Real-time System Identification of Unmanned Aerial Vehicles: A Multi-Network Approach

Vishwas Puttige, Sreenatha Anavatti
School of ACME, UNSW@ADFA, Australia
Email: v.puttige@adfa.edu.au, agsrenat@adfa.edu.au

Abstract— **In this paper, real-time system identification of an unmanned aerial vehicle (UAV) based on multiple neural networks is presented. The UAV is a multi-input multi-output (MIMO) nonlinear system. Models for such MIMO system are expected to be adaptive to dynamic behaviour and robust to environmental variations. This task of accurate modelling has been achieved with a multi-network architecture. The multi-network with dynamic selection technique allows a combination of online and offline neural network models to be used in the architecture where the most suitable outputs are selected based on a given criterion. The neural network models are based on the autoregressive technique. The online network uses a novel training scheme with memory retention. Flight test validation results for online and offline models are presented. The multi-network dynamic selection technique has been validated on real-time hardware in the loop (HIL) simulation and the results show the superiority in performance compared to the individual models.**

I. INTRODUCTION

In the recent past, unmanned aerial vehicles (UAVs) have been deployed for various defence operations. At the moment, use of UAVs for commercial civilian applications are widely being explored. The interdisciplinary nature of UAV control systems is attracting researchers from all fields of engineering. Most of the available research on UAVs is generally based on simulations [1]–[5]. Hence there is limited research available on low-cost, small-sized fixed wing UAVs for commercial applications. Currently, there are few universities working towards the development and implementation of different algorithms on UAVs in real-time [6]–[8]. One such fixed-wing platform is under development at the School of Aerospace, Civil and Mechanical Engineering (ACME) in UNSW@ADFA.

The UAV is a six degree of freedom (DOF), nonlinear, complex system. UAVs display inertial coupling and aerodynamic coupling effects and hence can be modelled as a MIMO system with six outputs and a minimum of four inputs. At the School of ACME, low-cost, radio controlled model planes are built to carry desired payloads forming the test-bed for various system identification and control experiments. Sensors and actuators are mounted on-board the UAV to provide information regarding change in behaviour of the platform during the course of flight. Behaviour of the UAV is quantified mainly in terms of the translational velocities and the angular rates with respect to a fixed frame of reference. The data from the sensors is processed and recorded on an on-board computer. This platform with the sensors and on-board computer is flown remotely to collect the six DOF data from it. The data from the sensors and the actuators are used to develop UAV simulation models and thereby used in the design of flight control system. The accuracy of the designed controller depends largely on the quality of the data collected. Hence, the sensors are calibrated at different conditions to obtain superior quality of data.

The flight control system (FCS) for the UAV performs tasks similar to those executed by a pilot for manned aircrafts. Robust control techniques, capable of adapting themselves to the changes in dynamics of the platform are necessary for the autonomous flight. Such controllers can be developed with the aid of suitable system identification (ID) techniques. This system ID based on flight data is also necessary for understanding the dynamics of the UAVs. Numerous system identification techniques have been proposed for the modelling of nonlinear systems. Fuzzy identification [9], state space identification [10], frequency domain analysis [11], artificial neural networks [12] are some of the prominent ones. The ability of the neural networks to learn makes them suitable for nonlinear applications.

Neural network architectures such as multi-layer perceptron (MLP), radial basis function networks (RBFN) have gained popularity in modelling linear and nonlinear system. Neural networks can be trained offline and used for identification or control. Offline trained models are robust to environmental noise but have restricted use because when the UAV dynamics changes beyond certain bounds the prediction error become significant. As an alternative smaller neural networks can be trained online during the UAV's flight. But training a network to model the entire 6 DOF during its flight involves large computations, hence, may not be implementable in real-time. One possible solution to the computation issue is to adhere to the conventional lateral and longitudinal dynamics and train smaller networks to imitate them online by suitably accounting for the cross couplings. The accuracy of these networks is dependent on the available time for training based on the processor speed. Hence a via-media is to switch to a well trained offline model for a short duration while the online model is adapting itself to changes in

Vishwas Puttige is a PhD student and Sreenatha Anavatti is a senior lecturer at the School of Aerospace, Civil and Mechanical Engineering, University of New South Wales at Australian Defence Force Academy, Canberra, ACT-2600, Australia

