System identification of an unmanned quadcopter system using MRAN neural

IOP Conf. Series: Materials Science and Engineering **270** (2017) 012019 doi:10.1088/1757-899X/270/1/012019

System identification of an unmanned quadcopter system using MRAN neural

M F Pairan and S S Shamsudin*

Department of Aeronautical Engineering, Faculty of Mechanical and Manufacturing, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Johor, Malaysia.

*syafiq@uthm.edu.my

Abstract. This project presents the performance analysis of the radial basis function neural network (RBF) trained with Minimal Resource Allocating Network (MRAN) algorithm for real-time identification of quadcopter. MRAN's performance is compared with the RBF with Constant Trace algorithm for 2500 input-output pair data sampling. MRAN utilizes adding and pruning hidden neuron strategy to obtain optimum RBF structure, increase prediction accuracy and reduce training time. The results indicate that MRAN algorithm produces fast training time and more accurate prediction compared with standard RBF. The model proposed in this paper is capable of identifying and modelling a nonlinear representation of the quadcopter flight dynamics.

1. Introduction

Quadcopter is a type of rotorcraft that consists of four main fixed pitch rotors and does not require any complex mechanical control mechanism for its propellers. Because of this, the mechanical system of quadcopter is much easier to maintain compared to conventional helicopter. Quadcopter has a unique capability of taking off and landing vertically, besides hovering and cruising at lower speed compared to fixed wing unmanned aerial vehicle (UAV). Due to these advantages, quadcopter has attracted very strong interest worldwide and has been used in both military and civil applications such as in real-time reconnaissance surveillance, aerial photography, delivery services and also traffic monitoring. These applications require the quadcopter to fly in close proximity relative to the targets and therefore, need the application of a robust control system.

A robust controller is capable of adapting to change in dynamics and dealing with the quadcopter's uncertainty parameters. A robust control system is of utmost importance for autonomous flight and a robust controller is designed to provide accurate flight performance within the acceptable disturbance ranges [1]. It is a challenging process to identify an accurate dynamics model for a quadcopter since it is highly nonlinear, has multi-input multi-output (MIMO) characteristics, strongly coupled, under-actuated and has an unstable system [2,3]. Development of a dynamics model using the first principle of modelling based on Newton's law and Euler's angles are inaccurate, time-consuming and require deep mathematical and physical calculations about the system behaviour [4–8]. On the other hand, the system identification (SI) provides a method to derive dynamics models based on experimental data of the quadcopter control inputs and measured outputs. Neural networks (NN) based SI approach is an alternative for modelling techniques for quadcopter and it does not rely on any specific mathematical model. The NN approach is typically consisted of a flexible model structure and is characterized by a

limited number of NN parameters. NN models are trained using a large number of efficient training methods with high prediction accuracy [8–11]. Nevertheless, this NN modelling method also has few disadvantages such as longer required training time, slow convergence rate and vulnerability to over-fitting problems.

Recursive training algorithms with a good selection of NN model structure can notably improve the prediction accuracy, reduce the NN training time and avoid data over-fitting [8,11–13]. The usage of recursive algorithms like Kalman Filter or recursive Gauss-Newton (rGN) with either Constant Trace (CT) or EFRA variation can be applied to reduce the computation complexity of the offline (batch) training method. However, the correct selection of hidden neurons remains to be a vital parameter to be considered since either excessive or insufficient neuron can lead to a poor prediction performance. Typical selection of hidden neurons is based on the trial and error method or rule of thumb approaches [11,14,15]. After several attempts or trials, the deduced number of hidden neurons obtained from the trial and error method is then used to estimate the weights using NN training algorithms [16–19]. In order to avoid the troublesome selection of hidden neurons to be used in NN model, automatic neurons selection, network growing and pruning strategy can also be employed to remove hidden neurons that consistently made little contributions to the network outputs. Fully-tuned radial basis function (RBF) with MRAN recursive training algorithm is proposed in this study to overcome the large numbers of hidden neurons and parameters selection dilemma. The MRAN algorithm introduces network growing and hidden neuron pruning approach to automatic tuning the neuron based error criteria thus reducing training time and over-fitting problem. In MRAN learning approach, new training sample is presented in recursive mode while all RBF network parameters will be continuously updated [20,21].

