Robotica (2012) volume 30, pp. 107–121. © Cambridge University Press 2011
doi:10.1017/S0263574711000312

Modeling and control of biologically inspired flying robots

Micael S. Couceiro†*, J. Miguel A. Luz‡, Carlos M. Figueiredo‡ and N. M. Fonseca Ferreira‡

†*Institute of Systems and Robotics, University of Coimbra, Pólo II, 3030-290 Coimbra, Portugal*
E-mail: micaelcouceiro@isr.uc.pt
‡*Department of Electrotechnics Engineering, Coimbra Institute of Engineering Rua Pedro Nunes, 3031-601 Coimbra, Portugal E-mails: miguel.luz@iol.pt, cfigueiredo@isec.pt, nunomig@isec.pt*

(Received in Final Form: March 16, 2011. First published online: April 27, 2011)

SUMMARY

This paper covers a wide knowledge of physical and dynamical models useful for building flying robots and a new generation of flying platform developed in the similarity of flying animals. The goal of this work is to develop a simulation environment and dynamic control using the high-level calculation tool MatLab and the modeling, simulation, and analysis of dynamic systems tool Simulink. Once created the dynamic models to study, this work involves the study and understanding of the dynamic stability criteria to be adopted and their potential use in the control of flying models.

KEYWORDS: Aerodynamics; Control; Robotics; Seagull; Dragonfly.

1. Introduction

Birds have many similar features to reptiles but they are distinguished from all the other animals because of their feathers[3] and other unique characteristics studied.[10] The seagull is one of the most well known and studied large birds. They rely on optimized glide and soar for a long time taking advantage of upward movements of air, in order to efficiently use their energy.

The study of dynamic models based on insects has been extended and shows some results that can be considered very close to the real model.[27,38] The dragonfly has been one of the models under study[31] because it is considered one of the major challenges in the field of aerodynamics. Recent studies show that dragonfly's aerodynamics is unstable, as they use it to fly in a totally different way than the steady flight of aircrafts and large birds.[17] The unsteady aerodynamics has not had proper attention due to the level of inherent complexity.

Robotics evolved from the field of automation in which there was a desire to emulate the characteristics and biologically inspired mobility. In the last few years, there were significant advances in robotics, artificial intelligence, and other fields allowing the implementation of biologically inspired robots.[6,4] With these new resources, researchers are increasingly investing in reverse engineering based on characteristics and behavior of real creatures. The evolution in research and technology resulted in machines that can recognize facial expressions, understand speech, and perform movements, such as walking, jumping, or swimming, in a very similar way as living beings.[35,29,19,12]

However, in order to take full advantage of the desired agility of a biological inspired flying robot and to achieve a wide envelope of allowed maneuvers, a control system capable of dealing with the nonlinearity inherent to the models is necessary. Such controller should be able to stabilize and follow the given trajectory in the presence of input saturation, state constraints, and disturbances (as presented in Section 4). In this paper, both integer and fractional order (*FO*) *PID* controllers are studied and implemented in both seagull and dragonfly simulated models. The controller gains are obtained using both gradient descent optimization method and biological inspired *PSO* in order to obtain minimum error when the model moves between two points within the workspace.

The paper is organized as follows. In Section 2, the mathematical modeling of the robotic models is presented that describes the implemented kinematics and dynamics. In Section 3, the architecture of the robot control is presented, and an approach to the optimization methods used to tune the controllers is given. In Section 4, the performance of the different controllers is compared. Finally, the main conclusions are outlined in Section 5.

2. Mathematical Modeling

In this section, it is analyzed the mathematical modeling of the biomechanical models 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.[14]

2.1. Kinematics

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

[*] Corresponding author. E-mail: micaelcouceiro@gmail.com

Table I. Some features from different birds.

	Weight (kg)	Wing area (m^2)	Flapping frequency (Hz)	Maximum velocity (m/s)
Common tern	117×10^{-3}	50×10^{-3}	6.3	7.8
Black-headed gull	235×10^{-3}	75×10^{-3}	4.0	9.0
Seagull	374×10^{-3}	115×10^{-3}	3.5	9.2
Royal tern	480×10^{-3}	108×10^{-3}	4.8	10.7
Herring gull	960×10^{-3}	181×10^{-3}	3.6	11.7
Great skua	1378×10^{-3}	214×10^{-3}	3.8	12.9
Great blacked-backed gull	1959×10^{-3}	272×10^{-3}	3.3	13.6
Sooty albatross	2857×10^{-3}	340×10^{-3}	2.9	14.7
Wandering albatross	8878×10^{-3}	620×10^{-3}	2.5	19.2

Table II. 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

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 fly in an unsteady state regime[5] as their wingtip speed is faster than their flight speed.

In order to choose the appropriate bird model to compare with the insect flight (i.e., quasi-steady flight versus unsteady flight), we analyzed several features from real birds (Table I). The features were extracted both experimentally and through the analysis of research, such as refs. [24, 34, 25, 18].

The most adequate one to our model was the seagull, which has the best relation between weight, wing area, maximum velocity, and flapping frequency, making it one of the most well-known examples of quasi-steady flying models.[15]

The insect model was based on a dragonfly, which flight is considered unsteady with a flapping frequency near 20 Hz.[37] Table II compares some of the most studied insect features.[32,40] Besides the tabled features, there are several unique characteristics of the dragonfly (e.g., two pair of wings, tail influence, and flying styles), which are the main reasons that the dragonfly was used as model. Those will be discussed later on this paper.

The forces and flows around a flapping wing still represent a challenge in fluid dynamics.[38]

In order to visualize the models' behavior, while in simulation, we developed 3D models in *AutoCAD* inspired in a seagull and a dragonfly.

The models are shown in Fig. 1, where each adjacent part represented with different colors corresponds to individual elements connected through joints. The used axis system is shown in Fig. 1, 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. 1. The method used to calculate the forces depending on the rotations arising from the model is based on the kinematic structures.

With the developed model, we analyzed the seagull flight movement and its behavior in different stages, such as taking

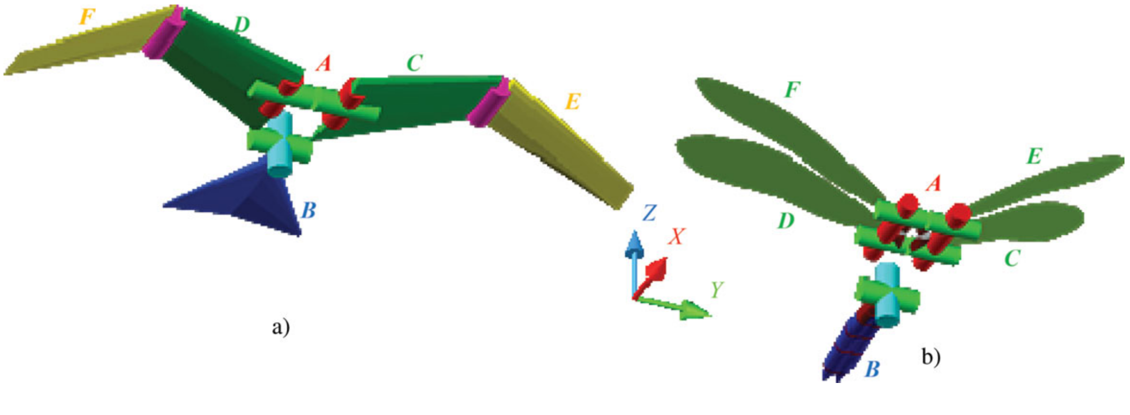

Fig. 1. (Colour online) Kinematic structure of the (a) seagull and the (b) dragonfly.

Table III. Kinematic Transformation for each link of the (a) seagull and the (b) dragonfly.

(a) Link	Kinematic transformation	(b) Link	Kinematic transformation
Body (*A*)	$T_4^0 = T_1^0 \cdot T_2^1 \cdot T_3^2 \cdot T_4^3$	Body (*A*)	$T_4^0 = T_1^0 \cdot T_2^1 \cdot T_3^2 \cdot T_4^3$
Tail (*B*)	$T_6^0 = T_4^0 \cdot T_5^4 \cdot T_6^5$	Tail (*B*)	$T_6^0 = T_4^0 \cdot T_5^4 \cdot T_6^5$
Left wing no. 1 (*C*)	$T_{10}^0 = T_4^0 \cdot T_7^4 \cdot T_9^7 \cdot T_{10}^9$	Left wing no. 1 (*C*)	$T_{10}^0 = T_4^0 \cdot T_7^4 \cdot T_9^7 \cdot T_{10}^9$
Right wing no. 1 (*D*)	$T_{11}^0 = T_4^0 \cdot T_8^4 \cdot T_9^8 \cdot T_{11}^9$	Right wing no. 1 (*D*)	$T_{11}^0 = T_4^0 \cdot T_8^4 \cdot T_9^8 \cdot T_{11}^9$
Left wing no. 2 (*E*)	$T_{12}^0 = T_{10}^0 \cdot T_{12}^{10}$	Left wing no. 2 (*E*)	$T_{15}^0 = T_4^0 \cdot T_{12}^4 \cdot T_{14}^{12} \cdot T_{15}^{14}$
Right wing no. 2 (*F*)	$T_{13}^0 = T_{11}^0 \cdot T_{13}^{11}$	Right wing no. 2 (*F*)	$T_{16}^0 = T_4^0 \cdot T_{13}^4 \cdot T_{14}^{13} \cdot T_{16}^{14}$

off and flying with twists and turns. Through these studies of flying motion, we obtained valuable initial specifications, which helped us choose the initial mechanical design.

