

## Research Article

# Modeling and Control of a Dragonfly-Like Robot

Micael S. Couceiro,<sup>1</sup> N. M. Fonseca Ferreira,<sup>1</sup> and J. A. Tenreiro Machado<sup>2</sup>

<sup>1</sup>Department of Electrotechnical Engineering, Institute of Engineering of Coimbra, Rua Pedro Nunes, Quinta da Nora, 3030-199 Coimbra, Portugal

<sup>2</sup>Department of Electrotechnical Engineering, Institute of Engineering of Porto, Rua Dr. António Bernardino de Almeida, 4200-072 Porto, Portugal

Correspondence should be addressed to N. M. Fonseca Ferreira, nunomig@isec.pt

Received 19 December 2009; Revised 31 March 2010; Accepted 10 June 2010

Academic Editor: Seul Jung

Copyright © 2010 Micael S. Couceiro et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Dragonflies demonstrate unique and superior flight performances than most of the 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, the dynamics of a dragonfly-inspired robot is studied. The system performance is analyzed in terms of time response and robustness. The development of computational simulation based on the dynamics of the robotic dragonfly allows the test of different control algorithms. We study different movements, the dynamics, and the level of dexterity in wing motion of the dragonfly. The results are positive for the construction of flying platforms that effectively mimic the kinematics and dynamics of dragonflies and potentially exhibit superior flight performance than existing flying platforms.

## 1. Introduction

The study of dynamic models based on insects is becoming popular and shows results that may be considered very close to reality [1, 2]. One of the models under study is based on the dragonfly [3] because it is considered a major challenge in terms of dynamics. Recent studies show that the aerodynamics of dragonflies is unstable because they use a flying method radically different from steady or quasisteady flight that occurs in aircrafts and flapping or gliding birds [4]. This unsteady aerodynamic has not received proper attention due to the inherent level of complexity.

The technological advances allow the construction of robotic systems that are able to perform tasks of some complexity. In the past, there were significant advances in robotics, artificial intelligence, and other areas, allowing the implementation of biologically inspired robots [5]. Therefore, researchers are investing in reverse engineering based on the characteristics of animals. The progress of technology resulted in machines that can recognize facial expressions, understand speech, and perform movements very similar to living beings.

Some interesting examples are spiders [6], snakes [7], insects [8], and birds [9, 10]. They all require an extensive

study of both the physical and the behavioral aspect of real animals.

Bearing these ideas in mind, the paper is organized as follows. Section 2 presents the state of the art in the area. Sections 3 provides an overview of the physical structure and the kinematics of the dragonfly. Sections 4 and 5 describe the dragonfly dynamics developing the dynamical analysis and the control algorithms, respectively. Finally, Section 6 outlines the main conclusions.

## 2. State of the Art

Inspired by the unique characteristics of animals, researchers have placed a great emphasis on the development of biological robots. This chapter addresses the studies and previous work done in this area focusing on the development of robots inspired in flying animals.

Modern airplanes are extremely effective for steady, level flight in still air. Propellers produce thrust very efficiently, and today's cambered airfoils are highly optimized for speed and/or efficiency. However, examining performance in more interesting flight regimes reveals why birds and insects are still the true masters of the sky.



FIGURE 1: Sequence of images illustrating the wings-beat of the robotic bird SIRB.

The evolution of powered flight from theropod dinosaurs (i.e., large bipedal dinosaurs) up to birds and insects is recognized as the key adaptive breakthrough that contributed to the biological success of this group. Some birds are capable of migrating thousands of kilometers with incredibly small energy consumption—the wandering albatross can fly for hours, or even days, without flapping its wings by exploiting the shear layer formed by the wind over the ocean surface in a technique called dynamic soaring. Remarkably, the flight metabolic cost for large birds is indistinguishable from the baseline metabolic cost, suggesting that they can travel incredible distances powered almost completely by gradients in the wind. Other birds achieve efficiency through similarly rich interactions with the air including formation flying, thermal soaring, and ridge soaring. Small birds and large insects, such as butterflies and dragonflies, use gust soaring to migrate hundreds or even thousands of kilometers carried primarily by the wind.

The flight of insects has been an interesting subject of, at least, half a century, but serious attempts to recreate it are much more recent [11]. Aircraft designers have been interested in increasing the morphic capabilities of wings and

this area received a major boost in 1996, when the Defense Advanced Research Projects Agency of the U.S. (DARPA) launched a MAV of three years in order to create a flying platform with less than 15 centimeters long for surveillance and reconnaissance.

Some other biological inspired platforms have been developed such as the *Dragonfly* from *Wow Wee!*. The *Dragonfly* toy was developed in 2007 and it is controlled by a radio transmitter. It looks like a dragonfly with a wingspan of 40.6 centimeters, with a lightweight body and strong double wings. As the dragonfly beats the wings to fly it does not need a propeller to generate a thrust force. It only uses a propeller in the tail to move left or right.

In 2008 a robotic platform inspired by the flight of birds was developed at ISEC. SIRB (*Simulation and Implementation of a Robotic Bird*) was built based on the results obtained using a simulator developed in *Matlab* [12] (Figure 1).

While the developments of robotic platforms described above are a positive step in the production of new biologically inspired flying robots, there is a subarea that does not have the proper attention of researchers: the control and autonomous navigation of robots.

Some studies have been appearing on the area of autonomous navigation of flying robots, studying new techniques of odometry and vision [13]. Fumiya Iida developed control algorithms with the Reichard model conducting experiments in an autonomous flying airship robot in an unstructured environment [13].

The control of flying robots, even if not inspired in flying animals, represents a high level of complexity. Puntunan and Parnichkun [14] compared the classical *PID* with a self-tuning *PID* algorithm for the control a small helicopter. The results obtained with the self-tuning *PID* proved that this type of control offers a better performance than the classical *PID*. However, it was possible to observe some relatively high overshoots in the system response.

