Neural controller of UAVs with inertia variations

J. Enrique Sierra-Garcia[1][0000-0001-6088-9954], Matilde Santos[2][0000-0003-1993-8368] ,
Juan G. Victores[3][0000-0002-3080-3467]

[1] Department of Civil Engineering, University of Burgos, 09006 Burgos, Spain
jesierra@ubu.es

[2] Computer Science Faculty, Complutense University of Madrid, 28040 Madrid, Spain
msantos@ucm.es

[3] University Carlos III of Madrid, 28911 Madrid, Spain
jcgvicto@ing.uc3m.es

Abstract. Floating offshore wind turbines (FOWT) are exposed to hard environmental conditions which could impose expensive maintenance operations. These costs could be alleviated by monitoring these floating devices using UAVs. Given the FOWT location, UAVs are currently the only way to do this health monitoring. But this means that UAV should be well equipped and must be accurately controlled. Rotational inertia variation is a common disturbance that affect the aerial vehicles during these inspection tasks. To address this issue, in this work we propose a new neural controller based on adaptive neuro estimators. The approach is based on the hybridization of feedback linearization, PIDs and artificial neural networks. Online learning is used to help the network to improve the estimations while the system is working. The proposal is tested by simulation with several complex trajectories when the rotational inertia is multiplied by 10. Results show the proposed UAV neural controller gets a good tracking and the neuro estimators tackle the effect of the variations of the rotational inertia.

Keywords: Inertia variations, neuro-estimator, UAV, neural networks, FOWT.

1 Introduction

In recent years, new and valuable applications of unmanned aerial vehicles (UAV) have emerged in different sectors such as defense, security, construction, agriculture, entertainment, shipping, etc. One of the most recent applications is the inspection and maintenance of offshore wind turbines and, particularly, floating devices (FOWT) [1]. Due to their location in deep seas, to take images and to get those places, efficient and robust UAV controllers are necessary. To address this complex task, soft computing has been proved an efficient approach [2-5], even more when internal parameters of the system changes during the operation.

In this work we address the problem of UAV rotational inertia variations while the system is working. There are few works that study the effect of payload or inertia variation on the quadrotor dynamics. In [6], an adaptive control is used to estimate the parameter variation. In [7], adaptive parameter estimation is used to update the control action according to the current UAV mass and inertia moment. More recently, Wang [8] estimates the variations in the UAV payload and the effect of the wind gusts. But this topic demands further research.

We propose the design of a control strategy based on neural networks to cope with variations in the UAV mass and the rotational inertia. The controller combines feedback linearization, PIDs and adaptive neural estimators. This approach is tested by simulation for several trajectories giving good results.

The paper is organized as follows. Section 2 is focused on the description of the system and the modelling of the UAV inertia variation. The neural controller and the neuro estimators are studied in Section 3. Results are discussed in Section 4. Conclusions and future works end the paper.

2 System and Disturbances Models

A quadrotor is composed by four rotors which lift and propel it. Its dynamics is described by the position (x, y, z) and Euler's angles (ϕ, θ, ψ). The angular and translational dynamics of the system are given by the following equations [9]:

$$\tau = J\dot{\omega} + \omega \times J\omega, \qquad J = \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix}, \qquad m_Q\dot{v} = RT - m_Q g e_3 \tag{1}$$

Where τ is the torque vector (N.m), J is the inertia tensor (Nm^2), ω is the angular velocity (rad/s), m_Q is the UAV mass (Kg), R is the rotation matrix, g is the gravitational acceleration (m/s^2), T is a vector of forces (N). Vectors τ and T are related to the propeller speeds, $\Omega_1, \dots, \Omega_4$, velocities of the rotors 1 to 4 (rad/s), the thrust coefficient b (N.s^2), the drag coefficient d (N.m.s^2), and the longitude of each arm l (m).

