

Integrative Model of *Drosophila* Flight

William B. Dickson,* Andrew D. Straw,* and Michael H. Dickinson†
California Institute of Technology, Pasadena, California 91125

DOI: 10.2514/1.29862

This paper presents a framework for simulating the flight dynamics and control strategies of the fruit fly *Drosophila melanogaster*. The framework consists of five main components: an articulated rigid-body simulation, a model of the aerodynamic forces and moments, a sensory systems model, a control model, and an environment model. In the rigid-body simulation the fly is represented by a system of three rigid bodies connected by a pair of actuated ball joints. At each instant of the simulation, the aerodynamic forces and moments acting on the wings and body of the fly are calculated using an empirically derived quasi-steady model. The pattern of wing kinematics is based on data captured from high-speed video sequences. The forces and moments produced by the wings are modulated by deforming the base wing kinematics along certain characteristic actuation modes. Models of the fly's visual and mechanosensory systems are used to generate inputs to a controller that sets the magnitude of each actuation mode, thus modulating the forces produced by the wings. This simulation framework provides a quantitative test bed for examining the possible control strategies employed by flying insects. Examples demonstrating pitch rate, velocity, altitude, and flight speed control, as well as visually guided centering in a corridor are presented.

I. Introduction

FLIGHT, like all forms of locomotion, involves a complex interaction between an animal and its environment. Although neural circuits, muscles, and wings make up the central physical plant of an animal's motor system, flight behavior does not result from a simple set of feed-forward commands. For example, most of an insect's nervous system is dedicated to the sensory information that is generated as the animal moves through its environment [1]. The insect's brain rapidly processes and fuses this rich information stream to create a motor code that can modify wing motion on a stroke-by-stroke basis [2]. Sensory feedback is essential, both for short-term stability as well as long-term guidance and navigation. What we view as behavior, such as a fly flitting across the room to land on the window, represents the output of a complex set of sensory-motor circuits that operates through the dynamics of muscles, skeleton, aerodynamics forces, and the environment. Although biologists have appreciated the central role of feedback in flight [3], conventional biological disciplines such as neurobiology or biomechanics are not endowed with the mathematical framework to deal with the feedback in a rigorous manner. Fortunately, recent progress in insect aerodynamics has fostered new engineering approaches such as the application of control theory to animal flight [4,5]. Such work will be critical in further developing an integrative view of flight biology, if for no other reason than it will provide a rigorous framework for incorporating observations from multiple disciplines within a single context, as well as permit experiments that are not possible on a real animal. Indeed, the very nature of feedback-dominated systems is that they are robust to ablation and perturbation, a fact that renders them resistant to standard reductionist methods.

In this paper we present our first attempts at constructing a 3-D dynamic model of the flight system of a flying insect, the fruit fly,

Drosophila melanogaster. Similar work has recently emerged to model artificial mechanical flappers [6,7], small, fixed wing aircraft [8], and hovercraft [9]. Flies provide a convenient scaffold for an integrated control model, because they have been subject to extensive investigations of aerodynamics, sensory processing, and motor control [10–25]. In constructing this model, we target both short-term and long-term goals. Immediately, we can ask if our understanding of fly aerodynamics, neurobiology, and biophysics is sufficient to create a simulated entity that can fly stably in a manner that captures some of the salient features of a real fly. Toward this end, we must be careful not to bias our conclusions by deliberately choosing free parameters that result in satisfactory performance but are not justified physiologically. The challenge here is that certain features of our model are well constrained by detailed observation and experiment and easily formulated (e.g., wing and body aerodynamics), whereas other components are poorly understood and may be only roughly hewed in mathematical terms (e.g., neural circuitry and wing hinge mechanics). Despite this limitation, we feel that enough is known regarding this particular organism to warrant such an attempt. The long-term goal, which is not contingent on immediate success, is to construct a model that will serve as a framework for future studies. As research on flight biology continues, black box simplifications that are currently necessary will be replaced by more rigorous “bottom-up” formulation, advancing both the predictive power and physical reality of the model. The most satisfying result would be to create a useful research tool that, in conjunction with experiments on real animals, provides new insight into the underlying biology.

Our model is based on a framework that combines a simulation of insect flight dynamics, a model of insect sensory systems, and a model of the flight environment. Using this framework, control strategies can be evaluated by allowing the outputs of the sensory systems to modulate the forces produced by the wings and the results assessed in a quantitative manner. The simulation framework consists of five main components (as shown in Fig. 1).

1) *Sensory systems model*: As it is not currently practical to attempt to model all of the sensory systems of a fruit fly, our model focuses on the two key sensory systems that are arguably most important for flight control, the visual system, and the halteres. These systems can provide information concerning the rate of change of key variables such as pitch, roll, and yaw as well as indicating the presence of approaching obstacles or distant targets. They thus make up the basis of both a control and a navigation system. Additional components, such as an olfactory system, could be readily incorporated in the future.

2) *Articulated rigid-body simulation*: A system of rigid bodies is used to model the dynamics of the fly. A two-winged insect, such as a

Presented as Paper 34 at the 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 9–12 January 2006; received 19 January 2007; revision received 11 July 2007; accepted for publication 5 November 2007. Copyright © 2007 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. Copies of this paper may be made for personal or internal use, on condition that the copier pay the \$10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/08 \$10.00 in correspondence with the CCC.

*Postdoctoral Scholar, Bioengineering, 1200 East California Boulevard, Mail-Code 138-78.

†Professor, Bioengineering, 1200 East California Boulevard, Mail-Code 138-78. Member AIAA.

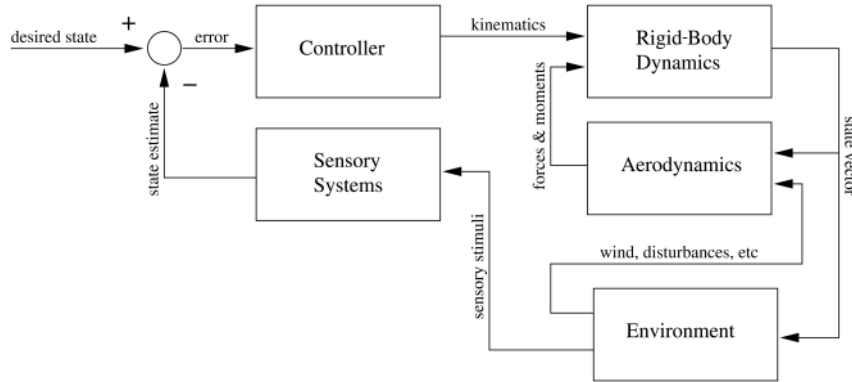


Fig. 1 Block diagram illustrating the overall architecture of the simulation framework.

fruit fly, is modeled as three separate rigid bodies representing the wings and the body. The wings are connected to the body with actuated ball joints that enable the wings to track the desired kinematics. Whereas the framework has been developed using a two-winged insect, it is easily modified to deal with four-winged animals.

3) *Aerodynamic model*: The aerodynamic model provides estimates of the aerodynamic forces and moments during flight. The force and moment model is derived from experimental measurements obtained using a dynamically scaled robotic model. A quasi-steady blade element model is used to estimate the forces and moments produced by the wings. The model of the forces and moments acting on the body of the fly was developed from force and moment coefficients measured using a dynamically scaled body model in a tow tank.

4) *Control model*: The control model consists of a set of control laws specifying the behavior of the insect with regard to sensory input. The control model provides a means by which the outputs of the sensory system modulate the wing kinematics thus providing actuation. Using this framework, different control strategies can be systematically evaluated and compared with the results of laboratory experiments to provide insight into insect flight control and behavior. Although the changes in wing motion generated by the control model are based on observations of real flies [23], the control model nevertheless represents the most coarsely rendered component of the model. No attempt was made, for example, to accurately model neural circuitry or musculoskeletal dynamics.

5) *Environment model*: The environment model provides input to both the sensory systems and the aerodynamics model. By simulating arbitrary visual surroundings with a computer graphics engine, we have the flexibility of placing the model fly in different virtual environments, including more naturalistic fieldlike habitats and simpler conditions such as those used in the laboratory for experiments on the fly visual system. We can also add physical features such as wind gusts to provide realistic or arbitrary perturbations.

II. Simulation Framework

The simulation framework consists of four separate software components encapsulating the aerodynamics, sensory, control, and environmental models. In addition, the framework provides an application programming interface (API) through which the model parameters can be accessed and the interaction among the various components can be specified. An overall goal is to provide a unified interface for rapidly developing insect flight mechanics, control, and sensory simulations. In this section we briefly describe the models underlying the main components of the simulation framework.

A. Sensory Systems

1. Visual System

The visual guidance of fly flight has been an active research area for many years. We sought to faithfully capture much of the known biology of the *Drosophila* visual system to provide input for the

guidance of our model. In the present work, the visual modeling is limited to a relatively simple implementation of motion-detection circuitry for the purposes of heading control.

Two related questions motivate much of this work, although the results and discussion presented here are insufficient to resolve such general issues. Numerous features of the fly motion-detection pathway are nonlinear and function quite differently than most engineered computer vision systems (for a review, see [16,26]). For example, rather than extracting accurate velocity estimates of image features, the fly visual system responds to local motion in a way that is sensitive to contrast and spatial pattern in addition to velocity. Is this due to fundamental constraints imposed by the available neural computational hardware, or (not mutually exclusively) is the nature of the problems faced by flies such that these properties are, in fact, advantageous from a computational or algorithmic perspective?