In this paper we address other control and optimization methods comparing the results obtained in order to make the system steadier and, thereby, obtaining a better performance.

### 3. Kinematic Analysis

The dragonfly model is being studied due to the unique juggling maneuvers of this creature. Jane Wang [2] developed a set of equations based on a real model of a dragonfly by watching its flight in laboratory.

The objective in defining the geometry is to develop a physical model that can be mathematically described as being comparable to the actual real dragonfly. Based on some works already developed in this area, and performing a geometric analysis of the dragonfly, it was possible to reach a relatively simple model with a high-quality response when comparing to what it is observed in nature.

As we can see, the major difference between the geometry of two-winged animals (e.g., birds) and the geometry of the dragonfly is 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 degrees), phased-stroking (in which the wings beat with a difference of 90 degrees), synchronized-stroking (in which the four wings are synchronized as a single pair of wings), and gliding such as that occurs in large birds. 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 degrees (counter-stroking) that will be explained in the sequel.

Based on the geometry, and following an analysis of the multi-link model, we estimated the location of every joint in the robot and obtained the kinematic model represented in Figure 2.

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, for minimizing the area of the wing when being on the downward movement. This

structure will provide a good mobility, making it a total of ten controllable links.

The 3D animation developed in *MatLab* was made following the Denavit-Hartenberg (D-H) notation as it is depicted in Table 1 and consequently represented by the transformation matrices (1).

$$\begin{aligned}
 T_1^0 &= \begin{bmatrix} s_1 & c_1 & 0 & 0 \\ -c_1 & s_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & T_2^1 &= \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 T_3^2 &= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & T_4^3 &= \begin{bmatrix} c_3 & -s_3 & 0 & -L_1 \\ 0 & 0 & 1 & 0 \\ -s_3 & -c_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 T_5^4 &= \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & T_6^5 &= \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 T_7^4 &= \begin{bmatrix} c_6 & -s_6 & 0 & L_1 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & \\
 T_9^7 &= T_9^8 = T_{14}^{12} = T_{14}^{13} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & \\
 T_{10}^9 &= \begin{bmatrix} c_8 & -s_8 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_8 & c_8 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & T_8^4 &= \begin{bmatrix} c_7 & s_7 & 0 & L_1 \\ -s_7 & c_7 & 0 & 0 \\ 0 & 0 & 1 & -L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 T_{11}^9 &= \begin{bmatrix} c_9 & s_9 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -s_9 & c_9 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & T_{15}^{14} &= \begin{bmatrix} c_{12} & -s_{12} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_{12} & c_{12} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 T_{13}^4 &= \begin{bmatrix} c_{11} & s_{11} & 0 & L_1 - L_3 \\ -s_{11} & c_{11} & 0 & 0 \\ 0 & 0 & 1 & -L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & T_{16}^{14} &= \begin{bmatrix} c_{13} & s_{13} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -s_{13} & c_{13} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{1}
 \end{aligned}$$

With the D-H transformation matrices, we can calculate the relationship between the links that compose the kinematic

TABLE 1: D-H Dragonfly model.

	<i>X</i>	<i>Z</i>		
	a	$\alpha$ [degrees]	d	$\theta$ [degrees]
1	0	0	0	$\theta_1 - 90^\circ$
2	0	$-90^\circ$	0	$\theta_2$
3	0	$90^\circ$	0	$90^\circ$
4	$-L_1$	$-90^\circ$	0	$\theta_3$
5	0	0	0	$\theta_4$
6	0	$90^\circ$	0	$\theta_5$
7	$L_1$	0	$L_2$	$\theta_6$
8	$L_1$	0	$-L_2$	$-\theta_7$
9	0	$90^\circ$	0	$-90^\circ$
10	0	$90^\circ$	0	$\theta_8$
11	0	$90^\circ$	0	$-\theta_9$
12	$L_1 - L_3$	0	$L_2$	$\theta_{10}$
13	$L_1 - L_3$	0	$-L_2$	$-\theta_{11}$
14	0	$90^\circ$	0	$-90^\circ$
15	0	$90^\circ$	0	$\theta_{12}$
16	0	$90^\circ$	0	$-\theta_{13}$

TABLE 2: Kinematic Transformation for each link of the dragonfly.

Link	Kinematic transformation
Body	$T_4^0 = T_1^0 \cdot T_2^1 \cdot T_3^2 \cdot T_4^3$
Tail	$T_6^0 = T_4^0 \cdot T_5^4 \cdot T_6^5$
Left wing no. 1	$T_{10}^0 = T_4^0 \cdot T_7^4 \cdot T_9^7 \cdot T_{10}^9$
Right wing no. 1	$T_{11}^0 = T_4^0 \cdot T_8^4 \cdot T_9^8 \cdot T_{11}^9$
Left wing no. 2	$T_{15}^0 = T_4^0 \cdot T_{12}^4 \cdot T_{14}^{12} \cdot T_{15}^{14}$
Right wing no. 2	$T_{16}^0 = T_4^0 \cdot T_{13}^4 \cdot T_{14}^{13} \cdot T_{16}^{14}$

structure of the dragonfly. Table 2 shows the kinematic transformation for each link of the dragonfly.

#### 4. Dynamical Analysis

The dragonfly dynamics is somehow similar to other flying creatures such as birds [15] 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 (Figure 3).

Recent studies reveal that dragonflies use a complex aerodynamics to fly, differently from aircrafts and large birds. A dragonfly flaps its wings to create a whirlwind of air that is controlled and used to provide lift. On the other hand, planes 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 mechanism similar to the flight of the dragonfly, but with a higher level of complexity, such as the hummingbird, that can surprisingly manipulate the feathers of the wings during the rapid flapping. However, the study of dragonfly flight shows that it can be as efficient as the hummingbird but with a much easier flight system. More than 200 million years of evolution provide evidences of a successful and infallible aerodynamics.

