SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348 – 8379 / https://doi.org/10.14445/23488379/IJEEE-V8I6P103

Volume 8 Issue 6, 15-22, June 2021

Original Article

Black box Modeling of Twin Rotor MIMO System by Using Neural Network

Huong T.M. Nguyen[1], Mai Trung Thai[1]

[1] *Thai Nguyen University of Technology, Thai Nguyen city, Viet Nam*

Received Date: 02 May 2021
Revised Date: 06 June 2021
Accepted Date: 16 June 2021

Abstract - *In model predictive control, building the correct model and solving the optimal problem are two jobs that always require a lot of time and effort. These are also two issues that many scientists are interested in studying when applying model-driven reporting control to certain objects. With a TRMS object we can build a white box model, a gray box model or a black box model. Some authors have built TRMS model published in [2], [3], [4], [5]. We have studied the optimal problem solving methods in model predictive control in articles [6], [7], [8]. In [9], we builds a white box model of TRMS object according to Newton method. Studying the effects of the interchannel effects of the white box model TRMS. In this paper, authors bulding black box modeling of Twin Rotor MIMO System by using neural network, compare the results of the black box model with the real model in order to choose a suitable algorithm and provide the ability to apply that model in simulation and object control.*

Keywords - *Black box model, Neural network, Yaw angle, Pitch angle, Gradient descent back-propagation.*

I. INTRODUCTION

Artificial Neural Networks (ANNs) are intended to mimic the behavior of biological neural networks (NNs). In fact, there are many types of neural networks and their applications are also different. These networks need to be trained using the appropriate learning algorithms for a particular application. A learning algorithm is an optimization procedure in which the synaptic weights between neurons are found to obtain an optimal mapping between a set of inputs and the corresponding desired outputs. In this paper, networks (Multi Layer Perceptron - MLP) were selected for modeling with many different training methods. Therefore, MLP networks and related learning algorithms have been briefly presented here.

The TRMS is a laboratory set-up designed by Feedback Instrument Ltd [1] and is a suitable test platform for assessment and implementation of advanced control techniques. The system is connected to a computer through a fast interface to transfer control signals to the actuators and to receive the corresponding feedback signals from the sensors. Moreover, the real-time workshop toolbox of Matlab/Simulink provides an opportunity for the designer to facilitate the controller design procedure using advanced control toolboxes and other useful built-in functions. The system possesses two propellers perpendicular to each other, one for vertical movement and the other for horizontal motion. However, each one of them significantly affects the motion of the other.

Therefore, the more accurate the TRMS model building, the higher the quality of control in general and predictive control in particular.

II. THE TRMS MODEL

The TRMS was given in fig 1, the physical model as follows:

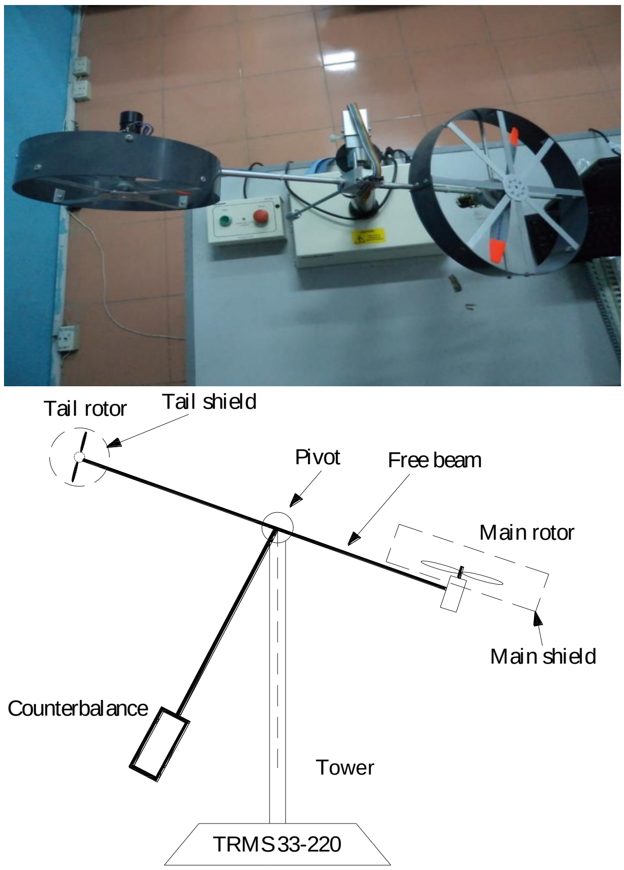

Fig. 1 The TRMS

A. The Fundamentals of MLP Networks

The function of a network is determined by the architecture of the network, its parameters include weights, biases and the type of its processing elements. The architecture of a typical MLP network is defined by the number of layers and the number of neurons in each layer. The structure of an MLP network is designed based on an iterative process. Some of the most common learning algorithms that can be used to train MLP networks include: Levenberg-Marquardt back-propagation, gradient descent back-propagation, quasi-Newton back-propagation, Bayesian regularisation back-propagation, conjugate gradient back-propagation, one step secantback-propagation, resilient back-propagation, and scaled conjugate gradient back-propagation. The paper mainly uses Gradient Descent back-propagation, and some of its conjugate algorithms to identifycation TRMS model.