In this work we modeled a single wide-field integrator, inspired by the matched filter hypothesis [27], using a scheme outlined next. Although not described here, our model framework is capable of more detailed simulations such as implementation of more detailed models of lobula plate neurons. Such models are capable of quite accurately predicting neural responses to a variety of simple or more naturalistic visual stimuli [28–31].

When considering motion detection, it is important to note that the velocities of image features moving across the retina cannot be transduced into neural responses directly, but must be computed through subsequent processing of luminance signals as encoded by photoreceptors. A brief outline of the present understanding of the nature of these computations is presented here. For these purposes, it is important to compute visual estimates of local velocity in a biologically realistic manner rather than simply using direct readouts of local angular velocity made possible by use of a modeling environment and often employed in analytic analyses of visual motion (e.g., [32]).

To attain sensitivity to motion over their large receptive fields, the outputs of many elementary motion detectors (EMDs), each sensitive to input from a small angular region of space, are summed spatially on the input dendrites of the lobula plate tangential cells (LPTCs). Although the anatomical underpinnings of these EMDs are still unresolved, physiological results from numerous experiments are in close agreement with predictions of elaborations of the Hassenstein–Reichardt correlator model for motion detection [33]. The various components of this model and a brief summary of the current biological understanding are outlined here.

The inputs to each correlator consist of two channels carrying visual information from nearby regions in visual space. These inputs originate as luminance encoded by photoreceptors but are further processed by additional neural elements. Insects have rather coarse spatial resolution, and as a general rule, each facet lens provides a significantly blurred image to its photoreceptors. Such blurring reduces the potential for spatial aliasing. A detailed eye map for *Drosophila* was made by Buchner [34] (as plotted in Fig. 2 of [35]). To a first approximation, the angular separation between adjacent photoreceptors ranges from about 4.5 to 6 deg. In our model, Buchner’s data for all 699 visual elements were used to generate the

orientation of each input and its spatial blur for each simulated photoreceptor of the two compound eyes.

At each time point in the simulation, six images spanning the entire visual world from the fly's point of view are taken from the simulated environment (see Sec. II.D). The blurred photoreceptor input luminances are formed by spatial low-pass filtering of these images with a Gaussian kernel. The Gaussian filter is given by

$$G(\zeta) = \exp\left[-\frac{4 \log(2)\zeta^2}{\Delta\rho^2}\right] \quad (1)$$

where ζ is the angle from the optical axis of the photoreceptor, and $\Delta\rho$ is the acceptance angle [36]. We use $\Delta\rho = k\Delta\phi$, where $\Delta\phi$ is the angle between adjacent elements and $k = 1.1$ as found by Buchner [37] for *Drosophila*. A rapid algorithm for performing this filtering was used [38]. A Mercator projection of the intensity and color of light falling on each photoreceptor in the simulated fly is shown in Fig. 2. As seen, the photoreceptors are arranged around the unit sphere, creating a panoramic visual system similar to that of *Drosophila*.

The transduction of light energy into neural signals filters out high temporal frequencies. For each facet lens in our model, we simulated a photoreceptor with temporal dynamics given by a linear filter that closely matches the temporal dynamics of a fly photoreceptor response [39]. The impulse response of this filter is given by

$$V(t) = \exp\left[-\frac{(\log \frac{t}{t_p})^2}{2\sigma^2}\right] \quad (2)$$

where t_p is the time to peak and σ specifies the width. For the simulations presented here, $t_p = 20$ ms and $\sigma = 0.355$, values fit to light-adapted photoreceptor impulse responses measured in *Drosophila* [40].

The outputs of adjacent simulated photoreceptors were then used as inputs to a simple implementation of a Hassenstein–Reichardt correlator. The correlator model works in the following way. Consider the time-varying output of two photoreceptors $V_A(t)$ and $V_B(t)$. Such a signal, when filtered with a linear, first-order low-pass filter, will be delayed and is denoted $V'_A(t)$ or $V'_B(t)$. Thus, $V'_A(t)V_B(t)$ is the product of the temporally delayed photoreceptor output, $V'_A(t)$, with the undelayed output, $V_B(t)$. The first-order delay filter is of the form:

$$f_\tau(t) = \frac{1}{\tau} \exp\left(-\frac{t}{\tau}\right) \quad (3)$$

where the time constant τ is set to 35 ms. Although regional differences in temporal properties are known to exist across the retina of another species of fly [41], we use only a single value for our *Drosophila* model. The output of an EMD is defined to be two such mirror-symmetric subunits subtracted as

$$V_{\text{EMD}}(t) = V'_A(t)V_B(t) - V_A(t)V'_B(t) \quad (4)$$

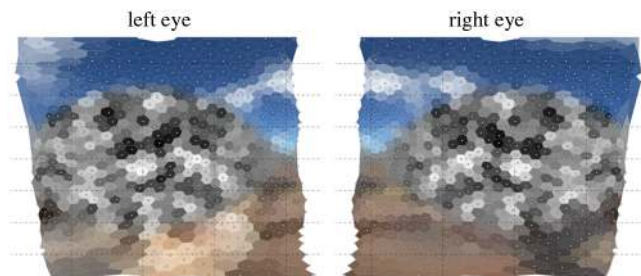


Fig. 2 A simulated fly's eye view. The intensity and color of light falling on each photoreceptor of the two compound eyes are plotted on a cylindrical (Mercator) projection. The precise direction of each visual element is denoted by a small circle.

The correlator model shares several properties with those found experimentally. In particular, output is dependent on contrast and spatial patterns of the stimulus in addition to velocity.

For each model eye (left and right) of 699 photoreceptors, there are 2059 possible EMDs between adjacent inputs. At each time step of the simulation, the activity of each EMD is computed.

2. The Halteres

In Diptera, including fruit flies, the pair of hind wings is modified into pair of dumbbell-shaped organs with a knoblike end and a stiff stalk. These organs are called halteres and play an important role in flight stabilization. It has long been known that flies are unable to fly when their halteres are removed or immobilized [42]. During flight, the halteres oscillate in antiphase with the wings in planes that are tilted back from the sagittal plane by approximately 30 deg. The stalk of the haltere is heavily innervated with approximately 400 mechanoreceptors [43–46]. A subset of these mechanoreceptors transduces information concerning the rotational velocity of the fly because they are exclusively sensitive to the lateral deflections of the haltere within its stroke plane. Lateral forces are not caused by the back and forth beating of the haltere, but are generated by Coriolis forces caused by the angular rotation of the fly [47]. An expression for the Coriolis force acting on the haltere of a fly during flight is given by [44]

$$F = -2m(\boldsymbol{\omega} \times \mathbf{v}) \quad (5)$$

where m is the mass of the haltere end knob, $\boldsymbol{\omega}$ is the angular velocity of the fly in a body frame, and \mathbf{v} is the velocity of the end knob of the haltere relative to the fly [47]. Note, a subscript L or R will be used when referring specifically to left or right halteres, respectively.

The haltere model assumes that the mass of the haltere is located in the end knob which is approximated as a point mass. The angular position of the haltere within the stroke plane is given by the periodic function $\lambda(t)$ which is antiphase with the wing stroke position. The velocities of the left and right haltere end knobs relative to the fly are given by

$$\mathbf{v}_L(t) = [\dot{\lambda}(t)d \sin \varepsilon \sin \lambda(t), \dot{\lambda}(t)d \cos \varepsilon \sin \lambda(t), \dot{\lambda}(t)d \cos \lambda(t)]^T \quad (6)$$

and

$$\mathbf{v}_R(t) = [\dot{\lambda}(t)d \sin \varepsilon \sin \lambda(t), -\dot{\lambda}(t)d \cos \varepsilon \sin \lambda(t), \dot{\lambda}(t)d \cos \lambda(t)]^T \quad (7)$$

respectively, where d is the length of the haltere and ε is the angle between the haltere stroke plane and the sagittal plane. In this description the roll, pitch, and yaw axes of the fly are given by the x (longitudinal), y (transverse), and z (dorsoventral) body axes, respectively. Using Eqs. (5–7), the lateral components of the Coriolis forces acting on the left and right halteres are then found to be given by

$$F_L(t) = -2m\dot{\lambda}(t)d[q(t) \cos \varepsilon \cos \lambda(t) - r(t) \sin \lambda(t) + p(t) \sin \varepsilon \cos \lambda(t)] \quad (8)$$

and

$$F_R(t) = -2m\dot{\lambda}(t)d[q(t) \cos \varepsilon \cos \lambda(t) + r(t) \sin \lambda(t) - p(t) \sin \varepsilon \cos \lambda(t)] \quad (9)$$

respectively, where q , r , and p are the x , y , and z components of the angular velocity vector. From Eqs. (8) and (9) it is apparent that the Coriolis force acting on the haltere is a complex waveform whose value at any given instant depends upon the velocity of the haltere, the stroke position of the haltere, and the three components of the angular velocity vector.

A region of the haltere stalk, called the basal plate, contains numerous fields of campaniform sensilla near the base of the haltere. One field, in particular (dF2), is known to be particularly sensitive to lateral deflection and is thought to act as the Coriolis detector [44,48]. In this manner the fly is able to sense the lateral component of the Coriolis forces. The neural processing of these signals is poorly understood. However, it is thought that flies can sense all three components of the angular velocity vector using both of their halteres [49,50]. To use the halteres for sensory feedback in our integrated flight control model a scheme for converting the forces sensed by the halteres into changes in wing kinematics is required. As the neural processing by which this is done in the fruit fly is in general not well understood, it is necessary for us to posit a plausible scheme instead. A first step in such a scheme is an estimate of the angular velocity from the Coriolis forces or resulting strains. Next we outline a method for extracting the three components of the angular velocity vector using the lateral components of the Coriolis force from both halteres. Although this method cannot be viewed as an accurate model of sensory processing in the fly, it does provide a method that is at least theoretically possible given the sensory information available.