Instead of using the rotor speeds to control the UAV, it is a common practice to use the control signals u_1, u_2, u_3 and u_4, as defined by (2).This matrix is invertible, hence we can calculate velocity references for the rotors from a set of control signals.

$$\tau = \begin{pmatrix} bl(\Omega_4^2 - \Omega_2^2) \\ bl(\Omega_3^2 - \Omega_1^2) \\ d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{pmatrix}, T = \begin{pmatrix} 0 \\ 0 \\ b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \end{pmatrix}, \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{2}$$

Finally, from equations (1) and (2), the following system of equations is derived:

$$\ddot{\phi} = \dot{\theta}\dot{\psi}\,(I_y - I_z)/I_x + (lb/I_x)u_2,\ \ddot{\theta} = \dot{\phi}\dot{\psi}\,(I_z - I_x)/I_y + (lb/I_y)u_3 \tag{3}$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta}\,(I_x - I_y)/I_z + (d/I_z)u_4\,, \ddot{X} = -(sin\theta cos\phi)\,(b/m_Q)u_1 \tag{4}$$

$$\ddot{Y} = (sin\phi)\,(b/m_Q)u_1, \ddot{Z} = -g + (cos\theta cos\phi)\,(b/m_Q)u_1 \tag{5}$$

The values used to simulate the model, extracted from [9], are the following: l = 0.232 m; m_Q = 0.52 Kg; d = 7.5e^{-7} N.m.s^2; b = 3.13e^{-5} N.s^2; I_x = 6.23e^{-3} Kg.m^2; I_y = 6.23e^{-3} Kg.m^2; and I_z = 1.121e^{-2} Kg.m^2.

Handling of a load by a UAV produces three main effects: the total mass is increased, the inertia changes and so does the center of gravity. Assuming the center of mass is invariant, we have focused on the influence of the total mass increment and mostly on the variation of the inertia.

The rotational inertia of a set of objects is given by $I = \sum_i m_i r_i^2$, where m_i is the mass (Kg) of object i and r_i is the distance (m) between its center of mass and the turning axle [10]. We consider an object with mass M_L that is taken to a distance L_L from the center of the quadrotor in the Z axis.

If the object is infinitesimal and the mass is concentrated in one point, the Z axle rotational inertia is invariant. The object is attached to the UAV by a without-mass rigid structure. Thus, the total rotational inertia, Ix_T, Iy_T, Iz_T is:

$$Ix_T = Ix + M_L L_L^2, \qquad Iy_T = Iy + M_L L_L^2, \qquad Iz_T = Iz \tag{6}$$

The terms $dist_m$ and $dist_I$ are introduced in (3-5) to represent the influence of the mass and rotational inertia variation in the UAV dynamics. These disturbances are modelled as a step function at the moment when the object is grabbed by the UAV, t_{dist}. After t_{dist} the total mass of the system is $m + M_L = K_m \cdot m$ and the rotational inertias are, respectively, $Ix + M_L L_L^2 = K_{Ix} \cdot Ix$ and $Iy + M_L L_L^2 = K_{Iy} \cdot Iy$, where K_m, K_{Ix} and K_{Iy} are multipliers gains, with the constraint $(K_{Ix} - 1) \cdot Ix = (K_{Iy} - 1) \cdot Iy$.

$$\ddot{\phi} = (\dot{\theta}\dot{\psi}\left(I_y + dist_{Iy} - I_z\right) + lb \cdot u_2)/(I_x + dist_{Ix}), \tag{7}$$

$$\ddot{\theta} = (\dot{\phi}\dot{\psi}\,(I_z - I_x - dist_{Ix}) + lb \cdot u_3)/(I_y + dist_{Iy}) \tag{8}$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta}\,(I_x + dist_{Ix} - I_y - dist_{Iy})/I_z + (d/I_z)u_4 \tag{9}$$

$$\ddot{X} = -(sin\theta cos\phi)\,(b/(m_Q + dist_m))u_1 \tag{10}$$

$$\ddot{Y} = (sin\phi)\,(b/(m_Q + dist_m))u_1 \tag{11}$$

$$\ddot{Z} = -g + (cos\theta cos\phi)\,(b/(m_Q + dist_m))u_1 \tag{12}$$

$$dist_m = M_L \cdot step(t - t_{dist}),\, dist_{Ix} = dist_{Iy} = M_L \cdot L_L^2 \cdot step(t - t_{dist}) \tag{13}$$

3 Description of the neural controller

3.1 Description of the architecture of the control system

Figure 1 shows the control system based on neuro-estimators architecture. It is made up of a position controller, an attitude controller and two neuro-estimators for the mass and the inertia.