a) Designing the Excitation Signal

The TRMS has been excited with various input signals of different shapes such as sine and square waves, different possible amplitudes and different frequencies. The frequency of the training data ranges from 0.01Hz to 1Hz and cover amplitude between -2.5V and 2.5V, the minimum and maximum applicable voltages to the system respectively.

b) Model Structure Selection and Training algorithms

The Neural Network Autoregressive External input Model Structure (NNARX) approach has been selected in order to model the system, due to the fact that the input-output data set of the real system is available. In an NNARX model structure,the inputs to the NN-based model are the past control inputs of the real system, $[u(t-d), u(t-d-1), ..., u(t-d-m)]$ and also the past observed outputs, $[y(t-1), y(t-2), ..., y(t-m)]$, where *d* is some multiple of the sampling period and set to one here, and *m* and *n* are input and output lag spaces which are assumed to be two and three respectively. Fig 2 shows a SISO NNARX model structure. It is noted that the output of the SISO model, $\hat{y}(t)$, can be expressed as

$$\hat{y}(t) = f_{NN}(u(t-d), u(t-d-1), ..., u(t-d-m), y(t-1), y(t-12), ..., y(t-n)) \quad (1)$$

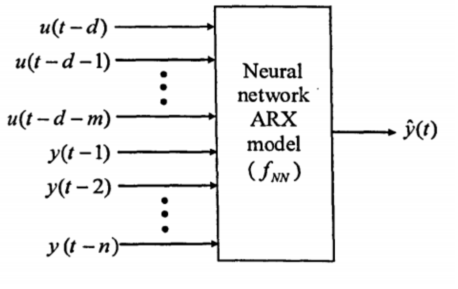

Fig. 2 The NNARX model structure of a SISO system

Some of the most common learning algorithms used in this article include:

+ Gradient descent (GD) back-propagation;
+ Gradient descent with momentum (GDM) back-propagation;
+ Gradient descent with adaptive learning rate (GDA) back-propagation;
+ Gradient descent with adaptive learning rate and momentum (GDX) back - propagatio;
+ Conjugate gradient back-propagation with Fletcher-Reeves (CGF);
+ Conjugate gradient back-propagation with Polak-Ribiere updates;
+ Conjugate gradient back-propagation with Powell-Beale restart (CGB).

All of these algorithms are based on the so-called Mean Squared Error (MSE) method. In MSE paradigms the objective is to determine the weights and the biases of an NN by minimising the following criterion:

$$V = \frac{1}{2Q}\sum_{q=1}^{Q} e_q^T e_q \quad (2)$$

Where:

$$e_q = t_q - e_q \qquad q = 1,2,....,Q \quad (3)$$

Q: Number of patterns or number of data set for training

t_q : Vector of target output (observed output) of pattern q.

y_q : Vector of network output (output prediction) of pattern q.

e_q : Error of pattern

V: Mean squared error

c) Test Model

Model validation is a procedure in which a model is tested to clarify whether the model adequately represents the characteristics of the corresponding real system. Note that each model is tested in accordance with its future application. If the residuals (prediction errors) of a model contain no or negligible information about the past residuals or about the dynamics of the system, it is likely that all information has been extracted from the training set, and conclusively the model approximates the system well. Hence, one needs, in principle, to check whether the residuals are uncorrelated with all the linear and nonlinear combinations of the past data. Such a test is of course completely unrealistic to be carried out in practice; thus, it is common to consider only a few wisely chosen auto-correlation and cross-correlation functions. In [5], equation (4) is the auto-correlation of prediction errors, and the cross-correlationbetween the inputs and the residuals of a model is presented in equation (5). The cross-correlation of the squared inputs and the squared errors is mentioned in equation (6). Equation (7) presents the cross-correlation between the squared inputs and the residuals of the model and finally the cross-correlation between the residuals and the multiplication of the residuals and the inputs is expressed in equation (8).

$$r_{ee}(\tau) = \frac{\sum_{i=1}^{N-\tau}(e_i - \bar{e})(e_{i+\tau} - \bar{e})}{\sum_{i=1}^{N}(e_i - \bar{e})^2} = \begin{cases} 1 & \text{if } \tau = 0 \\ 0 & \text{if } \tau \neq 0 \end{cases} \quad (4)$$

$$r_{ue}(\tau)=\frac{\sum_{i=1}^{N-\tau}(u_i-\bar{u})(e_{i+\tau}-\bar{e})}{\sqrt{\sum_{i=1}^{N}(u_i-\bar{u})^2\sum_{i=1}^{N}(e_i-\bar{e})^2}}=0 \quad \forall\ \tau \qquad (5)$$

$$r_{u^2e^2}(\tau)=\frac{\sum_{i=1}^{N-\tau}(u_i^2-\bar{u}^2)(e_{i+\tau}^2-\bar{e}^2)}{\sqrt{\sum_{i=1}^{N}(u_i^2-\bar{u}^2)^2\sum_{i=1}^{N}(e_i^2-\bar{e}^2)^2}}=0 \quad \forall\ \tau \qquad (6)$$

$$r_{u^2e}(\tau)=\frac{\sum_{i=1}^{N-\tau}(u_i^2-\bar{u}^2)(e_{i+\tau}-\bar{e})}{\sqrt{\sum_{i=1}^{N}(u_i^2-\bar{u}^2)^2\sum_{i=1}^{N}(e_{i+\tau}-\bar{e})^2}}=0 \quad \forall\ \tau \qquad (7)$$