Examining Eqs. (8) and (9), it is apparent that by adding the lateral components of the Coriolis forces of the left and right halteres, the roll p and yaw r components of the angular velocity cancel. Thus, we can solve for the pitch rate q as follows:

$$q(t) = -\frac{F_L(t) + F_R(t)}{4md\dot{\lambda}(t) \cos \varepsilon \cos \lambda(t)} \quad (10)$$

The denominator in Eq. (10) is equal to zero only at stroke reversals when $\dot{\lambda}(t)$ is equal to zero and at mid-half-stroke when $\cos \lambda(t)$ is equal to zero.

Subtracting the lateral components of the Coriolis forces of the left and right halteres, Eqs. (8) and (9), yields an expression containing only the roll p and yaw r rates:

$$F_L(t) - F_R(t) = 4m\dot{\lambda}(t)d[r(t) \sin \lambda(t) - p(t) \sin \varepsilon \cos \lambda(t)] \quad (11)$$

The roll and yaw rates, $p(t)$ and $r(t)$, can be estimated from Eq. (11) using a series of lagged measurements from a window of length δt . The measurement window is divided into $N + 1$ time points:

$$t_i = t - \frac{i\delta t}{N} \quad (12)$$

where $i = 0, \dots, N$. In the measurement window the roll and yaw rates are approximated by the k th order polynomials:

$$p(t) \approx \sum_{i=0}^k a_i t^i \quad (13)$$

and

$$r(t) \approx \sum_{i=0}^k b_i t^i \quad (14)$$

Inserting Eqs. (13) and (14) into Eq. (11) for $t = t_0, \dots, t_N$ yields a set of $N + 1$ linear equations with $2(k + 1)$ unknowns that can be solved for the unknowns a_i and b_i . The estimate for the roll and yaw rates at time $t = t_0$ is then given by $r(t_0)$ and $p(t_0)$, respectively. Values of $k = 1$ and $N = 20$ have been found to yield good results.

Temporal aspects of the neural processing of haltere sensory information are approximated with a filter based on observations of the haltere mediated responses of *Drosophila* [51]. The transfer function of the filter is given by

$$G(s) = \frac{1.873s + 10.08}{s^2 + 20.5s + 134.3} \quad (15)$$

which has complex poles at $-10.26 \pm 5.41i$ and is based on a fit to experimental data [51]. This filter is applied to the roll, pitch, and

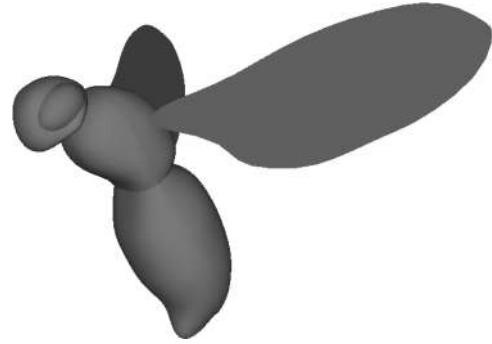


Fig. 3 Rendering of the articulated rigid-body structure used to represent a *Drosophila*.

yaw rates estimated from the lateral components of the Coriolis forces.

B. Articulated Rigid-Body Dynamics

The dynamics of the fly are provided by an articulated rigid-body simulation, in which body components are attached to each other using joints [52,53]. Several different types of joints can be used to connect the bodies. Joints differ from each other in the degrees of freedom of relative motion allowed. Examples of possible joints include ball joints, universal joints, and hinges with 3, 2, and 1 degree of freedom, respectively. In our simulation, a two-winged insect such as a fly is represented by a system of three separate rigid bodies representing the wings and the body. A rendering of the articulated rigid-body system used to represent a *Drosophila* is shown in Fig. 3. In our simulation the wings are connected to the body using ball joints that are actuated to provide relative motion between the bodies in the rigid-body system. The ball joints representing the wing hinges are actuated by virtual angular motors to allow the wings to track the desired kinematics. In real flies, wing motion is achieved through complex and not particularly well understood interactions between the neural, muscular, and skeletal systems in addition to the aerodynamics and mechanics [54]. For this reason we have chosen to use a highly simplified actuation system to achieve wing motion.

1. Physics Engine

At the heart of the articulated rigid-body simulation is a high performance physics library or physics engine. The physics engine provides a simulation of Newtonian physics for the rigid-body systems. For an introduction to the mathematical methods used in physics engines and articulated rigid-body dynamics, the reader is referred to [52,53]. Physics engines are commonly used in the computer gaming industry for developing interactive games as well as by the engineering community for studying manipulation and control strategies for robotic systems. For this reason there are many high-quality physics engines available both commercially (Havok, Meqon, Novodex) and as open source software (Open Dynamics Engine, Dynamechs).

Several criteria were evaluated when selecting a physics engine for the project including the existence usable API, good documentation, bindings to high-level languages such as Python or Ruby, an active developer community, and the availability of source code. Open Dynamics Engine (ODE) developed by Russell Smith and released under the GNU Lesser General Public License (LGPL) open source license was found to be an excellent fit to these criteria.[‡] ODE is a full-featured industrial-quality physics engine that includes many advanced joint types, integrated collision detection, and a C/C++ API. Bindings to ODE are available in both Python and Ruby, enabling rapid scripting using high-level languages. In addition to several commercial computer games, ODE has been used in both biomechanics and robotics research [55–57]. The equations of motion for rigid-body systems are derived from a Lagrange multiplier velocity-based model [58].

[‡]Data available online at <http://www.ode.org/> [retrieved 11 January 2008].

2. Mass Properties

To generate an accurate simulation of the flight dynamics of an insect, it is necessary to have an accurate estimate of the mass properties of the rigid bodies used in its representation. The mass properties of a rigid body are the total mass, center of mass, and inertia tensor. Thus for our fruit fly model, based on three rigid bodies, we need to estimate the mass, center of mass, and inertia tensor for each of the three bodies in the representation. In addition, to facilitate rapid changes in configuration, a systematic approach to estimating the mass properties of a given body from a polygonal boundary representation is desired. Our basic strategy for generating polygonal models and estimating their mass properties is outlined here.

1) Image collection: Calibrated digital images of the object are collected from sufficient vantage points to enable reconstruction of the object.

2) Boundary model: Using the collected images, a polygonal boundary representation of the object is developed with a CAD program (e.g., SolidWorks) or 3-D modeling software (e.g., Blender).

3) Mass properties: From the polygonal boundary representation of the object and an estimate of its density, the mass properties are calculated using Mirtich's algorithm [59].

The use of Mirtich's algorithm restricts this approach to objects or bodies which have uniform density. For the *Drosophila* simulation all three rigid bodies in the representation are assumed to have a uniform density equal to that of water. This produces a mass value similar to those measured in real flies. A schematic overview of the procedure outlined previously is shown in Fig. 4.

C. Aerodynamic Forces and Moments

The aerodynamic model is used to calculate the forces and moments acting on the body and wings of the insect during flight. The model is quasi-steady and is based on empirically determined force and moment coefficients. The quasi-steady assumption is based on the observation that the flow pattern and forces acting on a revolving wing display little time dependence even at angles of attack high enough to generate a leading edge vortex [21,60]. The aerodynamic forces and moments generated by the wings and body are considered separately and it is assumed that they are not influenced by wing–wing and wing–body interactions.

1. Wing Aerodynamics

The wing aerodynamics model is based on prior work using a dynamically scaled physical model [19,21,22]. In this model the wing is approximated by a flat plate with a planform based on a *Drosophila* wing. Using a standard blade element method, the wing is divided spanwise into a finite number of blade elements on which the forces and moments are calculated. The force on each blade element is calculated as the sum of the steady-state, rotational, and added mass components

$$\mathbf{F} = \mathbf{F}_s + \mathbf{F}_r + \mathbf{F}_a \quad (16)$$

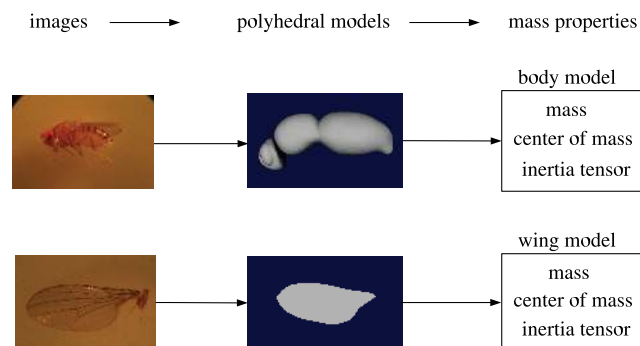


Fig. 4 Schematic of the procedure for generating polygonal boundary representations and determine the mass properties.

The total force acting on the wing is given by the sum of the blade element forces. However, to generate the appropriate moments acting on the wing, the force produced by each blade element must be applied to the rigid body representing the wing in the appropriate location. This location for a given blade element is roughly approximated by the mean spanwise location of the element and the chordwise position of the center of pressure for the element. The methods used for calculating the three force components in Eq. (16) as well as for predicting the chordwise position of the center of pressure are summarized next.