This allowed estimating the location of every joint in the robot. When compared with a real bird, the number of joints has been reduced, but this mechanical structure, with a total of six controlled joints, still has good mobility.

In order to implement the seagull's animation in *MatLab,* the Denavit–Hartenberg (D-H) notation[9] was followed to represent frame (joint) coordinates for a kinematic chain of revolute and translational joints.

On the basis of the D-H tables and transformation matrices of both seagull and dragonfly model presented in Appendix, we obtained the following kinematic transformation for each link of the models (Table III).

The dragonfly model is being studied by some researchers due to the unique juggling maneuvers of this creature. Wang[38] developed a set of equations based on a real model of a dragonfly by watching its flight in laboratory.

On the basis of 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. 1b).

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.

Similar 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° (counter-stroking) that will be explained latter in this paper.

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.

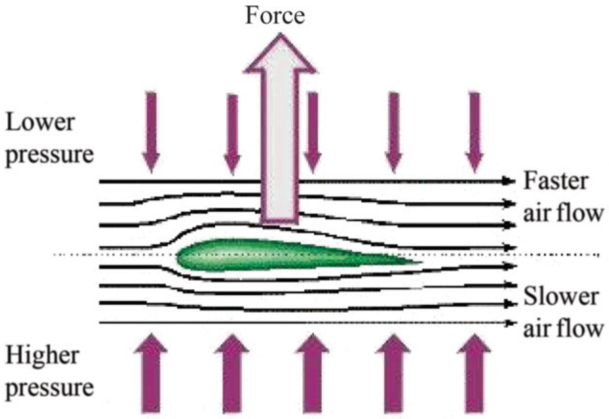

Fig. 2. (Colour online) Wing aerodynamics.

2.2. *The dynamics*

In order to establish a methodology and proper strategy to this project, it is important to study the aerodynamic principles knowing that those will be crucial to the physics behind the flight of birds and insects.

Several studies in this field have been made and are focused on the mathematical modeling of both linear and nonlinear dynamics.[7] The analysis of dynamical linear models is well established in the current literature.[13,30] However, the study of nonlinear flying models has gained increasing prominence. For most researchers, this study should be placed in the estimation of aerodynamic models and nonlinear analysis of the effects of nonlinearities in the specific aircraft systems.

Historically, the first application of nonlinear analysis of flight dynamics (i.e., a new technique in this field) was in 1977.[21] Since 1977, a large number of researchers from different countries have applied this new technique to better understand the dynamics of flight.

It is known that a mathematical model defining the dynamics of an aircraft is extremely important in both the study of dynamics and control.

2.2.1. The wings model. A bird's wing is curved along the top so that when air passes over the wing and divides, the curve forces the air on top to travel a greater distance than the air on the bottom (Fig. 2). Airflow tends to adapt itself in the presence of solid objects and to return to its original pattern as quickly as possible. Hence, when the air hits the front of the wing, the flow rate at the top increases to counterbalance the bigger distance it has to travel than the air below the wing, and as shown by Bernoulli's Principle,

https://doi.org/10.1017/S0263574711000312 Published online by Cambridge University Press

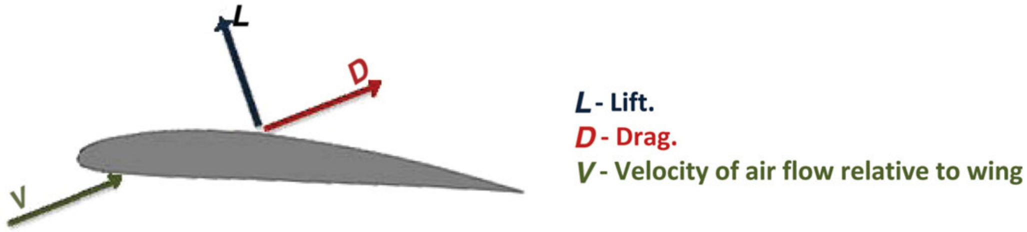

Fig. 3. (Colour online) Force acting on the wing.

fast-moving fluid exerts less pressure than slow-moving fluid; therefore, there is a difference in pressure between the air below and the air above. Whenever such a pressure difference exists in nature, a force is created in the direction of the lower pressure, in other words, downward.

Newton's third law of motion states that for every action there is an equal and opposite reaction. This is another way of saying that a force cannot be exerted without something to push against. So, if a wing deflects air downward, there must exist an equal and opposite reaction above and below the wing keeping it aloft.

The study of the airfoil could be enough if the wings of the models studied were static as is the case of aircraft. However, the structure of the wings of birds and insects is dynamic: they are not simple planar objects. The undulations in the wing change the airfoil around the wing, reducing the friction, and the wing bends around several axes, responding to both the actions of muscles and effects of inertia.

For some insects, including dragonfly, the *pterostigma* bends the wing during the beat improving the aerodynamic efficiency.

2.2.2. The gliding flight. 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 and Drag (Fig. 3).

The Lift L (1) is the component of aerodynamic force perpendicular to the relative wind and the Drag D (2) is the component of aerodynamic force parallel to the relative wind. Those components can be expressed by the following formulae:

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

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

The Lift and Drag on the wing depend on the wing area S, the density of air ρ, the freestream velocity υ_∞, and the Lift and Drag coefficients C_l and C_d, respectively, expressed as functions of the angle of attack α.

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 center of pressure.

Similar to ref. [23], we adopted the blade-element theory representing the Lift (3) and Drag (4) coefficients as functions of the angle of attack of the local wind.

$$C_l = C_{l\max} \cdot \sin(2 \cdot \alpha), \tag{3}$$

$$C_d = C_{do} + C_{d\max} \cdot \sin^2(\alpha). \tag{4}$$

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 both bird and insect model are depicted in Eqs. (5–7).

$$C_{l\max} = 2, \tag{5}$$

$$C_{do} = 0.05, \tag{6}$$

$$C_{d\max} = 1. \tag{7}$$

Some birds take advantage of the air currents to remain aloft for long periods without flapping their wings. Gliding has a lower metabolic cost than flapping flight.[2] The bird's aerodynamic characteristics determine how far and for how long it can glide, and how successfully it can soar in moving air. Those aerodynamic characteristics can be optimized by the bird in the flight by changing the wing spans and wing area. For optimal gliding, a bird's wing must maximize lift and minimize drag. As a rule, the smaller the bird, the shorter the distance it can glide and the faster it sinks. A good glider travels a long way horizontally with minimum loss of height, but eventually loses altitude due to the pull of gravity. Both large birds, such as seagulls or insects as dragonflies, use this type of methodology to reduce the energy during the flight. However, dragonflies use this to reduce the speed or perform maneuvers for a few seconds, while large birds can take advantage of this technique for much longer, even minutes. 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. The dragonfly and smaller insects end up unable to enjoy such an optimized gliding as large birds, such as seagulls, do.

2.2.3. The flapping flight. The aerodynamics involving flapping wings differs in many ways from conventional

https://doi.org/10.1017/S0263574711000312 Published online by Cambridge University Press

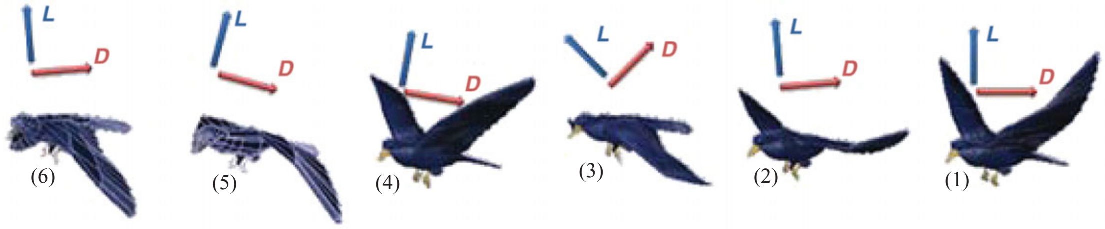

Fig. 4. (Colour online) Flapping flight sequence. Downstroke: 1-2-3; upstroke: 4-5-6.