The position controller has as input the desired trajectory $(Xref, Xref, Zref)$ and three output: u_1, Φ_{ref} and θ_{ref}. It is based on feedback linearization and three PIDs, one for each coordinate axis, to stabilize the trajectory. It requires the mass value that is changing so it is estimated with a neural network. The Z coordinate is directly controlled by u_1 while the control of X and Y coordinates is carried out by first obtaining the roll and the pitch references (Φ_{ref},θ_{ref}). These are the inputs of the attitude controller, plus an external reference of the yaw (Ψ_{ref}). The attitude controller has the same structure as the position one. The rotational inertia directly affects the angular dynamics and indirectly the translational dynamics. For this reason, the rotational inertia values (Ix_{est}, Iy_{est}) are real-time estimated by the inertia neuro-estimator and introduced as inputs of the controller. Assuming that the inertia in the Z axis does not suffer variations during the operation, the attitude controller generates three outputs: u_2, u_3 and u_4. The roll angle is controlled by u_2, the pitch angle by u_3 and the yaw angle by u_4.

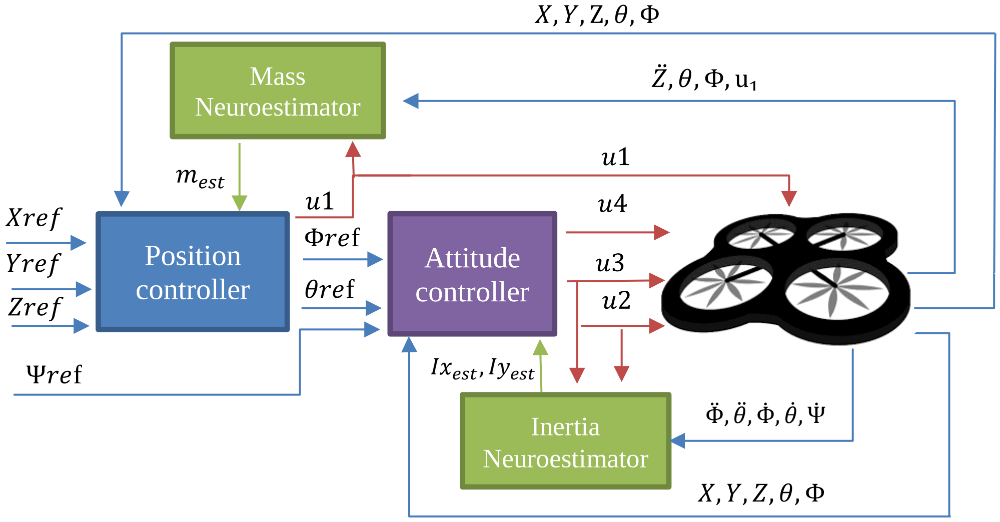

Fig. 1. Architecture of the neural controller.

The internal equations of the position controller are (14-19):

$$rZ(t_i) = \mathrm{K}_{PZ}\left(\mathrm{Z}_{ref}(t_i) - \mathrm{Z}(t_i)\right) + \mathrm{K}_{DZ}\left(\dot{\mathrm{Z}}_{ref}(t_i) - \dot{\mathrm{Z}}(t_i)\right) \tag{14}$$

$$\boldsymbol{u_1}(t_i) = \begin{cases} (rZ(t_i) + g)\frac{m_{est}}{b} & cos\theta_{i-1}cos\phi_{i-1} = 0 \\ (rZ(t_i) + g)\left(\frac{m_{est}}{b \cdot cos\theta_{i-1}cos\phi_{i-1}}\right) & cos\theta_{i-1}cos\phi_{i-1} \neq 0 \end{cases} \tag{15}$$

$$rY(t_i) = \mathrm{K}_{PY}\left(\mathrm{Y}_{ref}(t_i) - \mathrm{Y}(t_i)\right) + \mathrm{K}_{DY}\left(\dot{\mathrm{Y}}_{ref}(t_i) - \dot{\mathrm{Y}}(t_i)\right) \tag{16}$$