The steady-state component of the force acting on a blade element is conveniently expressed in terms of lift and drag. The steady-state lift and drag are proportional to the product of the air density ρ , the element chord length c , the element width Δr , and the square of the incident flow velocity in the plane of the blade element \mathbf{u}_e . Note, the plane of a blade element is defined as the subspace spanned by the wing normal and wing chord vectors that intersects the centerline of the element. Expressions for the magnitudes of the steady-state lift and drag are given as follows:

$$F_{s,\text{lift}} = \frac{1}{2}\rho c \|\mathbf{u}_e\|^2 C_L(\alpha) \Delta r \quad \text{and} \quad F_{s,\text{drag}} = \frac{1}{2}\rho c \|\mathbf{u}_e\|^2 C_D(\alpha) \Delta r \quad (17)$$

where α is the local angle of attack of the blade element, and the lift and drag coefficients $C_L(\alpha)$ and $C_D(\alpha)$ are shown in Fig. 5. The lift and drag coefficients were generated experimentally using a revolving dynamically scaled model *Drosophila* wing [19].

The magnitude of rotational force is proportional to the product of the fluid density, the square of the element chord length, the element width, the rate of change in angle of attack $\dot{\alpha}$, and the magnitude of the incident flow velocity in the plane of the blade element. The rotational force always acts in a direction normal to the surface of the wing and an expression for its magnitude is given by

$$F_r = C_r \rho \dot{\alpha} c^2 \|\mathbf{u}_e\| \Delta r \quad (18)$$

where C_r is the rotational force coefficient. A value of $C_r = 1.55$ is suggested by Sane and Dickinson in [22] as appropriate for the *Drosophila* wing. This value is essentially equivalent to the value derived by Fung [61] using the thin airfoil theory and assuming that the rotational axis of the wing is located at one-fourth chord.

The added mass force acting on a blade element is proportional to the density of the fluid, the square of the element chord length, the element width, and the acceleration of the incident flow normal to the surface of the element. The added mass force is assumed to always act normal to the surface of the wing. An expression for the magnitude of the added mass force is given by

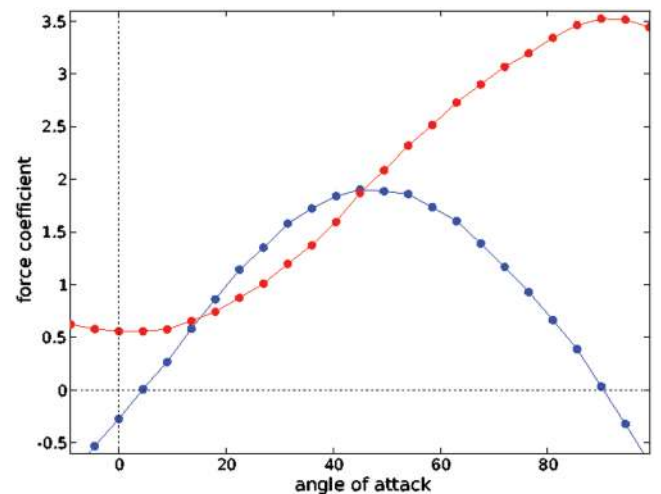


Fig. 5 Steady-state lift (blue) and drag (red) coefficients for revolving the model *Drosophila* wing [19].

$$F_a = \frac{\rho\pi c^2}{4} \left\{ \frac{\mathbf{u}_e \cdot \dot{\mathbf{u}}_e}{\|\mathbf{u}_e\|} \sin \alpha + \|\mathbf{u}_e\| \dot{\alpha} \cos \alpha \right\} \Delta r \quad (19)$$

This estimate for the added mass force acting on blade element is based on an approximation for the motions of an infinitesimally thin flat plate in an inviscid fluid, for example, see Sedov [62].

Measurements using a model *Drosophila* wing demonstrate that the spanwise location of the center of pressure is approximately constant with respect to the changes of the angle of attack. This is in good agreement with predictions by the quasi-steady model as shown in Fig. 6a.

An approximately linear relationship is found between angle of attack and the chordwise location of the center of pressure as shown in Fig. 6b. This linear relationship is well approximated by the following equation:

$$x_{cp}(\alpha) = c \left[0.82 \frac{|\alpha|}{\pi} + 0.05 \right] \quad (20)$$

To generate the appropriate moments acting on the entire wing, the force acting on each blade element is applied at the spanwise location (or center) of the element and at the center of pressure determined by $x_{cp}(\alpha)$, where α is the local angle of attack of the element.

2. Body Aerodynamics

The body aerodynamics model is used to calculate the forces and moments acting on the body of the insect during flight. During the rigid-body simulation all the separately calculated body forces and moments are applied at the center of mass of the insect's body. The model assumes that the body of the insect is bilaterally symmetric. Under this assumption, the force and moment coefficients for the body are a function of the angle of attack of the body α , the sideslip angle β , and the Reynolds number (Re). In practice, the dependence of force and moment coefficients on the Reynolds number can be generally ignored. This is due to the fact that variation of the coefficients with the Reynolds number is only significant at very low flight speeds where the forces and moments are negligibly small.

The forces parallel and normal to the body of the insect are proportional to the product of the fluid density, the body reference area S , and the square of the incident velocity \mathbf{u}_b . Expressions for the magnitudes of the parallel and normal forces are given by

$$F_P = \frac{1}{2} \rho C_P(\delta) S \|\mathbf{u}_b\|^2 \quad \text{and} \quad F_N = \frac{1}{2} \rho C_N(\delta) S \|\mathbf{u}_b\|^2 \quad (21)$$

where $C_P(\delta)$ and $C_N(\delta)$ are experimentally determined coefficients and δ is the angle between the longitudinal body axis and the incident flow. The angle δ can be expressed in terms of the angle of attack and sideslip angle of the body as follows:

$$\delta(\alpha, \beta) = \tan^{-1}(\sqrt{\tan^2 \alpha + \tan^2 \beta}) \quad (22)$$

The normal and parallel force coefficients, $C_P(\delta)$ and $C_N(\delta)$, exhibit a simple trigonometric dependence upon the angle δ as shown in Fig. 7a. This dependence can be adequately modeled using the following expressions:

$$C_P(\delta) = k_P \cos \delta \quad \text{and} \quad C_N(\delta) = k_N \sin \delta \quad (23)$$

where the constants k_P and k_N are determined using a linear regression. For *Drosophila*, values $k_P = 0.6$ and $k_N = 1.2$ are found to be suitable. Note, the results depend upon the choice reference area S , used when calculating the force coefficients from experimental data. For this study the reference area was selected to be that given by the projection onto the frontal plane of the body.

Moments generated during steady translation of the body about the roll axis were found to be negligibly small compared to the moments about the pitch and yaw axes. For this reason the rolling moment produced by the aerodynamic forces acting on the body is assumed to be zero. The pitching and yawing moments are found to depend upon the density of the fluid, the length of the body L , the area of the body S , and the incident velocity \mathbf{u}_b . Expressions for the pitching and yawing moments are given by

$$M_P = \frac{1}{2} \rho C_{pitch}(\alpha, \beta) S L \|\mathbf{u}_b\|^2 \quad \text{and} \quad M_Y = \frac{1}{2} \rho C_{yaw}(\alpha, \beta) S L \|\mathbf{u}_b\|^2 \quad (24)$$

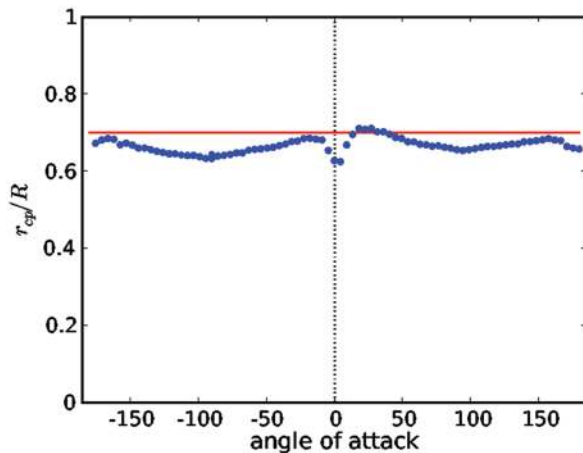
where $C_{pitch}(\alpha, \beta)$ and $C_{yaw}(\alpha, \beta)$ are experimentally determined pitching and yawing moment coefficients. As an example, the experimentally determined pitching moment for the body as a function of angle of attack is shown in Fig. 7b for a sideslip angle of zero. The dependence of pitching and yawing moments on angle of attack and sideslip angle can be approximated using the following functions:

$$C_{pitch}(\alpha, \beta) = \frac{(1 + \cos 2\beta)}{2} \left\{ f(\delta(\alpha, \beta)) \frac{(1 + \cos \beta)}{2} + f(-\delta(\alpha, \beta)) \frac{(1 - \cos \beta)}{2} \right\} \quad (25)$$

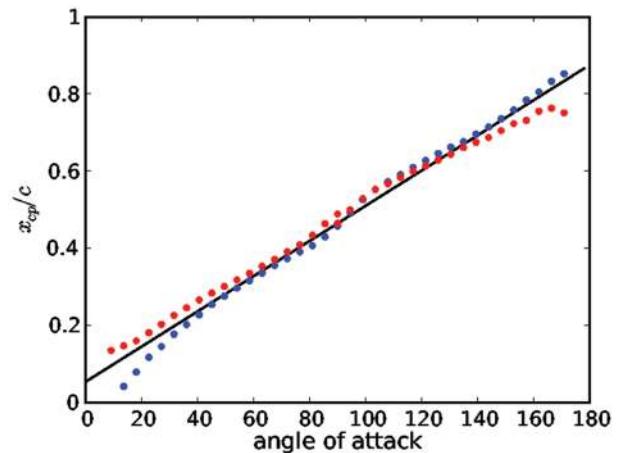
and

$$C_{yaw}(\alpha, \beta) = \frac{(1 - \cos 2\beta)}{2} \left\{ f(\delta(\alpha, \beta)) \frac{(1 + \cos \beta)}{2} + f(-\delta(\alpha, \beta)) \frac{(1 - \cos \beta)}{2} \right\} \quad (26)$$

where



a)



b)

Fig. 6 Position of the center of pressure as a function of angle of attack. a) Spanwise position of the center pressure (blue) blade element prediction (red). b) Chordwise position of the center of pressure. Positive angles of attack (blue) and negative angles of attack (red). The linear fit provided by Eq. (20) is shown for comparison (black).