This paper presents the recursive system identification method for a quadcopter based on the RBF neural network structure. RBF is trained with Constant Trace (RBF-CT) and MRAN (RBF-MRAN) algorithm to model attitude dynamics of the quadcopter. Attitude output and pilot stick input during flight are recorded and used as training and validation data for the neural network. MRAN algorithm aims to reduce the training time, increase prediction accuracy and produce the minimal structure of the standard RBF. For this purpose, the addition and removal neuron strategy is incorporated into the RBF architecture. To assess the performance of the proposed model, RBF trained with MRAN algorithm is compared with the standard RBF structure trained with Constant Trace algorithm (CT) with regards to the percentage of root mean square error (RMSE), the coefficient of determination (R^2) criteria and the mean training time. The RBF-CT structure is selected for comparison because it has the same basic structure (three fixed layer), activation function (radial basis function) and also optimizing parameter method (CT algorithm).

2. Quadcopter Platform and Avionics System

The XuGong v2 Profoldable quadcopter shown in Figure 1, which is manufactured by ImmersionRC Ltd, has been chosen as the rotorcraft UAV platform for this project. The quadcopter's main frame is made from lightweight carbon fiber with a weight of 96g and it is capable of approximately producing 20 minutes flight time. It is also equipped with four AIR 2213 KV920 (rpm/V) brushless motors with T9545 plastic self-tightening propellers and is controlled by four AIR20A electronic speed controller (ESC). Radio transmitter controller used for this quadcopter is FrSkyTaranis X9D 2.4 GHz transmitter with FrSky X8R receiver. In this study, the DJI NAZA v2 is used as the main flight controller for the XuGong quadcopter. The XuGong quadcopter platform is primarily selected because of its main frame compatibility, open source and its capability of future upgrades. It is also equipped with crash-resistant arms that will minimize the damage during hard landing and crash. These arms contain a break-away pin that will take the crash impact by bending back into the original shape without breaking the arm.

The data acquisition system is developed to record the desired attitude data during flight to assess the performance of the proposed system identification method. The collected data is recorded on the flash drive as text file format. The overview of the data acquisition for the onboard system and ground station is depicted in Figure 2, which consists of a computer as the ground station, a VN-100 Inertia Measurement Unit (IMU) and National Instrument (NI) myRIO 1900, which is a compact and portable

reconfigurable I/O device with built-in wireless devices that is used as a data logger. MyRIO can be configured in NI LabVIEW software as the wireless host that provides 10 analogue inputs (AI), six analogue outputs (AO), 40 digital inputs and outputs (DIO) and power output (3.3V and 5V). The VN-100 IMU consists of 3-axis accelerometers, 3-axis gyros, 3-axis magnetometer, a barometric pressure sensor and a 32-bit processor. VN-100 is capable of computing data in a real-time continuously over a 360-degree range of motion with no limitation on the angles and it also has a built-in microcontroller that runs a quaternion based Extended Kalman Filter (EKF), which offers faster and smoother attitude estimates and angular rate measurements.

Figure 1: XuGong quadcopter platform

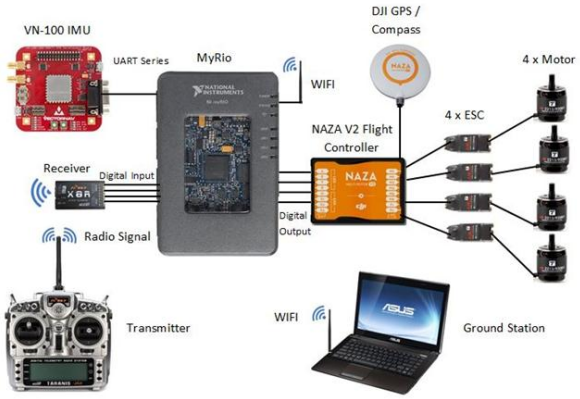

Figure 2: Quadcopter system identification setup

The task of the onboard system is to collect all necessary flight data from the IMU and receiver while the ground station is used to monitor the quadcopter's flight data during flight. Figure 3 shows a detailed overview of onboard SI system for the quadcopter under development. National Instrument myRIO 1900 board is used to collect flight information from the IMU through serial UART and store the data in a flash drive. The FPGA in the myRIO 1900 board is used to collect all control stick input signals from the receiver and these signals will be transferred to real-time processor for data saving.

3. System Identification based on MRAN Algorithm

Figure 4 illustrates the overall process of system identification based on neural network model. The RBF neural network model is located in parallel with the actual quadcopter plant where NN inputs are from actual inputs and outputs of the quadcopter. The NN model is trained with inputs data to predict the output of the quadcopter plant. The error between the predicted outputs of the RBF model and the outputs of the quadcopter is used in training algorithm to adjust the NN parameters. The overall neural network modelling approach consists of the flight test data gathering process, neural network model structure selection, and the model estimation and validation.