the dynamics [13]. By having multiple networks and a dynamic switching technique, accurate modelling for different flight conditions can be achieved. The concept of multiple neural network models for adaptive control was introduced by Narendra in [14]. In the past, multi-networks have been used for linear systems or offline modelling that are mainly based on numerical simulations [15]–[17]. In this work a real-time implementation of multi-networks for a UAV is presented. A well trained offline model based on previously collected flight data is used in conjunction with a simpler online model to achieve better accuracy. The switching between the online and the offline models is carried out based upon a suitable criterion. Two different selection criteria are considered for the dynamic selection process. In the first case, the instantaneous errors between the predicted outputs and the actual outputs are considered for selection. In the second case, the past performance of the two networks are also taken into consideration. A weighted sum of the instantaneous error and the mean square error (MSE) is used for selection. Each model uses a multi-input multi-output architecture. Modelling is performed with MLP networks based on the autoregressive technique with exogenous inputs (ARX) forming recurrent neural networks (RNN) introduced by Ljung in [18]. A novel training method is adapted for the online model where the network is trained with small batches of data and the weights from the previous batch are retained in memory [19]. The retraining of the online network is carried out only when the prediction error increases beyond a certain threshold. The online and offline identification techniques have been flight tested repeatedly. The multi-net model has been validated with a real-time hardware in the loop (HIL) simulation technique. The viability of the proposed technique in real-time is proved using the HIL simulation, hence allowing it to be implemented on flight tests.

The proposed multi-network architecture is explained in section II along with the basic dynamics of the UAV. The training scheme for the offline and the online networks are explained in section III. The results comparing an online model and an offline model for the 6 DOF UAV system is analysed in section IV. Here, the real-time HIL simulation technique is explained. Results from the multi-network modelling with two selection criteria are also presented in this section. The results indicate the superiority of the proposed multi-network modelling in predicting the UAV dynamics as compared to the online and offline models. Some concluding remarks are presented in the section V.

II. MULTI-NETWORK ARCHITECTURE

The mathematical model for the UAV can be represented as

$$\dot{x} = f(x, u); y = g(x, u) \tag{1}$$

where f and g are nonlinear functions, x, u and y represent the UAV states, the input vector and the outputs from the UAV respectively. In general, the outputs from the UAV are the angular rates, roll (p), pitch (q), yaw (r) and the linear velocities (u, v and w) along the three axes. The functions, f and g depend on factors such as the aerodynamics, inertia properties of the UAV and the effectiveness of the control surfaces and continuously change during the flight of a UAV. Moreover, the non-linearities and coupling inherent to the system add to the complexities. Hence, identification of the mathematical model as a relation between the outputs y and inputs u from the flight data may be more practical. This can be used for analysis as well as design of flight control systems.

The design of a reasonable offline model involves the availability of suitable flight data encompassing all possible regimes. Since this is not practical, a few sample flight tests are carried out for training the network. This offline model is implemented on the UAV for the consequent flight tests. The conventional batch wise Levenberg Marquardt (LM) method is adapted for training this model.

The UAV is a 6 DOF system with 4 inputs, hence, online training a single model with 6 outputs requires large training time. Thus single model structure is not suitable for online modelling. To circumvent this problem the UAV is modelled as two parallel networks representing the longitudinal and lateral dynamics similar to a conventional aircraft [20]. To account for the coupled dynamics the roll rate and the yaw rate are taken as the coupling terms for the longitudinal dynamics network and the pitch rate for the lateral network. The elevator and the throttle servo deflections are taken as the inputs to the longitudinal network. Similarly, the lateral network uses the aileron, rudder and the throttle deflections as inputs. Due to the restriction on the available time for online training simpler networks with less number of neurons are utilised. This restriction on the time arises from the capacity of the processor as well as the sampling time (training time needs to be lesser than the sampling time for the given processor speed). Small packets of flight data form the training set for these networks. The number of data points in the packets is decided based on the available processing time and the required accuracy.

The training data for the offline model is restricted. Hence, it may not perform as expected in different flight conditions. Due to the constraints on the training (particularly in terms of the available training time) in the online model it may not provide the desired accuracy under all flight conditions [21]. As a solution to this issue the offline and the online models are combined in the form of a multi-network architecture and individual outputs are compared with respect to a particular criterion and dynamically selected. This ensures a better performance compared to any of the individual models. Two different criteria are tested for the dynamic selection between networks.

In the first case, instantaneous error difference between the predicted output and the immediate past system output is used as the selection criterion. This is named criterion

1 and is given as

$$Err = E_{inst}. \tag{2}$$

Here Err is the error factor taking the value of the instantaneous error. This criterion provides accurate predictions, but involves frequent switching between networks. When this modelling architecture is used with a controller it may lead to large amounts of switching transients. To overcome this issue, the past performance of the network is taken into consideration along with the present performance in the second case. It was observed that a weighted sum of the instantaneous error and the mean square error between the network output and the actual output provides the most efficient performance with reduced frequency of switching. This weighted sum of errors, named as criterion 2, is shown in equation 3.

$$Err = w_1 * MSE + w_2 * E_{inst}. \tag{3}$$

Here the error factor Err is the weighted sum of the errors used for the dynamic selection, w_1 and w_2 are the weights assigned to the mean square error (MSE) and the instantaneous error (E_{inst}). The values of the weights is decided based on the noise level in the network inputs and experimental judgement. The network output with a smaller value of Err is used as the output of the multi-network. The errors are checked at every instant of time and the outputs are switched suitably. The results from the hardware in the loop simulation for neural network modelling of UAV longitudinal dynamics shown in section IV substantiate this theory. The overall MSE values are relatively smaller for the proposed multi-net method.