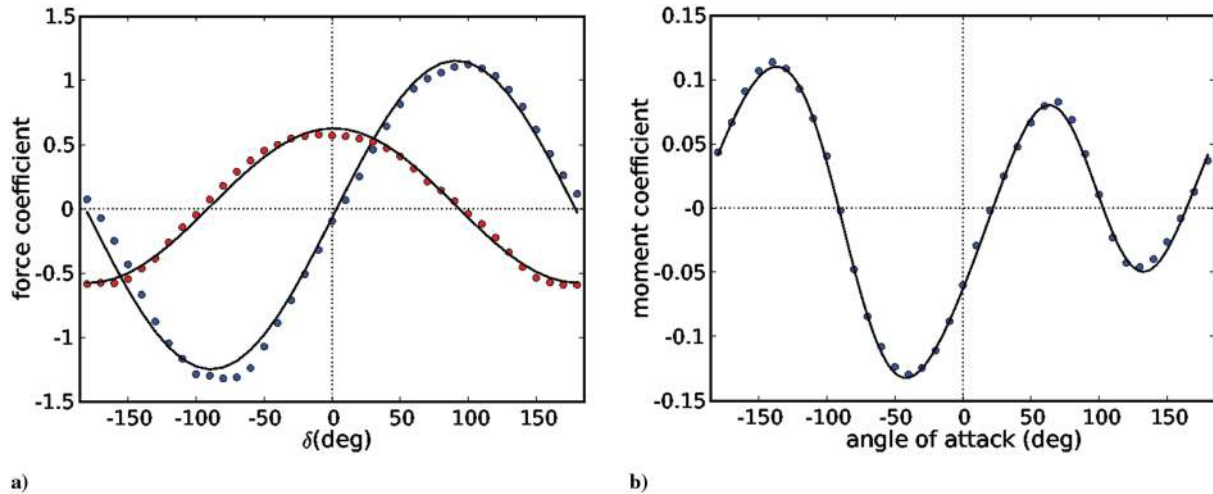


Fig. 7 Experimentally determined normal force, parallel force, and pitching moment coefficients for the fly body. a) Normal force coefficient (blue). Parallel force coefficient (red). A fit of the trigonometric functions (23) are shown for comparison (black). b) Pitching moment coefficient (blue) and a fit of Eq. (25) for comparison (black).

$$f(\delta) = \sum_{n=1}^N A_n \sin n\delta + B_n \cos n\delta \quad (27)$$

The values of the A_n and B_n in the function f are determined using a least-squares regression to the experimentally measured pitching and yawing moments for a suitable choice of N . In practice a value of $N = 4$ has been found to yield a reasonable fit.

D. Environment Model

The environment model is used to represent the environment external to the insect. It provides inputs to both the sensory models (visual, mechanosensory) and the aerodynamics models (wind, gusts). Currently, the environment model consists of two main components: a 3-D graphics engine that generates optically realistic images for processing by the visual system and a wind/gust model to provide steady and transient perturbations to the insect during flight.

1. Visual Environment + 3-D Graphics Engine

A computer model of three-dimensional terrain and other objects provides input to the simulated visual system. Environmental appearance models were constructed using scripts to generate arbitrary primitives, including textured ground planes and walls with a random checkerboard pattern. Such virtual objects are loaded into the rendering engine, which is built using OpenSceneGraph,[§] a free, open source library for displaying realistic scenes in real time for computer games and visual simulations (e.g., Fig. 8). At each sampling instant, the rendering engine produces six images such that a cube of snapshots is formed around the simulated fly's position and orientation. This "cube map" represents the color and luminance of the environment in any direction from the observer's perspective. From this information, the intensity of light falling on each simulated photoreceptor is calculated by the visual system model as described in Sec. II.A.1.

2. Wind and Gusts

The wind and gust models enable us to apply either a steady wind or various transient gusts to perturb the flight of the insect. The ability to apply steady wind effectively enables us to decouple the ground speed and the airspeed. As many of the proposed flight control strategies for insects rely on feedback from the visual system, effectively measuring ground speed [63], this decoupling is crucial in order to analyze their performance. The gust model enables the user

to apply transient perturbations to an insect as it flies. Using the gust model, the stability and robustness of various control strategies can be accessed. Gusts can be either temporal or spatial and may be applied to any combination of axes. For example, a temporal step gust is specified as follows:

$$\mathbf{V}(t) = \begin{cases} 0, & \text{if } t < t_g \\ \mathbf{V}_g, & \text{otherwise} \end{cases} \quad (28)$$

where \mathbf{V}_g is the gust velocity vector and t_g is the time at which the gust occurs. For discrete gusts, the velocity transition can be a step (as above) or a graded change using the standard 1-cosine model and specifying the gust length. Temporal gusts may also be sinusoidal, specified along a given direction with a given frequency and amplitude. Spatial gusts enable the user to model the effects of transitioning from still air into a moving airstream. As with temporal gusts a spatial gust may be abrupt or gradual.



Fig. 8 A view of the simulated 3-D environment within the OpenSceneGraph engine. The position and orientation of the fly's head are used each time step of the simulation to generate images that are optically filtered to provide the visual input to each model photoreceptor.

[§]Data available online at <http://www.openscenegraph.org> [retrieved 11 February 2008].

E. Flight Control Model

The flight control model specifies the baseline wing kinematics and a set of deformation modes that can be used to modify or deform the kinematics in order to alter the aerodynamic forces produced. In addition, the flight control model specifies a set of actions or control laws by which sensory input activates the deformation modes. The different deformation modes represent degrees of freedom with respect to actuation, whereas the control laws specify how much deformation to apply along the given modes.

1. Baseline Wing Kinematics and Deformation Modes

The angular motors controlling the motion of the wings are set up to allow essentially arbitrary wing kinematics by specifying the desired angular velocity of the wing about the wing hinge at each instant of the simulation. In this way, the simulated fly is capable of using both artificial and natural (digitized from high-speed video) wing kinematics. For each wing, the kinematics are described by three angles: the stroke position angle γ_0 , the deviation angle γ_1 , and the rotation angle γ_2 as described in [64]. The stroke position angle γ_0 is the given by the angular position of the projection of the wing axis in the stroke plane. The deviation angle γ_1 is given by the angle between the stroke plane and the wing axis, and the rotation angle γ_2 is given by the angular rotation of the wing about its axis. Normally, kinematics digitized from high-speed video sequences of free flying insects are used to create baseline wing kinematics for simulations [23]. Figure 9 shows a typical example of baseline wing kinematics used in our simulations.

The wing deformation modes enable actuation—the modification of the aerodynamic forces by a controller. Although greatly simplified, the deformation modes are meant to roughly approximate what is known with regard to how flies modify their wing kinematics to stabilize flight and maneuver. Four different deformation modes are defined: pitch, yaw, roll, and throttle. The changes in wing kinematics affected by the four deformation modes are described briefly below.

1) Pitch mode: The pitch mode deformation modulates the mean stroke position of the wings forward and backward. For example, to produce a forward pitch torque, the mean stroke positions of the wings are moved rearward. A controller that modifies the pitch torque based on sensory input sends the desired mean stroke position angle $\langle \gamma_0 \rangle$ to the pitch mode deformation mechanism. Note that the notation $\langle \cdot \rangle$ is used to denote average.

2) Yaw mode: The yaw mode deformation differentially adjusts the stroke amplitudes of the wings. For example, to produce a yaw torque to the left, the amplitude of the right wing is increased and the amplitude of the left wing decreased. A controller that modifies yaw torque sends the desired difference between left and right wing stroke amplitudes $\Delta \gamma_0$ to the yaw mode deformation mechanism.

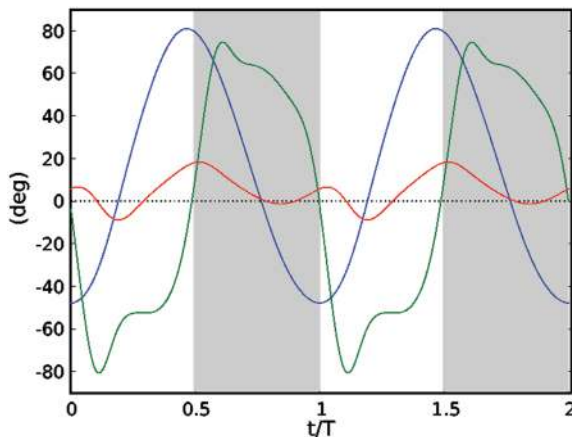


Fig. 9 Baseline wing kinematics digitized from a free flying *Drosophila*; stroke position angle γ_0 (blue), stroke deviation angle γ_1 (red), and rotation angle γ_2 (green).

3) Roll mode: The roll mode deformation differentially inclines the stroke planes of the wings. For example, to produce a roll torque to the left, the stroke plane of the right wing is inclined forward and the stroke plane of the left wing is inclined rearward. A controller that modifies the roll torque sends the desired difference in the stroke plane inclination $\Delta \eta$ to the roll mode deformation mechanism.