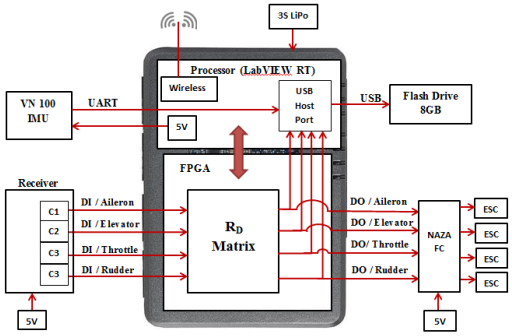

Figure 3: Details of onboard system

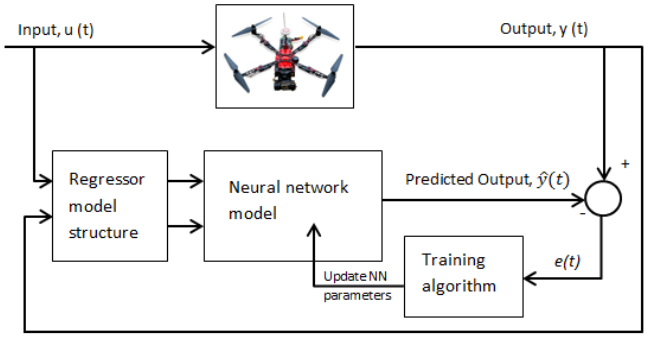

Figure 4: System identification based on neural network

3.1. Flight Test Data Collection

Flight test data collection is the experimental method where the input-output data pair is obtained from the quadcopter onboard system. Flight data collection is an important step to be conducted prior to the actual implementation of real-time system identification. For training and validation data collection, the frequency sweep excitation signals that have a quasi-sinusoidal shape of increasing frequency with variable amplitude are given to the quadcopter input [22,23]. The frequency sweeps signal issued by a pilot will start at low frequency and increases up to the higher frequency as desired by the user. Flight data for the quadcopter is collected for attitude information only. During the experiment, all control inputs and state measurements are recorded at a 50Hz sampling rate and the noises are filtered through a low-pass filter with 10Hz cut-off frequency. The data acquisition system on the quadcopter measures the Euler angles (i.e. roll, Φ; pitch, θ and yaw, ψ) and the angular rates (i.e. roll rate, p; pitch rate, q and yaw rate, r) as the output of the system. The control inputs associated with pilot commands are defined as elevator stick, δ_{lon}; rudder stick, δ_{pedal} and aileron stick, δ_{lat}. They are collected as input of the system and are stored in a flash drive. Table 1 tabulates the output responses based on frequency swept for each input channel for flight test data collection.

Table 1: Output responses from input channel

Input channel	**Euler angles**	**Angular rates**
δ_{lat}	Φ	p
δ_{lon}	θ	q
δ_{ped}	ψ	r

3.2. Model Structure Selection

A model structure selection is a step where the model to represent the relationship between inputs and outputs of a system is chosen. The model output should replicate the expected value of the true system output. The most preferred model structure is the one with capability to predict accurately the actual flight dynamics from a set of test data. In this study, the neural network representation of nonlinear Auto Regressive eXogenous inputs (ARX) model structure is used to predict the relationship between input-output variables.

As indicated in Figure 5, RBF architecture with one hidden layer is chosen to learn the nonlinear relationship of NNARX model. Input-output relationship (i.e. coefficient of ARX model) and lagged input-output data are expressed in terms of parameter vector, θ and time regression vector, φ. For a single hidden layer case, the outputs formulation from the output layer of an RBF network is given by Equation 1 to Equation 4, where $w1_{kj}$ is the weights between hidden and output layers, μ_j and σ_j refer to centre and width of the Gaussian function of jth hidden layer, h denotes the number of neurons in the hidden layer while B_k is the bias elements for output layer. The number of inputs, hidden and outputs of neural network is presented by i, j and k, respectively. Parameter vector of RBF contains all RBF parameters (centres, weights, widths and bias) and regression vector consists of past input and output. The best regressor of RBF structure for system identification that produces low percentage of RMSE and short mean training time based on trial and error method are selected in this study [11].