All the neural network models in this study have an autoregressive structure and use the LM training algorithm. The next section briefly talks about this neural network ARX structure and the training method.

III. ONLINE AND OFFLINE MODELLING

Different neural network structures and training methods were tried for modelling the nonlinear dynamics of the UAV. The ARX technique proved to be most suitable for this purpose [18], [22]. In the autoregressive neural network model the network retains information about the previous outputs and inputs to predict the next output. This provides equivalent retention capabilities of the dynamics of the UAV by the network. The predicted output of a nonlinear model can be obtained as [18], [19]

$$\begin{aligned}\widehat{y}(t|\theta) = g(a_1 y(t-1) + a_2 y(t-2) + ..\\ +a_{na} y(t-na) + b_1 u(t-1) + .. + b_{nb} u(t-nb))\end{aligned} \tag{4}$$

where θ is the coefficient matrix which gives the influence of past outputs $(a_1, .., a_{na})$ and influence of past inputs $(b_1, .., b_{nb})$ on each of the subsequent outputs. The nonlinear function is defined by g, the y and u terms correspond to past outputs and past inputs respectively. The above equation can be simplified as

$$\widehat{y}(t) = g(\theta, \phi(t)) \tag{5}$$

where,

$$\begin{aligned}\theta = (a_1, a_2, ...a_{na}, b_1, b_2, ..., b_{nb})\\ \phi(t) = (y(t-1), ...y(t-na), u(t-nk), ...,\\ u(t-nk-nb+1))\end{aligned}$$

Here ϕ is the matrix of past inputs and outputs called the regressor and is available from memory. To obtain the coefficients θ, many assumptions and detailed knowledge of the plant are necessary [23]. Hence for a dynamic nonlinear system such as the UAV it may not be feasible. This can be avoided by using black-box methods such as the neural networks. The output of a two layered neural network is given as

$$\widehat{y}_i(t) = F_i(\sum_{j=0}^{l_1} W_{2ij} G_j(\sum_{k=1}^{l_2} W_{1jk} x_k(t) + W_{1j0}) + W_{2i0}) \tag{6}$$

In the above equation F and G are the activation functions, l_1 and l_2 are the number of neurons in the two layers, W_{1j0} and W_{2i0} are the bias to the two layers and x_k is the network input. In most of the cases the nonlinearities are best represented by the hyperbolic tangent function as the activation function G and a linear relation F. W_{1jk} and W_{2ij} are the weights from the hidden layer and the output layer respectively. These weights correspond to the θ matrix in equation 5. Hence the problem of obtaining the best prediction depends on the network structure adapted and the training method used.

Iterative training is performed to minimize an error function using the Levenberg Marquardt's technique. The goal of the training is to obtain the most suitable values of weights for closest possible prediction through repetitive iterations. The LM method works on the principle of minimizing the mean squared error between the actual output of the system and the predicted output of the network. The mean square error as a function of θ to be minimized is given by [18]

$$V_N(\theta) = \frac{1}{N} \sum_{t=1}^{N} (y(t) - \widehat{y}(t))^2 \tag{7}$$

The updated value for θ after each iteration is given by

$$\theta_{i+1} = \theta_i - \mu_i R_i^{-1} G_i \tag{8}$$

where μ is the step length, R is a matrix responsible for the search direction and G is the gradient given by

$$G_i = \acute{V}_N(\theta_i) = -\frac{1}{N} \sum_{t=1}^{N} (y(t) - \widehat{y}(t|\theta_i))\psi(t, \theta_i) \tag{9}$$

where ψ is the derivative of the predicted output with respect to co-efficient matrix θ. The search direction is given by $R_i^{-1} G_i$ which is used to update the weights. For the LM method R_i is given as

$$R_i = \frac{1}{N} \sum_{t=1}^{N} (\psi(t, \widehat{\theta}_i), \psi^T(t, \widehat{\theta}_i)) + \mu I \tag{10}$$

In this equation the first part is the Hessian matrix and μ is the step size. For a zero value of this step size the LM algorithm changes to the Gauss-Newton method.

Conventional batch-wise LM training is used for offline models. Here, a large set of flight data is used to train the network. At the completion of training, the weights are frozen and the network is used for prediction during flight.

For the online models, the networks are required to be trained continuously. A mini-batch-wise algorithm is adapted based on the LM technique. During the course of flight, the data collected is fed as small packets to the online model which adapts its weights and predicts the future outputs. The trained weights are retained in the memory and form the starting point for the next set of training.

IV. EXPERIMENTS AND RESULTS

The UAV platform with the sensors, actuators and the on-board computer is remotely flown to collect flight data. A PC-104 based on-board computer is used as the data-logger and as the autopilot unit. Matlab Simulink's xPC target is used as the operating environment for all flight tests. A Simulink model of the data collection files and the identification (or control) algorithm is created. This Simulink model is built using Matlab real-time workshop and xPC target and included into the on-board computer before flight. The flight data collected facilitates the development of aircraft models and design of controllers.

