Adaptive Neuro Control for Altitude Stabilization of UAVs

Jesús Enrique Sierra[1], Matilde Santos[2], Victoria Lopez*[3]

[#]Computer Science Faculty, Complutense University of Madrid
C/Profesor García Santesmases 9, Madrid, Spain

[1]jesier01@ucm.es
[2]msantos@ucm.es
[3]vlopez@fdi.ucm.es

Abstract— **In this work an adaptive neuro-control strategy is proposed for the stabilization of an unmanned aerial vehicle (UAV). The influence of the quadrotor's mass variation and the wind disturbances are analyzed. Simulation results show how the on-line learning increases the robustness of the controller, reducing the effects of the changes in mass and the wind on the UAV altitude.**

***Keywords*— Neuro-control, online learning, neural networks, unmanned aerial vehicles (UAV), quadrotors**

I. Introduction

In recent years, new and valued applications of unmanned aerial vehicles have emerged in different sectors like: defenses, security, construction, agriculture, entertainment, shipping, etc, that demand the design of efficient and robust controllers for these and others applications. Thus, the modeling and control of these complex and unstable systems still motivate the research and the interest of the scientific community [1-3].

Having said that, the modeling and control of unmanned aerial vehicles (UAV) are not an easy job, its complexity comes from the randomness of the airstreams and of the exogenous forces, the high non-linearity of the dynamics, the coupling between the internal variables, the uncertainty of the measurements… These factors make the techniques based on artificial intelligence a promising approach for the identification and the control of these systems.

These techniques are especially interesting when the model's parameters vary while the system is working. For example, the variations suffered by the total mass of the system when the vehicles are used in logistic tasks, since the mass depends on the packages which are shipped.

There are some studies where neural networks have been applied to model these systems [4, 5] and to control them [6-8]. It is also possible to find examples of the application of these intelligent techniques to model other complex non-linear systems [9], as for example marine vehicles [10].

In this work an adaptive neuro-control strategy is proposed to stabilize an unmanned aerial vehicle. The influence of the mass and the external disturbances is studied. The results show how the online learning increases the robustness of the controller, by reducing the effects of the changes in mass and the external disturbances on the height.

The paper is organized as follows. In section 2 the equations which describe the dynamic behavior of the system are presented. Section 3 describes the adaptive neuro-control strategy that has been implemented. Simulation results are presented and discussed in section 4. The document ends with the conclusions and future works.

II. System Description

A quadrotor vehicle is composed by four perpendicular arms, each one with a motor and a propeller (figure 1). The four motors drive the lift and direction control.

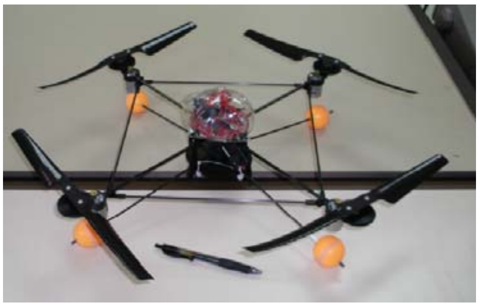

Fig. 1 Example of quadrotor vehicle

The system is based on two couples of propellers which are opposed each other (1,3) y (2,4) (Figure 2).

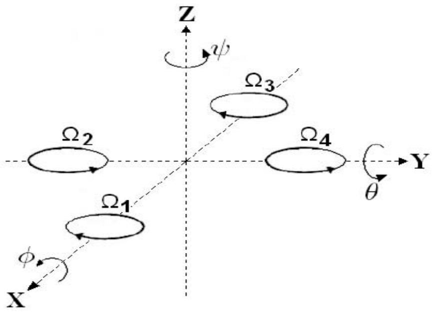

Fig. 2 UAV's coordinate system

In order to ensure the system is balanced, one pair of motors turns clockwise while the other one turns counterclockwise. The increment of the speed of rotor 3

ISSN 2356-5608

respect to rotor 1 produces a positive pitch ($\theta > 0$), while the increment of the speed of rotor 4 regarding rotor 2 produces a positive roll ($\phi > 0$).

The absolute position is described by three coordinates, (x, y, z) and the attitude by the three Euler's angles (ϕ, θ, ψ), under the conditions: $(-\pi \le \psi < \pi)$ for the yaw angle, $\left(\frac{\pi}{2} \le \phi < \frac{\pi}{2}\right)$ for the roll angle and $(\frac{\pi}{2} \le \theta < \frac{\pi}{2})$ for the pitch.

By using the Newton-Euler's method, the angular dynamics of the system is represented as follows:

$$\tau = J\dot{\omega} + \omega \times J\omega \qquad (1)$$

$$J = \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix} \qquad (2)$$

Where τ is a vector of torques in the three axis, J is the inertia tensor, ω is a vector of angular velocities and $\times$ represents the vectorial product.