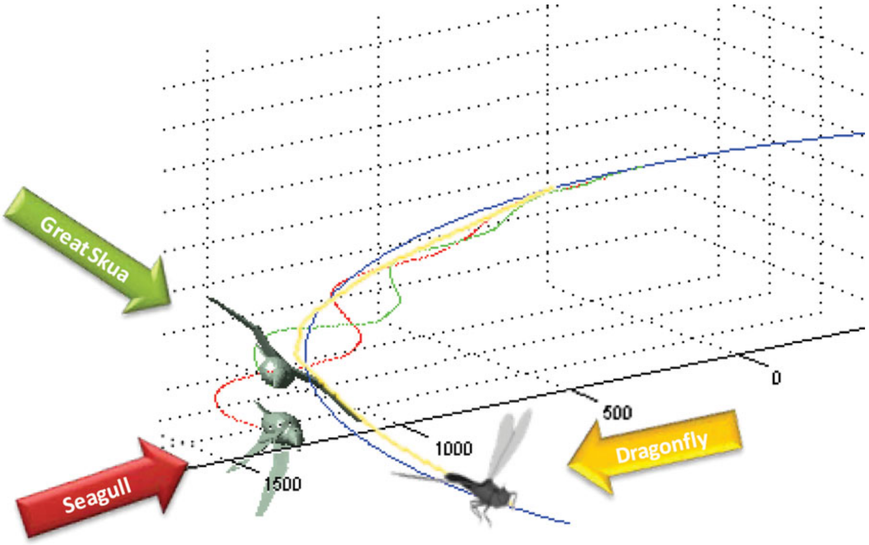

Fig. 5. (Colour online) Chart obtained through the developed simulator that shows the difference between the trajectory accomplished by a great skua (very large bird), a seagull (large bird) and a dragonfly. The stability of this last one when compared to the others is undeniable.

aerodynamics, but some conventional rules apply. A conventional airplane uses a propeller for thrust and fixed wings for lift while an ornithopter's and insect's wing must provide both of these forces.

The interaction between a flapping wing and the air is very complex since a bird must flap its wings to generate lift and thrust to overcome gravity and drag. The forces generated by this interaction are chaotic and their simulation is often unstable because of high sensitivity. The net aerodynamic force vectors acting on the wing[36] vary throughout the flapping cycle as it can be checked in the dynamical analysis (Fig. 4). On the downstroke, air is displaced in a downward and backward direction. On the upstroke, the situation is reversed being the area of the wing smaller than before in order to make a positive global thrust force. In order to minimize the wing area, birds use different techniques, such as manipulating the wings. As a simplification, we considered the area in the upstroke to be half the area in the downstroke.

As seen previously, the forces of Lift and Drag will depend on the angle of attack. However, which will be the behavior of these forces when flapping wings? As ref. [23], we considered the existence of an advance angle related with the flapping velocity and the freestream velocity (8)

$$\delta = a\ \tan\left(\frac{w_f}{v_\infty}\right). \tag{8}$$

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_∞, 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 Eq. (9):

$$\begin{cases} F_x = L \cdot \sin\delta - D \cdot \cos\delta, \\ F_z = L \cdot \cos\delta + D \cdot \sin\delta. \end{cases} \tag{9}$$

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 (Fig. 5).

The previous simulation was performed using the standard features (i.e., weight, wing area, and flapping frequency) of the great skua, the seagull and the dragonfly (Table I and II).

https://doi.org/10.1017/S0263574711000312 Published online by Cambridge University Press

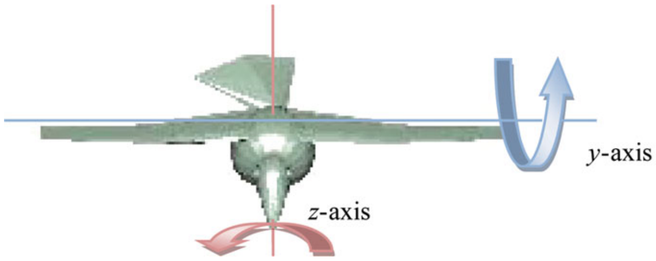

Fig. 6. (Colour online) Action of seagull tail.

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. Nevertheless, applying this principle to the development of flying platforms is complex because the effect has to be simple and predictable.

2.2.4. Tail influence. Although the majority of avian flight studies have focused on the wings, the tail also appears to be crucial to the evolutionary success of birds as flying organisms.

The precise use of the tail in birds has not been thoroughly documented.[39] The tail feathers are instrumental in stabilizing the flight, changing the direction of the forward movement, compensating for the lift force, and acting as a brake when the bird lands.

We are using the tail in order to cause a drag force changing the moment of the seagull and, consequently, producing a rotation around an axis equal to the rotation axis of the tail. That is, if the tail is bending up, the bird will rotate around the same joint bending up too. If the tail bends up and twists right (for example, Fig. 6), the seagull will then rotate around both the joints of the tail up and right. The angle of rotation of the tail is always relative to the movement of the bird.

The influence of the tail has relevance in the calculus of moments.[36] The bird will be able to rotate using the tail or using different angles of attack on each wing (or different velocities on each wing).

However, in the case of the dragonfly, the tail has an influence even more relevant than in birds. In addition to causing a significant drag force (as in the case of the seagull), the weight factor provides a more efficient use of the tail. Figure 7 depicts a simplified object diagram of masses (P) of the implemented dragonfly model.

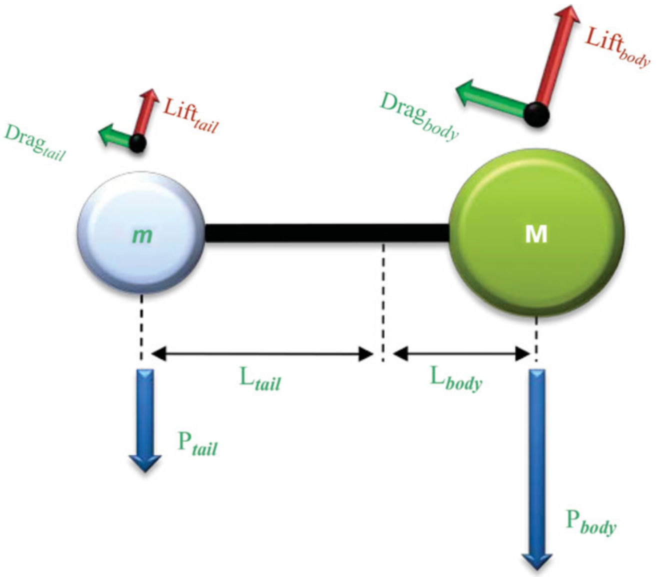

Fig. 7. (Colour online) Approximate decomposition of the body of a dragonfly in objects.

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.[11] 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 (10):

$$P_{\text{tail}} L_{\text{tail}} \cos(\phi_{\text{tail}}) = P_{\text{body}} L_{\text{body}} - (L \cdot \cos\delta + D \cdot \sin\delta). \tag{10}$$

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. The Architecture of the Robot Control

Conventional controllers, such as *PID* and many other advanced control methods, are useful to control linear processes. In practice, most processes are nonlinear. If a process is just slightly nonlinear, it can be treated as a linear process. If a process is severely nonlinear, it can be extremely difficult to control.

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

https://doi.org/10.1017/S0263574711000312 Published online by Cambridge University Press

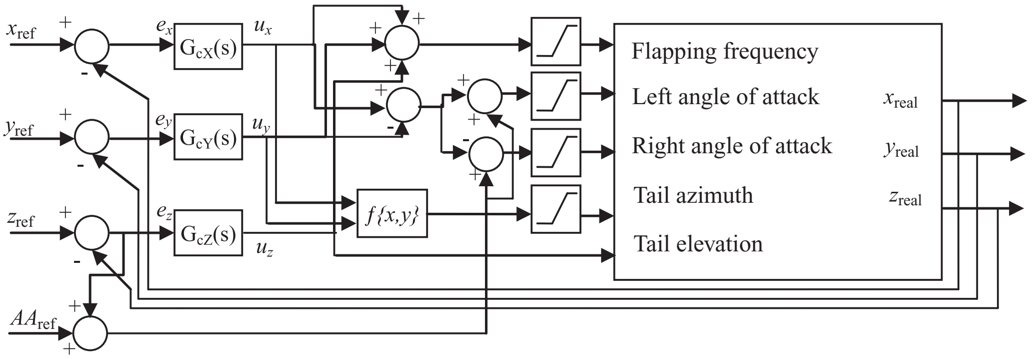

Fig. 8. Control diagram.

that present the most challenges. Nonlinear processes are difficult to control because there can be so many variations of the nonlinear behavior.