A real-time Hardware in the Loop (HIL) simulation has been developed which provides a virtual environment to test the concerned algorithm. Real-time validation of identification techniques or control logics before putting it to actual use has become a necessity in flight control. This scheme confirms that the algorithm under consideration can be implemented on a real-time system. During this testing some of the unseen issues in numerical simulation such as the computational time, faults in sensors or actuators, unstable controllers, etc become evident. The same on-board computer is used for the HIL testing of system identification models and various controller designs. A block diagram of the HIL simulation is shown in figure 1. The identification algorithm is programmed into the target terminal (PC-104). A host terminal has the Simulink model of the aircraft subsystem programmed in it. The aircraft parameters are transmitted to the target terminal and the identified outputs are received from it in real-time. The serial port is used as the medium for communication. A graphical user interface running on the host terminal displays the identified results as a three dimensional model as predicted by the technique. Figure 2 shows the HIL simulation under progress with the host and target terminal. The GUI displaying the behaviour of the flight as a 3D model can also be seen.

Training for the offline model is performed using a single large batch of data collected during different test flights of the UAV. Figures 5 and 6 provide one such set of flight data. Figure 5 shows the inputs provided to the UAV during a test flight and figure 6 shows the angular rate outputs (bold line with the legend 'system output') from the UAV. Different variables of the network structure are varied by trial and error in order to obtain an optimal solution to the approximation problem in terms of the accuracy. Four past outputs and inputs are used to obtain the predicted outputs for this model. The hidden layer for this offline model has 12 neurons and takes 83 seconds for training 25 seconds of flight data. The network is trained batch wise for either 1000 iterations or until the error is reduced to 10^{-7}. The offline network is trained with 4000 samples, which correspond to 200 sec of data combining a few different flight tests [13].

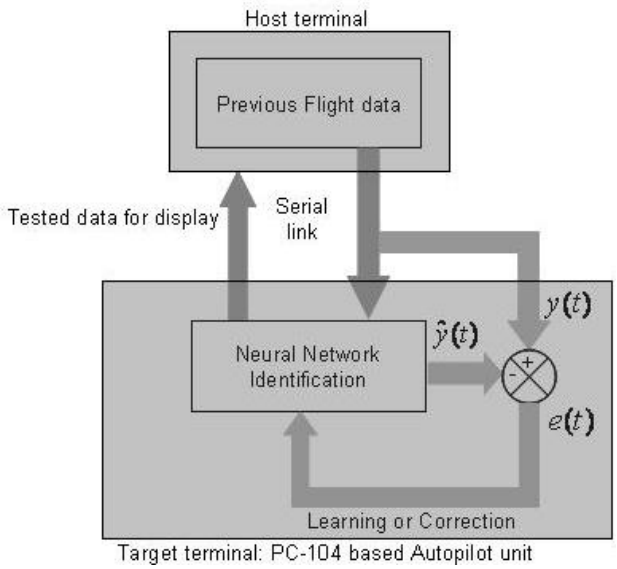


Figure 1. Block diagram of the hardware in the loop simulation

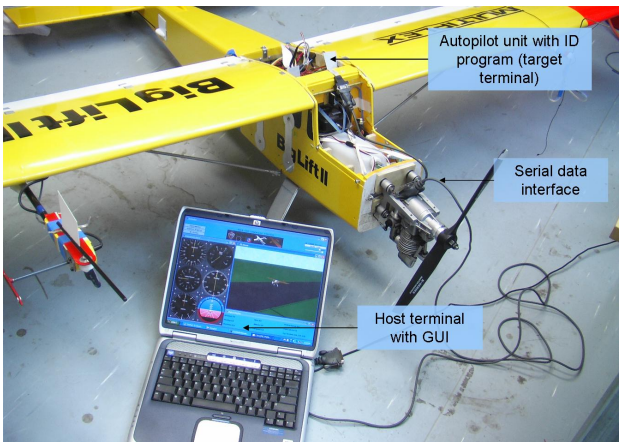


Figure 2. Hardware in the loop simulation of the multi-network modelling of the UAV

The online training is performed using small fixed batches of input and output data. Larger batch sizes take longer to train and hence the model does not function in real-time with the designated sample time. Very small sizes of the batch reduced the performance of training. For the batch size values ranging from five to ten, satisfactory prediction accuracies and speed of training are observed. The available training period is less than or equal to