The two pairs of wings 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 contrary to the weight. Nevertheless, applying this principle to the development of flying platforms is complex because the effect has to be simple and predictable. Less than ten years ago, people saw the flows generated by the insects as something uncontrollable. The turbulence was, and still is, often seen as something undesirable, causing failures in the turbines of the aircrafts and reducing their effectiveness. In the case of the rotor of helicopters, the blades sometimes fail because each blade is continuously affected by the turbulence generated by the preceding blade, causing vibrations that may weaken the metal. However, for the dragonfly, this type of flight is something natural and extremely efficient as we shall see in the next section.

We have undertaken a dynamical analysis to test the validity of the system model. In order to easily change the parameters (e.g., wing area, weight) we built a computer program highlighting the fundamentals of robot mechanics and control.

The computer programs emphasize capabilities such as the 3D graphical simulation and the programming language giving some importance to mathematical aspects of modeling and control [16].

We start by presenting several results of the dragonfly dynamics around the gliding flight. These results are based on different parameters of the dragonfly. In each simulation the wind has a constant velocity of  $v = 5.0$  m/s against the movement of the dragonfly that has an initial velocity of  $v_0 = 3.0$  m/s. We change the weight and the area of the wing parameters in order to analyze the dragonfly dynamics. The initial parameters are a total weight of  $m = 10^{-3}$  kg and the wing an area of  $S = 10^{-4}$  m<sup>2</sup>.

For the dragonfly to fly in a straight line, without flapping its wings, a continuously changing of the angle of attack (alpha) is needed to keep a vertical resulting force equal to zero. The angle of attack will then increase the lift and the drag forces. A higher drag force results in the reduction of the velocity. This process stops when the velocity reaches zero since we do not want the dragonfly to be dragged by the wind.

In the following experiments that can be seen in Figures 4–7 we will change the mass and the wing area in increments of 25% and 10% of the initial parameters, respectively.

As we can see, increasing the weight requires a higher angle of attack in order to fly. The dragonfly keeps gliding for a short amount of time when compared to large birds. Despite the weight that is also well below the weight of

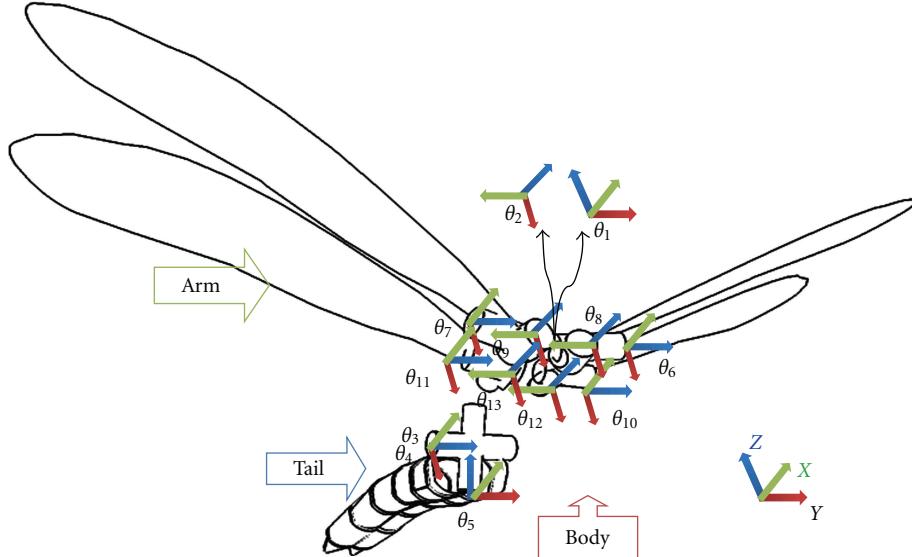


FIGURE 2: Kinematic structure of the system.

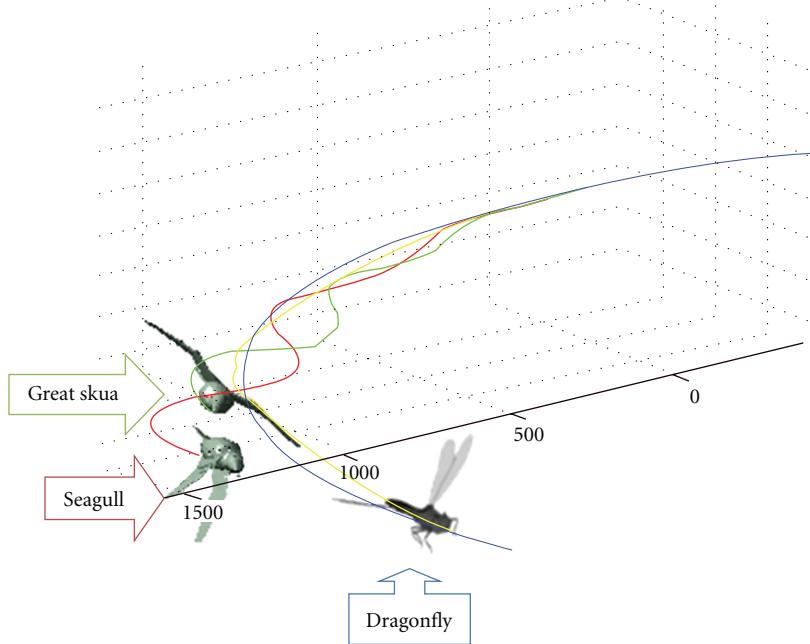


FIGURE 3: Chart obtained through the developed simulator that shows the difference between the trajectory accomplished by a great skua (very large bird), a seagull (large bird) and a dragonfly. The stability of this last one when compared to the others is undeniable.

the large flying creatures, like soaring birds, the area of the wings does not allow gliding for a long time. Obviously, the dragonfly, like all insects or small birds, does not have the same ability to glide as a large bird.

