NEURAL NETWORKS TRAINING ARCHITECTURE FOR UAV MODELING

Antonio Barrientos, Rodrigo San Martin, Pedro Gutierrez, Jaime del Cerro
Departamento de Automática, Ingeniería Electrónica e Informática Industrial
Universidad Politécnica de Madrid
Spain
barrientos@etsii.upm.es, rsan@etsii.upm.es, pgutierrez@etsii.upm.es jcerro@etsii.upm.es

ABSTRACT

This work proposes the use of hybrid models of supervised neural networks for modeling of a dynamical complex system and analyze different training architectures, in this case a scale helicopter, whose attitude and position identification is performed. This model will be useful for the development and utilization of the helicopter as Unmanned Aerial Vehicle (UAV). Throughout this work the supervised hybrid networks is examined, as well as the characterization of the treatment of the training commands, with which the present results are achieved.

KEYWORDS: Artificial Intelligence, Supervised Neural Networks, Modeling, Unmanned Aerial Vehicle, Helicopter.

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have demonstrated their utility in the military field. Different types of vehicles, but mainly planes, are being used for observation, mapping, mobile target acquisition and air-to-ground warfare missions. Their use in civil applications is recent, being appropriate for surveillance, inspection and aerial imagery acquisition tasks, among many others. The most suitable vehicle for many of the aforementioned tasks is the helicopter because it offers a good compromise between maneuverability, forward-flight speed and the capacity of hover.

Mathematical models of the helicopter's flight dynamics are critical for the development of robust controllers that allow for autonomous flight. Control algorithms and strategies are first evaluated in a simulator and second on hardware and real-flight conditions. Flight dynamics may be modeled with analytical, empirical or mixed models. An analytical model is complex [1], both at its conception and implementation stages, and so computationally expensive that it is not viable for real-time applications. Empirical models are simpler and capable of real-time simulation but have limited precision. Mixed models combine analytical expressions and empirical approximations to reach a good compromise between accuracy and speed.

In this paper a neural-network based model is presented as a reliable and fast empirical model of a radio-controlled mini-helicopter. Its main advantages are the low-computational cost and ability to store different models as matrixes.

The radio-signals sent as commands to the helicopter's servos define, through coupling interactions, the attitude of the airframe that, through coupling and distortion, defines the position and velocity along the XYZ axes. Therefore two neural networks are needed: one for the radio-signals-to-attitude system and another for the attitude-to-position system. The on-board hardware records the radio-signals, attitude and position variables during flight so several training architectures are possible. In particular, this work illustrates the differences between two training architectures: the first uses only the radio-signals, while the second uses both the radio-signals and recorded data for the attitude variable (roll, pitch and yaw). The NN-based models, from the two training methods, are evaluated using the flight data, and an assessment of the validity of the neural-network as an empirical model of a mini-helicopter is given. The modeled vehicle is a BENZIN-trainer radio-controlled mini-helicopter from Vario, with a 5Kg payload capacity, a 26 cc. gasoline engine, and custom-designed avionics.

1. HYBRID MODEL USING NEURAL NETWORKS

There are two kinds of supervised neural networks: recurrent and non-recurrent. For the modelling of a dynamic system, like a mini-helicopter, the characteristics of the two types are needed [2][3][4]. There are several hybrid networks that mix the recurrent and non-recurrent architectures, for example the Elman [5] and Jordan [6] networks. Basically they use a forward-propagation block (like non-recurrent networks) and context-neurons that store previous states (like recurrent networks).

Due to the characteristics of the system a new architecture was designed (see Figure 1). The first block contains context-neurons that store past input and output data. The number of past-states to be stored is defined automatically at the training stage or with a stochastic pattern. The second block features one of the types of non-recurrent network (as will be explained later). Figure 1 shows **Block A** and **Block B** that represent the non-recurrent and context-neuron blocks, respectively.