Variable	Batch size = 5	Batch size = 10
θ	0.0707	0.2285
u	0.0779	0.2737

TABLE I.
MEAN SQUARE ERROR VALUES FOR TWO DIFFERENT BATCH SIZE ONLINE MODELS

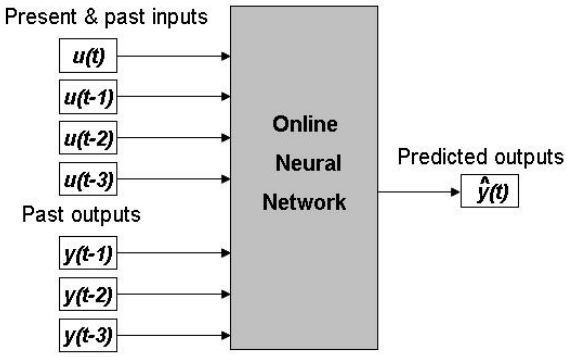


Figure 3. Online network structure with a dependency of 3 past inputs and past outputs

the sampling period due to one step prediction provided by the network. The effect of different batch sizes on performance of the network is presented in [24]. The MSE values for two different batch sizes of the online network are presented in table I. It can be seen that the online model with batch size of 5 has lower MSE values mainly because this network can be trained for more iterations for a fixed sample time. Three past outputs and inputs are used for the prediction and has four neurons in the hidden layer. The network structure is shown in figure 3. The variation in the dynamics is measured in terms of the error between the previous output and predicted output obtained from the trained network. As long as the dynamics does not vary beyond a threshold the weights of this trained network are retained in memory. The network is repeatedly trained only when the error from the new batch is greater than the predefined threshold. Figure 4 shows the variation of one of the weights during training. It can be seen that for a steady flight the weights are not changed, indicating retraining is not necessary during this time.

In figure 6, the flight test results of the online model for the angular rates are compared with that of the offline model. The offline model is trained with data from 0 to 25 sec and untrained data is used for validation during the next 25 sec. In figures 7 and 8 the same results are plotted individually for better visibility of the outputs. It can be seen that for the trained section of the offline model (0 to 25 sec), the approximation is much smoother than the online model results. But the offline model is less efficient in adapting to the large variation in the system inputs during the validation part (25 to 50 sec). The online model is more adaptive to changes in the inputs which can be seen from the results.

A Simulink block diagram of the multi-network model with online and offline networks is shown in figure

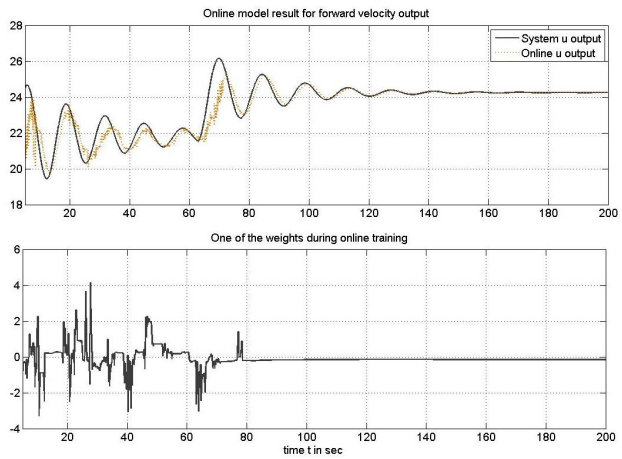


Figure 4. Online training output for forward velocity with the variations in one of the weights; velocity in meter/sec

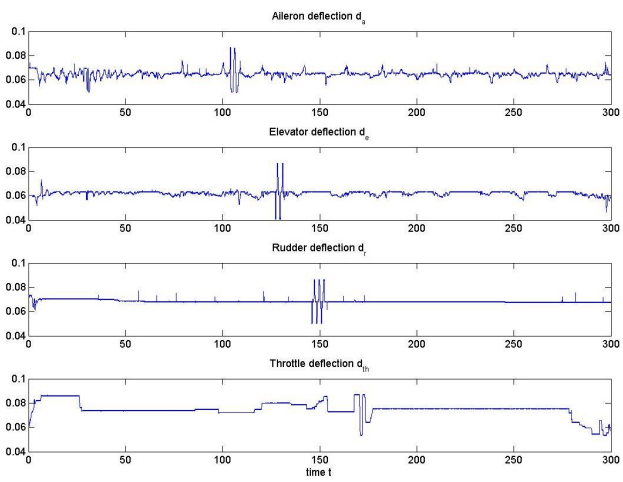


Figure 5. Servo inputs from one of the UAV test flights as duty cycles of the PWM signals

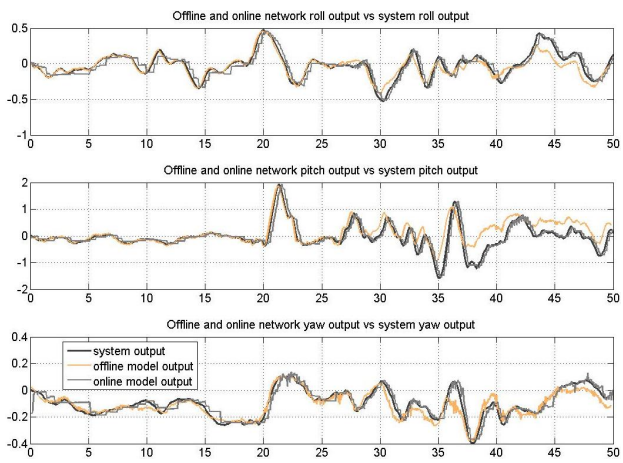


Figure 6. Angular rates data collected during one of the flight tests superimposed with the predicted outputs from the online and offline models; units in rad/sec