4) Throttle mode: The throttle mode deformation modulates the overall aerodynamic force produced by the wings by adjusting the wing beat frequency and stroke amplitude of both wings simultaneously. A controller that modifies the overall aerodynamic forces sends the desired change (from the baseline kinematics) in wing beat frequency Δf and stroke amplitude $\Delta \gamma_0$ to the throttle mode deformation mechanism.

2. Angular Velocity Control

It is known that flies control angular velocity in a compensatory manner, countersteering in response to visual and mechanical stimulation [65]. A basic control strategy developed for our framework emulates this by controlling the angular velocity of the fly via a simple set of proportional controllers acting directly on the pitch, yaw, and roll deformation modes. Suppose the desired or set-point angular velocity vector is given by

$$\omega^* = (p^*, q^*, r^*)^T \quad (29)$$

To control the pitch rate, the input to the pitch mode deformation mechanism is set to a value proportional to the difference between the set-point pitch rate q^* and the pitch rate as estimated from the halteres q as follows:

$$\langle \gamma_0 \rangle = G_q (q^* - q) \quad (30)$$

where G_q is the gain of the controller. The difference in the set point and estimated pitch rates is referred to as the pitch rate error. In a similar fashion the inputs to the yaw and roll deformation mechanisms are set to values proportional to the yaw rate and roll rate errors, respectively,

$$\Delta \gamma_0 = G_r (r^* - r) \quad \text{and} \quad \Delta \eta = G_p (p^* - p) \quad (31)$$

where G_r , and G_p are the gains of the controllers. This simple strategy produces a compensatory angular velocity control similar to that seen on real flies [65] when given a set-point angular velocity equal to zero. In addition, by setting the angular velocity set point to a nonzero or time-varying value, simple flight maneuvers can be affected. For example, Fig. 10 demonstrates tracking of a sinusoidally varying pitch rate command by this simple controller. The oscillations in pitch rate, seen in the figure, are due to the time-

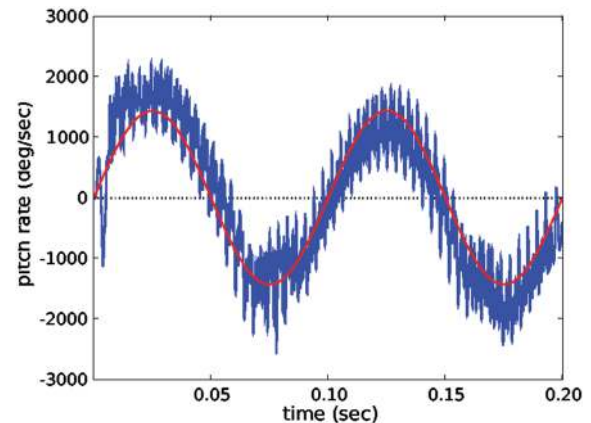


Fig. 10 Example demonstrating the tracking of a sinusoidally varying pitch rate command using angular velocity control. Actual pitch rate (blue); pitch rate command (red). High frequency oscillations are due to the periodic forces produced by the fly's wings and not instability in the controller.

varying forces and torques produced by the flapping wings rather than instability in the controller.

F. Bank, Elevation, and Azimuth Control

How flies control orientation during flight is not, at present, well understood. A possible scheme for controlling the Euler angles representing bank φ , elevation θ , and azimuth ψ uses three controllers that determine the set points for roll, pitch, and yaw rates. The roll, pitch, and yaw rates can be expressed in terms of the Euler rates as follows:

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{bmatrix} 0 & -\sin \varphi & \cos \varphi \cos \theta \\ 0 & \cos \varphi & \sin \varphi \cos \theta \\ -1 & 0 & \sin \theta \end{bmatrix} \begin{pmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\varphi} \end{pmatrix} \quad (32)$$

Note that the bank, elevation, and azimuth Euler angles described here differ from those usually defined in flight mechanics in that before applying the rotations for ψ , θ , and φ , the fly is rotated by -90 deg about the world frame y axis. This modified definition of bank, elevation, and azimuth eliminates the problems that occur with the conventional Euler angle definitions near the fly's typical flight posture while retaining the usual definitions of the roll, pitch, and yaw axes found in the insect flight literature. Note also that the usual hovering flight posture for a small fly, such as *Drosophila*, is with the longitudinal body axis (the x axis in the body frame) near vertical.

The flight controller of the simulated fly is currently designed to operate under conditions where the bank and elevation are small, whereas azimuth is expected to take on any value. When the bank and elevation angles are small, the matrix relating the roll, pitch, and yaw rates to the Euler rates in Eq. (32) simplifies so that $p \approx \dot{\psi}$, $q \approx \dot{\theta}$, and $r \approx -\dot{\varphi}$. Thus under these conditions it is feasible to control azimuth by setting the roll rate, elevation by setting pitch rate, and bank by setting yaw rate. Under more general conditions the matrix in Eq. (32) can be inverted to provide control vectors for bank, elevation, and azimuth in terms of the roll, pitch, and yaw rates. However, in what follows we consider only the simple control strategy that is applicable to flight with small bank and elevation angles. In practice this strategy has been successful for flight with elevation angles in the range of ± 60 deg and bank angles in the range of ± 20 deg. The bank, elevation, and azimuth controllers determine the roll, pitch, and yaw rate set points using simple proportional integral and derivative (PID) control strategy. For example, the roll rate set point is determined using the following equation:

$$p^* = G_{\psi,0}(\psi^* - \psi) + G_{\psi,1}(\dot{\psi}^* - \dot{\psi}) + G_{\psi,2} \int_0^t (\psi^* - \psi) ds \quad (33)$$

where ψ^* is the azimuth set point, and $G_{\psi,0}$, $G_{\psi,1}$, and $G_{\psi,2}$ are the proportional, derivative, and integral gains. The expressions for pitch rate and roll rate set points, q^* and r^* , are of the same form as Eq. (33) but with azimuth ψ replaced by elevation and bank, θ and φ , respectively.

1. Forward Velocity Control

Forward flight velocity in flies is known to be strongly correlated with the angle of the body [63]. This suggests that forward velocity may be controlled by adjusting body angle. During flight, the angle between the stroke plane of the wings and the body is relatively constant. Thus, the entire fly, wings and body together, pitches forward to increase forward velocity and pitches backward to decrease it. A possible control scheme employing this strategy, which is being tested using our simulation framework, uses this idea to control forward velocity by adjusting the elevation angle. This is achieved by adjusting the elevation angle set point using a simple PID control strategy based on the error between the set-point forward velocity v_f^* and the perceived forward velocity v_f . Again, the equation for the elevation set point θ^* is of the same form as Eq. (33) but with ψ replaced by v_f .

An example of the response of the forward velocity controller to a step input in forward velocity is shown in Fig. 11. In response to the step input, the fly, initially at rest, pitches forward and accelerates. It gains forward velocity and, as it approaches the set-point velocity, pitches back, eventually reaching a stable pitch angle. The final pitch angle depends upon the set-point velocity in a manner similar to that found in experiments with flies in a wind tunnel [63]. Using an analogous controller, the lateral velocity of the fly can also be controlled by setting the yaw rate set point using the error between the desired lateral velocity and the perceived lateral velocity. Combining these two control schemes gives a method by which the simulated fly can perform simple maneuvers.

2. Altitude Control

Although the strategies used by flies to control altitude are not well understood at this time, it is useful when performing long simulations to have control schemes on hand that can be used to maintain or control the altitude of the simulated fly. Similarly, when considering issues of flight performance it is also of value to be able to control the rate of climb or descent. For this reason we have developed vertical velocity and altitude controllers that can be used during simulations. The vertical velocity controller uses the error between the vertical velocity set point v_z^* and the estimated vertical velocity v_z to adjust the wing beat frequency of the wings using the Δf throttle deformation mode. To keep the initial implementation of the altitude controller simple, only Δf is modified by the controller. Future

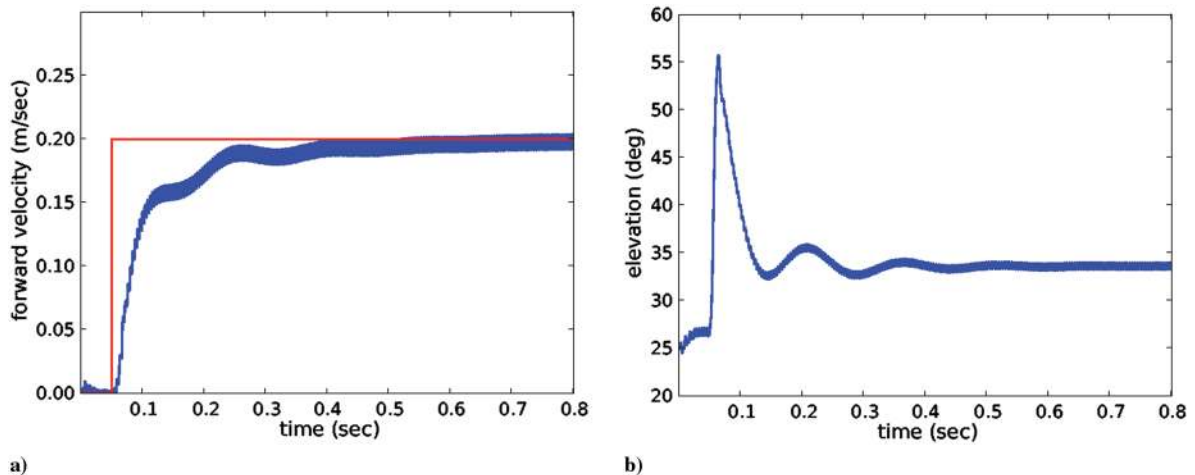


Fig. 11 Example of the response of the fly to step input when using the forward velocity controller. a) Forward velocity (blue) and step input (red). b) Elevation angle of the body (blue). High frequency oscillations are due to the periodic forces produced by the fly's wings and not instability in the controller.