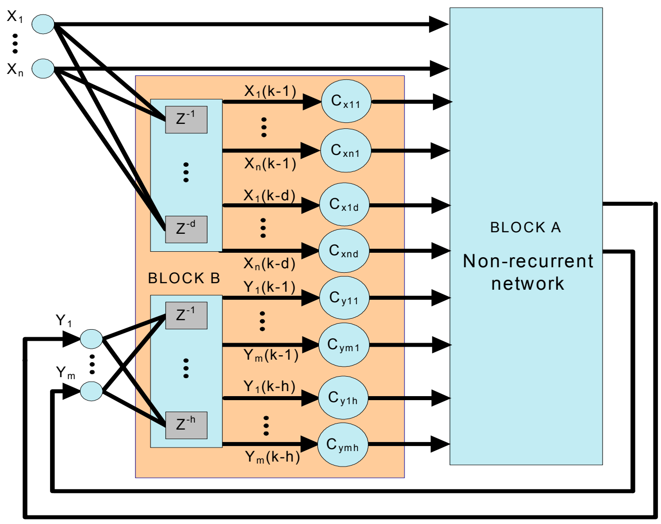

Figure 1. Model hybrid of neural network defined

The external inputs of the system are represented by vector $\mathbf{X} = [x_1,x_2,\ldots ,x_n]$, which generates context neurons $\mathbf{C_{xi1}}$ to $\mathbf{C_{xid}}$, depending on the number of past-states to be stored for each input. The outputs of the system are collected in the vector $\mathbf{Y}=[y_1,y_2,\ldots ,y_m]$ that also generates context neurons of the type $\mathbf{C_{yi1}}$ to $\mathbf{C_{yih}}$, depending on the number of past-states to be stored for each output.

The context neurons define the past-states to store. In the case of the inputs, up to the d-th previous sample is stored, being for the i-th input $\mathbf{C_{xi}=[x_i(k-1),\ldots ,x_i(k-d)]}$, where $\mathbf{C_{xir}=x_i(k-r)}$. For the outputs up to the h-th previous sample is stored, thus obtaining vector $\mathbf{C_{yi}=[y_i(k-1),\ldots ,y_i(k-h)]}$, where $\mathbf{C_{yir}=y_i(k-r)}$. Therefore the input vector for the non-recurrent network of **Block A** is $\mathbf{U=[X, C_{x1}, \ldots, C_{xn}, C_{y1},\ldots, C_{ym}]}$.

The hybrid network works with a number of previous states (order of state variables), **d** for the case of the external inputs and **h** for the outputs, thus creating an input vector of the following form: input vector **X** plus each element of **X** multiplied by their **d** previous states, plus the **m** outputs multiplied by their **h** previous states. The values of **d** and **h** are automatically defined.

2. TRAINING ARCHITECTURES

The training patterns (Data Base DB) are composed by the attitude, position and radio-signals (Croll, Cpitch, Cyaw and Cole) variables. The on-board avionics use an inertial measurement unit (IMU and magnetic compass) and a GPS to record many flight variables, while the radio-signals are recorded by the ground-station.

As mentioned before two neural networks are needed: one for the radio-signals-to-attitude system and another for the attitude-to-position system. Therefore two training architectures may be considered: the first uses the output of one network to train the other, while the second places them in *parallel* by using recorded data for both (see Figure 2). These training structures are called Daisy chain Architecture and Decoupled Architecture, respectively.