$$\boldsymbol{\Phi_{ref}}(t_i) = \begin{cases} \Phi_{ref}(t_{i-1}) & u_1(t_i) = 0 \\ -\mathrm{asin}\left(rY(t_i)\left(\frac{m_{est}}{b \cdot u_1(t_i)}\right)\right) & u_1(t_i) \neq 0 \end{cases} \tag{17}$$

$$rX(t_i) = \mathrm{K}_{PX}\left(\mathrm{X}_{ref}(t_i) - \mathrm{X}(t_i)\right) + \mathrm{K}_{DX}\left(\dot{\mathrm{X}}_{ref}(t_i) - \dot{\mathrm{X}}(t_i)\right) \tag{18}$$

$$\boldsymbol{\theta_{ref}}(t_i) = \begin{cases} \theta_{ref}(t_{i-1}) & u_1(t_i)\cos(\Phi(t_i)) = 0 \\ \mathrm{asin}\left(rX(t_i)\left(\frac{m_{est}}{b \cdot u_1(t_i)\cos(\Phi(t_i))}\right)\right) & u_1(t_i)\cos(\Phi(t_i)) \neq 0 \end{cases} \tag{19}$$

Where m_{est} is the estimated mass (Kg) and $(\mathrm{K}_{PX}, \mathrm{K}_{DX}, \mathrm{K}_{PY}, \mathrm{K}_{DY}, \mathrm{K}_{PZ}, \mathrm{K}_{DZ}) \in \mathbb{R}^6$ are the gains of the internal PIDs. If the neuro-estimator is not used, the value m_{est} is substituted by the mass of the quadrotor m_Q.

The equations of the attitude controller are obtained in a similar way, but in this case the angular dynamic is considered in the linearization (20-25):

$$r\Phi(t_i) = \mathrm{K}_{P\Phi}\left(\Phi_{ref}(t_i) - \Phi(t_i)\right) + \mathrm{K}_{D\Phi}\left(\dot{\Phi}_{ref}(t_i) - \dot{\Phi}(t_i)\right) \tag{20}$$

$$\boldsymbol{u_2}(t_i) = \left(\mathrm{Ix}_{est} \cdot r\Phi(t_i) - (\mathrm{Iy}_{est} - \mathrm{Iz}) \cdot \dot{\theta}(t_i)\dot{\Psi}(t_i)\right)/(l \cdot b) \tag{21}$$

$$r\theta(t_i) = \mathrm{K}_{P\theta}\left(\theta_{ref}(t_i) - \theta(t_i)\right) + \mathrm{K}_{D\theta}\left(\dot{\theta}_{ref}(t_i) - \dot{\theta}(t_i)\right) \tag{22}$$

$$\boldsymbol{u_3}(t_i) = \left(\mathrm{Iy}_{est} \cdot r\theta(t_i) - (\mathrm{Iz} - \mathrm{Ix}_{est}) \cdot \dot{\Phi}(t_i)\dot{\Psi}(t_i)\right)/(l \cdot b) \tag{23}$$

$$r\Psi(t_i) = \mathrm{K}_{P\Psi}\left(\Psi_{ref}(t_i) - \Psi(t_i)\right) + \mathrm{K}_{D\Psi}\left(\dot{\Psi}_{ref}(t_i) - \dot{\Psi}(t_i)\right) \tag{24}$$

$$\boldsymbol{u_4}(t_i) = \left(\mathrm{Iz} \cdot r\Psi(t_i) - (\mathrm{Ix}_{est} - \mathrm{Iy}_{est}) \cdot \dot{\Phi}(t_i)\dot{\theta}(t_i)\right)/(d) \tag{25}$$

Where Ix_{est} and Iy_{est} (Kg.m2) are the rotational inertias in the X and Y axis, respectively, estimated by the inertia neuro-estimator and $(\mathrm{K}_{P\Phi}, \mathrm{K}_{D\Phi}, \mathrm{K}_{P\theta}, \mathrm{K}_{D\theta}, \mathrm{K}_{P\Psi}, \mathrm{K}_{D\Psi}) \in \mathbb{R}^6$ are the gains of the internal PIDs. The values Ix_{est} and Iy_{est} are substituted by Ix and Iy when the neuro-estimator is not used.