$$r_{e\beta}(\tau)=\frac{\sum_{i=1}^{N-\tau}(e_i-\bar{e})(\beta_{i+\tau}-\bar{\beta})}{\sqrt{\sum_{i=1}^{N}(e_i-\bar{e})^2\sum_{i=1}^{N}(\beta_i-\bar{\beta})^2}}=0 \quad \tau\ge 0 \qquad (8)$$

where the bar over the symbols represents the average of a signal as

$$\bar{x}=\frac{1}{N}\sum_{q=1}^{Q}x_i \qquad (9)$$

N is the total number of data and

$$\beta_i=u_ie_i \qquad (10)$$

It is common to check whether the functions for lags in the interval $\tau\in[-20;20]$ are zero within an asymptotical 95% confidence interval, i.e. if the following condition is held:

$$-1.96/\sqrt{N}<r<1.96/\sqrt{N} \qquad (11)$$

III. SIMULATION RESULTS

The black box model of the TRMS object with 2 degrees of freedom on Matlab/Simulink and the real model when using the Gradient descent algorithm to train and test the network is shown in Figures 3 to 6.

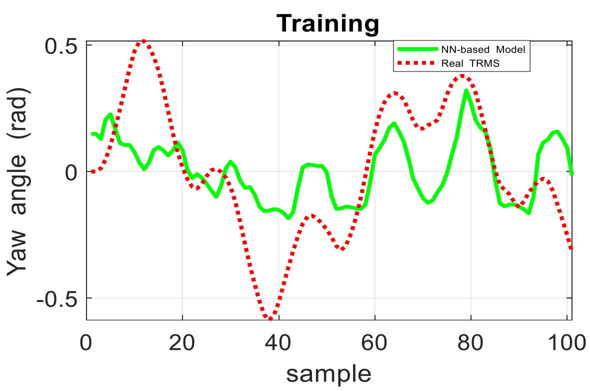

Fig. 3 The yaw angle of the TRMS model when using the Gradient descent algorithm to train the network

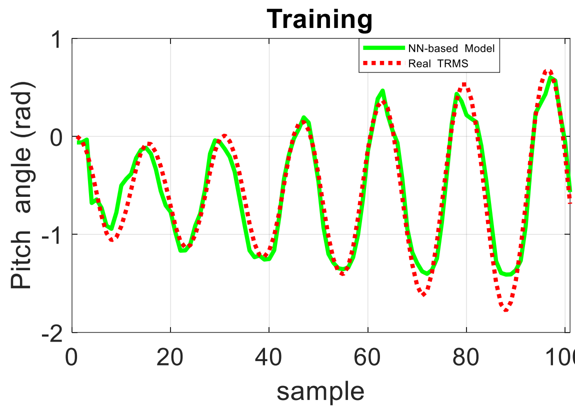

Fig. 4 The pitch angle of the TRMS model when using the Gradient descent algorithm to train the network

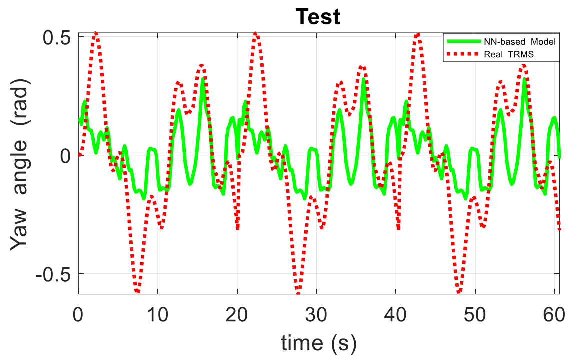

Fig. 5 The yaw angle of the TRMS model when using the Gradient descent algorithm to test the network

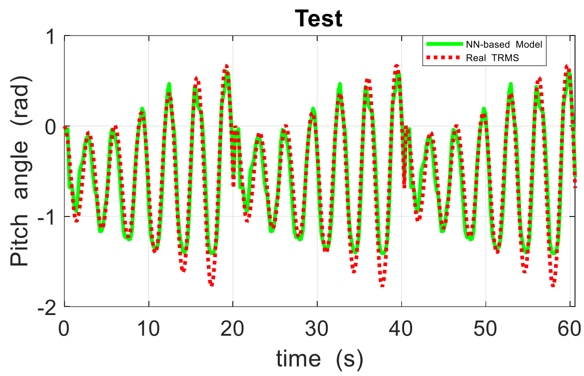

Fig. 6 The pitch angle of the TRMS model when using the Gradient descent algorithm to test the network

Comment:
The pitch mean squared error of training is: 2.518215. 10^{-2}
The pitch mean squared error of test is: 2.518215.10^{-2}
The yaw mean squared error of training is: 4.669712. 10^{-2}
The yaw mean squared error of test is: 4.669712. 10^{-2}

The black box model of the TRMS object with 2 degrees of freedom on Matlab/Simulink and the real model when using the Gradient descent with momentum and adaptive rate algorithm to train and test the network is shown in Figures 7 to 10.