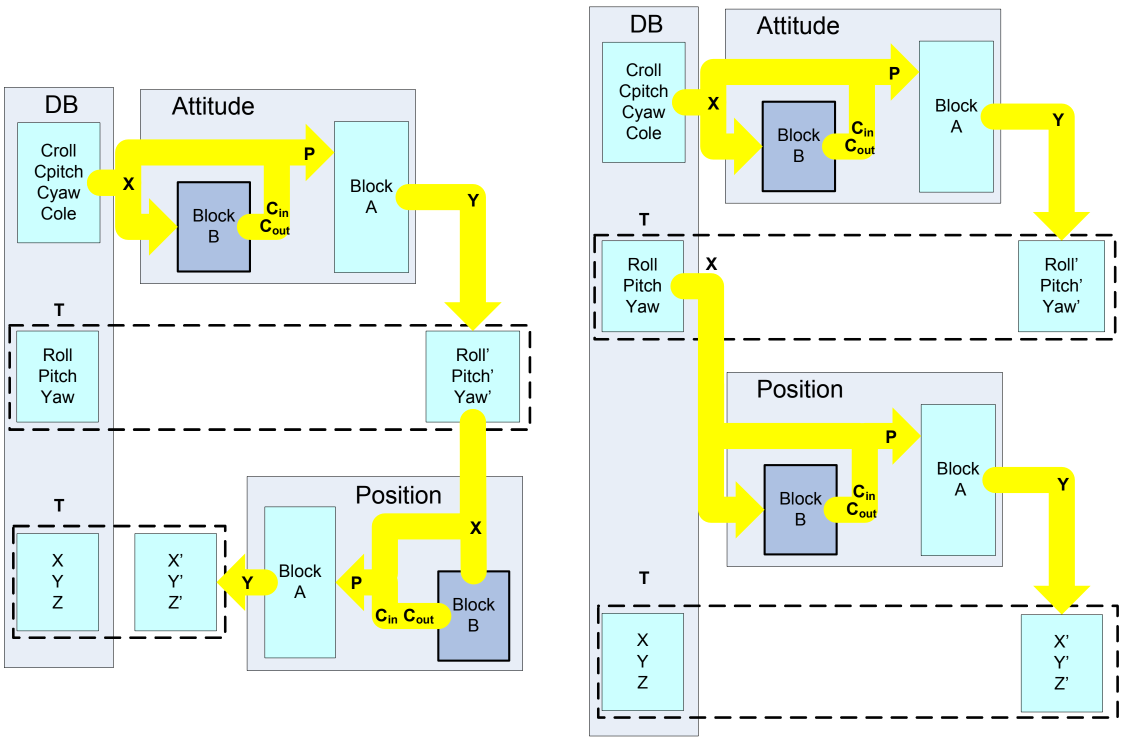

Figure 2. Daisy chain Architecture (left) and Decoupled Architecture (right)

2.1 Daisy chain Architecture

The first network (radio-signals-to-attitude system) is trained as an isolated system. This is possible because the recorded flight-data includes the attitude variable (roll, pitch, yaw). The second network (attitude-to-position system) is trained with the roll', pitch' and yaw' values, which are the outputs as modelled by the first network. In summary, the output vector of a previously trained radio-signals-to-attitude network is used to train the attitude-to-position network. For this architecture it is assumed that the modelling errors of the first network are inherited and corrected by the training of the second network.

2.2 Decoupled Architecture.

For this architecture the recorded attitude variable is used to train the attitude-to-position network. Therefore both networks are trained as isolated systems.

It is important to note that a decoupled architecture cannot be used for simulation as the real values for the roll, pitch and yaw variables are unknown. This may be a handicap for the model trained with the Decoupled Architecture because the attitude-to-position network is not trained to *compensate* the modelling errors of the radio-signals-to-attitude network.

It is expected that the error for the radio-signal-to-attitude network will be identical, regardless of the training architecture used. On the other hand the attitude-to-position system of the Decoupled Architecture will be better when fed with recorded attitude data because it will not inherit the previous' network error. As for the quality of the full model experimental data should confirm which architecture is better.

3. DATA AQCUISITION

The success of an empiric identification system is dependent on good experimental data. For a model based on neural-networks data-acquisition is even more relevant. It is a well-established fact that the training vectors of a neural-network must include data for the full operational range of the dynamic system. Therefore, the flight types [7] (take-off, landing, hovering, different manoeuvres) and data-quality are the criteria for the selection of training patterns.

The mini-helicopter is radio-controlled by an expert human pilot that executes predefined manoeuvres. For each experiment the radio-signals and several variables are recorded. The avionics (IMU, GPS and compass) send data wirelessly to a ground-station (PC) at 10 Hz app, while the radio-controller is attached directly to the PC (with a USB adapter). The system inputs and outputs are stored as a text file together with a GPS-signal quality indicator and time between samples (which is not necessarily constant).