An interesting aspect is the fact that by increasing the weight of the dragonfly it can glide longer. This can easily be explained: if you throw a feather against the wind it will not go as far as if you throw a stone. As we increase the weight of the dragonfly we are giving it the chance to fight against

the wind more easily; however, we are also ensuring that it needs a higher angle of attack of the wings which, on the other hand, will eventually reduce the speed anyway.

By increasing the area of the wings the dragonfly does not need to significantly increase the angle of attack because it can keep gliding more easily (Figures 6 and 7).

Birds, particularly large ones, adopt this technique much more frequently than insects do. Nevertheless, insects also use it, although not with the purpose of saving energy, since

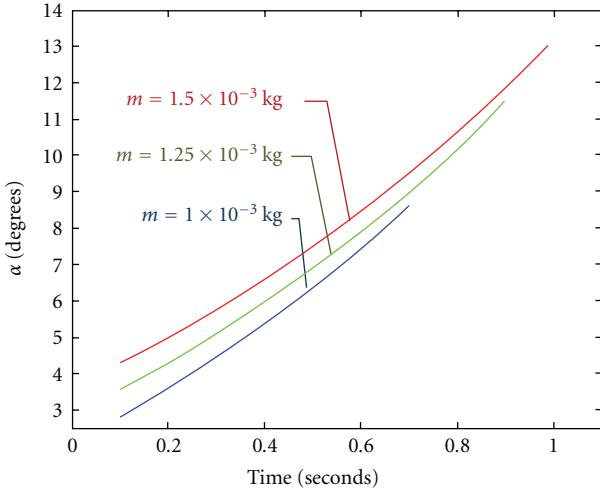


FIGURE 4: Dragonfly gliding straight—changing the weight. Angle of attack versus time.

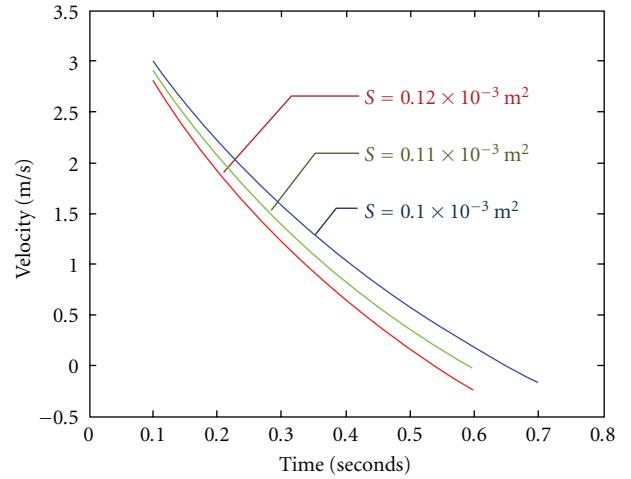


FIGURE 7: Dragonfly gliding straight—changing the wing area. Velocity versus time.

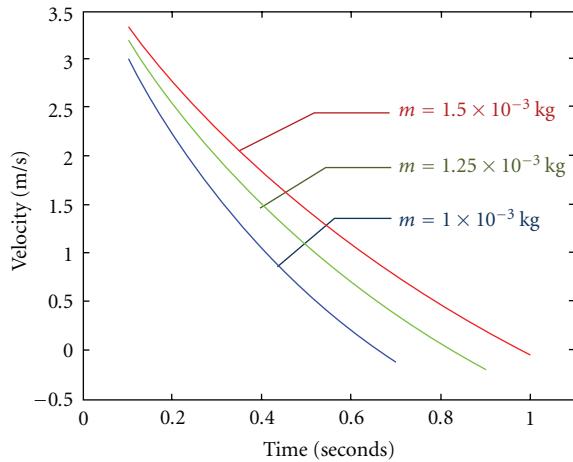


FIGURE 5: Dragonfly gliding straight—changing the weight. Velocity versus time.

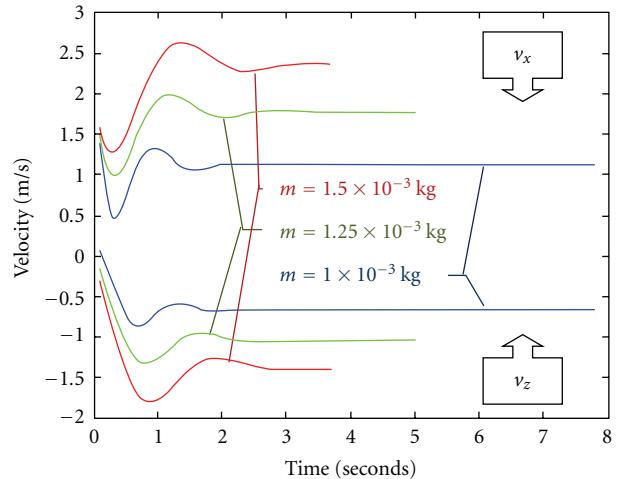


FIGURE 8: Dragonfly gliding down—changing the weight. Velocity versus time.

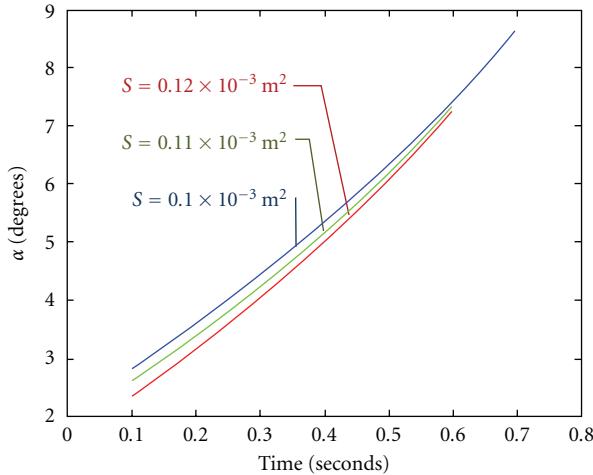


FIGURE 6: Dragonfly gliding straight—changing the wing area. Angle of attack versus time.