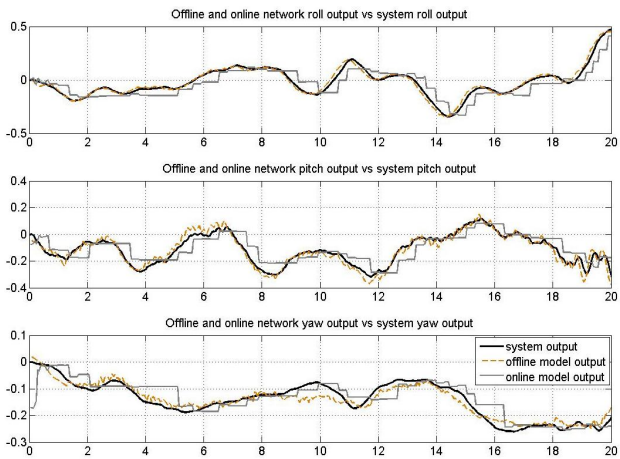


Figure 7. Training section of the outputs from offline model and online model against system outputs; units in rad/sec

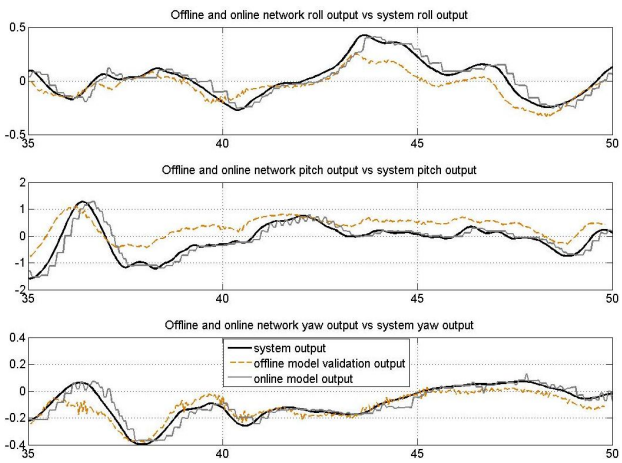


Figure 8. Validation section of the outputs from offline model against online model outputs and system outputs; units in rad/sec

9. For simplicity the aerodynamic coupling terms have been neglected here. In modelling the UAV longitudinal dynamics, online and offline networks have two inputs each, elevator and throttle, and two outputs each, forward velocity and pitch angle. Both the networks have a single hidden layer but different number of neurons and training criteria. The network parameters are provided in table II for comparison. Here Ts is the sample time of the system which is 0.05s. For the weighted average criterion 2, a weight of 0.8 was considered for the instantaneous error and a weight of 0.2 for the MSE. These values are chosen experimentally to reduce the switching between the two networks and maintaining accuracy of prediction.

Typical sections of the result from multi-network modelling of UAV longitudinal dynamics are shown in the figure 10. Here, two different sets of multi-network outputs obtained from different switching criteria are superimposed with the online and offline model outputs, and the actual system outputs. Even though the offline model has initial transients involved, it displays superior steady state behaviour. The online models are fast to catch up with the system response but displays small amounts

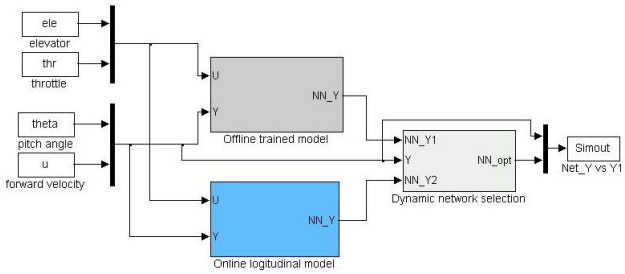


Figure 9. The multi-network architecture with dynamic selection for modelling UAV longitudinal dynamics

Parameters	Offline model	Online model
Hidden neurons	12	4
Iterations	1000	$18-25$
Error threshold	$1e-7$	0.001
Data Samples	>4000	$Batchlength$
Training time	$>> Ts$	$<= Ts$
Activation function	$tanh$	$tanh$

TABLE II.
OFFLINE AND ONLINE NETWORK INFORMATION

of steady state errors at times. It can also be seen that the outputs from the first multi-net model (blue dot-dashed line) has lesser prediction error as compared to those from either of the two individual models. Even though the results from the second multi-net model (grey dashed line) at few instances, have a small steady state error with respect to the actual output, this system is invariant to instantaneous erratic behaviour in either of the two individual models as it considers the past performance while switching.

The overall MSE values for the multi-network models with the two criteria are tabulated separately in tables III and IV with MSE values of individual online and offline models. It is evident from the two tables that the overall MSE is lesser with the criterion 1 but this method involves frequent switching which creates transients in the control process. By altering the weights assigned to each of the two errors the overall MSE and the switching transients can be altered.

