *Proceedings of IEEE SMC UK-RI Third Workshop on Intelligent Cybernetic Systems*, pp. 7–8, IEEE.

- Bensidhoum, T., F. Bouakrif, and M. Zasadzinski, 2023 A high order p-type iterative learning control scheme for unknown multi input multi output nonlinear systems with unknown input saturation. *International Journal of Computational and Applied Mathematics & Computer Science* **3**: 27–31.
- Castillo, E., B. Guijarro-Berdinas, O. Fontenla-Romero, A. Alonso-Betanzos, and Y. Bengio, 2006 A very fast learning method for neural networks based on sensitivity analysis. *Journal of Machine Learning Research* **7**.
- Chu, S. R., R. Shoureshi, and M. Tenorio, 1990 Neural networks for system identification. *IEEE Control systems magazine* **10**: 31–35.
- Coelho, J., R. Neto, D. Afonso, C. Lebres, H. Fachada, *et al.*, 2007a Helicopter system modelling and control with matlab. *CEE'07-2nd INTERNATIONAL CONFERENCE ON ELECTRICAL ENGINEERING* pp. 110–117.
- Coelho, J., R. M. Neto, C. Lebres, H. Fachada, N. M. Ferreira, *et al.*, 2007b Application of fractional algorithms in the control of a helicopter system. In *Symposium on Applied Fractional Calculus*, pp. 1–12.
- Coelho, J., R. M. Neto, C. Lebres, V. Santos, N. M. Ferreira, *et al.*, 2008 Application of fractional algorithms in the control of a twin rotor multiple input-multiple output system. *ENOC 08* pp. 1–8.
- Darus, I. Z. M. and Z. A. Lokaman, 2010 Dynamic modelling of twin rotor multi system in horizontal motion. *Jurnal Mekanikal*.
- Demuth, H. B. and M. H. Beale, 2000 *Neural Network Toolbox: For Use with MATLAB®*. MathWorks.
- El-Fakdi, A. and M. Carreras, 2013 Two-step gradient-based reinforcement learning for underwater robotics behavior learning. *Robotics and Autonomous Systems* **61**: 271–282.
- Ezekiel, D. M., R. Samikannu, and O. Matsebe, 2021 Pitch and yaw angular motions (rotations) control of the 1-dof and 2-dof trms: A survey. *Archives of Computational Methods in Engineering* **28**: 1449–1458.
- Ezekiel, D. M., R. Samikannu, and M. Oduetse, 2020 Modelling of the twin rotor mimo system (trms) using the first principles approach. In *2020 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–7, IEEE.
- Ghasemiyeh, R., R. Moghdani, and S. S. Sana, 2017 A hybrid artificial neural network with metaheuristic algorithms for predicting stock price. *Cybernetics and systems* **48**: 365–392.
- Khishe, M., M. R. Mosavi, and A. Moridi, 2018 Chaotic fractal walk trainer for sonar data set classification using multi-layer perceptron neural network and its hardware implementation. *Applied Acoustics* **137**: 121–139.
- Lin, S. and A. A. Goldenberg, 2001 Neural-network control of mobile manipulators. *IEEE Transactions on Neural Networks* **12**: 1121–1133.
- Ljung, L. and S. Gunnarsson, 1990 Adaptation and tracking in system identification—a survey. *Automatica* **26**: 7–21.
- Moness, M. and T. Daa-Eldeen, 2017 Experimental nonlinear identification of a lab-scale helicopter system using mlp neural network. In *2017 13th International Computer Engineering Conference (ICENCO)*, pp. 166–171, IEEE.
- Mosbah, H. and M. E. El-Hawary, 2017 Optimization of neural network parameters by stochastic fractal search for dynamic state estimation under communication failure. *Electric Power Systems Research* **147**: 288–301.
- Nasir, A. N. K. and M. O. Tokhi, 2014 A novel hybrid bacteria-chemotaxis spiral-dynamic algorithm with application to modelling of flexible systems. *Engineering Applications of Artificial Intelligence* **33**: 31–46.
- Palepogu, K. R. and S. Mahapatra, 2023 Pitch orientation control of twin-rotor mimo system using sliding mode controller with state varying gains. *Journal of Control and Decision* pp. 1–11.
- Patan, K. and M. Patan, 2023 Fault-tolerant design of non-linear iterative learning control using neural networks. *Engineering Applications of Artificial Intelligence* **124**: 106501.