$$\hat{y}(t) = B_k + \sum_{j=1}^{h} w1_{kj}\left(exp\left(-\frac{1}{{\sigma_j}^2}\left\|x-\mu_j\right\|^2\right)\right) \tag{1}$$

$$\hat{y}(t+k|t,\theta) = g(\varphi(t+k),\theta) \tag{2}$$

$$\theta = \left[w1_{kj}\mu_j\sigma_j B_k\right] \tag{3}$$

$$\varphi = [\varphi_1\varphi_2 \ \cdots \ \varphi_i] = \left[y(t-1) \ \cdots \ y\left(t-n_y\right) \ \ u(t-1) \ \cdots \ u(t-n_u)\right] \tag{4}$$

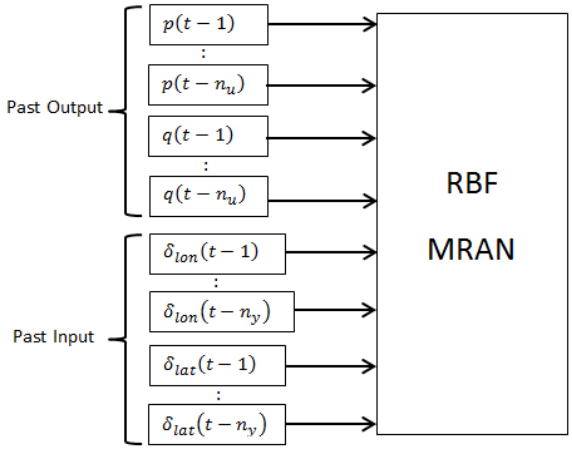

Figure 5: Regressor structure for RBF and MRAN

3.3. Model Estimation

The third step is to determine the NN parameters and minimize the error between the identified system and true model. The system collects the measurement data in each time step and, at the same time, a new model of the system is formed. In this study, RBF NN structure is trained with recursive training algorithm of Constant Trace (CT) and MRAN algorithm. Detailed methodology of MRAN algorithm is shown in Figure 6. The MRAN algorithm calculates the output of the RBF network with three error criteria, which are used to check if a new hidden unit should be added from the existing network. If all three criteria are satisfied, one neuron will be added and the algorithm will remove any hidden neurons that have the least contribution to the RBF output. If the error criteria do not satisfy any specification, the algorithm routine will proceed with Jacobian matrix and CT training calculation. After that, the MRAN method proceeds to prune step and continues with another set of input-output data pair.