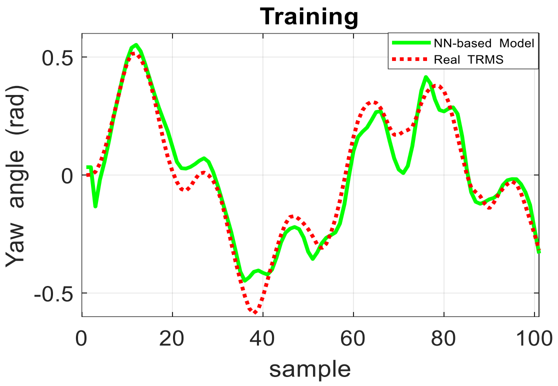

Fig. 7 The Yaw angle of the TRMS model when using the Gradient descent with momentum and adaptive rate algorithm to train the network

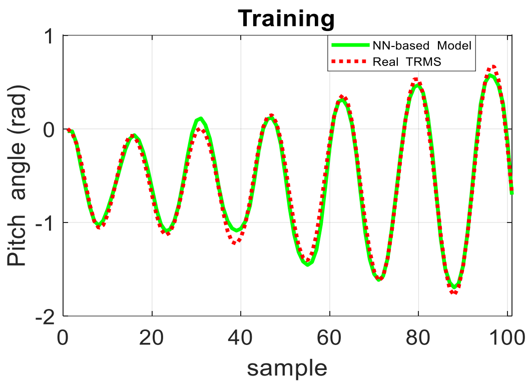

Fig. 8 The Pitch angle of the TRMS model when using the Gradient descent with momentum and adaptive rate algorithm to train the network

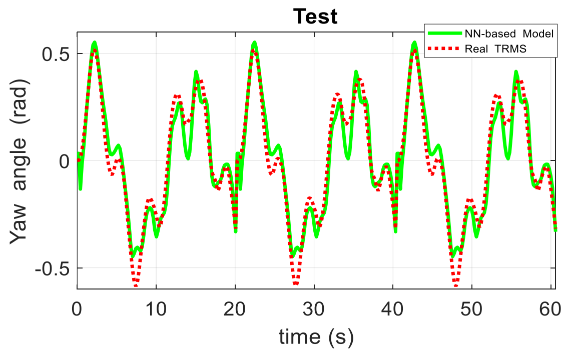

Fig. 9 The Yaw angle of the TRMS model when using the Gradient descent with momentum and adaptive rate algorithm to test the network

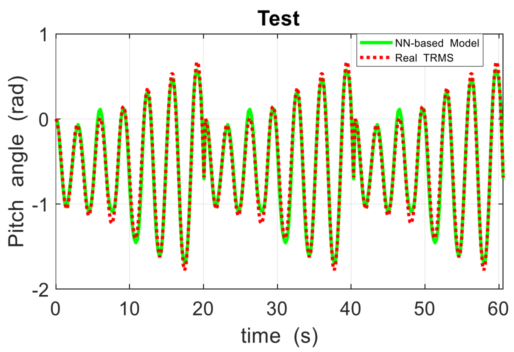

Fig. 10 The pitch angle of the TRMS model when using the Gradient descent with momentum and adaptive rate algorithm to test the network

Comment:

The pitch mean squared error of training is: 3.106907. 10^{-3}

The pitch mean squared error of test is: 3.106907.10^{-3}

The yaw mean squared error of training is: 5.579323. 10^{-3}

The yaw mean squared error of test is: 5.579323. 10^{-3}

The black box model of the TRMS object with 2 degrees of freedom on Matlab/Simulink and the real model when using the Conjugate gradient with Powell-Beale restarts algorithm to train and test the network is shown in Figures 11 to 14.

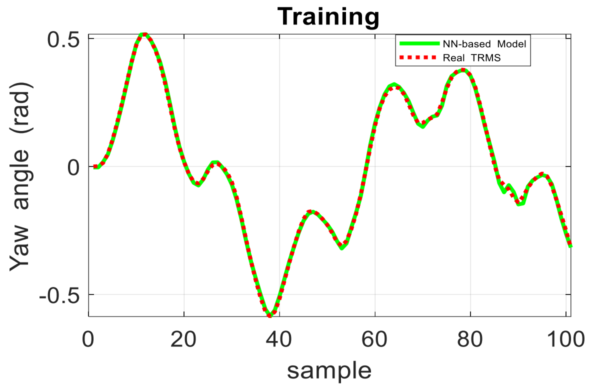

Fig. 11 The yaw angle of the TRMS model when using the Conjugate gradient with Powell-Beale restarts algorithm to train the network

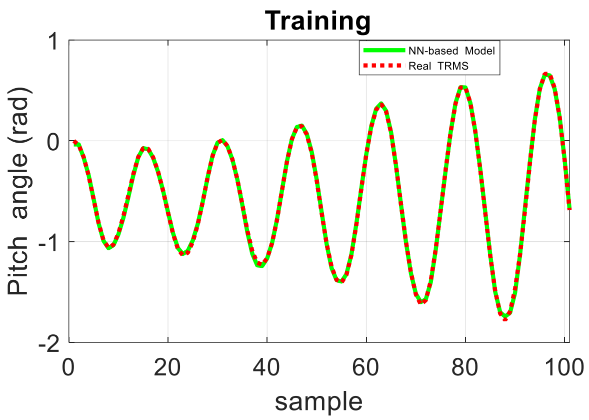

Fig. 12 The pitch angle of the TRMS model when using the Conjugate gradient with Powell-Beale restarts algorithm to train the network