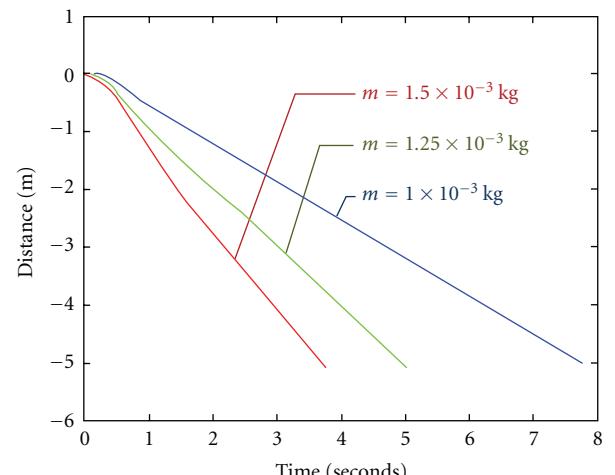


FIGURE 9: Dragonfly gliding down—changing the weight. Distance versus time.

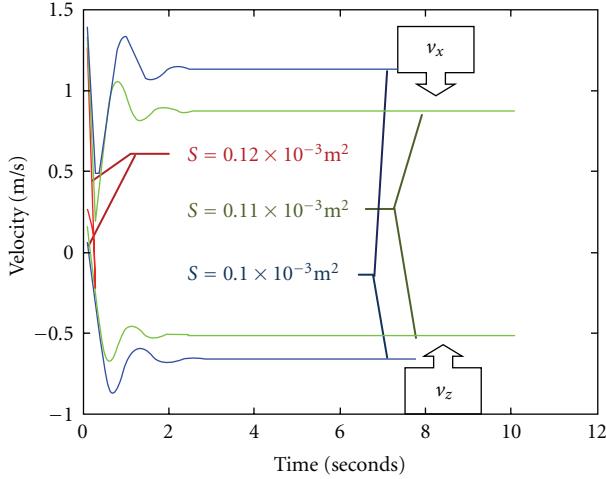


FIGURE 10: Dragonfly gliding down—changing the wing area. Velocity versus time.

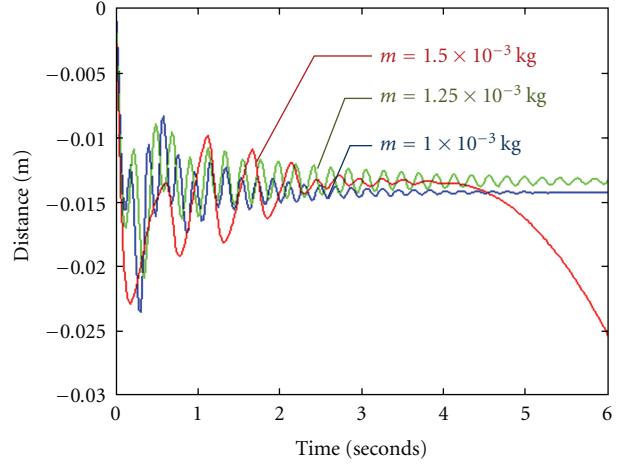


FIGURE 13: Dragonfly flapping straight—changing the weight. Distance versus time.

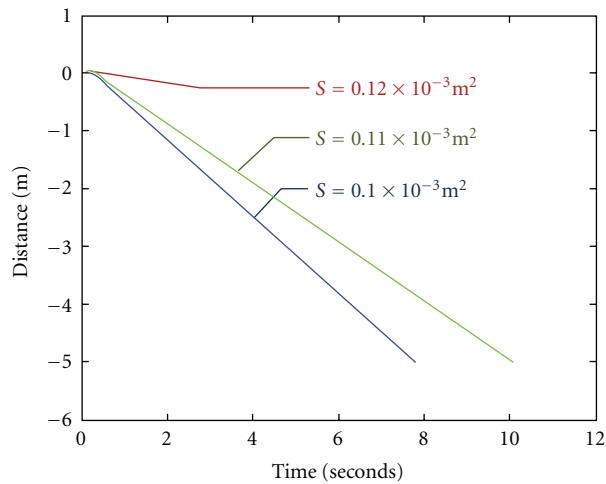


FIGURE 11: Dragonfly gliding down—changing the wing area. Distance versus time.

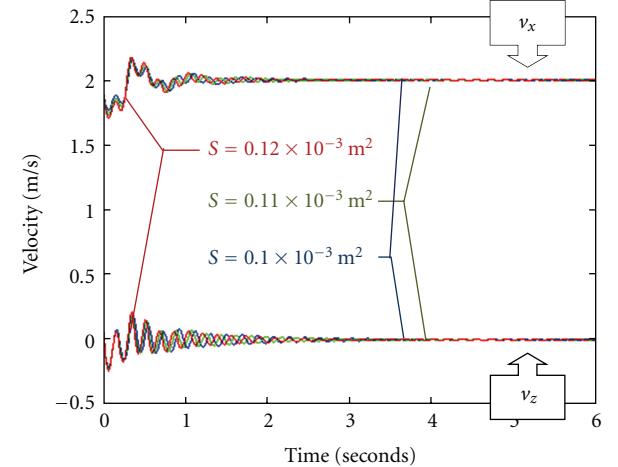


FIGURE 14: Dragonfly flapping straight—changing the wing area. Velocity versus time.

