Hybrid adaptive control of a dragonfly model

Micael S. Couceiro, Nuno M.F. Ferreira, J.A. Tenreiro Machado

ABSTRACT

Dragonflies show unique and superior flight performances than most of other insect species and birds. They are equipped with two pairs of independently controlled wings granting an unmatchable flying performance and robustness.

In this paper, it is presented an adaptive scheme controlling a nonlinear model inspired in a dragonfly-like robot. It is proposed a hybrid adaptive (*HA*) law for adjusting the parameters analyzing the tracking error. At the current stage of the project it is considered essential the development of computational simulation models based in the dynamics to test whether strategies or algorithms of control, parts of the system (such as different wing configurations, tail) as well as the complete system. The performance analysis proves the superiority of the *HA* law over the direct adaptive (*DA*) method in terms of faster and improved tracking and parameter convergence.

Keywords: Control, Robot, Fuzzy, Dragonfly, Adaptive, Hybrid

1. Introduction

In the last few years, there were significant advances in robotics, artificial intelligence and other fields allowing the implementation of biologically inspired robots [1,2] being one of the major challenges the controllability of those systems since they are nonlinear or even chaotic. Engineers have long been stymied in their attempts to build flying robots that can match the amazing flight capabilities of nature's most advanced flying insects and birds. Such robots could be used for a variety of tasks, from spying, to mine detection or even search and rescue missions in collapsed buildings.

The study of dynamic models based on insects is becoming a field of active research and shows results that can be considered very close to the real systems [3,4]. The dragonfly has been one of the systems under study [5] because it is considered one of the major challenges in the field of aerodynamics. Recent studies show that the aerodynamics of dragonflies is unstable because they use it to fly one way radically different from the steady flight of aircrafts and large birds [6]. Although the flight of insects has been an interesting subject of, at least, half academic century, the control of flying robots inspired in their design represents a high level of complexity since unsteady aerodynamics has not had proper attention.

Fuzzy controllers (*FC*) are supposed to work in situations where there is a large uncertainty or unknown variation in plant parameters and structures [7]. Fuzzy logic systems (*FLS*) provide nonlinear mappings from an input data vector space into a scalar output space, that are general enough to perform any nonlinear control or identification actions [8], for the control and identification of linear and nonlinear systems. However, in order to maintain consistent performance, fuzzy controllers should be equipped with appropriate online adaptive algorithms to form adaptive fuzzy controllers. In [8] it was presented a ‘‘direct fuzzy controller’’ based on *FC* rules, and an ‘‘indirect fuzzy controller’’ based on fuzzy modeling rules. Generally, the basic objective of adaptive control is to maintain consistent performance of a system in the presence of uncertainties.

This paper presents an adaptive methodology to control a simulated model of a dragonfly-like robot. The controller considered in this paper is constructed from fuzzy modeling rules based on [9]. For adjusting the parameters, it is proposed a hybrid adaptive scheme, combining adaptive fuzzy identification and adaptive fuzzy control. In the hybrid scheme, the adaptive algorithm utilizes a combination of two types of error for adjustment. We will apply and compare the performance analysis of the direct and the hybrid adaptive *FC* to control the nonlinear mathematical model based on the dragonfly kinematics and dynamics implemented in *MatLab*/*Simulink* [10].

The paper is organized as follows. In the Section 2 we present the implemented adaptive *FC*. Section 3 presents the robotic dragonfly mathematical modeling and control architecture. In Section 4 we compare the performance of both the direct adaptive and the hybrid adaptive *FC*. Finally, Section 5 outlines the main conclusions.

2. Adaptive fuzzy control

The basic configuration of an adaptive fuzzy control system is shown in Fig. 1. The *Reference Model* is used to specify the ideal response that the *FC* system should follow. The *Plant* is assumed to contain unknown components. The fuzzy controller is contrasted from fuzzy systems whose parameters *h* are adjustable. The *Adaptation Law* adjusts the parameters *h* online such that the plant output $y(t)$ tracks the reference model output $y_m(t)$.

A common approach for constructing adaptive controllers is the self-tuning method [11,12]. In this strategy, first a design method (for known plants) is used to provide a controller structure and a relationship between plant and controller parameters. The plant parameters are estimated using an online parameter identification algorithm. The controller parameters are then obtained from the estimates of the plant parameters as if these were the true plant parameters. This idea is often called the certainty equivalence principle.

Consider the nth-order nonlinear system of the controllability canonical form [13]:

$$\begin{cases} x^{(n)} = f(x, \dot{x}, \ldots, x^{(n-1)}) + g(x, \dot{x}, \ldots, x^{(n-1)})u \\ y = x \end{cases} \qquad (1)$$

where f and g are $_T$ unknown real continuous functions, u and y are the input and output of the system, respectively, and x ¼ ðx; x; . . . ; $x^{ðn-1Þ}$Þ is the state vector of the system which is assumed to be available for measurement. The controllability of (1) requires that gðxÞ – 0 for all x in a certain controllability region U_cCR^n . The control objective is to find a feedback control law u ¼ uðx; tÞ such that to make the state xðtÞ track a given desired bounded reference trajectory ym ðtÞ ¼ ðy_m; y; . . . ; $y_m^{ð}$ $_{n-1Þ}$ ^{T} Þ .

We assume the following collection of *If-Then* rules that describe the input–output behaviour of $f(x)$ and $g(x)$:

$$If x_1 is A_1^r and \ldots and x_n is A_n^r, Then f(\bar{x}) is C^r \qquad (2)$$

$$If x_1 is B_1^s and \ldots and x_n is B_n^s, Then g(\bar{x}) is D^s \qquad (3)$$

respectively, where A_i^r's and B_i^s's are fuzzy sets in R; C^r, and D^s are fuzzy sets in R which achieve membership value one at some point, $r = 1, 2, \ldots, N_f$, and $s = 1, 2, \ldots, N_g$.