The translational dynamics is given by:

$$m\dot{v} = RT - mge_3 \qquad (3)$$

Where m is the mass of the quadrotor, R is the rotation matrix, g is the gravitational acceleration, T is a vector of forces and $e_3 = [0,0,1]^T$ is a unit vector which describes the rotor orientation.

The vectors τ and T are a function of the velocities of the propellers:

$$\tau = \begin{pmatrix} bl(\Omega_4^2 - \Omega_2^2) \\ bl(\Omega_3^2 - \Omega_1^2) \\ d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{pmatrix} \qquad (4)$$

$$T = \begin{pmatrix} 0 \\ 0 \\ b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \end{pmatrix} \qquad (5)$$

In the equations 4 and 5, b is the thrust coefficient, d is the drag coefficient and l is the longitude of each arm; Ω_1, ...,Ω_4 are the velocities of the rotors 1 to 4, respectively.

In order to simplify the calculations, instead of using the speed of the rotors it is possible to define a set of control signals u_1, u_2, u_3 y u_4 as follows:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \qquad (6)$$

This matrix may be inverted, and then it is possible to generate speed references for the rotors from a set of control signals.

From equations 1 to 6, the following system of equations is derived:

$$\ddot{\phi} = \dot{\theta}\dot{\psi}\,(I_y - I_z)/I_x + (lb/I_x)u_2 \qquad (7)$$
$$\ddot{\theta} = \dot{\phi}\dot{\psi}\,(I_z - I_x)/I_y + (lb/I_y)u_3 \qquad (8)$$
$$\ddot{\psi} = \dot{\phi}\dot{\theta}\,(I_x - I_y)/I_z + (d/I_z)u_4 \qquad (9)$$
$$\ddot{X} = -(sin\theta cos\phi)\,(b/m)u_1 \qquad (10)$$
$$\ddot{Y} = (sin\phi)\,(b/m)u_1 \qquad (11)$$
$$\ddot{Z} = -g + (cos\theta cos\phi)\,(b/m)u_1 \qquad (12)$$

The constants used during the simulations are listed in Table 1, and were extracted from [11].

TABLE I
CONSTANTS OF THE MODEL

Parameter	Description	Value
l	Longitude of an arm	0.232 m
m	Mass of the quadrotor	0.52 Kg
d	Drag coefficient	7.5e-7 N m s^2
b	Thrust coefficient	3.13e-5 N s^2
I_x	Inertia in X	6.228e-3 Kg m^2
I_y	Inertia in Y	6.225e-3 Kg m^2
I_z	Inertia in Z	1.121e-2 Kg m^2
ρ_{air}	Density of the air	1.2 Kg/ m^3
A	Area in the direction of the wind	0.0186 m^2
Cd	Wind Drag coefficient	1

III. Description of the neuro-controller

A. Control Strategy

There are different control strategies with neural networks. In our case, a variant of the generalized learning algorithm (GLA) has been used. The modification consists of the refinement of the network during the execution of the controller by adaptive learning [11].

The first step is the application of the GLA algorithm to off line training the neural network in order to identify the inverse dynamic of the plant (Figure 3).

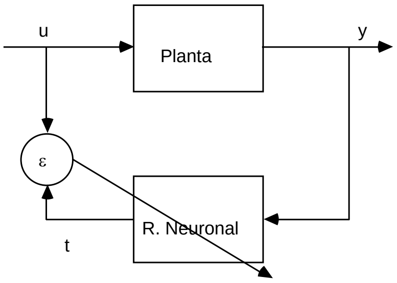

Fig. 3 Off-line training to identify the plant's inversed dynamic

The neural network that will be used as controller should be the inverted dynamic of the plant; therefore, from the desired response (reference signal, r) the control signal u must be generated, that will lead the real system's output to the reference signal.

Once the network has been off-line trained, it is placed in cascade connection with the plant. Then the configuration of the network is on-line refined. In order to do this, during each control interval two processes are sequentially applied to the network:

1. **Simulation**: the output *u* is generated from the input reference and it set as the input of the plant. (Figure 4, switch in the upper position).
2. **Learning**: From the current and past outputs of the plant the neural network is trained to generate the control output *u* (Figure 4, switch in the lower position).

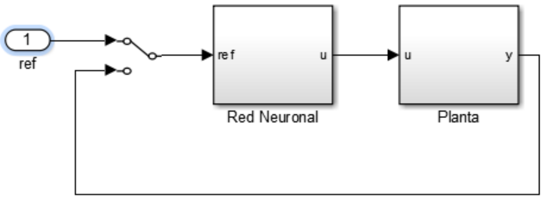

Fig. 4 Control phase +on-line learning

B. Altitude Control

In order to test the validity of this technique we have focused on the control of altitude variable. UAV are normally provided with accelerometers, so it is assumed that the acceleration in the z-axis (is available.