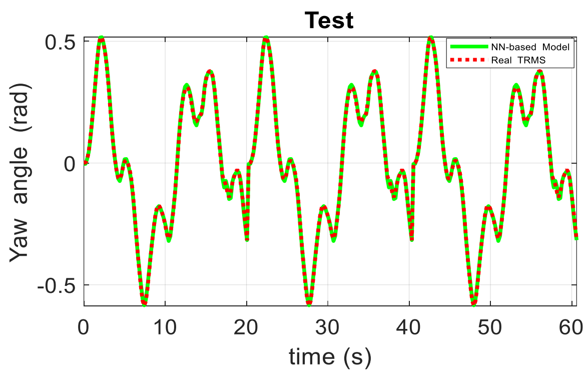

Fig. 13 The yaw angle of the TRMS model when using the Conjugate gradient with Powell-Beale restarts algorithm to test the network

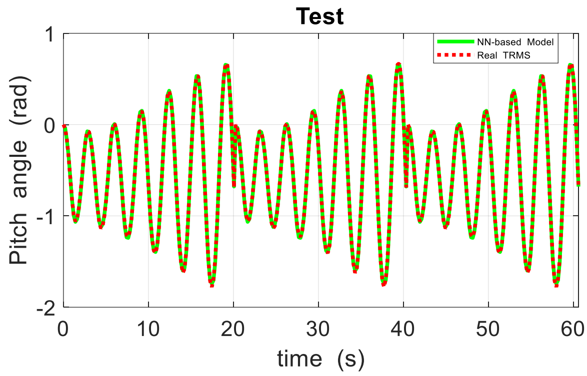

Fig. 14 The pitch angle of the TRMS model when using the Conjugate gradient with Powell-Beale restarts algorithm to test the network

Comment:

The pitch mean squared error of training is: 1.188296. 10^{-4}

The pitch mean squared error of test is: 1.188296.10^{-4}

The yaw mean squared error of training is: 6.483548. 10^{-5}

The yaw mean squared error of test is: 6.483548. 10^{-5}

The black box model of the TRMS object with 2 degrees of freedom on Matlab/Simulink and the real model when using the Conjugate gradient with Fletcher-Reeves algorithm to train and test the network is shown in Figures 15 to 18.

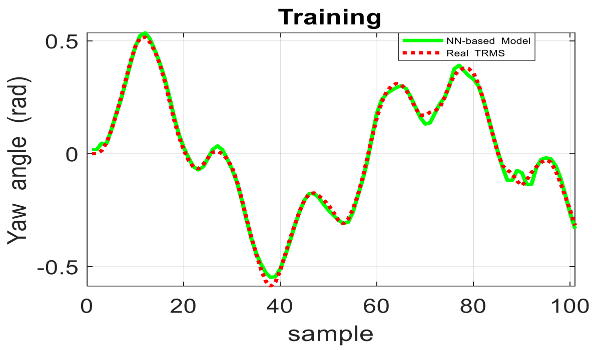

Fig. 15 The yaw angle of the TRMS model when using the Conjugate gradient with Fletcher-Reeves algorithm to train the network

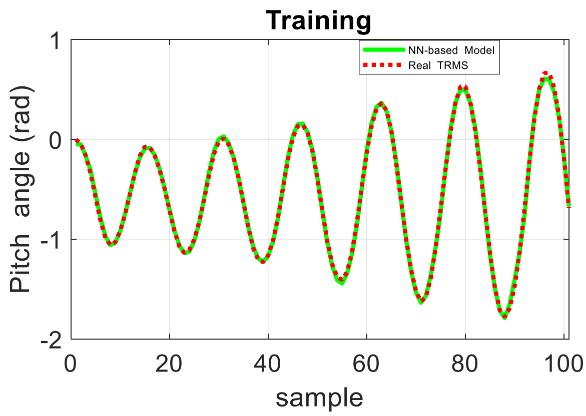

Fig. 16 The pitch angle of the TRMS model when using the Conjugate gradient with Fletcher-Reeves algorithm to train the network

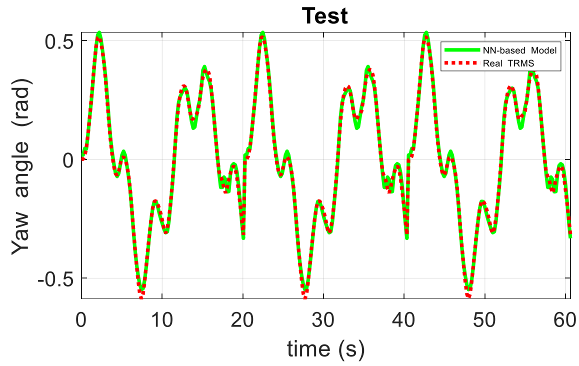

Fig. 17 The yaw angle of the TRMS model when using the Conjugate gradient with Fletcher-Reeves algorithm to test the network

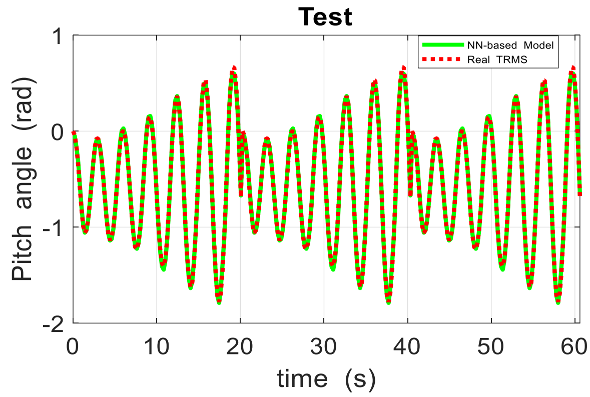

Fig. 18 The pitch angle of the TRMS model when using the Conjugate gradient with Fletcher-Reeves algorithm to test the network