If the plant model is not known, it is intuitively reasonable to replace it by an estimated model and use this model for designing the controller. This is the basic idea of a self-tuning adaptive controller, in which the controller is designed based on an estimated model of the plant (assuming this model is the true model of the plant) and the estimated model parameters are updated by an online algorithm.

Now consider the problem of controlling the system (1). If the plant dynamics is known, *i.e.*, the functions f and g are known, we can solve the control problem stated above by the so-called feedback linearization method. In this method, the functions f and g are used to construct the following feedback control law:

g x

$$u = u(\bar{x}, t) = \frac{1}{g(\bar{x})}\left[-f(\bar{x}) + y_m^{(n)}(t) + \bar{k}^T\bar{e}\right] \qquad (4)$$

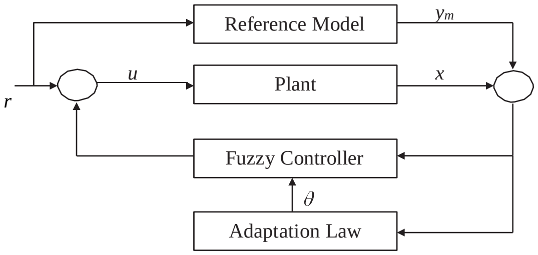

Fig. 1. Basic configuration of the adaptive fuzzy control system.

where $e = y_m(t) - y(t)$ is tracking error, (e ¼ e; e; . . . ; e^ðn−1Þ), and ðk ¼ k_n; . . . ; k_2; k_1Þ^T is chosen such that all roots of the polynomial $h(s) = s^n + k_1 s^{n-1} + \ldots + k_n$ are in the open left-half of the complex plane. Applying the control law (2) to the system (1) results in the following error dynamics:

$$e^{(n)} + k_1 e^{(n-1)} + \cdots + k_n e = 0 \tag{5}$$

This implies that starting from any initial conditions, we have lim n ! 1jeðtÞj ¼ 0, *i.e.*, tracking of the reference trajectory is asymptotically achieved. However, since *f* and *g* are unknown, we cannot use them to build the control law (4). Therefore, we replace c¼ g x u them by their estimates ^f and g^ to construct a self-tuning controller:

$$u_c = \frac{1}{\hat{g}(\bar{x}|\bar{\theta}_g)}\left[-\bar{f}(\bar{x}|\bar{\theta}_f) + y_m^{(n)}(t) + \bar{k}^T\bar{e}\right] \tag{6}$$

where -hf and -hg > d are parameters of the approximating systems ^f and g^, respectively.

The following hybrid adaptive law was proposed by Hojati in [9] in order to adjust the parameters:

$$\begin{cases} \dot{\bar{\theta}}_f = -\gamma_1\left[\gamma\varepsilon + \bar{e}^T P\bar{b}_c\right]\bar{\xi}(\bar{x}) \\ \dot{\bar{\theta}}_g = -\gamma_2\left[\gamma\varepsilon + \bar{e}^T P\bar{b}_c\right]\bar{\xi}(\bar{x})u_c \end{cases} \tag{7}$$

where nðxÞ is the vector of fuzzy basis functions, c_1, c_2 and c are positive constants, e is the modeling error defined in (8), P is a matrix that satisfies the Lyapunov equation [9] and b$_c$ ¼ ð0; . . . ; 0; 1Þ^T.

The modeling error can be written as:

$$\varepsilon = \left[\bar{\phi}_f + \bar{\phi}_f u_c\right]^T\bar{\xi}(\bar{x}) + w \tag{8}$$

where $\bar{\phi}_f = \bar{\theta}_f - \bar{\theta}_f^*$, $\bar{\phi}_g = \bar{\theta}_g - \bar{\theta}_g^*$, and $w = \left(\hat{f} - f\right) + (\hat{g} - g)u_c$.

3. Dragonfly

The mathematical model of the dragonfly system is the same analyzed in [10] with the support of [14] and the MSc thesis [15]. Next subsection presents an overview of the mathematical modeling that will be used as the plant to be controlled following the control architecture presented in sub Section 3.2.

3.1. Mathematical modeling

In this section it is analyzed the mathematical analysis of the biomechanical model and this can be made from two different perspectives. The first is the kinematic perspective that considers the movement characteristics and studies the movement from a spatial and temporal perspective. The second perspective is the dynamics which analyzes the forces acting in the system defining the forces that origin the movements [16].

Two types of flight can be considered: quasi-steady and unsteady states. For larger birds, the flights can be approximated by quasi-steady state assumptions because their wings flap at lower frequency during cruising. This means the wingtip speed is lower when compared to the flight speed. Thus larger birds, such as eagles and seagulls, tend to have a soaring flight. Their wings behave closely to fixed-wings. On the other hand, smaller birds and insects (*e.g.*, dragonflies) fly in an unsteady state regime [17] as their wingtip speed is faster than their flight speed.

The model presented in this paper was based on a dragonfly which flight is considered unsteady with a flapping frequency near 20 Hz [18]. Tables 1 and 2 compares some of the most studied insect features [19,20]. Besides the tabled features, there are several unique characteristics of the dragonfly (*e.g.*, two pair of wings, tail influence and flying styles) that are the main reasons the dragonfly was used as model. Those will be discussed later on this paper.

In order to visualize the models' behaviour, while in simulation, we developed a 3D model in *AutoCAD* inspired in a dragonfly. The model is presented in Fig. 2 and 3, where each adjacent part represented with different colors correspond to individual elements connected through joints. The used axis system is also presented in Fig. 2, in order to make it easier to understand the equations and dynamic analysis that follows. This axis system is relative to the *World Coordinate System* in which the model is located, which means that the formulae obtained for the calculation of the forces are only valid if the model is in agreement with Fig. 2. The method used to calculate the forces depending on the rotations arising from the model are based on the kinematic structures.