The switching between the offline and the online model for pitch angle and forward velocity for the two selection criteria are shown in figure 11. On the y-axis 0 indicates output from the selected online model and 1 indicates output from the selected offline model. The dark line shows the switching with only the instantaneous error (criterion 1). The lighter dashed line indicates the switching between the two models with mean square error and the instantaneous error (criterion 2). At every sampling time, outputs from the UAV are compared with predicted outputs from the two networks. The output with the smaller value of this error factor from either of the networks is selected. The multi-net models with dynamic selection show promise in reducing the MSE values compared to the online or the offline models.

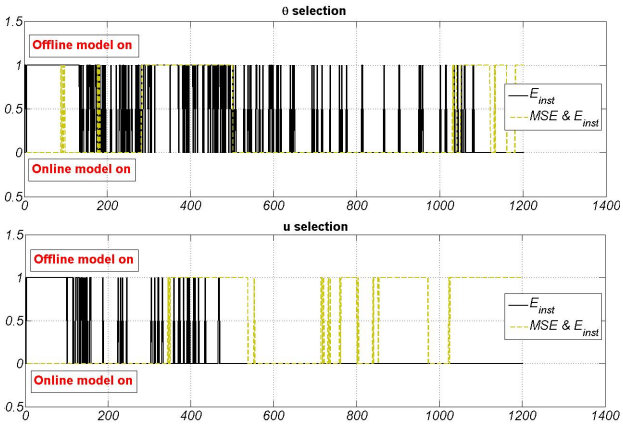


Figure 11. Selection between the offline model and online model for the theta as multi-network output

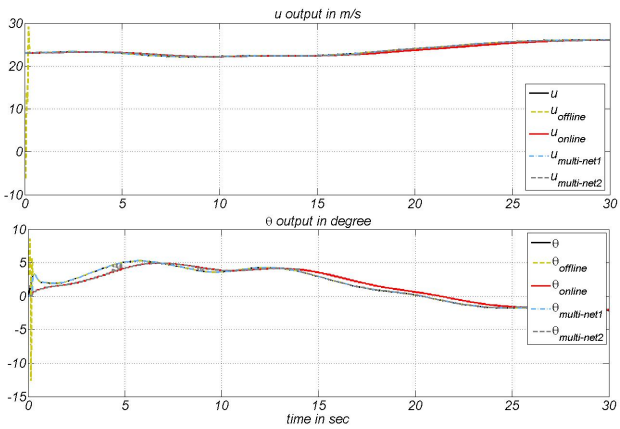


Figure 10. Multi-network architecture output vs individual networks outputs and system output for forward velocity and pitch angle

Variable	Online model	Offline model	Multi-net model
θ	0.1759	0.2437	0.0095
u	0.0534	1.6414	0.0001

TABLE III.
MEAN SQUARE ERROR VALUES FOR AN ONLINE MODEL, OFFLINE TRAINED MODEL AND THE MULTI-NET MODEL WITH CRITERION 1

V. CONCLUSIONS

Neural network based online and offline modelling for a nonlinear, complex system such as the UAV has been presented. Real-time simulations and flight test results for online and offline modelling of the multi-input multi-output UAV system highlight the merits and demerits of the online and offline models. To utilise the merits of both these models, a multi-network based system identification was proposed. This multi-network model dynamically selects the predicted outputs from the two individual models based on a criterion. The effect of two different criteria are studied and results for both are presented. It was observed that the multi-network model has smaller prediction errors and display superior performance as compared to either of the two individual models. Currently, this architecture is being used in the development of adaptive controllers for the autonomous flight of the UAV.

REFERENCES

[1] Myriam Manai, Andre Desbiens, and Eric Gagnon. Identification of a UAV and design of a hardware-in-the-loop system for nonlinear control purposes. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 2005–6483, San Francisco, USA, 2005. AIAA.

Variable	Online model	Offline model	Multi-net model
θ	0.1759	0.2437	0.1019
u	0.0534	1.6414	0.0088

TABLE IV.
MEAN SQUARE ERROR VALUES FOR AN ONLINE MODEL, OFFLINE TRAINED MODEL AND THE MULTI-NET MODEL WITH CRITERION 2

[2] Y. Kang and J. Hedrick. Design of nonlinear model predictive controller for a small fixed-wing unmanned aerial vehicle. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Colorado, 2006. AIAA.

[3] Doitsidis, Valavanis, Tsourveloudis, and Kontitsis. A framework for fuzzy logic based UAV navigation and control. In *IEEE International Conference on Robotics and Automation*, volume 4, pages pp 4041–4046, 2004.

[4] I. Schiller and J.S. Draper. Mission adaptable autonomous vehicles. In *IEEE Conference on Neural Networks for Ocean Engineering*, pages 143 – 150, Washington D.C, 1991. IEEE.