Comment:

The pitch mean squared error of training is: 5.216955. 10^{-4}

The pitch mean squared error of test is: 5.216955.10^{-4}

The yaw mean squared error of training is: 3.315153. 10^{-4}

The yaw mean squared error of test is: 3.315153. 10^{-4}

The black box model of the TRMS object with 2 degrees of freedom on Matlab/Simulink and the real model when using the Conjugate gradient with Polak-Ribiere updates algorithm to train and test the network is shown in Figures 19 to 22.

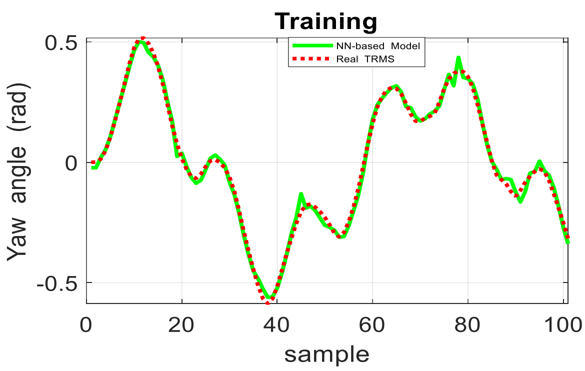

Fig. 19 The yaw angle of the TRMS model when using the Conjugate gradient with Polak-Ribiere updates algorithm to train the network

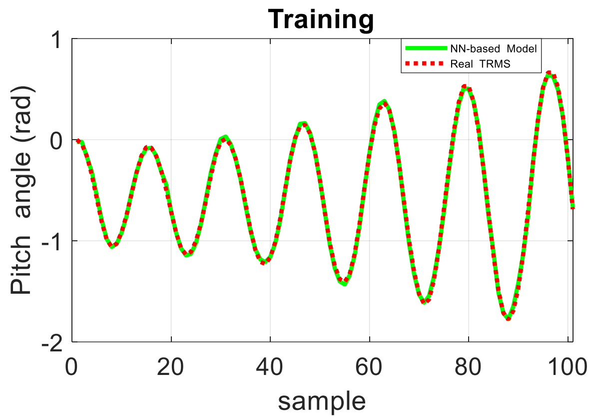

Fig. 20 The pitch angle of the TRMS model when using the Conjugate gradient with Polak-Ribiere updates algorithm to train the network

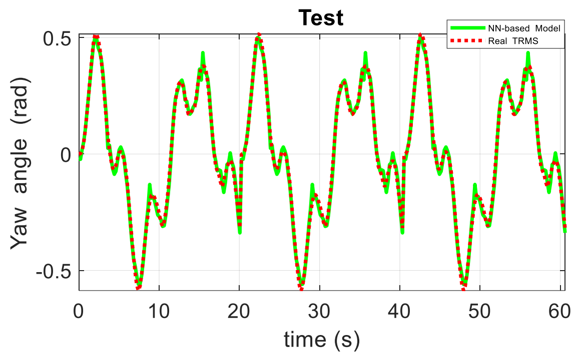

Fig. 21 The yaw angle of the TRMS model when using the Conjugate gradient with Polak-Ribiere updates algorithm to test the network

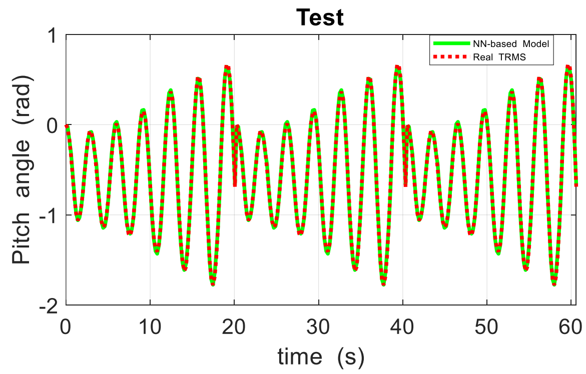

Fig. 22 The pitch angle of the TRMS model when using the Conjugate gradient with Polak-Ribiere updates algorithm to test the network

Comment:

The pitch mean squared error of training is: 3.889634. 10^{-4}

The pitch mean squared error of test is: 3.889634.10^{-4}

The yaw mean squared error of training is: 4.608945. 10^{-4}

The yaw mean squared error of test is: 4.608945. 10^{-4}

The black box model of the TRMS object with 2 degrees of freedom on Matlab/Simulink and the real model when using the Gradient descent with adaptive learning rate algorithm to train and test the network is shown in Figures 23 to 26.