Table 1
Some features from different insects.

	Weight [kg]	Wing area [m^2]	Flapping frequency [Hz]	Maximum velocity [m/s]
Bumblebee	0.32×10^{-3}	0.19×10^{-3}	130.0	3.0
Dragonfly	1.00×10^{-3}	1.00×10^{-3}	20.0	12.5
Butterfly	3.00×10^{-3}	2.80×10^{-3}	12.5	2.5

Table 2
Dragonfly's kinematic transformation for each link.

Link	Kinematic transformation
Body (*A*)	T^0_4 ¼ $T^0_1 \cdot T^1_2 \cdot T^2_3 \cdot T^3_4$
Tail (*B*)	T^0_6 ¼ $T^0_4 \cdot T^4_5 \cdot T^5_6$
Left wing $n^o1(C)$	T^0_{10} ¼ $T^0_4 \cdot T^4_7 \cdot T^7_9 \cdot T^9_{10}$
Right wing n^o1 (*D*)	T^0_{11} ¼ $T^0_4 \cdot T^4_8 \cdot T^8_9 \cdot T^9_{11}$
Left wing n^o2 (*E*)	T^0_{15} ¼ $T^0_4 \cdot T^4_{12} \cdot T^{12}_{14} \cdot T^{14}_{15}$
Right wing n^o2 (*F*)	T^0_{16} ¼ $T^0_4 \cdot T^4_{13} \cdot T^{13}_{14} \cdot T^{14}_{16}$

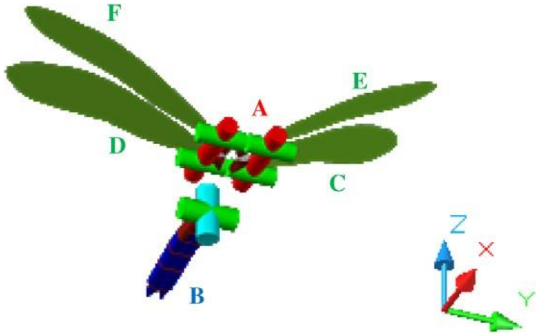

Fig. 2. Control diagram of the dragonfly system.

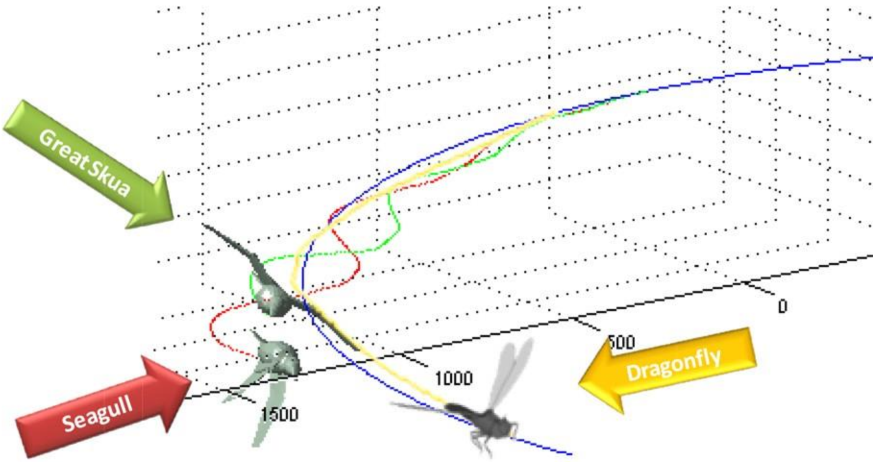

Fig. 3. Chart obtained through the simulator *SIRB* & *LIB SIM* that shows the difference between the trajectory accomplished by a great skua (very large bird), a seagull (large bird) and a dragonfly. The stability of this last one when compared to the others is undeniable.

In order to implement the dragonfly's animation in *MatLab,* the Denavit–Hartenberg (*DH*) notation [21] was followed to represent frame (joint) coordinates for a kinematic chain of revolute and translational joints. Based on the *DH* tables and transformation matrices the dragonfly model presented in appendix A we obtained the following kinematic transformation for each link of the models Table 2.

The dragonfly model is being studied by some researchers due to the unique juggling maneuvers of this creature. Jane Wang [22] developed a set of equations based on a real model of a dragonfly by watching its flight in laboratory.Based on research already developed in this field, and performing a geometric analysis of the dragonfly, it was possible to reach a simpler model with a high-quality response when comparing to what it is seen in nature Fig. 2. The major difference between the geometry of two-winged animals (*e.g.*, birds) and the geometry of the dragonfly are reflected in two pairs of wings. Similarly to birds, the dragonfly also has several movements and flying styles. The flight capabilities of dragonflies are prodigious. In addition to the individual states of take-off, gliding and flapping, this last one is divided into four different styles due to the two pairs of wings: counter-stroking (where the front and rear wings beat with a delay of 180°), phased-stroking (in which the wings beat with a difference of 90°), synchronized-stroking (in which the four wings are synchronized as a single pair of wings), and gliding such as occurs in large birds (*e.g.*, seagull). We will give special attention to the most common style in which the two pairs of wings of the dragonfly beat with a delay of 180° (*i.e.*, counter-stroking). Also, the tail and each pair of wings have the same degrees of freedom (rotational) found in other flying models such as birds. The wings will be treated as a flexible link, similarly to what is seen in the nature, minimizing the area of the wing when on a downward movement. This structure will provide a good mobility, making it a total of ten controllable links.