Each experiment is constrained with a set of operational conditions: wind speeds lower than 10 kph and a GPS signal that should be equal-to or above the status of Narrow Float [8]. Additionally the difference between landing and take-off for the values of roll, pitch and yaw should be close to zero (considering a flat surface).

4. RESULTS

The following results were obtained from a series of training patterns with data collected on 9 flight sessions with an approximate duration of 3 minutes each. All flight sessions comply with the quality criteria and operational constraints that ensure high quality data, and are representative of the vehicle's operational range. As justified by previous work [7] a Multi-Layer Perceptron (MLP) in Block A, with one-hidden layer [9], is used as the architecture for both the radio-signal-to-attitude and attitude-to-position networks.

To compare the staggered and decoupled architectures the radio-signal-to-attitude network should be trained first. This network has a mean square error (MSE) of the order of 10^{-6} and its outputs are then used to train the attitude-to-position network (for the Daisy chain Architecture). On the other hand the Decoupled Architecture uses the recorded attitude data for training.

Due to different physical conditions a *complete* flight session must be considered as three stages: take-off, flight manoeuvres and landing [7]. Results and discussion focuses on the second stage because it lasts longer (in time) and covers a broader operational range of the mini-helicopter (manoeuvres diversity).

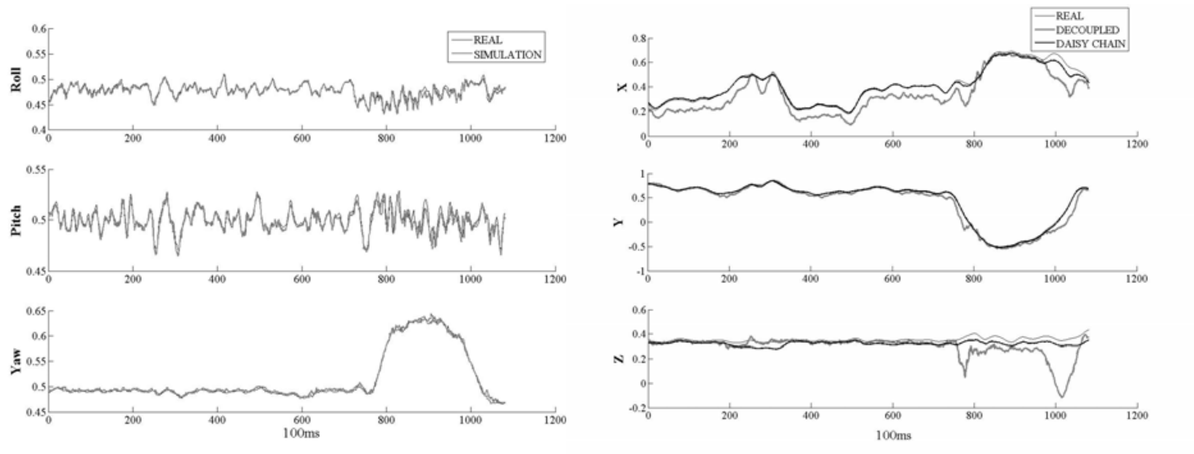

Figure 3: Attitude simulation for flight manoeuvres (left) and Position simulation for flight manoeuvres (right), with Daisy chain and Decoupled architectures.

Figure 3 shows the values of simulated vs. real data for both systems (radio-signals-to-attitude and attitude-to-position) for the two training architectures. It is important to note that for the attitude no distinction is made between the architectures because the results are equivalent. Table 1 shows the quantitative results, in terms of the MSE, for six different flights sessions. It is important to note that these sessions are different than the ones used for training. Both the graphical and quantitative results show that the Daisy chain Architecture has better performance.