The first attempt to control our system will be changing the flapping frequency, angle of attack, and tail rotations accordingly to the position error (Fig. 8).

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 flapping frequency 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 is seen in nature, if it is a large bird (e.g., seagulls), the flapping frequency is limited between 0 Hz (gliding) and 4 Hz. For the dragonfly model, the flapping frequency is limited between 0 and 20 Hz.

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 depends on the position error in the xy-plane, i.e., 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 $f(errorX, errorY)$, 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 considered although the dynamics inherent in the use of the tail is different). The nonlinear function $f(errorX, errorY)$ will systematically adjust the angle of azimuth of the tail in order to adjust the actual position on the xy-plane. For example, if you wish 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).

In this paper, we will compare the performance of the integer and fractional *PID* controllers.

The controller parameters depend on characteristics of the process and must be tuned accordingly to yield satisfactory control. Properly tuned *PID* controllers provide adequate control for a large number of applications. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the setpoint and the degree of system oscillation. Note that the use of the *PID* algorithm for control does not guarantee optimal control of the system or system stability.

This tuning, or optimization, can occur at a number of levels. At the highest level, the design may be optimized to make best use of the available resources. The implementation of this design will benefit from the use of efficient algorithms.[28] Three of the four optimization methods used to adjust the parameters of the different controllers can be found in the toolbox of *Simulink Response Optimization*. The fourth method of optimization addressed is the well-known Particle Swarm Optimization (*PSO*).

The *PSO* was developed by Kennedy and Eberhart.[16] This optimization technique, based on a population research, is inspired by the social behavior of birds. An analogy is established between a particle and an element of a swarm. These particles fly through the search space by following the current optimum particles. At each iteration of the algorithm, a movement of a particle is characterized by two vectors representing the current position x and velocity v (Fig. 9).

The velocity of a particle is changed according to the cognitive knowledge b (the best solution found so far by the particle) and the social knowledge g (the best solution found by the swarm). The weight of the knowledge acquired in the refresh rate is different according to the random values $\phi_i, i = \{1, 2\}$. These values are a random factor that follow a uniform probability function $\phi_i \sim U[0, \phi_{i\,\max}]$.

```
Initialize Swarm
repeat
    forall particles do
        Calculate fitness $f$
    end
    forall particles do
        $v_{t+1} = I\, v_t + \phi_1(b-x) + \phi_2(g-x)$
        $x_{t+1} = x_t + v_{t+1}$
    end
until stopping criteria
```

Fig. 9. *PSO* method.

https://doi.org/10.1017/S0263574711000312 Published online by Cambridge University Press

where *I* and *t* are the inertia and the time of iteration, respectively.

The *PSO* is a very attractive technique among many other algorithms based on population since it has only some few parameters to adjust. Although recent, it has been widely studied because it is robust, easy to implement, and has a low computational cost.

The *PSO* has been successfully used in many applications, such as robotics[8,33,26] and electric systems.[1]

In order to evaluate the control architecture, we can use performance criteria, such as the integral absolute error (*IAE*) or the integral time absolute error (*ITAE*). However, in the present case, the integral square error (*ISE*) and the integral time square error (*ITSE*) criteria have produced the best results and are adopted in the study

$$ISE = \int_0^\infty [r(t) - c(t)]^2\,dt, \tag{11}$$

$$ITSE = \int_0^\infty t \cdot [r(t) - c(t)]^2\,dt. \tag{12}$$

3.1. Integer PID controller

The *PID* combines the advantages of *PI* and *PD* controller. The integral action is related with the precision of the system being responsible for the permanent regime error. The unstable effect of *PI* controller is counterbalanced by the derivative action that tends to increase the relative stability of the system at the same time that makes the system response faster due to its anticipatory effect.

The *PID* action is given by

$$u(t) = Ke(t) + \frac{K}{T_i}\int e(t)\,dt + KT_d\frac{de(t)}{dt}, \tag{13}$$

whose Laplace is

$$G_c(s) = \frac{U(s)}{E(s)} = K\left(1 + \frac{1}{T_i s} + T_d s\right), \tag{14}$$

where parameters K, T_i, and T_d represent, respectively, the proportional gain, the integral time, and the derivative time.

The control signal *u*(*t*) will be applied to the system, implying a new value for the exit signal being immediately compared with the reference signal causing a new error signal *e*(*t*). The controller processes this new error signal generating a new control signal modifying the exit signal.

3.2. Fractional PID controller

The fractional order (*FO*) controllers are controllers whose dynamic behavior is described with differential equations whose order is not an integer number. In other words, the *FO PID* controllers have five parameters, and the derivative and integral orders improve design flexibility.

The mathematical definition of a derivative of fractional order α has been the subject of several different approaches. For example, we can mention the Laplace and the Grünwald–Letnikov definitions

$$D^\alpha[x(t)] = L^{-1}\{s^\alpha X(s)\}, \tag{15}$$

$$D^\alpha[x(t)] = \lim_{k\to 0}\left[\frac{1}{h^\alpha}\sum_{k=1}^{\infty}\binom{\alpha}{k}x(t-kh)\right], \tag{16a}$$

$$\binom{\alpha}{k} = \frac{(-1)^k\Gamma(\alpha+1)}{\Gamma(k+1)\Gamma(\alpha-k+1)}, \tag{16b}$$

where Γ is the gamma function and h is the time increment.

Grünwald–Letnikov definition is perhaps the best known one, being more suitable to perform discrete control.

In our case, to implement *FO* algorithms of the type

$$G_c(s) = K\left(1 + \frac{1}{T_i s^\lambda} + T_d s^\mu\right), \tag{17}$$

we adopt a fourth-order discrete-time Pade approximation ($a_i, b_i, c_i, d_i \in \Re$, $k = 4$):

$$G_p(s) \approx K_p\left(\frac{a_0 z^k + a_1 z^{k-1} + \cdots + a_k}{b_0 z^k + b_1 z^{k-1} + \cdots + b_k}\right), \tag{18}$$

where K_P is the position loop gain.

Fractional *PID*'s are also known as $PI^\lambda D^\mu$ controllers. If both λ and μ are 1, the result is a usual *PID* (henceforth called integer *PID* as opposed to a fractional *PID*). If $\lambda = 0\,(T_i = 0)$, a PD^μ controller is obtained. All these types of controllers are particular cases of the $PI^\lambda D^\mu$ controller.

It can be expected that $PI^\lambda D^\mu$ controller may enhance the systems control performance due to more tuning knobs introduced. Actually, in theory, $PI^\lambda D^\mu$ itself is an infinite dimensional linear filter due to the fractional order in differentiator or integrator. For controller tuning techniques, refer to refs. [22, 20].

4. Controller Performance

We will now perform some experiments comparing different controller architecture in order to achieve the better control algorithm to use in our model. The simulations were made using the standard features (i.e., weight, wing area, and flapping frequency) of both seagull and dragonfly (Table I and II).

To initially tune the controllers, we used a medium-scale gradient descent method with 200 maximum iterations.

In order to study the system dynamics, during the contact we apply, separately, rectangular pulses, at the references. This kind of input is commonly used to analyze the step response of a system.

The trajectory used to optimize the controllers is then a straight-line flight with a seagull velocity of $v_x = 3.0$ m/s and a dragonfly velocity of $v_x = 1.0$ m/s for 20 s. The seagull and the dragonfly will then need to instantaneously reach a velocity of $v_x = 5.0$ m/s and a velocity of $v_x = 3.0$ m/s, respectively. Finally, 20 s later, they will instantaneously reduce the velocity to the initial value.

Under the same circumstances, we obtained the integer and fractional *PID* parameters from Tables IV and V with the gradient descent optimization method.

https://doi.org/10.1017/S0263574711000312 Published online by Cambridge University Press

Table IV. *PID* and $PI^{\lambda}D^{\mu}$ controller parameters tuned with the gradient descent method for the seagull.

	K_{pX}	K_{iX}	K_{dX}	μ_X	λ_X	K_{pZ}	K_{iZ}	K_{dZ}	μ_Z	λ_Z
PID	5	0.1	7	–	–	100	80	12	–	–
$PI^{\lambda}D^{\mu}$	13	0.1	3	0.9	1	30	10	6	0.9	0.4

Table V. *PID* and $PI^{\lambda}D^{\mu}$ controller parameters tuned with the gradient descent method for the dragonfly.

	K_{pX}	K_{iX}	K_{dX}	μ_X	λ_X	K_{pZ}	K_{iZ}	K_{dZ}	μ_Z	λ_Z
PID	60	0	13	–	–	125	65	25	–	–
$PI^{\lambda}D^{\mu}$	36	0	5	0.85	0.9	106	70	25	0.8	0.6

Table VI. Time response parameters of the seagull model under the action of the integer and fractional *PID* controller tuned with the gradient descent method.