The relative wind acting on a wing produces a certain amount of force which is called the total aerodynamic force. This force can be resolved into components, called Lift (9) and Drag (10).

$$L = \frac{1}{2} \cdot \rho \cdot v_\infty^2 \cdot S \cdot C_l(\alpha) \tag{9}$$

$$D = \frac{1}{2} \cdot \rho \cdot v_\infty^2 \cdot S \cdot C_d(\alpha) \tag{10}$$

The Lift L is the component of aerodynamic force perpendicular to the relative wind and the Drag D is the component of aerodynamic force parallel to the relative wind. Both forces depends on the wing area S, the density of air q, the freestream velocity v_1 and the Lift and Drag coefficients named as C_l and C_d respectively, expressed as functions of the angle of attack a.

The Lift and Drag coefficients depend on the shape of the airfoil and will alter with changes in the angle of attack and other wing trimmings. The characteristics of any particular airfoil section can conveniently be represented by graphs showing the amount of lift and drag obtained at various angles of attack, the lift-drag ratio, and the movement of the centre of pressure.

Similarly to [23] we adopted the blade-element theory representing the Lift (11) and Drag (12) coefficients as functions of the angle of attack of the local wind.

$$C_l = C_{lmax} \cdot sin(2 \cdot \alpha) \tag{11}$$

$$C_d = C_{do} + C_{dmax} \cdot sin^2(\alpha) \tag{12}$$

Since we are not considering any particular wing aerodynamics at this point, the wing aerodynamics properties of maximum lift $C_{l\max}$ and drag $C_{d\max}$ coefficients as well as zero drag C_{d0} coefficient used in simulations for the dragonfly model are depicted in Eqs. (13)–(15).

$$C_{lmax} = 2 \tag{13}$$

$$C_{do} = 0.05 \tag{14}$$

$$C_{dmax} = 1 \tag{15}$$

The dragonfly's aerodynamic characteristics presented above determine how far and for how long it can glide, and how successfully it can soar in moving air in order to reduce the flight energy. In the case of the dragonfly, and even many of the insects, gliding can be divided in three types: free flight, where the dragonfly simply stops flapping its wings in order to lose altitude for a few seconds; adjusting the shape of wings, where the dragonfly is adjusting the angle of attack of the wings to float in the air without the need to flap, in order to perform a specific operation; gliding with the help of another insect, in which the female usually performs the control direction without flapping their wings while the male provides the driving force.

However, in order to create a positive global thrust force, flapping flight is required. As seen previously, the forces of Lift and Drag will depend on the angle of attack. However, which will be the behaviour of these forces when flapping wings? As [23] we considered the existence of an advance angle related with the flapping velocity and the freestream velocity (16).

$$\delta = atan\left(\frac{w_f}{v_\infty}\right) \tag{16}$$

The advance angle will then be zero when the velocity of the wings is zero, falling in the previously analyzed situation in gliding flight. This means that through the angle of attack it is possible to control the amplitude of the forces of Lift and Drag. On the other hand, the angle that these forces have relatively to the air flow can be controlled through the flapping velocity. If the wing is placed into a flow velocity, v_1, a thrust force will develop due to the horizontal component of the Lift that appears in the downstroke. So, in order to have a positive thrust, the wing will have to increase its velocity to overcome the opposing horizontal force generated in the aerodynamic Drag. The horizontal (x-axis) and vertical (z-axis) forces are related with the Lift, Drag and advance angle by the following equations (17):

$$\begin{cases} F_x = L \cdot sin\delta - D \cdot cos\delta \\ F_z = L \cdot cos\delta + D \cdot sin\delta \end{cases} \tag{17}$$

The dragonfly dynamics is somehow similar to other flying creatures such as birds and, consequently, the same equations may be considered. Nevertheless, when it comes to the flapping flight, the dragonfly takes a great advantage over birds and other two-winged creatures. The following figure shows the result obtained using a previously developed simulator *SIRB* & *LIB SIM* performed using the standard features presented in [24] and Table 1 (*i.e.*, weight, wing area and flapping frequency) of the great skua, the seagull and the dragonfly.

Recent studies reveal that dragonflies use a complex aerodynamics in order to fly, different from aircrafts and large birds. A dragonfly flaps its wings to create a whirlwind of air that is controlled and used to provide lift as aircrafts depend on good air flow over the top and bottom surfaces of their wings. For these machines the turbulence can be fatal. There are other creatures with a similar mechanism to the flight of the dragonfly, but with a higher level of complexity. Creatures such as the hummingbird, surprisingly can manipulate the feathers of the wings during the rapid flapping. The two pairs of wings of the dragonfly allow different independent flight techniques (as mentioned above) and the most common style is the counter-stroking. This type of flight allows that, when a pair of wings beats down creating a vortex of air, the other pair, which is still down, captures the energy of that vortex. Therefore, the air flow over the surface of the wings of the dragonfly has a much higher rate along the bottom of the wing creating more lift. In other words, the different states of flight, downstroke and upstroke, are indistinguishable creating an almost steady force positive to the movement and opposite to the weight. However, in order for the dragonfly to be able to accomplish some of the most amazing flight techniques, the use of the tail is inevitable.

Although the majority of avian flight studies have focused on the wings, the tail also appears to be crucial to the evolutionary success of dragonflies as flying organisms. In addition to causing a significant drag force (as in the case of the birds) the weight factor provides a more efficient use of the tail. Fig. 4 depicts a simplified object diagram of masses (P) of the implemented dragonfly model.