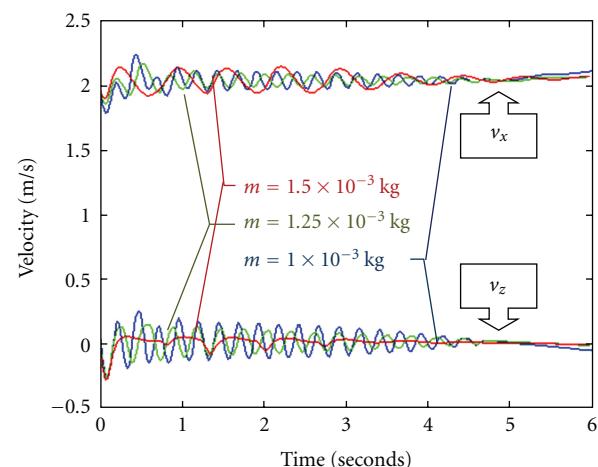


FIGURE 12: Dragonfly flapping straight—changing the weight. Velocity versus time.

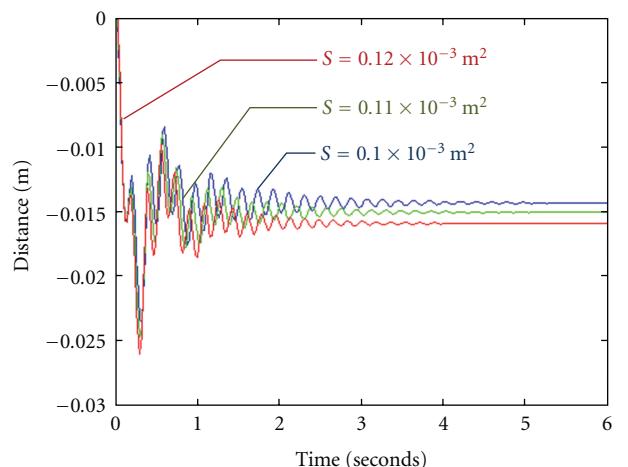


FIGURE 15: Dragonfly flapping straight—changing the wing area. Distance versus time.

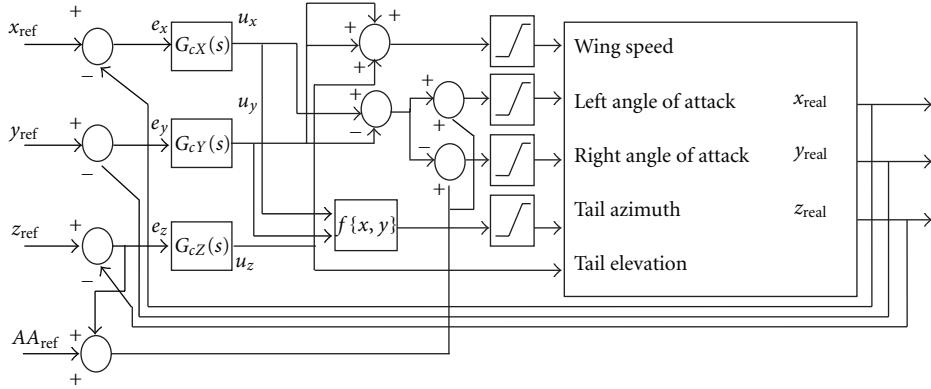
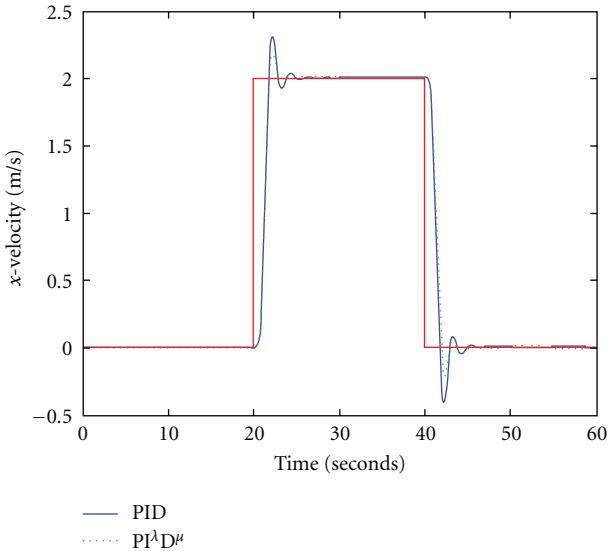


FIGURE 16: Control diagram of the dragonfly

FIGURE 17: Time response of the system under the action of the PID and  $PI^{\lambda}D^{\mu}$  controllers.

the difference is not relevant, but to accomplish some specific maneuvers.

The second experiment (Figures 8–11) shows the horizontal ( $v_x$ ) and vertical ( $v_z$ ) velocities of the bird as well as the vertical distance obtained when the bird is gliding down a vertical distance of 5.0 meters, when considering a fixed angle of attack in both wings.

It is obvious that, when we increase the weight of the dragonfly, it reaches the desired vertical distance faster. However, there is a ubiquitous aspect that must be emphasized: the movement is much more linear than the movement of larger creatures such as birds. The reason is the relation between the area of the wings and the weight.

Let us compare the flight with the one of a large bird: while the wings of the dragonfly are, let us suppose, 100 times smaller than the wings of the bird, the weight of the dragonfly is about 400 times smaller. By doing this imbalance in the weight/area of the wings we assure that the flying movement is more linear. Based on what we just said and taking into

account the large difference between the weight/area of the wings of the dragonfly, if we increase the area of the wing even more then the movement will be even more linear. We can confirm the idea in Figures 10 and 11.

Nevertheless, this relationship is not as straight as it seems in the previous charts. It is true that increasing the area of the wings by 10% the movement becomes more linear and it can eventually perform the desired trajectory smoothly and with a lower speed. However, increasing the area over 10% the dragonfly cannot achieve the desired position. This is due to the fact that the size of the wings is so large, when compared to the weight, that the drag caused by the wings is too high so that the resultant force in  $x$ -axis reaches zero.