	Flight A	Flight B	Flight C	Flight D	Flight E	Flight F
Daisy chain						
X	1,50E-04	1,13E-04	2,13E-04	2,06E-04	1,25E-04	9,13E-05
Y	2,36E-04	1,44E-04	4,55E-04	1,13E-04	8,34E-05	1,49E-04
Z	8,58E-05	9,38E-05	8,23E-05	1,09E-04	1,09E-04	1,06E-04
Decoupled						
X	5,85E-03	2,00E-04	6,08E-03	5,99E-03	1,97E-04	3,87E-03
Y	1,28E-04	3,28E-03	6,35E-04	1,19E-03	9,20E-04	1,09E-03
Z	9,84E-02	3,77E-04	1,73E-02	6,85E-04	1,12E-04	2,55E-03

Table 1: Comparison of Mean Square Error for real vs. simulated position, for different flight sessions (only the flight manoeuvres stage is considered)

5. DISCUSSION

As expected the quality of the simulated attitude is identical, as both MLPs were trained with the same set of input data. There are, however, significant differences in the quality of the simulated position variables (X, Y, Z) between the two architectures.

If the attitude-to-position network of the Decoupled Architecture is fed with real flight attitude data its MSE is lower, as the network is trained to model the interactions between real attitude and real position data. But at simulation-time only the simulated attitude data is available and the attitude-to-position network is unable to compensate for the errors induced by the first network. On the other hand the Daisy chain Architecture uses the simulated attitude to train the attitude-to-position network, so the inherited error is *fully corrected* at training time and *partially corrected* at simulation time.

6. CONCLUSIONS

The neural-network based model of a mini-helicopter is capable of accurate and real-time simulation regardless of the training architecture used. The utility of supervised neural networks for modelling sophisticated dynamic systems has been shown, particularly for complex systems such as a mini-helicopter. The model is capable of producing good-quality simulated data in real time as demonstrated by the comparison of real vs. simulated variables for flight sessions not used in the training stage.

As the dynamic system to be modelled presents two distinct subsystems (radio-signal-to-attitude and attitude-to-position) several training architectures are possible. Therefore it is possible to evaluate the convenience of daisy chain training versus decoupled training. The first architecture considers that the attitude-to-position network will inherit modelling errors, while the second tries to leverage the availability of real attitude data at the training stage. The Daisy chain Architecture is, as expected, better because it compensates the simulated attitude error.

This work is relevant as it paves the way for research on alternative modelling methods, especially those based on neural networks. NN are a powerful and easy-to-use tool because the right architecture and high-quality training data make real-time models possible.

In the future new architectures (both for training and for the network) that take into account different situations originated by the particular characteristics of the flight and external disturbances will be investigated and evaluated.

7. ACKNOWLEDGEMENTS

Authors would like to thank Spanish Administration (Ministerio de Educación y Ciencia), for financing the Project "*Guiado, Navegación y Control de Vehiculo Aéreo Autónomo (VAMPIRA)*", (DIP 2003-01767)

8. REFERENCES

[1] López Ruiz, José Luis "*Helicópteros: Teoría y diseño conceptual*". ETSI Aeronáuticos, Madrid, 1993

[2] Narendra, K.S. and Parthasarathy, K. "*Identification and Control of Dynamical Systems Using Neural Networks*". IEEE Transactions on neural networks, Vol.1 Nº 1, March 1990

[3] Barreto, G.A. and Araújo, A.F.R. "*Identification and Control of Dynamical Systems Using the Self-Organizing Map*". IEEE Transactions in neural networks, Vol.15, nº 5, Sep 2004

[4] Norgaard, M.; Ravn, O.; Poulsen, N.K. and Hansen, L.K. "*Neural networks for Modelling and Control of Dynamic Systems*". Springer-Verlag, London, 2001

[5] Jordan, M. "*Serial order: A parallel distributed processing approach*". Technical report, Institute for Cognitive Science. University of California, 1986

[6] Elman, J. "*Finding structure in time*". Cognitive Science, 14:179-211, 1990

[7] R. San Martin, A Barrientos, P. Gutierrez, J. del Cerro "*Unmanned Aerial Vehicle (UAV) Modelling Based on Supervised Neural Networks*", ICRA2006, Orlando FL. USA.

[8] Novatel "*OEM4 Family of Receivers: User Manual- Volume 1and 2. Installation and Operation*". NovAtel Inc., Canada, 2002

[9] Li, J.H.; Michel, A.N. and Porod, W. "*Qualitative Analysis and Synthesis of a Class of Neural Netwroks*". IEEE Trans. Circuits Syst., Vol.35. Nº 8, Aug. 1988