As it can be seen, when moving the tail, *i.e.*, the object of mass m, the global center of mass of the dragonfly will suffer a strong variation. Consider, for example, P_{body} equal to twice P_{tail} and L_{body} equal to half L_{tail}, which seams accurate with real dragonfly anatomy [25]. The equilibrium is reached when the tail is stretched and the force from the resultant F_z forces of Lift and Drag is zero. When the dragonfly flaps its wings it causes a positive force in the z-axis and the center of mass is modified. In order for the dragonfly to remain parallel to its horizontal path it needs to change the rotation of the tail establishing the following relationship (18):

$$P_{tail}L_{tail}cos(\emptyset_{tail}) = P_{body}L_{body} - (L.cos\delta + D.sin\delta) \tag{18}$$

To change the direction in the xy-plane something similar can be seen: to generate an imbalance in the overall mass of the system, the dragonfly will tend to rotate the tail in the x-axis. This is the principle of the pendulum and the tail of the dragonfly can be considered as a bidimensional-pendulum (with two degrees of freedom – according to the xy-plane and xz-plane).

3.2. Control architecture

Nonlinear control is one of the biggest challenges in modern control theory. While linear control system theory has been well developed, it is the nonlinear control problems that present the most challenges. Nonlinear processes are difficult to control because there can be so many variations of the nonlinear behaviour.

The first attempt to control our system will be changing the flapping frequency, angle of attack and tail rotations accordingly to the position error. The following control diagram in Fig. 5 depicts the dragonfly system.

In order to analyze the previous control diagram it is needed to understand the behaviour of our system for certain variations of the error (in this case, the position error). As it can be seen the wing speed inevitably depends on the sum of the position errors in x, y and z-axis being limited to a minimum and maximum saturation which in turn is associated to the simulated model. Based on what we see in nature the dragonfly model the wing speed is limited between 0 cycles/s and 10 cycles/s. The Left (wing) and Right (wing) Angle of Attack will allow the execution of different maneuvers (*e.g.*, turn/

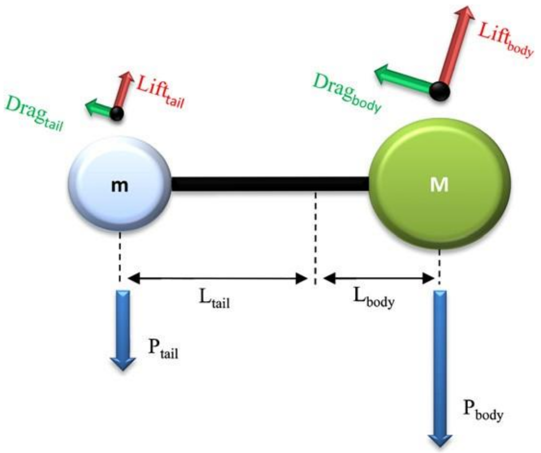

Fig. 4. Approximate decomposition of the body of a dragonfly in objects.

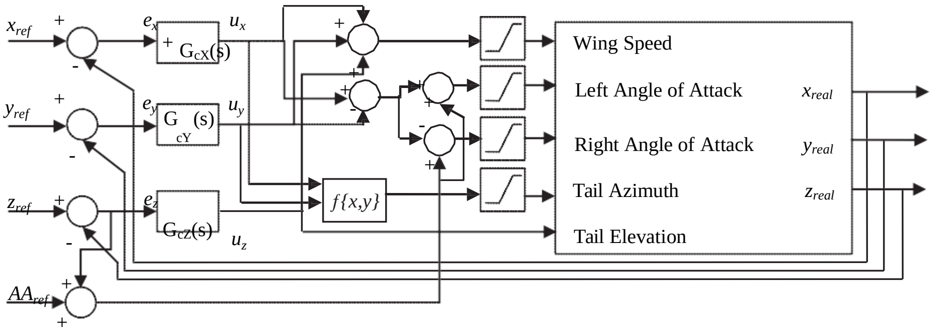

Fig. 5. Control diagram of the dragonfly system.

change direction, spin on its axis) and will depend on the position error in the *xy*-plane, *i.e.*, the difference between the position error in *x* and the position error in *y*. In this perspective, we add two references: a reference value (AA_{ref}) being the value considered to be ideal for the model to follow a path without deviation from the *xy*-plane (straight path) and the position error in the *z*-axis error (elevation) to ensure that the model follows the desired trajectory (*e.g.*, going up while changing direction). The Tail Azimuth angle will depend on a function *f*(*errorX*,*errorY*) which depends on the position error in *x*-axis and in the *y*-axis. This angle is only intended to assist the rotation maneuvers (regardless on the model in question, although the dynamics inherent to the use of the tail are different). The nonlinear function *f*(*errorX*, *errorY*) will systematically adjust the azimuth angle of the tail in order to adjust the actual position on the *xy*-plane. For example, if it is intended to turn left (*i.e.*, if the *xy*-plane error starts to increase), it will result in an incremental azimuth angle of the tail to the left (negative spin along the *z*-axis) until the error decreases. The Tail Elevation angle depends only on the position error in the *z*-axis (elevation).

4. Controller performances

We choose$d$ = 0.4, c_1 = 2, c_2 = 1 and c = 4 for the hybrid adaptive fuzzy controller and defined ten fuzzy sets over each axis and the following membership functions for i = 1, 2:

$$\begin{cases} \mu_{F_i^1}(x_i) = 1/[1+e^{5(x_i+4)}] \\ \mu_{F_i^2}(x_i) = e^{-(x_i+3.5)^2} \\ \mu_{F_i^2}(x_i) = e^{-(x_i+2.5)^2} \\ \mu_{F_i^2}(x_i) = e^{-(x_i+1.5)^2} \\ \mu_{F_i^3}(x_i) = e^{-(x_i+0.5)^2} \\ \mu_{F_i^4}(x_i) = e^{-(x_i+0.5)^2} \\ \mu_{F_i^5}(x_i) = e^{-(x_i+1.5)^2} \\ \mu_{F_i^5}(x_i) = e^{-(x_i+2.5)^2} \\ \mu_{F_i^5}(x_i) = e^{-(x_i+3.5)^2} \\ \mu_{F_i^6}(x_i) = 1/[1+e^{5(x_i+4)}] \end{cases} \tag{19}$$