	$PO(\%)$	$t_r(s)$	$t_p(s)$	t_s(s)
PID	23.29	1.58	2.58	11.02
$PI^{\lambda}D^{\mu}$	19.04	1.28	4.26	9.00

Table VII. Time response parameters of the dragonfly model under the action of the integer and fractional *PID* controller tuned with the gradient descent method.

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

Table VIII. *PID* and $PI^{\lambda}D^{\mu}$ controller parameters tuned with the *PSO* method for the seagull.

	K_{pX}	K_{iX}	K_{dX}	μ_X	λ_X	K_{pZ}	K_{iZ}	K_{dZ}	μ_Z	λ_Z
PID	10.8	1.3	10.4	–	–	54.3	14.5	15.3	–	–
$PI^{\lambda}D^{\mu}$	20	1.1	8.1	0.86	0.9	25.6	10.2	5.6	0.9	0.6

Table IX. *PID* and $PI^{\lambda}D^{\mu}$ controller parameters tuned with the *PSO* method for the dragonfly.

	K_{pX}	K_{iX}	K_{dX}	μ_X	λ_X	K_{pZ}	K_{iZ}	K_{dZ}	μ_Z	λ_Z
PID	40	0	6.5	–	–	117	80	39	–	–
$PI^{\lambda}D^{\mu}$	30	0.1	7.5	0.82	0.87	105	65	28	0.8	0.6

where K_p, K_i, K_d are, respectively, the proportional, integral, and derivative gain of the controller while μ and λ are the integration and differentiation order of the controller, respectively.

The step response of our system in the current situation can be analyzed in Figs. 10 and 11.

Time response characteristics of the integer and the fractional *PID* controller, namely, the percentage overshoot *PO*, the rise time t_r, the peak time t_p, and the settling time t_s, are presented in Tables VI and VII.

The advantages of the *FO* algorithm are clear, having a lower percentage overshoot and a smaller settling time, presenting better time response to the step perturbation.

On the basis of the already optimized gains through the gradient descent method, we will run a *PSO* algorithm that will determine the optimal parameters of the controller in order to obtain minimum error when the model moves between two points within the workspace.

In the experiments, it is used a $\phi_{i\,\max} = 1.6$, $I = 2$ and a population of 60 particles with 100 iterations (Tables VIII and IX). These values were obtained experimentally in order to attain a solution in the same time frame that would be necessary if using the gradient descent method from the *MatLab*'s toolbox *Simulink Response Optimization*.

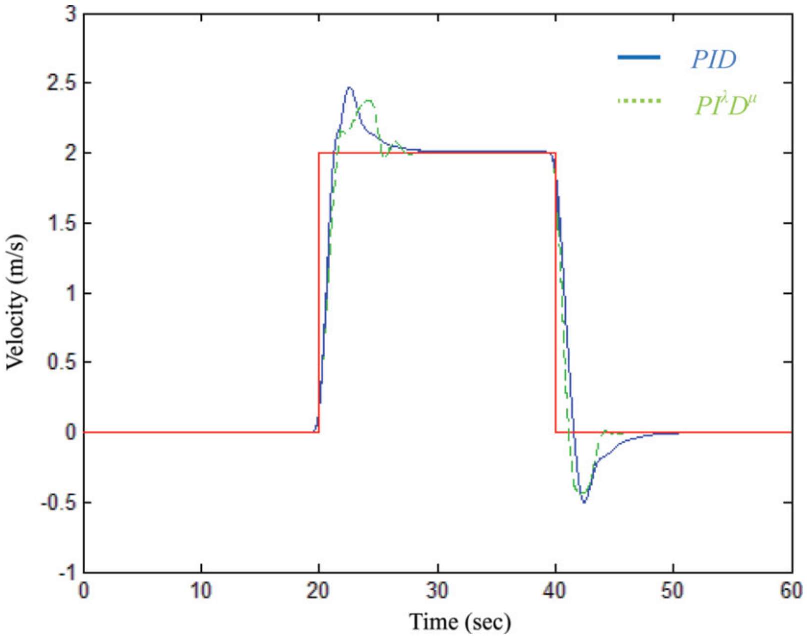

Fig. 10. (Colour online) Time response of the seagull model under the action of the integer and fractional *PID* controllers tuned with the gradient descent method.

https://doi.org/10.1017/S0263574711000312 Published online by Cambridge University Press

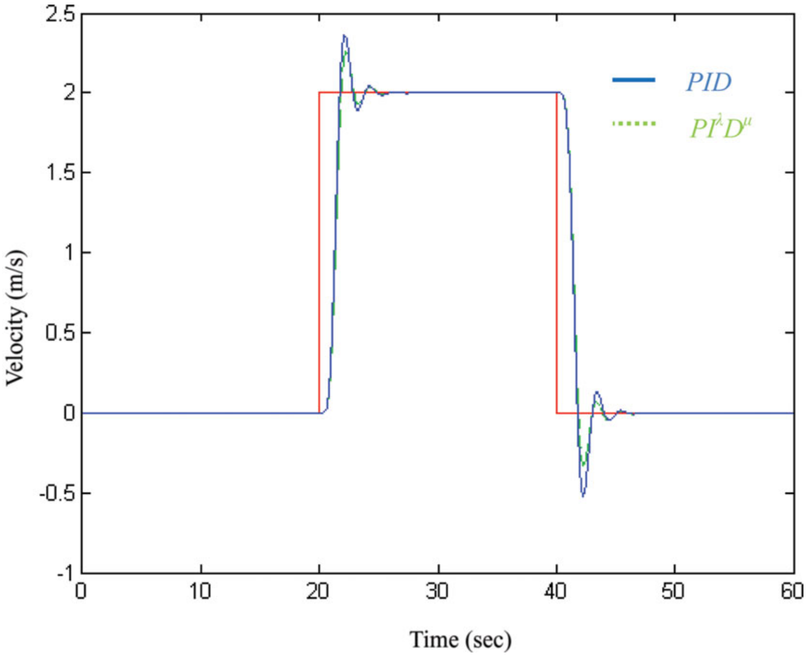

Fig. 11. (Colour online) Time response of the dragonfly model under the action of the integer and fractional *PID* controllers tuned with the gradient descent method.

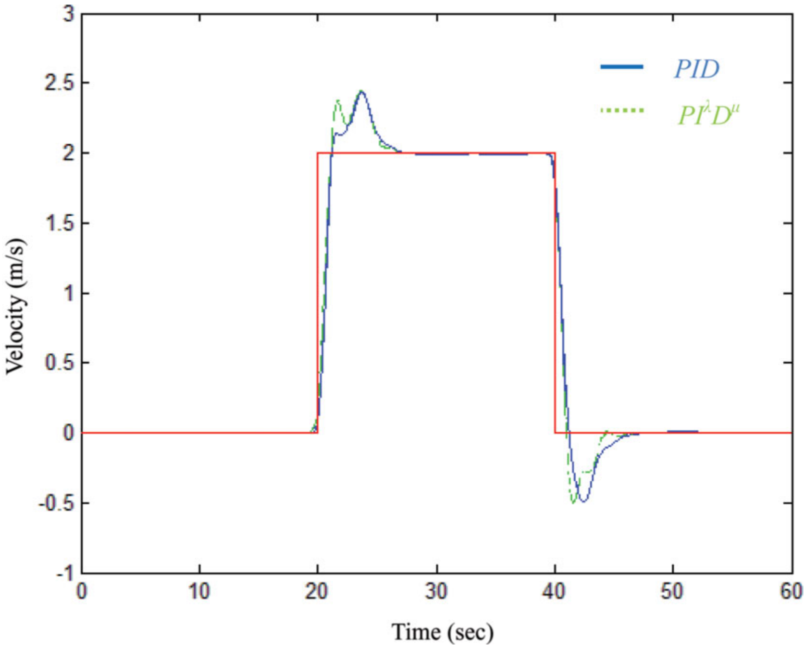

Fig. 12. (Colour online) Time response of the seagull model under the action of the integer and fractional *PID* controllers tuned with the *PSO* method.

Figure 12 and Table X show the time response of the system for the seagull model. The same can be seen for the dragonfly model in Fig. 13 and Table XI.

The optimization using the *PSO* technique shows more favorable results than the optimization using the gradient descent method. It can also be noticed that the *FO* algorithm leads to a reduction of the overshoot, at the cost of a slight improvement of the algorithm.

We now analyze two indices that measure the response error, namely, the *ISE* and the *ITSE*.