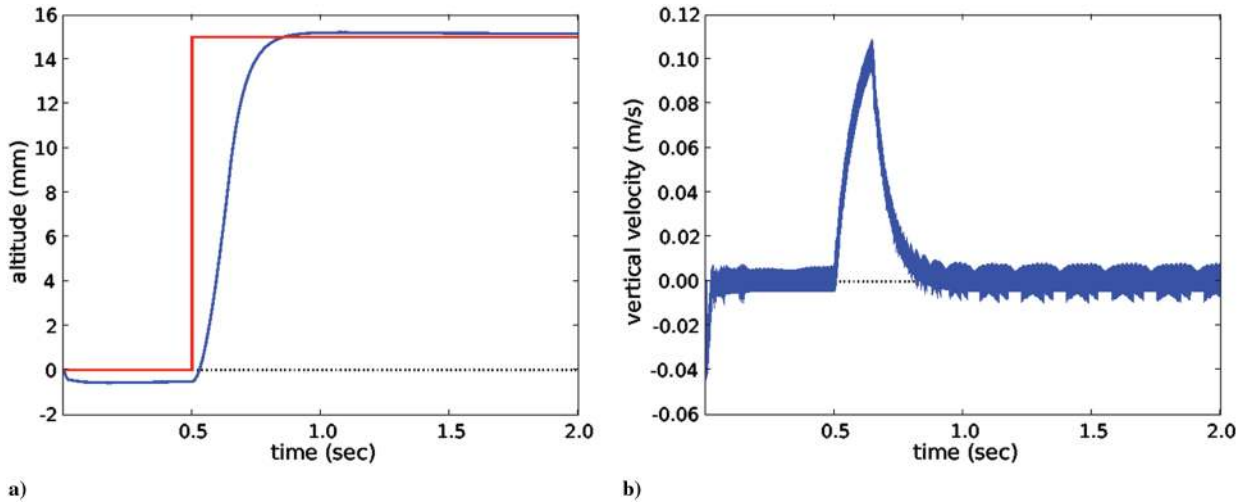


Fig. 12 Example of the response of the fly to step input when using the altitude controller. a) Altitude (blue) and step input (red). b) Vertical velocity of the fly (blue).

implementations will consider controllers that modify both Δf and $\Delta \gamma_0$ simultaneously. A PID control scheme of the same form as Eq. (33) is used to determine the magnitude of Δf . In addition, the values of Δf are clamped to keep the wing beat frequency of the fly within a physically reasonable range (150–250 Hz).

Altitude control is achieved by wrapping an outer control loop around the vertical velocity controller. In this fashion the altitude controller uses the error between the set-point altitude z^* and the estimated altitude z to adjust the vertical velocity set point. Again a PID controller of the same form as Eq. (33) is used for this purpose. An example of the response of the altitude controller to a step response is shown in Fig. 12. Note, the small steady-state error in altitude before and after the step response is due to the fact that the integral gain of the altitude controller is set to zero in the example.

G. Small Signal Frequency Analysis

The overall control system outlined thus far consists of four outer-loop controllers for forward velocity, altitude, azimuth, and bank. The remaining controllers exist as inner loops whose set points are determined by outer-loop controllers. For example, the forward velocity controller has the elevation controller as an inner loop and the elevation controller in turn has the pitch rate controller as an inner loop. Similarly, the altitude, azimuth, and bank controllers have the vertical velocity, roll rate, and yaw rate controllers as inner loops, respectively. Initially, the gains of the inner- and outer-loop controllers were tuned by examining the responses of the system to step inputs and adjusting the gains until performance appeared satisfactory. The initial performance goal was simply to achieve a stable step response with little or no overshoot.

To further assess the stability and performance of the proposed control system as well as to refine the tuning of the control gains, a small signal frequency analysis was performed for the four outer-loop controllers. Although the dynamics of the simulated fly are clearly nonlinear, a small signal frequency analysis was possible due to the fact that the outer-loop controllers, when acting in both open and closed loops, yield essentially sinusoidal outputs when presented with sufficiently small amplitude sinusoidal inputs. The open-loop tests were performed by disconnecting only the feedback to the controller under examination. For example, when performing open-loop testing on the forward velocity controller, only the forward velocity feedback signal v_f was removed. The other controllers (altitude, azimuth, and bank) operated in closed loop as normal. This was necessary to maintain flight stability and a normal flight attitude during the testing. In addition, all tests are performed with respect to a hovering flight condition.

The frequency range of the test signals for the small amplitude analysis was chosen to encompass the 0 (dB) unity gain crossing as well as the -180 deg phase crossing in order to help access the

stability of the controllers. The input amplitudes were selected to be small enough so that the outputs remained sinusoidal throughout the entire frequency range in both the open- and closed-loop configurations. The input conditions for the open- and closed-loop tests of the four outer-loop controllers are summarized in Table 1.

Plots of the open- and closed-loop frequency responses of four outer-loop controllers are shown in Figs. 13–16. For each of the four controllers the frequency at which the gain reaches 0 (dB) is lower than the frequency at which the phase becomes -180 deg. As the dynamics of the simulated fly are nonlinear, this test is not sufficient to determine whether or not the controller is truly stable. However, it does provide some confidence that for small inputs the sign of the feedback loop will not invert and cause instability. While the open-loop frequency response of the controllers indicates that the closed-loop system should be stable, it is important to recall that the response of the system may vary with flight conditions. For example, the system may respond differently at a forward velocity of 0.25 m/s than at the 0.0 m/s hovering condition at which the tests were performed. Therefore, it is desirable to insure that the controllers are robust to small changes in system parameters. Two numerical measures of how robust the controllers are to small changes in system parameters are the gain and phase margins. The gain margin is the amount of gain required to move the 0-dB crossing to the same frequency as the -180 -deg phase crossing. The phase margin is the difference between the -180 -deg phase and the actual phase at the 0-dB gain crossing. The gain margins of the controllers range from 11–20 dB and the phase margins range from 32–86 deg indicating that the controllers should be reasonably robust to small changes in the system parameters. The results of the small signal frequency analysis are summarized in Table 2.

H. Closed-Loop, Visually Based Flight Within a Corridor

A visually based flight controller for centering within a corridor geometry was developed as a preliminary demonstration of simulating more sophisticated flight behaviors. This controller was tested in a simulated corridor environment similar to that shown in Fig. 8. The basic idea behind this controller follows from the work of Humbert [32], who investigated the output of various schemes of wide-field integration (WFI) of idealized local velocity detectors. He

Table 1 Summary of the input conditions for the small amplitude frequency analysis of the forward velocity v_f , altitude z , azimuth ψ , and bank ϕ controllers

	v_f	z	ψ	ϕ
Input frequencies, Hz	0.25–45	1–100	1–115	3–57
Input amplitude	0.1, m/s	0.001, m	28 deg	7 deg

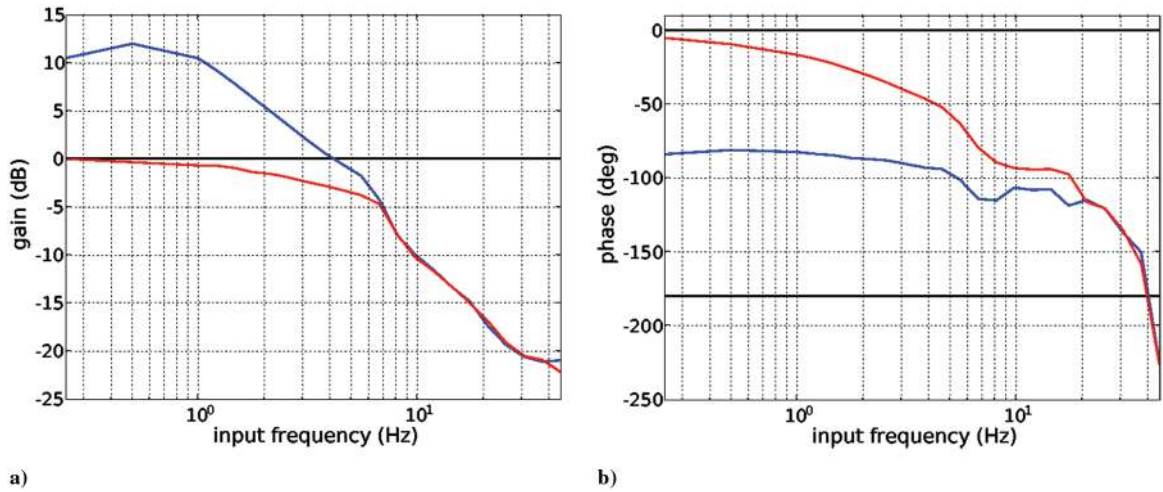


Fig. 13 Open-loop (blue) and closed-loop (red) small signal frequency responses of the forward velocity controller.