The initial conditions $\hbar_f$ð0Þ and $\hbar_g$ð0Þ were chosen randomly in the intervals [−30; 30] and [d; 30], respectively. The signal $x_1(t)$ and $x_2(t)$ represents the velocity and the acceleration in the x-axis (horizontal axis), respectively and the initial conditions are $x_1(0)$ = 2 and $x_2(0)$ = 0. The reference trajectory is:

$$\bar{y}_m(t) = ((1/2)sin(1/2t),(1/4)cos(1/2t))^T \tag{20}$$

Figs. 6 and 7 depict the response of the system under the action of the direct and hybrid adaptive fuzzy control, respectively.

As it is possible to conclude through the analysis of Figs. 6 and 7 the hybrid adaptive controller provides a faster tracking convergence with lower initial overshoots for both the velocity and acceleration of the dragonfly robot. While the direct adaptive law makes the system to become stable at a peak-to-peak velocity of 0.55 m/s (against a peak-to-peak velocity of 1 for the ideal situation) and a peak-to-peak acceleration of 0.33 m/s^2(against a peak-to-peak acceleration of 0.5 for the ideal situation), the system under the action of the hybrid adaptive law stabilizes at a peak-to-peak velocity of 0.85 m/s and a peak-to-peak acceleration of 0.45 m/s^2.

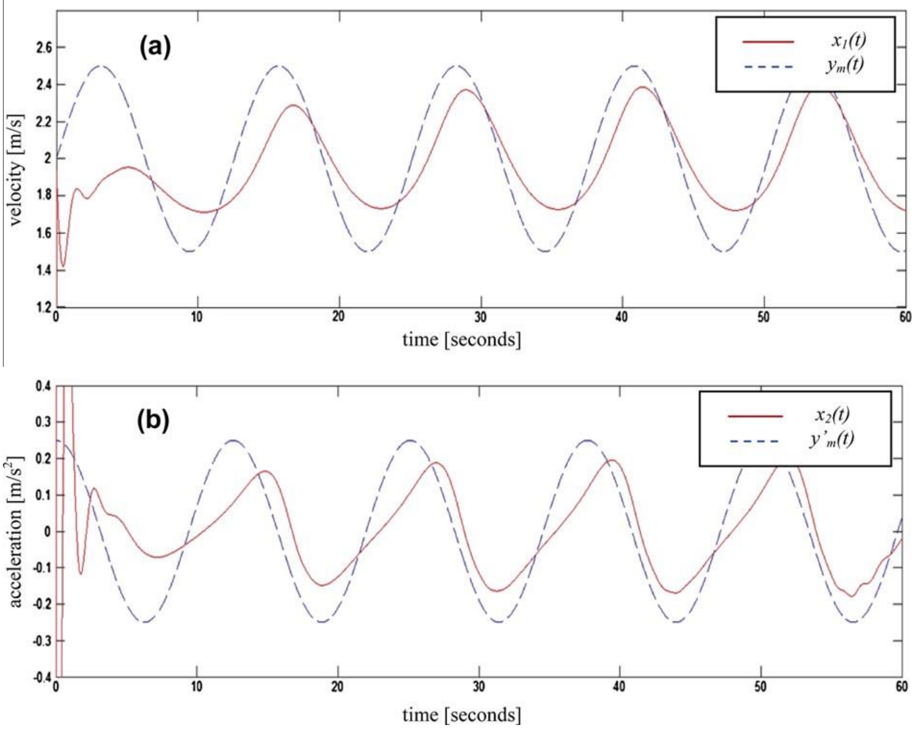

Fig. 6. Time response of the dragonfly system under the action of the direct adaptive fuzzy controller: (a) x_1 signal; (b) x_2 signal.

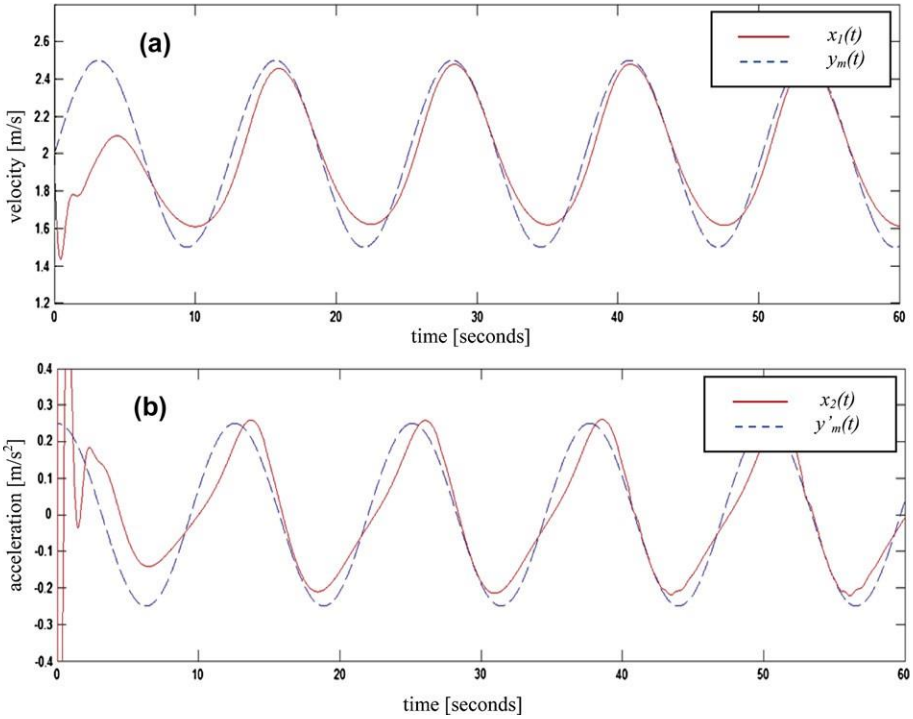

Fig. 7. Time response of the dragonfly system under the action of the hybrid adaptive fuzzy controller: (a) x_1 signal; (b) x_2 signal.