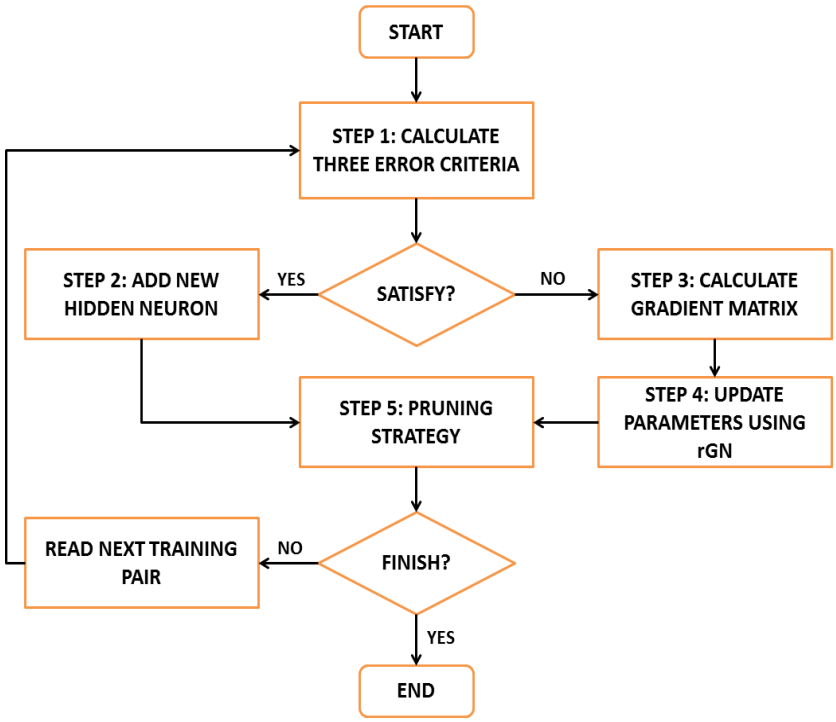

Figure 6: Summary of the MRAN algorithm flow

The first step of the algorithm is to check if the error criteria satisfy the condition for the addition of a new hidden unit. Error criteria E_1, E_2 and E_3 are user-defined thresholds that need to be satisfied to trigger the addition of a new hidden neuron condition. The calculations of the three error criteria are as given by Equation 5, Equation 6 and Equation 7, where e_t is error prediction, y_t is the output from measurement quadcopter output, $\hat{y}_t$ is the RBF prediction output, μ_{tr} is the centre of the hidden unit that is closest to the current input, γ is decay constant, i is number of iterations, eta_{max} and eta_{min} are respectively the maximum and minimum threshold to bound value of E_3, x_t is current input, N_w is sliding window size and e_j is current prediction error.

$$\|e_t\| = \|y_t - \hat{y}_t(x_t)\| > E_1 \tag{5}$$

$$e_{rmsi} = \sqrt{\sum_{j=t-(N_w-1)}^{t} \frac{\|e_j\|^2}{N_w}} > E_2 \tag{6}$$

$$d_t = \|x_t - \mu_{tr}\| > E_3 = max\{eta_{max}\gamma^i, eta_{min}\} \; 0 < \gamma < 1 \tag{7}$$

When all the criteria in the first step are met, a new hidden neuron is added with initial parameters associated with it in the second step. This is given by Equation 8, Equation 9 and Equation 10 where $\underline{w}_{h+1}^T$, μ_{h+1}, σ_{h+1} are respectively the initial parameters value of weight, center and width for new hidden neuron, e_t is current output error, x_t is the current input introduced into the network and k_p is the overlap factor that determines the overlap of the responses of the hidden units in the input space.

$$\underline{w}_{h+1}^T = e_t \tag{8}$$

$$\mu_{h+1} = x_t \tag{9}$$

$$\sigma_{h+1} = k_p\|x_t - \mu_{tr}\| \tag{10}$$

If any of the three criteria for adding a new hidden unit cannot be satisfied, the Jacobian matrix is calculated. The Jacobian matrix, Π is calculated by partial differentiating Equation 1 with respect to parameter vector, θ. The result from gradient vector calculation is shown by Equation 11.

$$\Pi = \begin{bmatrix} \mathrm{I}, \phi_1(x_t)\mathrm{I}, \phi_1(x_t)(2\,w_1^T/\sigma_1^2)(x_t-\mu_1)^T, \phi_1(x_t)(2\,w_1^T/\sigma_1^3)\|x_t-\mu_1\|^2, \cdots, \\ \phi_1(x_h)\mathrm{I}, \phi_h(x_t)\big(2\,w_h^T/\sigma_h^2\big)(x_t-\mu_h)^T, \phi_h(x_h)\big(2\,w_h^T/\sigma_h^3\big)\|x_h-\mu_h\|^2 \end{bmatrix}^T \tag{11}$$

Next, all RBF parameters $\theta = \left[w_0, w_1, \mu_1^T, \sigma_1, \cdots, w_h, \mu_h^T, \sigma_h\right]^T$ are updated using the CT algorithm. CT method is proposed to prevent covariance blow-up without loss of the tracking ability by limiting the eigenvalues value [13]. The CT training algorithm is shown by Equation 12 to Equation 15.

$$K(t) = P(t-1)\Pi(t)\big(1 + \Pi^T(t)P(t)\Pi(t)\big)^{-1} \tag{12}$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)e_t \tag{13}$$

$$\bar{P}(t) = P(t-1) - K(t)\Pi^T(t)P(t-1) \tag{14}$$

$$P(t) = \frac{\alpha_{max} - \alpha_{min}}{trace\big(\bar{P}(t)\big)}\bar{P}(t) - \alpha_{min}I \tag{15}$$

Next, in the pruning step, the MRAN algorithm will remove neurons that make little contribution to the RBF output for N_w consecutive observations. The output, y_{nj} (j = 1 ... n_y) of the n^{th} hidden neuron is

normalized with the highest output obtained in order to reduce inconsistency in the output data. The calculation in this step is made using Equation 16 and Equation 17.

$$y_{nj} = w_{nj}\, exp\left(-\frac{1}{{\sigma_n}^2}\|x - \mu_n\|^2\right) \tag{16}$$

$$r_{nj} = \frac{y_{nj}}{max\left\{y_{1j}, y_{2j}, \cdots, y_{hj}\right\}} \tag{17}$$

The neuron is removed if r_{nj} value decreases below desired threshold value, δ for N_w consecutive inputs. After this pruning step, next input-output data pair is trained and the overall calculation routine is repeated.