[5] F. Borrelli, T Keviczky, and G.J. Balas. Collision-free uav formation flight using decentralized optimization and invariant sets. In *IEEE Conference on Decision and Control*. IEEE, 2004.

[6] S. Howitt, D. Richards, and A. White. Supervisory control of autonomous UAVs in networked environments. In *AIAA Infotech@Aerospace*, Arlington, Virginia, 2005. AIAA.

[7] Vijayakumar Janardhan, Derek Schmitz, and S. N. Balakrishnan. Development and implementation of new nonlinear control concepts for a ua. In *23rd Digital Avionics Systems Conference*, volume 2, pages 12.E.5– 121–10, 2004.

[8] Anthony J. Calise, Eric N. Johnson, Matthew D. Johnson, and J. Eric Corban. Applications of adaptive neural-network control to unmanned aerial vehicles. In *AIAA/ICAS International Air and Space Symposium and Exposition: The Next 100 Years*, Dayton, OH, 2003. AIAA.

[9] Li-Xin Wang. Design and analysis of fuzzy identifiers of nonlinear dynamic systems. *IEEE Transactions on Automatic Control*, 40(1), 1995.

[10] Shaaban Salman, Vishwas Puttige, and Sreenatha Anavatti. Real-time validation and comparison of fuzzy identification and state-space identification for a UAV platform. In *IEEE International Conference on Control Applications*, pages 2138–2143, Munich, 2006.

[11] R. Pintelon and J. Schoukens. *System Identification : A Frequency Domain Approach*. Wiley-IEEE Press, 1st edition, 2001.

[12] Kumpati S. Narendra and Kannan Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE transaction on Neural Networks*, 1(1), 1990.

[13] Vishwas Puttige and Sreenatha Anavatti. Real-time multi-network based identification with dynamic selection implemented for a low cost UAV. In *IEEE Conference on System, Man and Cybernetics*, Montreal, Canada, 2007. IEEE.

[14] K.S. Narendra and J. Balakrishnan. Adaptive control using multiple models. *IEEE Transactions on Automatic Control*, 42(2), 1997.

[15] R.R. Selmic and F.L. Lewis. Multimodel neural networks identification and failure detection of nonlinear systems. In *Proccedings of the 40th IEEE Conference on Decision and Control*, 2001.

[16] T. Fujinaka and S. Omatu. A switching scheme for adaptive control using multiple models. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 1999.

[17] L. Giovanini, M. Benosman, and A. Ordys. Adaptive control using multiple models switching and tuning. In *Proceedings of the International Conference on Industrial Electronics and Control Applications*, 2005.

[18] Lennart Ljung. *System Identification - Theory for the User*. Prentice Hall, 2nd edition, 1999.

[19] Vishwas Puttige and Sreenatha Anavatti. Real-time neural network based online identification technique for a UAV platform. In *International Conference on Computational Intelligence for Modeling Control Application*, page 92, Washington, DC, USA, 2006. IEEE Computer Society.

[20] John D. Anderson. *Introduction to Flight*. McGraw-Hill, 1989.

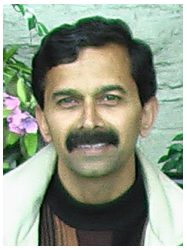
[21] Vishwas Puttige and Sreenatha Anavatti. Comparison of real-time online and offline neural network models for a UAV. In *International Joint Conference on Neural Networks*, Orlando, U.S.A., 2007. IEEE.

[22] Lennart Ljung and J Sjoberg. A system identification perspective on neural networks, 1992.

[23] Chunhua Hu, Xingli Huang, Jinchun Hu, and Jihong Zhu. System identification of a small UAV's speeding up process before take-off. In *Proceedings of the 5th Asian Control Conference*, pages 392 – 395, 2004.

[24] Vishwas Puttige, Sreenatha Anavatti, and Tapabrata Ray. Comparative analysis of multiple neural networks for online identification of a UAV. In *Twentieth Australian Joint Conference on Artificial Intelligence*, Gold Coast, Australia, 2007. Springer.

Vishwas Puttige is currently a Ph.D candidate at the School of Aerospace, Civil and Mechanical Engineering (ACME), University of New South Wales at Australian Defence Force Academy (UNSW@ADFA), Australia. He received his Master of Engineering degree from the Australian National University in 2004 and his Bachelor of Engineering degree from P.E.S.I.T affiliated to the Visvesvaraya Technological University, India in 2003. His research interests include control systems, flight dynamics and artificial neural networks. He has been an IEEE student member since the year 2006.



Sreenatha Anavatti received his Ph.D degree in aerospace engineering from the Indian Institute of Science in 1990, his Bachelor of Engineering degree in mechanical engineering from the Mysore University, India in 1984. He is currently a Senior Lecturer at the School of Aerospace, Civil and Mechanical Engineering (ACME), University of New South Wales at Australian Defence Force Academy (UNSW@ADFA), Australia. His current research interests include control systems, flight dynamics, robotics, aeroelasticity, artificial neural networks, fuzzy systems and unmanned systems.