Fig. 8 and 9 depicts the norm of the tracking error vector defined in (11) versus time in the logarithmic scale for the direct adaptive law and the hybrid adaptive law where it can be verified the supremacy of this last one.

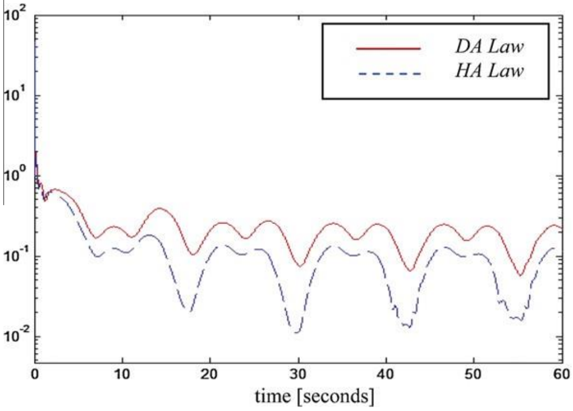

Fig. 8. Norm of tracking error vector in controlling the dragonfly system using the direct adaptive fuzzy controller (dashed line), and the hybrid adaptive fuzzy controller (solid line).

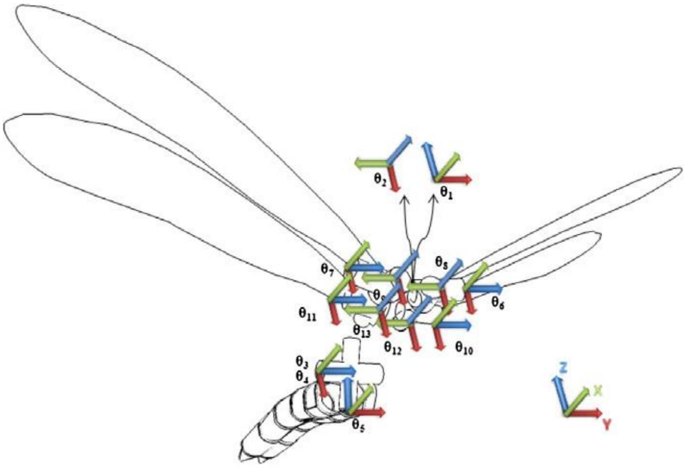

Fig. 9. Dragonfly’s kinematics.

Table 3
DH model of the dragonfly.

	X		*Z*	
	a	*a* [°]	d	*h* [°]
1	0	0	0	$h_1 - 90°$
2	0	−90°	0	h_2
3	0	90°	0	90°
4	$-L_1$	−90°	0	h_3
5	0	0	0	h_4
6	0	90°	0	h_5
7	L_1	0	L_2	h_6
8	L_1	0	$-L_2$	$-h_7$
9	0	90°	0	−90°
10	0	90°	0	h_8
11	0	90°	0	$-h_9$
12	$L_1 - L_3$	0	L_2	h_{10}
13	$L_1 - L_3$	0	$-L_2$	$-h_{11}$
14	0	90°	0	−90°
15	0	90°	0	h_{12}
16	0	90°	0	$-h_{13}$

$$|e(t)| = \sqrt{e^2(t) + \dot{e}^2(t)} = \sqrt{(y_m(t) - x_1(t))^2 + (\dot{y}_m(t) - x_2(t))^2} \tag{21}$$

As it can be seen, and emphasizing the previous conclusion, Fig. 8 shows that the tracking error analyzed in the hybrid adaptive controller reaches a value lower than the one obtained using the direct adaptive controller.

5. Conclusion

In this paper we developed a hybrid combined direct and indirect adaptive fuzzy controller to control the nonlinear mathematical model based on the dragonfly dynamics. In direct adaptive control, the controller parameters are directly adjusted and no effort is made for identifying the plant parameters while that in indirect adaptive control, the controller parameters are based on the estimated model parameters.

The obtained results appeared to be satisfactory, proving that the development of the kinematical and dynamic model can show the behaviour of different flying creatures. The information concerning the physical nature of the flapping flight in the dragonfly proved to be important to analyze solutions. Despite all simplifications, our model is still quite complex, and further research needs to be conducted to explore additional abstractions.

The simulation results confirm the superiority of the *HA* law (fast tracking error convergence, fast and improved parameter convergence). They also show that the hybrid adaptive fuzzy controller could perform successful control without incorporating any linguistic description into the design.

Appendix A

The 3D animation of the dragonfly developed in *MatLab* was made following the Denavit–Hartenberg (*DH*) notation as it is depicted in Table 3 and consequently represented by the transformation matrices Eqs. (22)–(27).