3.4. Model Validation

One step ahead (OSA) prediction and mean square error (MSE) are used to validate the performance of the proposed network models. One step ahead prediction is a simple plot that compares the actual measurement data with the model prediction over a test data set. The overall accuracy of the prediction error is calculated according to the percentage of root mean square error (RMSE) and R^2 criterion as given by Equation 18 and Equation 19, respectively. The training performance of the NN models is gauged using the mean training time in each iteration.

$$\%RMSE = \sqrt{\frac{\sum(\hat{y}_i(t) - y_i(t))^2}{\sum(y_i(t) - \bar{y}_i(t))^2}} \tag{18}$$

$$R^2 = 1 - \frac{\sum(\hat{y}_i(t) - y_i(t))^2}{\sum(y_i(t) - \bar{y}_i(t))^2} \tag{19}$$

4. Results and Discussion

The performance of the proposed RBF architecture trained with MRAN algorithm is compared to the standard RBF to determine whether the proposed RBF model is capable of improving the prediction performance over the standard RBF model. The NN models are trained to predict the future response of the quadcopter attitude dynamics with two output variables i.e. roll and pitch rates. The best model structures for RBF and MRAN are selected for comparison by trial and error approach as tabulated in Table 2.

Table 2: Regression structure selection

NN model	Regression structure	Roll rate (%RMSE)	Pitch rate (%RMSE)	Mean training time
RBF-CT	1-1	20.39	20.59	6.71
	2-1	18.33	22.77	5.39
	2-2	17.04	23.05	12.42
	3-1	16.69	16.86	6.97
	3-2	18.92	21.23	16.96
	3-3	22.62	24.04	23.72
RBF-MRAN	1-1	12.61	16.42	2.34
	2-1	12.58	16.65	2.11
	2-2	12.73	15.55	6.35
	3-1	10.99	13.53	5.23
	3-2	10.31	14.86	6.18
	3-3	9.81	15.13	12.56

In this study, both RBF structures are selected with three past outputs and one past input lag structure for system identification because this structure type provides the lowest percentage of RMSE and the shortest mean training time. Table 3 displays all RBF parameters used for system identification of the quadcopter. The MRAN training parameters selected for this study using the trial and error method are shown in Table 4.

Table 3: RBF NN parameters

Network Specifications	**RBF-CT**	**RBF-MRAN**
Number of hidden neuron	5	1 (initial)
Number of past output	3	
Number of past input	1	
Activation function in hidden layer	Radial basis function	
Activation function in output layer	Linear function	
CT parameter	$\boldsymbol{\alpha_{max}}$= 10 α_{min} = 0.001	$\boldsymbol{\alpha_{max}}$= 20 $\boldsymbol{\alpha_{min}}$ = 0.1
Number of regressors	8	

Table 4: MRAN training parameters

E_1	E_2	Eta_{max}	Eta_{min}	γ	NW	K_p	$Threshold$
0.04	0.2	4	2	0.96	15	0.3	0.001

Table 5 shows the comparison of the results of percentage RMSE, R^2, mean training time and final neuron of RBF and MRAN structure to identify roll and pitch rate of the quadcopter. From the table, it can be seen that MRAN algorithm produces better prediction accuracy with an RMSE of 10.99% and 13.53% for roll rate and pitch rate, respectively, compared to standard RBF. This table also indicates that the R^2 value of MRAN is closer to one and offers the best fit with the actual output compared to RBF trained with CT. Table 5 also highlights that MRAN structure offers faster system identification with 5.23 ms mean training time compared with RBF-CT (6.97 ms). This can be attributed to the fact that MRAN requires only four hidden neurons compared to five fixed RBF-CT that resulted in faster completion of the training process.

Table 5: Overall RBF and MRAN prediction performance

NN model	**Percentage of RMSE (%)**		**R^2**		**Mean training time(ms)**	**Final hidden neuron (unit)**
	Roll Rate (q)	**Pitch Rate (p)**	**Roll Rate (q)**	**Pitch Rate (p)**		
RBF	16.69	16.86	0.9696	0.9690	6.97	5
MRAN	10.99	13.53	0.9868	0.9801	5.23	4

Figure 7 presents the hidden neuron growth history along with MRAN training. Hidden neurons are added and pruned to obtain the best RBF structure throughout the sampling time. This result implies that the MRAN model adapts itself to the changes in flight dynamics very well. It is apparent that the proposed RBF model has achieved the maximum growth of hidden neurons at eight units before being reduced and stabilized at four hidden neurons.