The network must be able to simulate the control signal by using acceleration measurements. The set of input data of the network is taken each time instant t, with the value of the control signal in the 10 past time samples where i = 1…10, and is the sampling time, and the acceleration in the Z-axis at the current time is (t). The set of output data has been generated with the value of the control signal at the current time (t). The sampling time used in the experiments is 10 ms.

In order to control the altitude Z, a PID controller, which generates the references of the acceleration in the z-axis, is added.

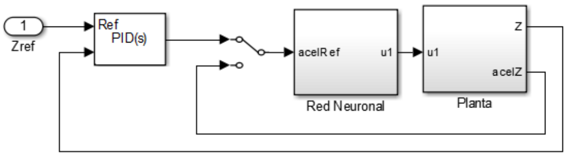

Fig. 5 Proposed altitude controller

A train of pulses with variable amplitude is generated during the off-line phase to train the network. The duration of the train of pulses is 2s. Before the training, the input and output sets are normalized to adjust the values to the range 0-1.

C. Neural Network

During this work, the selected network has been a multilayer perceptron (MLP). The MLP is an artificial neural network composed by an input layer, a set of hidden layers and an output layer. Each layer is composed by a set of nodes; each node in one layer connects with a certain weight to every node in the following layer. The figure 6 depicts the structure of an example of a MLP network.

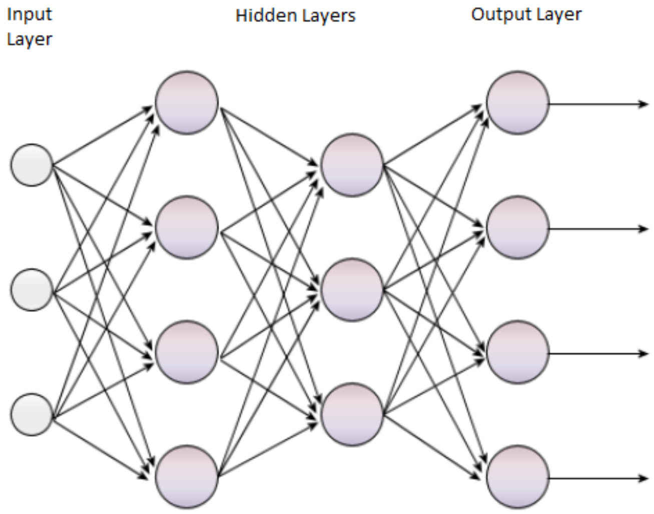

Fig. 6 Structure of a MLP

The nodes in the hidden layers and the output layer are based on the perceptron. In the perceptron the inputs coming from the previous layers are firstly weighted summed, then the result is incremented by a threshold, and the outcome is introduced in an activation function. The output of the activation function feeds the inputs of the subsequent layers. The equation 13 represents this process.

(13)

Where are the weights of the inputs, is the threshold and is the activation function. Typically the activation functions most used are the Log-sigmoid transfer function, the linear transfer function and the Hyperbolic tangent sigmoid transfer function.

The figure 7depicts the structure of the perceptron

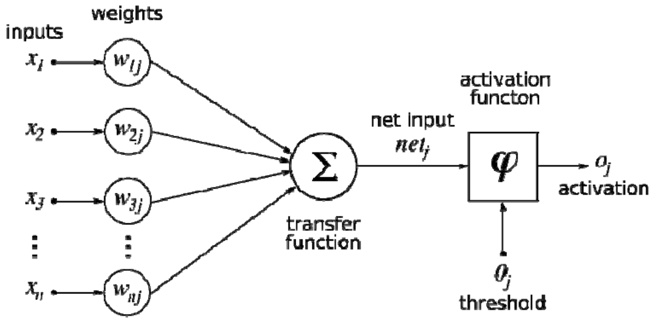

Fig. 7 Structure of the perceptron

In our case the network has been configured with 5 hidden layers. Levenberg-Marquardt with has been used as optimization algorithm.

IV. Results

Positive results have been obtained by simulation with the software Matlab/Simulink. The duration of each simulation has been 15 seconds. The controller has been off-line trained for the first 2 seconds, and the on-line learning algorithm has been then applied for the remaining 13 seconds by introducing the signal Zref as reference to the controller.

The controller has been trained from the beginning up to t=2 s. The control signals used to off-line train the network produce changes in the altitude during this interval. The control phase begins in t=2, when the Z reference is set to 5, and then acceleration references are continuously generated to stabilize the Z over this value. During the simulation the signals acelRef and acelZ match better over time, therefore the controller is able to stabilize the altitude to the desired value.