3.2 Description of the neuro-estimators

The generic structure of the neuro-estimators is show in Figure 2. They need an analytic model of the UAV and a neural network. The model is used to obtain a measurement of the value to be estimated in order to train the network. But even when the model does not find a right value due to any singularity, the neural network is still able to provide a valid output, generating new knowledge.

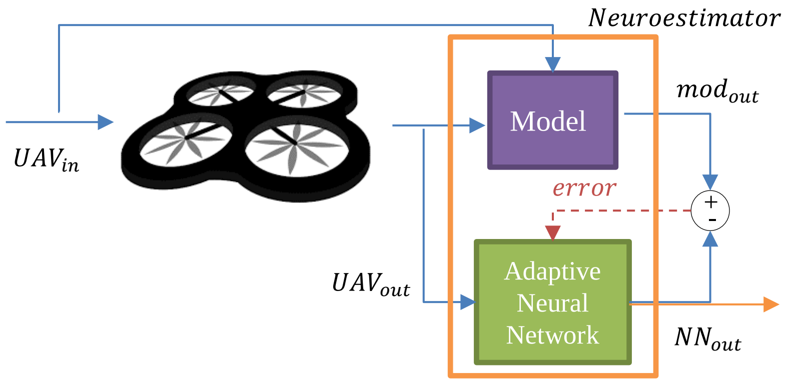

Fig. 2. Structure of the neuro-estimators.

First, the neural network is offline trained with the outputs of the model (mod_{out}) as targets and the outputs of the UAV (UAV_{out}) as inputs. Then, while the UAV system is working, the network is on-line learning from new inputs and targets. So it adapts to changes in the system parameters, and the controller can reject these disturbances.

The equations that represent the performance of the neuro-estimator are given by:

$$NN_{out}(t_i) = \mathrm{f}_{NN}\left(\mathrm{w}_{NN}(t_i), UAV_{out}(t_i)\right) \tag{26}$$

$$\mathrm{w}_{NN}(t_i) = \mathrm{f}_{adapt}(\mathrm{w}_{NN}(t_{i-1}), UAV_{out}(t_{i-1}), mod_{out}(t_{i-1})) \tag{27}$$

$$mod_{out}(t_i) = \mathrm{f}_{mod}(UAV_{in}(t_{i-1}), UAV_{out}(t_{i-1})) \tag{28}$$

Where f_{NN} represents the neural network during the simulation phase, w_{NN} is the set of internal parameters of the network, f_{adapt} is the adaption function, f_{mod} is the analytic model, UAV_{in} is the set of inputs from the UAV and UAV_{out} the set of UAV. For each estimator (mass or inertia), UAV_{in}, UAV_{out}, and the f_{mod} are different.

- **Mass adaptive neuro-estimator:**

$$f_{mod} = \mathrm{b}/((Z + g)/(u_1 \cos\theta \cos\phi)), NN_{out}(t_i) = \mathrm{f}_{NN}\left(\mathrm{w}_{NN}, \ddot{Z}, \cos\theta\,, \cos\phi\right) \tag{29}$$

- **Inertia adaptive neuro-estimator:**

$$f_{mod} = \frac{\ddot{\phi} lbu_3 + \ddot{\phi}\dot{\phi}\dot{\psi} Iz - \dot{\phi}\dot{\psi} lbu_2 + \dot{\phi}\dot{\psi}\dot{\theta}\dot{\psi} Iz}{\dot{\phi}\dot{\psi}\dot{\theta}\dot{\psi} + \ddot{\phi}\ddot{\theta}}, NN_{out}(t_i) = \mathrm{f}_{NN}\left(\mathrm{w}_{NN}, \ddot{\phi}, \ddot{\theta}\right). \tag{30}$$

4 Discussion of the simulation results

Simulation results have been obtained during 16 seconds. The tuple of multipliers (K_m, K_{Ix}, K_{Iy}) has been set to (2, 10, 10) and t_{dist} to 5 seconds. The neural controller is offline trained during the first 3 seconds; from them on the online learning is applied. In both neuroestimators the neural network is a MLP with one hidden layer of 30 neurons. The Levenberg-Marquardt algorithm has been used for the training with μ=0.001.