This shows that the relationship weight/area of the wings of the dragonfly is ideal and that manipulating this relationship can eventually have unexpected results and may compromise the good efficiency of the dragonfly flight.

We will now analyze the flapping flight of the dragonfly to understand how it works in order to implement a control algorithm. The analysis of the flapping flight is not as simple as for the case of the gliding flight. In the next experiment, we must note that our first priority is to fly in a straight line.

Following a similar line of thought of the gliding flight we change the weight and wing area. Figures 12 and 13 show how the velocities and vertical distance react while changing the bird weight.

The previous figures show that the dragonfly can maintain a very straight trajectory except for a weight 50% higher, because it begins to slightly lose some altitude. However, the flight starts with an initial velocity  $v_0 = 2 \text{ m/s}$  and remains near this value even with the significant increase in the weight.

It is easy to understand that if we increase the area of the wings of the dragonfly (Figures 14 and 15), then the flapping wings response will be enhanced. This effect is opposed to the previous experiment, where the significant increase of the area of the wing brought some inconvenience in the gliding flight, because of the lack of thrust force. A larger area of the wings means a smaller settling time of the dragonfly velocity as can be easily seen in Figure 15.

The difference of effectiveness between the dragonfly and large birds mainly focuses on the flight stability.

The dragonfly can eventually overcome variations in the parameters (e.g., weight, area of the wings) more easily than birds and other two-winged creatures. The dragonfly maintains a regular wing-beat of 3.0 to 5.0 flaps/s (depending on the weight and wing area) not making use of the gliding flight such as large birds do. The experiments in the next section with the optimized controllers will give us a better understanding about the real stability and performance of the dragonfly flight.

## 5. Controller Performances

In this section we develop several experiments for comparing the performances of the *FO* (Fractional Order) *PID* algorithms [17, 18].

The first attempt to control our system will be changing the wing speed velocity, angle of attack and tail rotations accordingly to the position error (Figure 16).

In order to analyze the previous control diagram we need to understand the behavior of our system for certain variations of the error (in this case, the position error).

The wing speed inevitably depends on the sum of the position errors in  $x$ -,  $y$ - and  $z$ -axes being limited to a minimum and maximum saturation which in turn is associated to the simulated model. Experimentally, and based on what we see in nature, the wing speed is limited between 0 cycles/s and 10 cycles/s.

The Left (wing) and Right (wing) Angles of Attack are what will allow the execution of different maneuvers (e.g., turn/change direction, spin on its axis) and depend on the position error in the  $xy$ -plane, that is, the difference between the position error in  $x$  and the position error in  $y$ . To this result we add two references: a reference value ( $AA_{ref}$ ) being the value considered to be ideal, so the model can 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 can follow the desired trajectory (e.g., going up while changing direction).

The Tail Azimuth angle will depend on a function  $f(error_x, error_y)$  which depends on the position error in  $x$ -axis and in the  $y$ -axis. This angle is only intended to assist the rotation maneuvers. The nonlinear function  $f(error_x, error_y)$  will systematically adjust the angle of azimuth of the tail in order to adjust the actual position on the  $xy$ -plane. For example, if the dragonfly turns 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).

In this paper we will compare the performance of the integer and fractional order (*FO*) *PID* controllers. *FO* controllers are algorithms whose dynamic behavior is described through differential equations of non integer order. Contrary to the classical *PID*, where we have three gains to adjust, the *FO PID*, also known as  $PI^{\lambda}D^{\mu}$  ( $0 < \lambda, \mu \leq 1$ ), has five tuning parameters, including the derivative and the integral orders to improve the design flexibility.

TABLE 3: *PID* and  $PI^{\lambda}D^{\mu}$  controller parameters.

	$K_{pX}$	$K_{iX}$	$K_{dX}$	$\mu_X$	$\lambda_X$	$K_{pZ}$	$K_{iZ}$	$K_{dZ}$	$\mu_Z$	$\lambda_Z$
<i>PID</i>	60	0	13	—	—	125	65	25	—	—
$PI^{\lambda}D^{\mu}$	36	0	5	0.85	0.9	106	70	25	0.8	0.6

The mathematical definition of a derivative of fractional order  $\alpha$  has been the subject of several different approaches such as the Laplace: the Grünwald-Letnikov definition is perhaps the best suited for designing directly discrete time algorithms:

$$D^{\alpha}[x(t)] = \lim_{k \rightarrow 0} \left[ \frac{1}{h^{\alpha}} \sum_{k=1}^{\infty} \binom{\infty}{k} x(t - kh) \right],$$

$$\binom{\infty}{k} = \frac{(-1)^k \Gamma(\alpha + 1)}{\Gamma(k+1)\Gamma(\alpha - k + 1)},$$

where  $\Gamma$  is the sgamma function and  $h$  is the time increment.

For the implementation of the  $PI^{\lambda}D^{\mu}$  given by

$$G_c(s) = K \left( 1 + \frac{1}{T_i s^{\lambda}} + T_d s^{\mu} \right), \quad (3)$$

we adopt a 4th-order discrete-time Pade approximations in the  $Z$ -Domain.

To tune the controllers' parameters we used a medium-scale *Gradient Descent* method with 200 maximum iterations. To find a local minimum of a function of the position error using gradient descent, one takes steps proportional to the negative of the gradient (or the approximate gradient) of the function at the current point.

The first attempt to control our system will be changing the wing speed velocity, angle of attack, and tail rotations accordingly with the cartesian position error.

In order to study the system response to perturbations, during the experiment we apply, separately, rectangular pulses, at the references. Therefore, the trajectory used to optimize the controllers consists in a straight line flight with a velocity of  $v_x = 1$  m/s during the first 20 seconds. The dragonfly will then need to instantaneously achieve a velocity of  $v_x = 3$  m/s. Finally, 20 seconds later, the system will instantaneously reduce the velocity to  $v_x = 1$  m/s again.