Furthermore, the results obtained by one step ahead prediction responses from the RBF and MRAN model are respectively plotted against the actual attitude responses of the quadcopter in Figure 8 and

Figure 9. The red and green solid line represents the predicted output from the MRAN and RBF-CT, respectively, meanwhile the black line represents the measured rate from IMU. The predicted outputs from MRAN and RBF-CT are capable to track the actual attitude measurement of the quadcopter. As can be observed in Figure 8 and 9, MRAN outputs closely match the actual measured outputs of the quadcopter compared with that of the RBF-CT. Therefore, when there are sudden changes in attitude dynamics, MRAN is capable of following and tracking the change of the dynamics. Results from the graph tally with the RMSE and R^2 results that indicate MRAN is better than RBF-CT in predicting the attitude dynamics of the quadcopter.

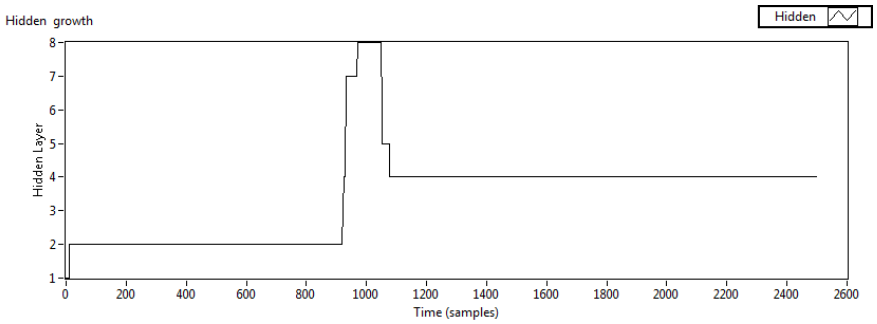

Figure 7: MRAN hidden neuron growth

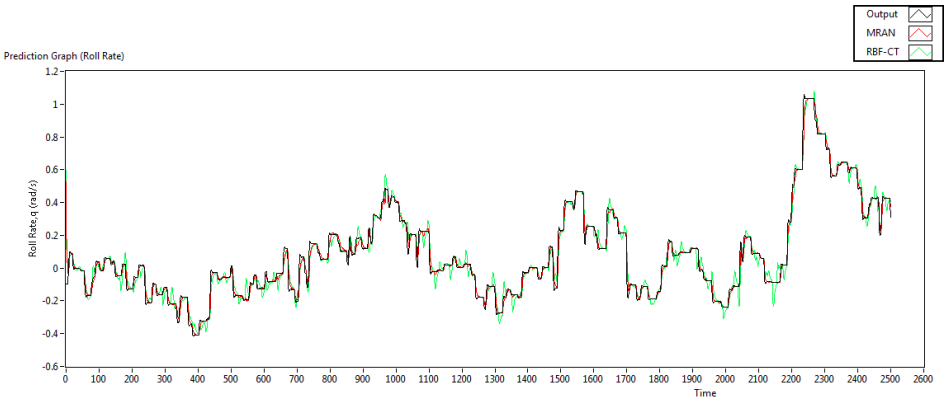

Figure 8: Prediction graph of roll rate for RBF-CT and MRAN

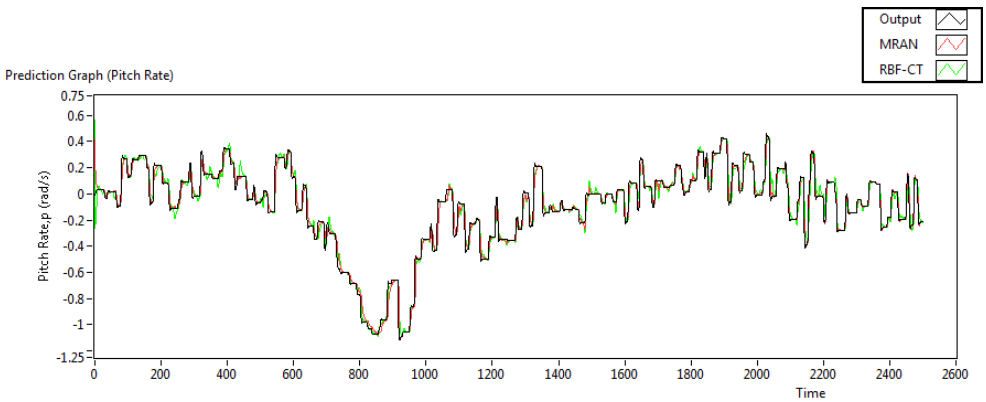

Figure 9: Prediction graph of pitch rate for RBF-CT and MRAN

5. Conclusion

In this study, nonlinear attitude dynamics of the Xugong quadcopter is identified from the flight data using recursive neural network approach. The RBF and MRAN models are proposed for the real-time