Figure 3 (left) shows the UAV tracking of a helical lemniscate trajectory (reference in blue and trajectory with neuro-estimators in red). Without neuro-estimators (blue line) the trajectory tracking was wrong (Figure 3, right).

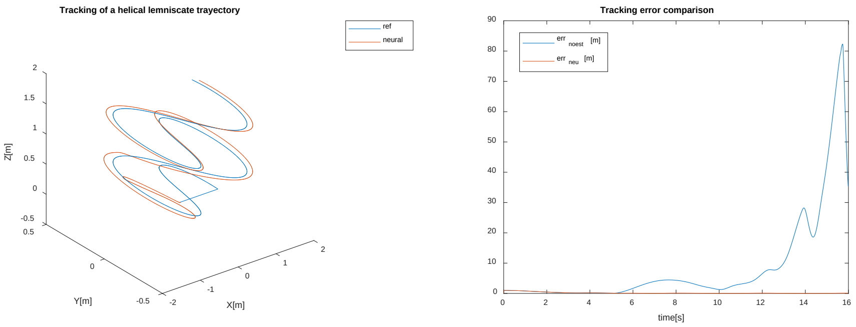

Fig. 3. Tracking of a helical lemniscate (left) and tracking error (right).

Figure 4 shows the UAV trajectory tracking of each coordinate. The results with neuro-estimators (red line) and without them (blue line) regarding the reference (yellow line) proves the efficiency of the neuro-estimators and how the error grows and the system becomes unstable without them. The worst error is in the y-axis.

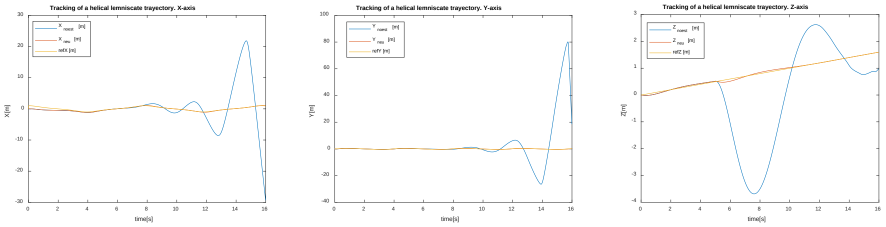

Fig. 4. UAV tracking of X, Y and Z coordinates, respectively, with mass and inertia variation.

Results for different trajectories are shown in Table 1. The columns labelled “Neur” means results with the neuro-estimators vs “Nest”, without them. Values below 0.0001 have been fixed to 0. The best results per column have been underlined. The evaluation criteria are the MSE, U1 (control effort), and PERF (performance of the controller). The latter is calculated multiplying the inverse of $MSE_{T,}$ $MAX_{T,}$ and U1, therefore bigger values indicate better results.

For every trajectory and evaluation criteria the best results are obtained with the neuro-estimators. In general, the best performance is obtained for the helical lemniscate trajectory that also shows the biggest difference depending on the application of the neuro-estimators; indeed, regarding the MAX_T criterium, this value is up to 80 times smaller with the neuro-estimators.

Table 1. Tracking error, control effort and performance comparison for different trajectories.