In this optimization, the use of a controller in the  $y$ -axis is unnecessary since there will be no movement in this axis; therefore, we will ignore it for now.

Let us then compare the *PID* and  $PI^{\lambda}D^{\mu}$  controllers. Under the last conditions we obtained the *PID* and  $PI^{\lambda}D^{\mu}$  controller parameters depicted in Table 3.

To analyze more clearly the dynamical response to the step perturbation we subtract the dynamic response without perturbation to the step dynamic response with perturbation under the action of both *PID* and  $PI^{\lambda}D^{\mu}$  algorithms (Figure 17).

Table 2 compares the time response characteristics of the integer and the fractional *PID* controllers, namely the percent overshoot  $PO$ , the rise time  $t_r$ , the peak time  $t_p$  and

TABLE 4: Time response parameters of the system under the action of the PID and  $PI^{\lambda}D^{\mu}$  controllers.

	$PO(\%)$	$t_r$	$t_p$	$t_s$
PID	18.25	0.74	1.16	5.52
$PI^{\lambda}D^{\mu}$	13.16	0.86	1.26	5.58

the settling time  $t_s$  (there was used a 5% band in order to determine the settling time).

We can see that the FO algorithm leads to a reduction of the overshoot, at the cost of a slight increase of the algorithm.

## 6. Conclusion

The functionalities presented in this work are implemented in a simulation platform. We obtain satisfactory results proving that the development of the kinematical and dynamic model can lead to the implementation of an artificial machine with a behavior close to the dragonfly.

The design methodology and implementation can be deemed successful in this project. By obtaining a balance between physical modeling and the objective of animation, a strong advance in the system design has been achieved. Despite all simplifications, our model is still incomplete, and further research needs to be conducted to explore additional abstractions.

## References

- [1] L. Schenato, X. Deng, W. C. Wu, and S. Sastry, "Virtual insect flight simulator (VIFS): a software testbed for insect flight," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '01)*, pp. 3885–3892, Seoul, Korea, May 2001.
- [2] Z. J. Wang, "Dissecting insect flight," *Annual Review of Fluid Mechanics*, vol. 37, pp. 183–210, 2005.
- [3] M. Tamai, W. Zhijian, G. Rajagopalan, H. Hui, and H. Guowei, "Aerodynamic performance of a corrugated dragonfly airfoil compared with smooth airfoils at low reynolds numbers," in *Proceedings of the 45th AIAA Aerospace Sciences Meeting and Exhibit*, pp. 5800–5811, Reno, Nev, USA, January 2007.
- [4] A. B. Kesel, "Aerodynamic characteristics of dragonfly wing sections compared with technical aerofoils," *Journal of Experimental Biology*, vol. 203, no. 20, pp. 3125–3135, 2000.
- [5] Y. Bar-Cohen and C. Breazeal, *Biologically-Inspired Intelligent Robots*, SPIE Press Monograph Vol. PM122, SPIE Publications, 2003.
- [6] N. Vallidis, *A Hexapod Robot and Novel Training Approach for Artificial Neural Networks*, CiteSeer, 2008, 2008.
- [7] B. W. Spranklin, *Design, analysis, and fabrication of a snake-inspired robot with a rectilinear gait*, M.S. thesis, MS University of Maryland, College Park, Md, USA, 2006.
- [8] G. V. Lauder, "Aerodynamics: flight of the robofly," *Nature*, vol. 412, pp. 688–689, 2001.
- [9] A. Ellison, "Cybird," *Product Review in FlyingToys*, pp. 92–94, August 2006.
- [10] M. S. Couceiro, C. M. Figueiredo, N. M. F. Ferreira, and J. A. T. Machado, "Biological inspired flying robot," in *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE '09)*, San Diego, Calif, USA, August–September 2009.
- [11] R. Zbikowski, "Fly like a fly," *IEEE Spectrum*, vol. 42, no. 11, pp. 40–45, 2005.
- [12] M. S. Couceiro, C. M. Figueiredo, N. M. F. Ferreira, and J. A. T. Machado, "Simulation of a robotic bird," in *Proceedings of the 3rd IFAC Workshop on Fractional Differentiation and Its Applications*, Ankara, Turkey, November 2008.
- [13] F. Iida, "Goal-directed navigation of an autonomous flying robot using biologically inspired cheap vision," in *Proceedings of the 32nd International Symposium on Robotics (ISR '01)*, pp. 1404–1409, 2001.
- [14] S. Puntunan and M. Parnichkun, "Online self-tuning pre-compensation for a PID heading control of a flying robot," *International Journal of Advanced Robotic Systems*, vol. 3, no. 4, pp. 323–330, 2006.
- [15] M. S. Couceiro, C. M. Figueiredo, N. M. F. Ferreira, and J. A. T. Machado, "The dynamic modeling of a bird robot," in *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions (Robotica '09)*, Castelo Branco, Portugal, May 2009.
- [16] N. M. F. Ferreira and J. A. T. Machado, "RobLib: an educational program for analysis of robots," in *Proceedings of the 4th Portuguese Conference on Automatic Control*, pp. 406–411, Guimaraes, Portugal, October 2000.
- [17] N. M. F. Ferreira, R. Barbosa, J. A. T. Machado, and J. A. Tenreiro, "Fractional-order position/force control of mechanical manipulators," in *Proceedings of Conférence Internationale Francophone d'Automatique (CIFA '02)*, Nantes, France, July 2002.
- [18] M. S. Couceiro, N. M. F. Ferreira, and J. A. T. Machado, "Application of fractional algorithms in the control of a robotic bird," *Journal of Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 4, pp. 895–910, 2009.