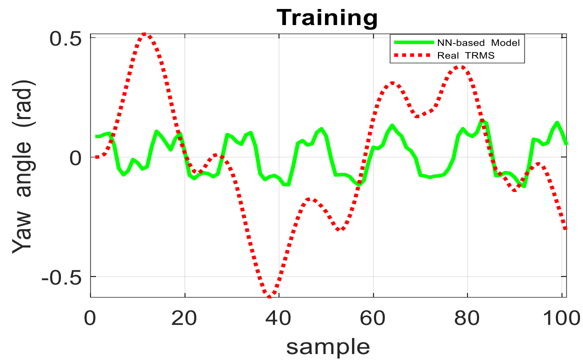

Fig. 23 The yaw angle of the TRMS model when using the Gradient descent with adaptive learning rate algorithm to train the network

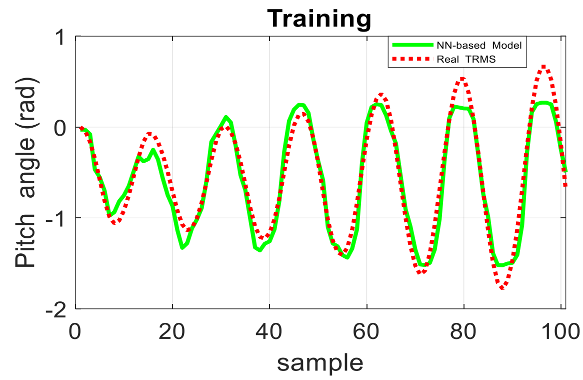

Fig. 24 The pitch angle of the TRMS model when using the Gradient descent with adaptive learning rate lgorithm to train the network

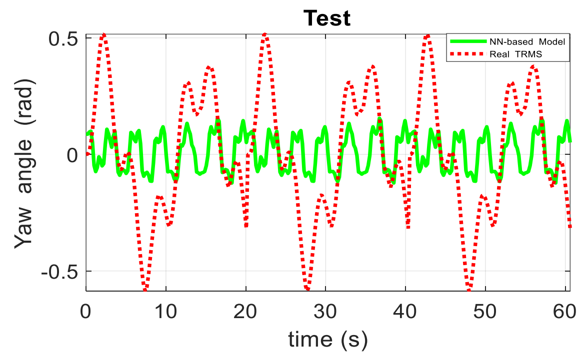

Fig. 25 The yaw angle of the TRMS model when using the Gradient descent with adaptive learning rate algorithm to test the network

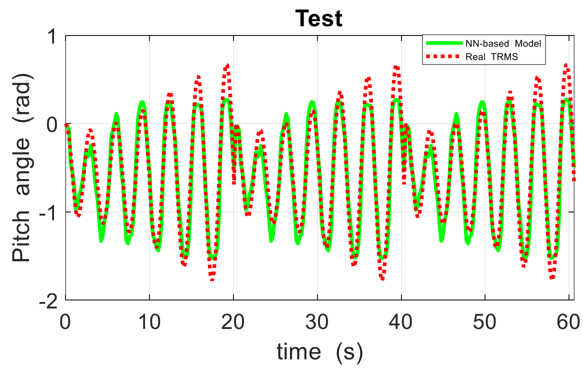

Fig. 26 The pitch angle of the TRMS model when using the Gradient descent with adaptive learning rate algorithm to test the network

Comment:

The pitch mean squared error of training is: 2.760583. 10^{-2}

The pitch mean squared error of test is: 2.760583.10^{-2}

The yaw mean squared error of training is: 6.769054. 10^{-2}

The yaw mean squared error of test is: 6.769054. 10^{-2}

The black box model of the TRMS object with 2 degrees of freedom on Matlab/Simulink and the real model when using the Scaled conjugate gradient algorithm to train and test the network is shown in Figures 27 to 30.

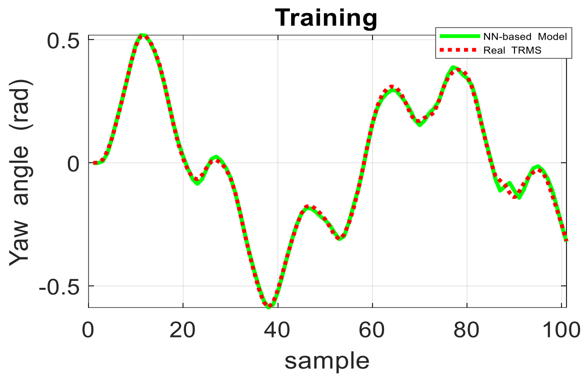

Fig. 27 The yaw angle of the TRMS model when using the Scaled conjugate gradient algorithm to train the network

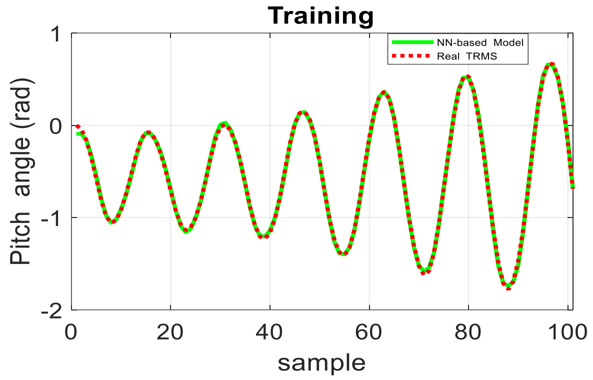

Fig. 28 The pitch angle of the TRMS model when using the Scaled conjugate gradient algorithm to train the network