$$\mathrm{T}_1^0 = \begin{bmatrix} s_1 & c_1 & 0 & 0 \\ -c_1 & s_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathrm{T}_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathrm{T}_3^2 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{22}$$

$$T_4^3 = \begin{bmatrix} c_3 & -s_3 & 0 & -L_1 \\ 0 & 0 & 1 & 0 \\ -s_3 & -c_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_5^4 = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_6^5 = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{23}$$

$$T_7^4 = \begin{bmatrix} c_6 & -s_6 & 0 & L_1 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_9^7 = T_9^8 = T_{14}^{12} = T_{14}^{13} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{24}$$

$$T_{10}^9 = \begin{bmatrix} c_8 & -s_8 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_8 & c_8 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_8^4 = \begin{bmatrix} c_7 & s_7 & 0 & L_1 \\ -s_7 & c_7 & 0 & 0 \\ 0 & 0 & 1 & -L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{11}^9 = \begin{bmatrix} c_9 & s_9 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -s_9 & c_9 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{25}$$

$$T_{12}^4 = \begin{bmatrix} c_{10} & -s_{10} & 0 & L_1 - L_3 \\ s_{10} & c_{10} & 0 & 0 \\ 0 & 0 & 1 & L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{15}^{14} = \begin{bmatrix} c_{12} & -s_{12} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_{12} & c_{12} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{26}$$

$$T_{13}^4 = \begin{bmatrix} c_{11} & s_{11} & 0 & L_1 - L_3 \\ -s_{11} & c_{11} & 0 & 0 \\ 0 & 0 & 1 & -L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{16}^{14} = \begin{bmatrix} c_{13} & s_{13} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -s_{13} & c_{13} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{27}$$

With the *DH* transformation matrices, the relationship between the links that compose the kinematic structure of the dragonfly can be calculated Table 2.

References

[1] Couceiro MS, Carlos M. Figueiredo, NM, Fonseca Ferreira JA, Tenreiro Machado, (2009a). Biological inspired flying robot. In: Proceedings of IDETC/CIE 2009 ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference August 30 – September 2, San Diego.

[2] Bar-Cohen Y, Breazeal C. Biologically-Inspired Intelligente Robots, Vol. 2PM12. SPIE Press; 2003.

[3] Schenato L, Deng X, Wu WC, Sastry S. Virtual Insect Flight Simulator (VIFS): A Software Testbed for Insect Flight. In: IEEE Int. Conf. Robotics and Automation, Seoul, Korea, (2001).

[4] Jane Z, Wang. Dissecting insect flight. Annu Rev Fluid Mech 2005:183–210.

[5] Tamai M, Zhijian Wang, Ganesh Rajagopalan, Hui Hu, 2007. Aerodynamic Performance of a Corrugated Dragonfly Airfoil Compared with Smooth Airfoils at Low Reynolds Numbers. In: 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, Jan.

[6] Kesel Antonia B. Aerodynamic characteristics of dragonfly wing sections compared with technical aerofoils. J Experiment Biol 2000;203(20):3125–35.

[7] Wang Li-Xin. A course in fuzzy systems and control. Upper Saddle River, NJ: Prentice-Hall Inc.,; 1996.

[8] Wang Li-Xin. Stable adaptive fuzzy control of nonlinear systems. IEEE Trans Fuzzy Syst 1993;1:146–55.

[9] Hojati M, Gazor S. Hybrid adaptive fuzzy identification and control of nonlinear systems. IEEE Trans Fuzzy Syst 2002;10(2):198–210.

[10] Couceiro MS, Fonseca Ferreira NM, Tenreiro Machado JA. Analysis and Control of a Dragonfly-Inspired Robot. In: International Symposium on Computational Intelligence for Engineering Systems. Porto; 2009. p. 18–9.

[11] Astrom KJ, Wittenmark B. Adaptive Control. Reading. MA: Addison-Wesley; 1989.

[12] Sastry S, Bodson M. Adaptive control: stability, convergence, and robustness. Upper Saddle River, NJ: Prentice-Hall; 1989.

[13] Isidori A. Nonlinear control systems. New York: Springer-Verlag; 1989.

[14] Couceiro MS, Carlos M. Figueiredo, Fonseca Ferreira NM, Tenreiro Machado JA, 2009c. The Dynamic Modeling of a Bird Robot. In: 9th Conference on Autonomous Robot Systems and Competitions, Robotica 2009, Castelo Branco, Portugal, 07 May.

[15] Couceiro MS, 2010. Desenvolvimento e Controlo de Robôs Voadores Biologicamente Inspirados. In: MSc Thesis at the Institute of Engineering of Coimbra.

[16] Hamill J, Knutzen KM, 2009. Biomechanical Basics of Human Movement. 3rd ed., Bastimore. MD: Lippincott. Williams and Wilkins.

[17] Colozza A. Fly Like a Bird. IEEE Spectrum 2007;44(5):38–43.

[18] Wakeling JM, Ellington CP. Dragonfly flight II. Velocities, accelerations, and kinematics of flapping flight. J Exp Biol 1997;200:557–82.

[19] Tanaka H, Hoshino K, Matsumoto K, Shimoyama I, 2005. Flight Dynamics of a Butterfly-type Ornithopter''. 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '05), pp. 310–315, Edmonton, Alberta, Canada, August 2–6.

[20] Zuo D, Peng S, Chen W, Zhang W. Numerical Simulation of Flapping-Wing Insect Hovering Flight at Unsteady Flow. Int J Numerical Methods Fluid, Vol. 53. Wiley InterScience; 2007. p. 1801–17.

[21] Denavit J, Hartenberg RS. A kinematic notation for lower-pair mechanisms based on matrices. Trans ASME 1955 J Appl Mech 1955:215–21.

[22] Wang Z Jane. Dissecting insect flight. Annu Rev Fluid Mech 2005:183–210.

[23] Parslew Ben, 2005. Low Order Modelling of Flapping Wing Aerodynamics for Real-Time Model Based Animation of Flapping Flight. MSc Thesis in School of Mathematics, University of Manchester.

[24] Couceiro MS, Ferreira NMF, Machado JAT. Aplication of Fractional Algoritms in the Control of a Robotic Bird. In: the Journal of Comunications in Nonlinear Science and Numerical Simulation-Special Issue. Elsevier; 2010.

[25] Duncan Jim, 1999. Manitoba Dragonfly Survey: Basic Biology, NatureNorth.com.