Experiments *S1* and *S2* correspond to *PID* and $PI^{\lambda}D^{\mu}$ tuned with the gradient descent optimization method, while experiments *S3* and *S4* correspond to *PID* and $PI^{\lambda}D^{\mu}$ tuned with the *PSO* method, respectively.

Table X. Time response parameters of the seagull model under the action of the integer and fractional *PID* controller tuned with the *PSO* method.

	$PO(\%)$	$t_r(s)$	$t_p(s)$	$t_s(\text{s})$
PID	22.28	1.20	3.68	7.20
$PI^{\lambda}D^{\mu}$	21.40	1.16	3.74	7.04

https://doi.org/10.1017/S0263574711000312 Published online by Cambridge University Press

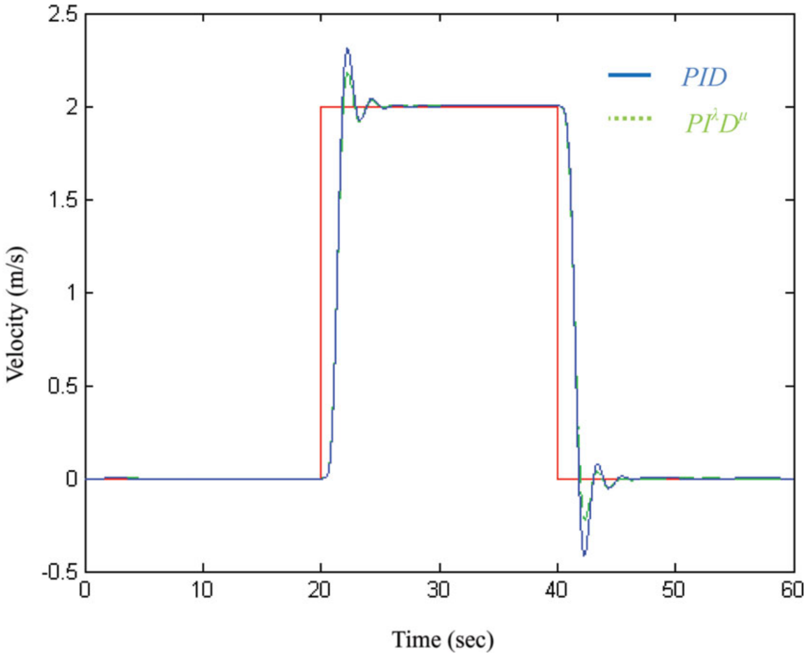

Fig. 13. (Colour online) Time response of the dragonfly model under the action of the integer and fractional *PID* controllers tuned with the *PSO* method.

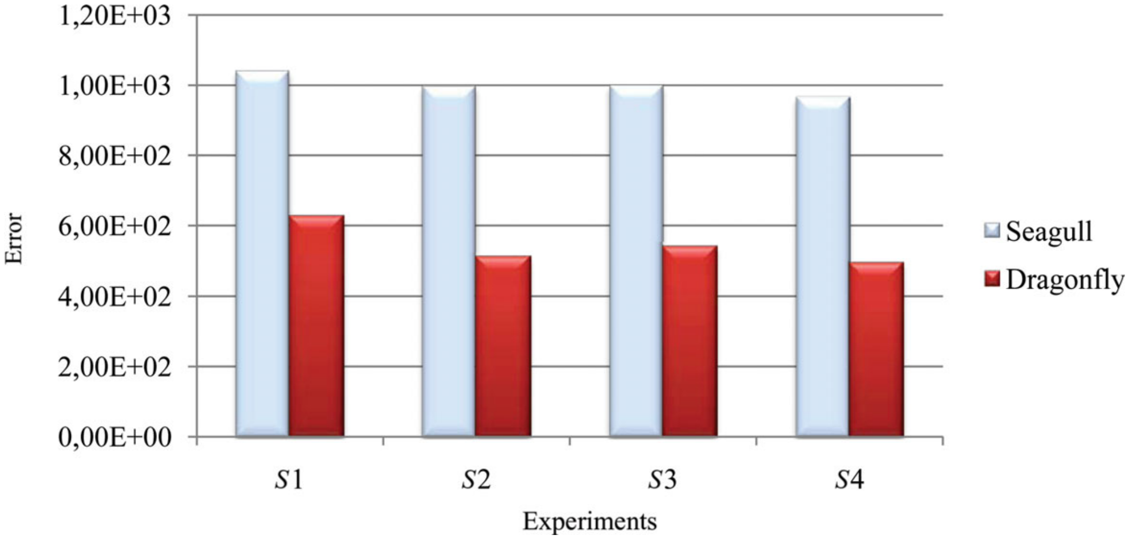

Fig. 14. (Colour online) Performance criteria of the system using the *ISE*.

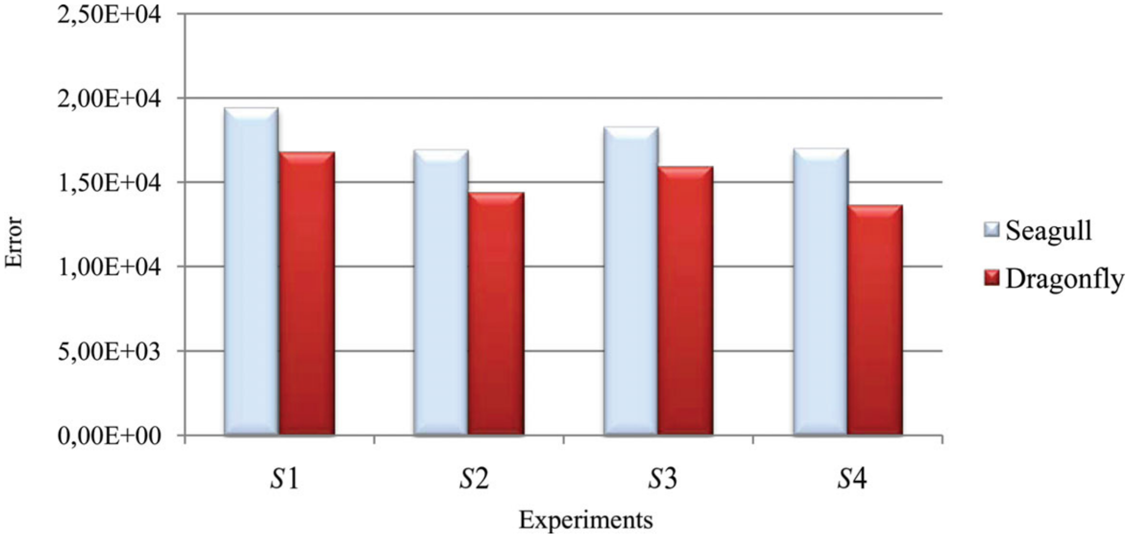

Fig. 15. (Colour online) Performance criteria of the system using the *ITSE*.

https://doi.org/10.1017/S0263574711000312 Published online by Cambridge University Press

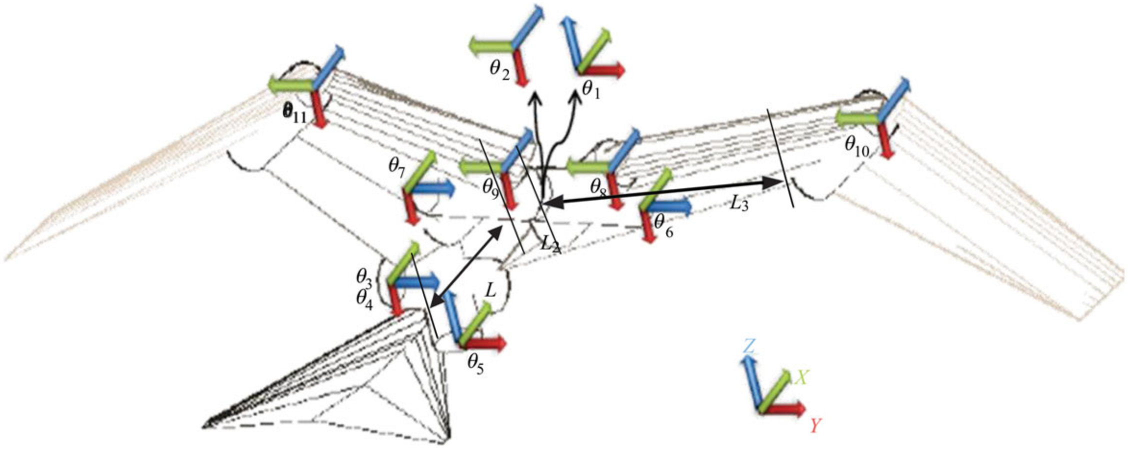

Fig. 16. (Colour online) Seagull's kinematics.