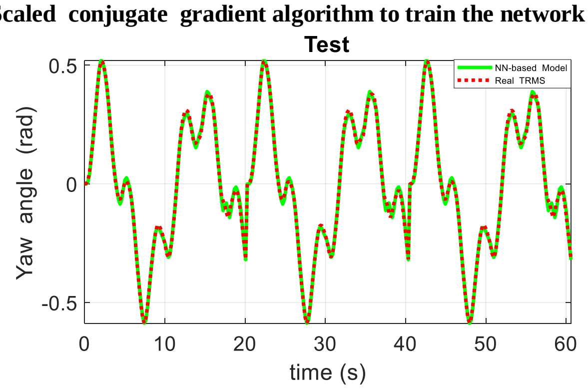

Fig. 29 The yaw angle of the TRMS model when using the Scaled conjugate gradient algorithm to test the network

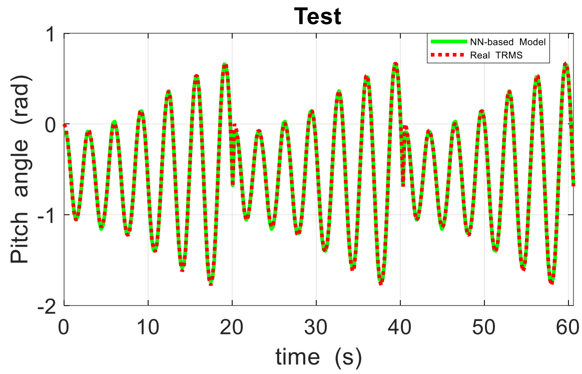

Fig. 30 The pitch angle of the TRMS model when using the Scaled conjugate gradient algorithm to test the network

Comment:

The pitch mean squared error of training is: 3.441739. 10^{-4}

The pitch mean squared error of test is: 3.441739.10^{-4}

The yaw mean squared error of training is: 1.333501. 10^{-4}

The yaw mean squared error of test is: 1.333501. 10^{-4}

Comment**:** The simulation results show that when using the same data sets with different training algorithms to recognize the TRMS object model for different results, the mean squared error between the models. The real object and recognition model of the training model and the test model decreases from 10^{-2} to 10^{-5}. With these datasets, if using the training algorithms are: Conjugate gradient with Powell-Beale restarts, Conjugate gradient with Fletcher-Reeves, Conjugate gradient with Polak-Ribiere updates or Scaled conjugate gradient, the squared deviation is about 10^{-4} or 10^{-5} are both acceptable.

V. CONCLUSION

Building a model of the TRMS according to the black box model using Neural network through the simulation on Matlab and comparing with the real model shows that the mean square error between the training and the testing model in both yaw and pitch angles when using the same network training structure and datasets, we can use the models using the training and testing algorithms: Conjugate gradient with Powell-Beale restarts, Conjugate gradient with Fletcher-Reeves, Conjugate gradient with Polak-Ribiere updates hoặc Scaled conjugate gradient because of small model bias, and the model trained and tested by the experts should not be used algorithm: Gradient descent, Gradient descent with momentum and adaptive rate or Gradient descent with adaptive learning rate because of large model error. The next research is expected that the authors will build a black box model or a gray box model thanks to GA identification in order to exploit and test modern control methods to control TRMS objects.

VI. ACKNOWLEDGEMENTS

This research was supported by the Thai Nguyen University of Technology, Viet Nam (http://www.tnut.edu.vn/), under grant number T2020-B72.

REFERENCES

[1] Twin Rotor MIMO System 33-220 User Manual, 1998 (Feedback Instruments Limited, Crowborough, UK).

[2] A. Rahideh, M.H. Shaheed, Mathematical dynamic modelling of a twin rotor multiple input–multiple output system, Proceedings of the IMechE, Part I. Journal of Systems and Control Engineering 221 (2007) 89–101.

[3] Ahmad, S. M., Shaheed, M. H., Chipperfield, A. J., and Tokhi, M. O. Nonlinear modelling of a twin rotor MIMO system using radial basis function networks. IEEE National Aerospace and Electronics Conference, (2000) 313–320.

[4] Ahmad, S. M., Chipperfield, A. J., and Tokhi, M. O. Dynamic modelling and optimal control of a twin rotor MIMO system. IEEE National Aerospace and Electronics Conference, (2000) 391–398.

[5] Shaheed, M. H. Performance analysis of 4 types of conjugate gradient algorithm in the nonlinear dynamic modelling of a TRMS using feedforward neural networks. IEEE International Conference on Systems, man and cybernetics, (2004) 5985–5990.

[6] Huong T.M. Nguyen, Thai. Mai.T, Anh. Do.T.T, Lai Lai K. (2014), Stabilization for Twin Rotor MIMO System based on BellMan's Dynamic Programming Method, Journal of science and Technology of Thai Nguyen University, 128(14) (2014) 161-165.

[7] Huong. Nguyen.T.M, Thai. Mai.T, Chinh. Nguyen. H, Dung. Tran.T and Lai. Lai.K, Model Predictive Control for Twin Rotor MIMO system, The University of Da Nang Journal of science and Technology, 12(85) (2014) 39 – 42 .

[8] Huong T.M. Nguyen, Thai Mai T., Lai Lai K., Model Predictive Control to get Desired Output with Infinite Predictive Horizon for Bilinear Continuous Systems, International Journal of Mechanical Engineeringand Robotics Research, 4(4) (2015) 299 - 303.

[9] Huong T.M. Nguyen, Interchannel Interference in While box Model of Twin Rotor MIMO System. SSRG International Journal of Electrical and Electronics Engineering (SSRG-IJEEE), 8(4) (2021) 36-40.