- Rahideh, A., A. H. Bajodah, and M. H. Shaheed, 2012a Real time adaptive nonlinear model inversion control of a twin rotor mimo system using neural networks. *Engineering Applications of Artificial Intelligence* **25**: 1289–1297.
- Rahideh, A., A. H. Bajodah, and M. H. Shaheed, 2012b Real time adaptive nonlinear model inversion control of a twin rotor mimo system using neural networks. *Engineering Applications of Artificial Intelligence* **25**: 1289–1297.
- Rahideh, A., M. H. Shaheed, and H. J. C. Huijberts, 2008 Dynamic modelling of a trms using analytical and empirical approaches. *Control Engineering Practice* **16**: 241–259.
- Rere, L. R., M. I. Fanany, and A. M. Arymurthy, 2016 Metaheuristic algorithms for convolution neural network. *Computational intelligence and neuroscience*.
- Salimi, H., 2015 Stochastic fractal search: a powerful metaheuristic algorithm. *Knowledge-based systems* **75**: 1–18.
- Sivadasan, J. and J. R. J. Shiney, 2023 Modified nondominated sorting genetic algorithm-based multiobjective optimization of a cross-coupled nonlinear pid controller for a twin rotor system. *Journal of Engineering and Applied Science* **70**: 133.
- Sjöberg, J., H. Hjalmarsson, and L. Ljung, 1994 Neural networks in system identification. *IFAC Proceedings Volumes* **27**: 359–382.
- Tavakolpour, A. R., M. Mailah, I. Z. M. Darus, and O. Tokhi, 2010 Self-learning active vibration control of a flexible plate structure with piezoelectric actuator. *Simulation Modelling Practice and Theory* **18**: 516–532.
- Tijani, I. B., R. Akmeliawati, A. Legowo, and A. Budiyo, 2014 Nonlinear identification of a small scale unmanned helicopter using optimized narx network with multiobjective differential evolution. *Engineering Applications of Artificial Intelligence* **33**: 99–115.
- Toha, S. F. and M. O. Tokhi, 2009 Dynamic nonlinear inverse-model based control of a twin rotor system using adaptive neuro-fuzzy inference system. In *2009 Third UKSim European Symposium on Computer Modeling and Simulation*, pp. 107–111, IEEE.
- Toha, S. F. and M. O. Tokhi, 2010 Augmented feedforward and feedback control of a twin rotor system using real-coded moga. In *IEEE Congress on Evolutionary Computation*, pp. 1–7, IEEE.
- TRahman, T. A., A. As'array, N. A. Jalil, and R. Kamil, 2019 Dynamic modelling of a flexible beam structure using feedforward neural networks for active vibration control. *International Journal of Automotive and Mechanical Engineering* **16**: 6263–6280.
- Wai, R. J., 2003 Tracking control based on neural network strategy for robot manipulator. *Neurocomputing* **51**: 425–445.
- Widrow, B. and M. E. Hoff, 1960 Adaptive switching circuits. In *IRE WESCON convention record*, volume 4, pp. 96–104.
- Wu, H., Y. Zhou, Q. Luo, and M. A. Basset, 2016 Training feedforward neural networks using symbiotic organisms search algorithm. *Computational intelligence and neuroscience*.
- Xia, Y. and J. Wang, 2001 A dual neural network for kinematic control of redundant robot manipulators. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **31**: 147–154.
- Yaghini, M., M. M. Khoshraftar, and M. Fallahi, 2013 A hybrid algorithm for artificial neural network training. *Engineering Applications of Artificial Intelligence* **26**: 293–301.
- Yoo, S. J., Y. H. Choi, and J. B. Park, 2006 Generalized predictive

control based on self-recurrent wavelet neural network for stable path tracking of mobile robots: adaptive learning rates approach. *IEEE Transactions on Circuits and Systems I: Regular Papers* **53**: 1381–1394.

**How to cite this article:** Ezekiel, D. M., Samikannu, R., and Matsebe, O. Improved Set-point Tracking Control of an Unmanned Aerodynamic MIMO System Using Hybrid Neural Networks. *Chaos Theory and Applications*, 6(1), 51-62, 2024.

**Licensing Policy:** The published articles in *Chaos Theory and Applications* are licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).