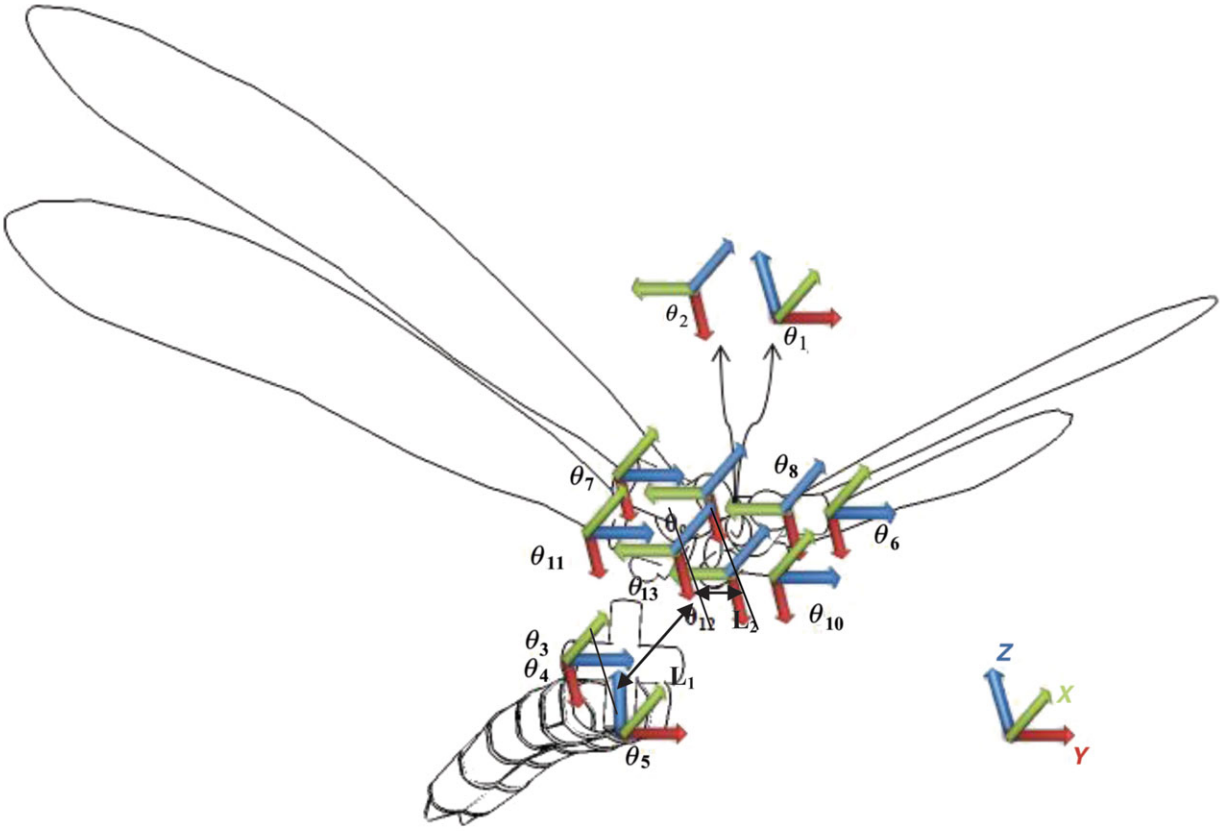

Fig. 17. (Colour online) Dragonfly's kinematics.

Table XI. Time response parameters of the dragonfly model under the action of the integer and fractional *PID* controller tuned with the *PSO* method.

	$PO(\%)$	$t_r(s)$	$t_p(s)$	t_s(s)
PID	15.49	0.82	1.24	3.96
$PI^{\lambda}D^{\mu}$	9.02	0.90	1.26	5.76

Analyzing both *ISE* and *ITSE*, we can observe that the system response under the action of the *FO* controller is more favorable than the integer controller. Another clear conclusion is a better system response for the dragonfly when compared with the seagull proving that the dragonfly can follow an imposed trajectory in a more reliable way (as it can be seen in Figs. 14 and 15, where it presents lower error value in all the performed experiments).

5. Conclusions

In this paper, we have proposed the design of an accurate simulation software and implementation for large bird and dragonfly flight that includes all major components involved: aerodynamics, kinematics, and external environment.

The obtained results appeared to be satisfactory proving that the development of the kinematical and dynamic model can show the behavior of different flying creatures. The

https://doi.org/10.1017/S0263574711000312 Published online by Cambridge University Press

Table XII. D-H model of the seagull.

	X		Z	
	a	α (degrees)	d	θ (degrees)
1	0	0	0	$\theta_1 - 90$
2	0	-90	0	θ_2
3	0	90	0	90
4	$-L_1$	-90	0	θ_3
5	0	0	0	θ_4
6	0	90	0	θ_5
7	L_1	0	L_2	θ_6
8	L_1	0	$-L_2$	$-\theta_7$
9	0	90	0	-90
10	0	90	0	θ_8
11	0	90	0	$-\theta_9$
12	$-L_3$	0	0	θ_{10}
13	L_3	0	0	$-\theta_{11}$

information concerning the physical nature of the flapping flight 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 apparent complexity inherent in the dragonfly flight proved to be more stable and controllable than the flight of the seagull. It is as if, somehow, we could compare an airplane to a helicopter: comparing with the aircraft (which has fixed wing), the helicopters are much more complex, expensive, and operate with limited speed and capacity. The advantage is that they can hover, reverse the trend, and achieve a vertical flight.

Just as we can make that comparison, we could reach a similar conclusion to that of some scientists who developed hybrid models of aircraft and helicopters in order to maintain the strengths of each model, addressing the limitations of the other.

References

1. M. R. Alrashidi and M. E. El-Hawary, "A survey of particle swarm optimization applications in power system operations," *Electr. Power Compon Syst.* **34**(12), 1349–1357 (2006).
2. R. V. Baudinette and K. Schmidt-Nielsen, "Energy cost of gliding flight in herring gulls," *Nature* **248**, 83–84 (1974).
3. M. Cianchi, *Leonardo da Vinci's Machines* (Becocci Editore, Florence, 1988) ISBN 8882000036.
4. Y. B. Cohen and C. Breazeal, *Biologically-Inspired Intelligente Robots* (SPIE Press, Bellingham, WA, USA, 2003) vol. PM122.
5. A. Colozza, "Fly like a bird," *IEEE Spectr.* **44**(5), 38–43 (2007).
6. M. S. Couceiro, C. M. Figueiredo, N. M. F. Ferreira and J. A. T. Machado, "Biological Inspired Flying Robot," *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE '09)*, San Diego (Aug. 30–Sep. 2, 2009).
7. M. S. Couceiro, C. M. Figueiredo, N. M. F. Ferreira and J. A. T. Machado, "The Dynamic Modeling of a Bird Robot," *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions - Robotica*, Castelo Branco, Portugal (May 7, 2009).
8. M. S. Couceiro, R. M. Mendes, N. M. F. Ferreira and J. A. T. Machado, "Control Optimization of a Robotic Bird," *Proceedings of the European Workshop on Movement Science (EWOMS '09)*, Lisbon, Portugal (Jun. 4–6, 2009).
9. J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *Trans. ASME J. Appl. Mech.* **23**, 215–221 (1955).
10. K. P. Dial, R. J. Randall and T. R. Dial, "BioScience - What use is half a wing in the ecology and evolution of birds," *BioScience* **56**(5), 437–445 (2006).
11. J. Duncan, *Manitoba Dragonfly Survey: Basic Biology*, NatureNorth.com (1999).
12. A. Ellison, "Cybird," In: *Product Review in FlyingToys, Manufactured by Skytech International and distributed by FlyingToys* (2006) pp. 92–94.
13. B. Etkin and L. D. Reid *Dynamics of Flight: Stability and Control*, 3rd ed. (John Wiley & Sons, New York, USA, 1995) 400 pp., ISBN: 978-0-471-03418-6.
14. J. Hamill and K. M. Knutzen, *Biomechanical Basics of Human Movement*, 3rd ed. (Lippincott, Williams and Wilkins, Baltimore, MD, USA, 2009).
15. H. Hu, A. Gopa-Kumar, G. Abate and R. Albertani, *An Experimental Study of Flexible Membrane Wings in Flapping Flight* (Aerospace Science and Technology, 2010).
16. J. Kennedy and R. C. Eberhard, "Particle Swarm Optimization," *Proceedings of the IEEE International Conference on Neural Networks* (1995) pp. 1942–1948.
17. Antonia B. Kesel, "Aerodynamic characteristics of dragonfly wing sections compared with technical aerofoils," *J. Exp. Biol.* **203**(20), 3125–3135 (2000).
18. G. P. Kulemin, *Millimeter-Wave Radar Targets and Clutter* (Artech House, Norwood, 2003) pp. 58–65.
19. G. V. Lauder, *Flight of the Robofly* (Nature, Macmillan Publishing Ltd., 2001) pp. 688–689, ISSN 0028-0836.
20. J. Liang, Y. Q. Chen and R. Fullmer, "Simulation Studies on the Boundary Stabilization and Disturbance Rejection for Fractional Diffusion-Wave Equation," *Proceedings of the 2004 American Control Conference*, Boston, Massachusetts (2004) pp. 5010–5015.
21. R. D. Mehra, W. C. Kessel and J. V. Carrol, *Global stability and control analysis of aircraft at high angles-of-attack*, Annual Technical Reports 1/2/3, ONR-CR215-(1/2/3), Scientific Systems Inc., USA (1977).
22. C. A. Monje, B. M. Vinagre, Y. Q. Chen, V. Feliu, P. Lanusse and J. Sabatier, "Proposals for Fractional $PI^{\lambda}D^{\mu}$ Tuning," *Proceedings of the First IFAC Workshop on Fractional Differentiation and Its Application Bordeaux*, Francepp (2004) pp. 156–161.
23. B. Parslew, Low Order Modelling of Flapping Wing Aerodynamics for Real-Time Model Based Animation of Flapping Flight *M.Sc. Thesis* (Manchester, UK: School of Mathematics, University of Manchester, 2005).
24. C. J. Pennycuick, "Wingbeat frequency of birds in steady cruising flight: New data and improved predictions," *J. Exp. Biol.* **199**, 1613–1618 (1996), printed in Great Britain, The Company of Biologists Limited.
25. C. J. Pennycuick, "Speeds and wingbeat frequencies of migrating birds compared with calculated benchmarks". *J. Exp. Biol.* **204**, 3283–3294 (2001), printed in Great Britain, The Company of Biologists Limited.
26. E. J. S. Pires, P. B. M. Oliveira, J. A. T. Machado and J. B. Cunha, "Particle Swarm Optimization versus Genetic Algorithm in Manipulator Trajectory Planning," *Proceedings of the 7th Portuguese Conference on Automatic Control* (Sep. 11–13, 2006).
27. L. Schenato, X. Deng, W. C. Wu and S. Sastry, "Virtual Insect Flight Simulator (VIFS): A Software Testbed for Insect Flight," *Proceedings of the IEEE International Conference Robotics and Automation*, Seoul, Korea (2001).
28. J. A. Snyman, *An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms* (Springer, New York, USA, 2005), ISBN 038729824X.
29. B. W. Spranklin, Design, Analysis, and Fabrication of a Snake-Inspired Robot with a Rectilinear Gait *M.S. Thesis* (College Park: University of Maryland, 2006) 218 pp., AAT 1436363.