system identification implementation. Fully-tuned RBF network is proposed in this project to improve prediction performance and training time of standard RBF model. The proposed MRAN model also offers advantages in the automatic tuning of RBF model parameters, which includes neurons selection as part of training updates. The real-time has predicted the outputs from the proposed neural network models and these outputs are compared with the actual output responses of the Xugong quadcopter for performance assessment. Validation of the results conducted in this study confirmed that the proposed MRAN model is suitable for modelling the quadcopter's attitude dynamics correctly with improvement in training time. The MRAN algorithm offers good attitude prediction with minimal NN structure (less neuron) and shorter training time compared to RBF-CT. Dynamics models obtained from the MRAN model can be further used for research on stability, control and design of the adaptive flight control for autonomous flight of a quadcopter.

Acknowledgement
This research is funded by research grant from the Ministry of Science, Technology and Innovation (MOSTI) and is technically supported by members of Aircraft System and Design Research (ASDR), UTHM.

References
[1] Zulu A and John S 2014 *Open J. Appl. Sci.* **4** 547–56
[2] Zhang X, Li X, Wang K and Lu Y 2014 *Abstr. Appl. Anal.* **2014** 320526
[3] McKerrow P 2004 *IEEE International Conference on Robotics and Automation*
[4] Abas N, Legowo A and Akmeliawati R 2011 *4th International Conference On Mechatronics*
[5] Putro I E, Budiyono A, Yoon K J and Kim D H 2008 *IEEE International Conference on Robotics and Biomimetics*
[6] Wei W 2015 *Development of an Effective System Identification and Control Capability for Quadcopter UAVs* PhD Thesis University of Cincinnati
[7] Tischler M and Remple R K 2006 *Aircraft and Rotorcraft System Identification: Engineering Methods with Flight Test Examples* American Institute of Aeronautics and Astronautic
[8] Shamsudin S S and Chen X 2014 *Int. J. Intell. Syst. Technol. Appl.* **13** 56–80
[9] Zurada J M 1996 *IEEE Aerospace Applications Conference*
[10] Lawryńczuk M 2014 *Computationally Efficient Model Predictive Control Algorithms* Springer International Publishing Switzerland
[11] Shamsudin S S and Chen X 2012 *Int. J. Model. Identif. Control* **17** 223–41
[12] Hunter D, Yu H, Member S, Pukish M S, Kolbusz J and Wilamowski B M 2012 *IEEE Trans. Ind. Informatics* **8** 228–40
[13] Shamsudin S S 2013 *The Development of Neural Network Based System Identification and Adaptive Flight Control for an Autonomous Helicopter System* PhD Thesis University of Canterbury
[14] Panchal F S and Panchal M 2014 *Int. J. Comput. Sci. Mob. Comput.* **3** 455–64
[15] Peyada N K and Ghosh A K 2009 *AIAA Atmospheric Flight Mechanics Conference*
[16] Bansal S, Jiang F J and Tomlin C J 2016 *IEEE Conference on Decision and Control*
[17] Dief T N and Yoshida S 2015 *IRACST - Int. J. Comput. Sci. Inf. Technol. Secur.* **5** 314–19
[18] Zhong H, Li S, Wang Y and Liu H 2016 *IEEE/ASME Int. Conf. Mechatron. Embed. Syst. Appl.*
[19] Wu J, Peng H, Chen Q and Peng X 2014 *ISA Trans.* **53** 173–85
[20] Suresh S, Sundararajan N and Saratchandran P 2008 *Neurocomputing* **71** 1345–58
[21] Sundararajan N, Saratchandran P and Li V 2002 *Fully Tuned Radial Basis Function Neural Networks For Flight Control* Kluwer Academic Publishers
[22] Cai G, Taha T, Dias J and Seneviratne L 2016 *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **231** 30-46
[23] Adiprawita W, Ahmad A S and Sembiring J 2007 *J. Bionic Eng.* **4** 237–44