Trajectory	MSE_X		MSE_Y		MSE_Z		MSE_T		MAX_T		U1		PERF	
	Neur	Nest	Neur	Nest	Neur	Nest	Neur	Nest	Neur	Nest	Neur	Nest	Neur	Nest
Linear	**0,094**	0,098	**0,094**	0,098	**1,208**	2,362	**1,223**	2,377	**10,10**	10,10	**8,650**	8,681	**0,009**	0,004
Circular	**0,192**	0,226	**0,098**	0,172	**1,208**	2,362	**1,246**	2,418	**10,24**	10,24	**8,656**	8,736	**0,009**	0,004
Helical	**0,161**	0,350	**0,037**	0,452	**0,040**	1,120	**0,185**	1,557	**1,007**	4,642	**8,644**	9,993	**0,618**	0,013
Cyc. helical	**0,140**	0,192	**0,012**	0,151	**0,574**	1,335	**0,651**	1,476	**1,141**	5,850	**8,685**	8,832	**0,154**	0,013
Lemniscate	**0,098**	0,271	**0,013**	0,219	**1,208**	2,363	**1,220**	2,521	**10,05**	10,05	**8,652**	9,201	**0,009**	0,004
Hel. lemnisci	**0,156**	3,029	**0,037**	6,635	**0,025**	1,098	**0,180**	8,629	**1**	82,32	**8,684**	42,34	**0,638**	0
Step	**0,983**	1,018	**0,985**	1,012	**1,212**	2,367	**2,011**	3,173	**17,32**	17,32	**8,949**	8,993	**0,003**	0,002

5 Conclusions

In this work a new control strategy based on the hybridization of feedback linearization, PIDs and adaptive neural network estimators is proposed. This approach helps to deal with changes in the rotational inertia of UAV while flying.

The neuro-estimators combine an analytic model of the UAV and MLP neural networks. Online learning of the neural network allows the system to improve the estimation of the changing mass and inertia of the quadrotor.

Simulation results with different trajectories prove the validity of this control strategy with neuro-estimator to stabilize the UAV even under noticeable variations in the mass and in the rotational inertia of the quadrotor.

Among other possible future works, we may highlight the study of the influence of other disturbances such as the ones generated by the engines and the desirable application of this methodology to a real system.

Acknowledgment

This work was partially supported by the Spanish Ministry of Science, Innovation and Universities under Project number RTI2018-094902-B-C21.

References

1. Schäfer, B. E., Picchi, D., Engelhardt, T., & Abel, D. Multicopter unmanned aerial vehicle for automated inspection of wind turbines. In 2016 24th Mediterranean Conference on Control and Automation (MED) (pp. 244-249). IEEE. (2016, June).
2. Sierra, J. E., & Santos, M. Modelling engineering systems using analytical and neural techniques: Hybridization. Neurocomputing, 271, 70-83. (2018).
3. San Juan, V., Santos, M., & Andújar, J. M. Intelligent UAV Map Generation and Discrete Path Planning for Search and Rescue Operations. Complexity. (2018).
4. Santos, M., Lopez, V., & Morata, F. Intelligent fuzzy controller of a quadrotor. In Intelligent Systems and Knowledge Engineering (ISKE), 2010 International Conference on (pp. 141-146). IEEE. (2010, November).
5. Yanez-Badillo, H., Tapia-Olvera, R., Aguilar-Mejia, O., & Beltran-Carbajal, F. On line adaptive neurocontroller for regulating angular position and trajectory of quadrotor system. Revista Iberoamericana de Automática e Informática Industrial, 14(2), 141-151. ISSN: 1697-7912, https://doi.org/10.1016/j.riai.2017.01.001. (2017).
6. Min, B.-C., Hong, J.-H., Matson, E.T.: Adaptive robust control (ARC) for an altitude control of a quadrotor type UAV carrying an unknown payloads. In: 2011 11th International conference on control, automation and systems Korea, pp. 26–29. (2011)
7. Wang, C., Nahon, M., Trentini, M., Nahon, M., Trentini, M.: Controller development and validation for a small quadrotor with compensation for model variation. In: 2014 International conference on unmanned aircraft systems (ICUAS). IEEE. (2014)
8. Wang, C., Song, B., Huang, P., & Tang, C. Trajectory tracking control for quadrotor robot subject to payload variation and wind gust disturbance. Journal of Intelligent & Robotic Systems, 83(2), 315-333. (2016).
9. Nicol, C., Macnab, C. J. B., & Ramirez-Serrano, A. Robust neural network control of a quadrotor helicopter. In 2008 Canadian Conference on Electrical and Computer Engineering (pp. 001233-001238). IEEE. (2008, May).
10. Serway, R. A., & Jewett, J. W. Physics for scientists and engineers with modern physics. Cengage learning. (2018).