30. B. L. Stevens and F. L. Lewis *Aircraft Control and Simulation*, 2nd ed. (John Wiley & Sons, New York, USA, 2003) 680 pp., ISBN: 978-0-471-37145-8.

31. M. Tamai, Z. Wang, G. Rajagopalan and H. Hu, "Aerodynamic Performance of a Corrugated Dragonfly Airfoil Compared with Smooth Airfoils at Low Reynolds Numbers," *Proceedings of the 45th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada (2007) pp. 1–12.

32. H. Tanaka, K. Hoshino, K. Matsumoto and I. Shimoyama, "Flight Dynamics of a Butterfly-Type Ornithopter," *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, Edmonton, Alberta, Canada (Aug. 2–6, 2005) pp. 310–315.

33. J. Tang, J. Zhu and Z. Sun, "A Novel Path Panning Approach Based on Appart and Particle Swarm Optimization," *Proceedings of the 2nd International Symposium on Neural Networks*, LNCS 3498 (2005) pp. 253–258.

34. J. Tong and A. Schwab, *The Flight of Birds* (MIT Press, Cambridge, MA, USA, 1997).

35. N. Vallidis, *A Hexapod Robot and Novel Training Approach for Artificial Neural Networks*, CiteSeer (2008).

36. C. Van Den Berg and J. M. V. Rayner, "The moment of inertia of bird wings and the inertial power requirement for flapping flight," *J. Exp. Biol.* **198**(8), 1655–1664 (1995).

37. J. M. Wakeling and C. P. Ellington, "Dragonfly flight II. Velocities, accelerations, and kinematics of flapping flight," *J. Exp. Biol.* **200**, 557–582 (1997).

38. Z. J. Wang, "Dissecting Insect Flight,"*Annu. Rev. Fluid Mech.* **37**, 183–210 (2005).

39. C. Zhu, K. Muraoka, T. Kawabata, C. Cao, T. Fujimoto and N. Chiba *Real-Time Animation of Bird Flight Based on Aerodynamics* (Institute of Technology, Iwate University, 2006).

40. D. Zuo, S. Peng, W. Chen and W. Zhang, "Numerical simulation of flapping-wing insect hovering flight at unsteady flow" *Int. J. Numer. Methods in Fluids* **53**, 1801–1817 (2007).

Appendix

Seagull Kinematics

The 3D animation of the seagull developed in *MatLab* was made following the D-H) notation as it is depicted in Table XII and consequently represented by the following transformation matrices (19)–(23):

$$T_1^0 = \begin{bmatrix} s_1 & c_1 & 0 & 0 \\ -c_1 & s_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_3^2 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{19}$$

$$T_4^3 = \begin{bmatrix} c_3 & -s_3 & 0 & -L_1 \\ 0 & 0 & 1 & 0 \\ -s_3 & -c_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_5^4 = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_6^5 = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{20}$$

Table XIII. D-H model of the dragonfly.

	X		Z	
	a	α (degrees)	d	θ (degrees)
1	0	0	0	$\theta_1 - 90$
2	0	-90	0	θ_2
3	0	90	0	90
4	$-L_1$	-90	0	θ_3
5	0	0	0	θ_4
6	0	90	0	θ_5
7	L_1	0	L_2	θ_6
8	L_1	0	$-L_2$	$-\theta_7$
9	0	90	0	-90
10	0	90	0	θ_8
11	0	90	0	$-\theta_9$
12	$L_1 - L_3$	0	L_2	θ_{10}
13	$L_1 - L_3$	0	$-L_2$	$-\theta_{11}$
14	0	90	0	-90
15	0	90	0	θ_{12}
16	0	90	0	$-\theta_{13}$

$$T_7^4 = \begin{bmatrix} c_6 & -s_6 & 0 & L_1 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_9^7 = T_9^8 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{10}^9 = \begin{bmatrix} c_8 & -s_8 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_8 & c_8 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{21}$$

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

$$T_{12}^{10} = \begin{bmatrix} c_{10} & -s_{10} & 0 & -L_3 \\ s_{10} & c_{10} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{22}$$

$$T_{13}^{11} = \begin{bmatrix} c_{11} & s_{11} & 0 & L_3 \\ -s_{11} & c_{11} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{23}$$

With the D-H transformation matrices, we can calculate the relationship between the links that compose the kinematic structure of the seagull (Table IIIa).

Dragonfly Kinematics

Following the same process of the seagull kinematic analysis, we obtain the following dragonfly D-H model shown in Table XIII and Eqs. (24)–(29):

$$T_1^0 = \begin{bmatrix} s_1 & c_1 & 0 & 0 \\ -c_1 & s_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_3^2 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{24}$$

https://doi.org/10.1017/S0263574711000312 Published online by Cambridge University Press

$$T_4^3 = \begin{bmatrix} c_3 & -s_3 & 0 & -L_1 \\ 0 & 0 & 1 & 0 \\ -s_3 & -c_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_5^4 \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_6^5 = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{25}$$

$$T_7^4 = \begin{bmatrix} c_6 & -s_6 & 0 & L_1 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_9^7 = T_9^8 = T_{14}^{12} = T_{14}^{13} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{26}$$

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

$$T_{11}^9 = \begin{bmatrix} c_9 & s_9 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -s_9 & c_9 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{27}$$

$$T_{12}^4 = \begin{bmatrix} c_{10} & -s_{10} & 0 & L_1 - L_3 \\ s_{10} & c_{10} & 0 & 0 \\ 0 & 0 & 1 & L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{15}^{14} = \begin{bmatrix} c_{12} & -s_{12} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_{12} & c_{12} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{28}$$

$$T_{13}^4 = \begin{bmatrix} c_{11} & s_{11} & 0 & L_1 - L_3 \\ -s_{11} & c_{11} & 0 & 0 \\ 0 & 0 & 1 & -L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{16}^{14} = \begin{bmatrix} c_{13} & s_{13} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -s_{13} & c_{13} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{29}$$

With the D-H transformation matrices, we can calculate the relationship between the links that compose the kinematic structure of the dragonfly (Table IIIb).

https://doi.org/10.1017/S0263574711000312 Published online by Cambridge University Press