These results could be extended to control the Euler's angles for path following.

A. Robustness when varying the mass

In this experiment the effect of the variation of the mass is simulated. The mass of the quadrotor is duplicated at t=4. The error at the output of the neural network worsens the system response. Indeed, the stationary error when on-line learning is not applied is relevant.

B. Robustness with wind disturbances

A new term ($dist$) is added to equation 12 to model external disturbances.

$$\ddot{Z} = -dist - g + (cos\theta cos\phi)\,(b/m)u_1 \qquad (14)$$

The disturbance represents the variation of the acceleration caused by the external wind in the movement direction [12]:

$$dist = sgn(v_w) \cdot \rho_{air} \cdot A \cdot Cd \cdot \left(\dot{Z} - v_w\right)^2/(2m) \qquad (15)$$

Where v_w is the wind speed, ρ_{air} is the air density, A is the area of the quadrotor, Cd is the drag coefficient respect to the wind, $\dot{Z}$ is the velocity in the z-axis and sgn denotes the sign function.

In this experiment, from t=4 the wind speed is simulated by a step with Gaussian noise added. The SNR between the average wind and the noise is 10dB. The average wind is 12 m/s. This value matches to number 6 in the Beaufort's scale (Strong Breeze).

Likewise in the previous experiment, the controller without on-line learning cannot react to this disturbance, but in this case the stationary error is not so big.

V. Conclusions and Future Works

UAVs are complex systems to model and control. The complexity comes from their strong non-linear dynamics and also from external disturbances.

In this work an adaptive neuro-controller is proposed. The proposal is validated by the simulation of the altitude control of an UAV. The results show how the online learning of the network increases the robustness of the controller, by reducing the effects of changes in mass and wind disturbances.

Among others possible future lines we may highlight: the control of the complete system (Euler's angles) and the study of the influence of the neural network topology on the control performance.

Acknowledgment

The author Victoria Lopez would like to thank the 2017 International Conference on Control & Signal Processing Organization for the opportunity to deliver a lecture during the conference.

References

[1] J.J. Ortega, M. Sigut. (2016) *A low-cost mobile prototype for high-realism flight simulation*. RIAI - Revista Iberoamericana de Automática e Informática Industrial, 13(3):293-303.

[2] A. A. Fahmy, and A.A. Ghany. (2015) *Adaptive functional-based neuro-fuzzy PID incremental controller structure*. Neural Computing and Applications, 26(6), 1423-1438, 2015

[3] P. Garcia-Auñón, M. Santos, J.M. Cruz. (2017)*A new UAV ship-tracking algorithm*. Preprints of the 20th IFAC World Congress, pp. 13632-13637.

[4] J.E. Sierra and M. Santos. (2018) *Modeling engineering systems using analytical and neural techniques: hybridization* Neurocomputing, 271, 70-83.

[5] S. Bansal, A.K. Akametalu, F.J. Jiang, F. Laine, and C.J. Tomlin. (2016) *Learning quadrotor dynamics using neural network for flight control*. In: Decision and Control (CDC), IEEE 55th Conf. on (pp. 4653-4660).

[6] H. Boudjedir, O. Bouhali, and N. Rizoug, N. (2014) *Adaptive neural network control based on neural observer for quadrotor unmanned aerial vehicle*. Advanced Robotics, 28(17), 1151-1164.

[7] N.A. Bakshi, and R. Ramachandran, R. (2016). *Indirect model reference adaptive control of quadrotor UAVs using neural networks*. In: Intelligent Systems and Control (ISCO), 20116 10th Int. Conf. on (pp. 1-6).

[8] J.E. Sierra, M. Santos. (2017) *Control de un vehículo cuatrirrotor basado en redes neuronales*, Actas de las XXXVIII Jornadas de Automática, CEA, pp. 431-436.

[9] M. Santos, M. (2011) *Aplicaciones Exitosas de Control Inteligente a Casos Reales*. Revista Iberoamericana de Automática e Informática Industrial.

[10] M. Santos, R. López, and J.M. de la Cruz, J. M. (2006) *A neuro-fuzzy approach to fast ferry vertical motion modelling*. Engineering Applications of Artificial Intelligence, 19(3), 313-321, 2006.

[11] Wise, K. A., Lavretsky, E., & Hovakimyan, N. (2006, June*). Adaptive control of flight: theory, applications, and open problems*. In American Control Conference, 2006 (pp. 6-pp). IEEE.

[12] NASA Glenn Research Center. https://www.grc.nasa.gov/WWW/K-12/airplane/ldrat.html