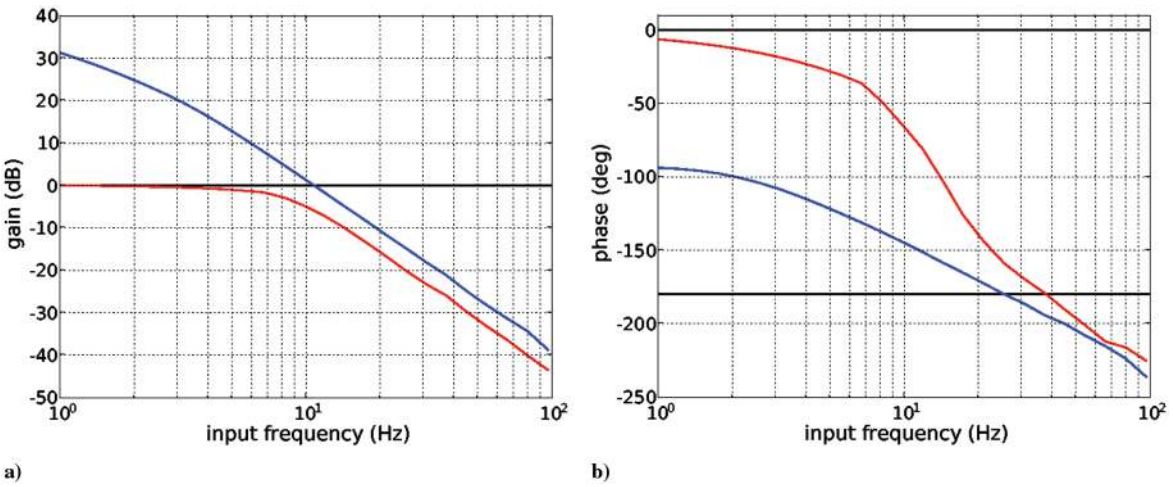


Fig. 14 Open-loop (blue) and closed-loop (red) small signal frequency responses of the altitude controller.

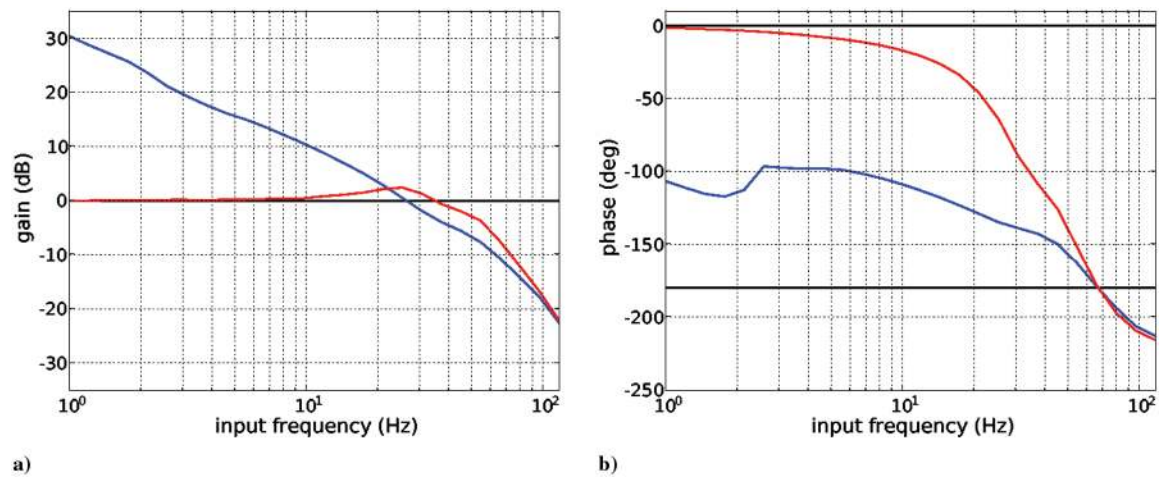


Fig. 15 Open-loop (blue) and closed-loop (red) small signal frequency responses of the azimuth controller.

showed that, when linearized about a midcorridor trajectory, a wide-field integrator that sums the magnitudes of horizontal velocities measured around the equator produces the response:

$$V_{WFInt_{ideal}}(t) = A\dot{\psi} + By \quad (34)$$

where A and B are constants (when flying in a corridor of fixed width at constant velocity), $\dot{\psi}$ is the yaw rotation rate, and y is the lateral

position of the fly with respect to the midcorridor. For negligible yaw rates and constant flight velocity, the output of this wide-field integrator is thus proportional to the lateral position of the fly with respect to midcorridor y . Under these conditions the output of the wide-field integrator can be used to produce a countersteering control strategy that turns right when the fly has deviated from the midcorridor to the left and turns left when deviated from the midcorridor to the right.

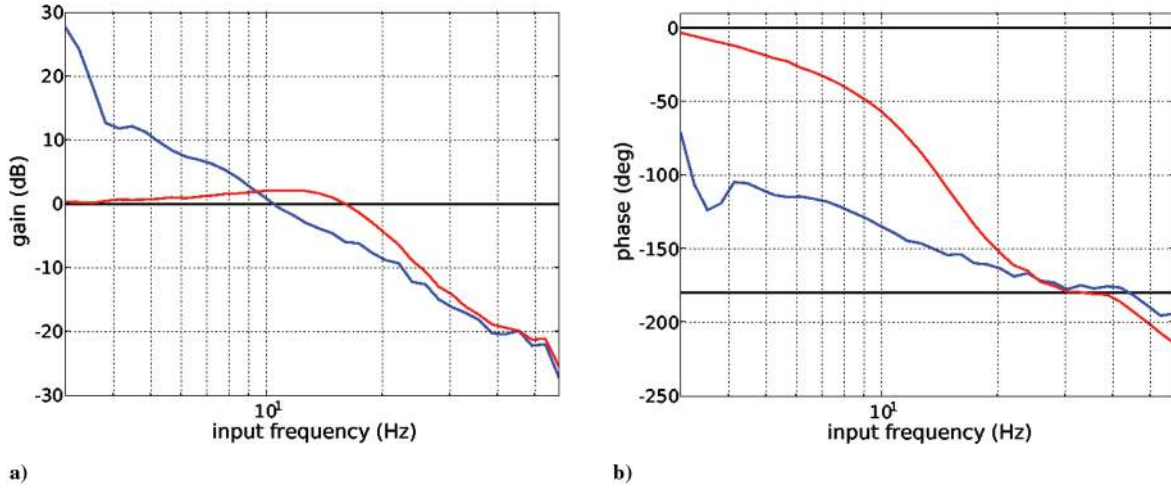


Fig. 16 Open-loop (blue) and closed-loop (red) small signal frequency responses of the bank controller.

In this work we have built upon these ideas by Humbert in two ways. First, by using estimates of visual velocity derived from simplified, biologically inspired motion detectors as described in Sec. II.A.1, we tested whether a control system based on such inputs might be successful despite the differences from perfect estimates of velocity. Second, we implemented a control strategy that creates a pattern of behavior, which under the conditions tested, remains near the assumptions under which the theory was developed.

The wide-field integrator response in our simulation is the weighted sum of numerous elementary motion detectors:

$$V_{\text{WFI}_{\text{simulation}}}(t) = \sum_{i=1}^N k_i V_{\text{EMD}_i}(t) \quad (35)$$

where k_i is the weight of each EMD contribution and $V_{\text{EMD}_i}(t)$ is the EMD response computed as described in Sec. II.A.1. The weights used are shown in Fig. 17.

As the basis for a visually based centering controller, an estimate of lateral position y , within a corridor can be found by rearranging Eq. (34) and using visually derived motion detector outputs

$$y = \frac{V_{\text{WFI}_{\text{simulation}}}(t) - A\dot{\psi}}{B} \quad (36)$$

Simulations were performed in a corridor of fixed, known width and in which forward velocity was maintained near 0.5 m/s using state feedback. The azimuthal angular velocity $\dot{\psi}$ was taken directly from the state variables of the simulation, although in principle, a haltere-derived, or perhaps a visually derived, estimate could also be used. The values of A and B were estimated from open-loop flights down the same corridor as used for the closed-loop flights. Finally, this visual estimate of y was temporally low-pass filtered to minimize variance (first-order low-pass τ , 200 ms).

In an ad hoc process, we developed and tested several control laws to regulate lateral tunnel position. In this framework, a zero set point corresponds to the centering response described above as the basis for Eq. (34). The most successful control law produced behavior shown in Fig. 18. The simulated fly navigated through the corridor

Table 2 Summary of the results of the small amplitude frequency analysis for the forward velocity v_f , altitude z , azimuth ψ , and bank ϕ controllers

	v_f	z	ψ	ϕ
0 (dB) gain crossing, Hz	4	11	27	10
-180- deg phase crossing, Hz	40	26	67	44
Gain margin, dB	20	15	11	20
Phase margin, deg	86	32	44	43
Bandwidth, Hz	6	12	34	13

without crashing into the walls from a variety of starting positions. Furthermore, the size of the checks in the random checkerboard pattern was doubled, and the trajectories flown are similar, illustrating a degree of robustness to visual perturbations. The performance was achieved with the control law:

$$\psi^* = -\frac{2k_1}{\pi} \tan^{-1}\left(\frac{y^* - y}{k_2}\right) \quad (37)$$

where ψ^* is the azimuth set point input to the lower-level controllers described above and y^* is the lateral position set point. The constant k_1 is proportional to the maximum angle formed between the fly's trajectory and the corridor direction, and the constant k_2 is proportional to the corridor width. Both k_1 and k_2 were hand-tuned to produce a controller which acts as a saturating proportional controller. As seen in Fig. 18, this has the effect that flight is broken up into periods of straight flight and periods of relatively rapid turning. Compared to other controllers we tested, this controller was most successful because it isolated periods of large yaw rate $\dot{\psi}$ to infrequent turns, allowing the wide-field integrator output to be used more reliably to estimate the lateral position.

To test the control law itself, the stability of the countersteering control strategy given by Eq. (37) was investigated using a small signal frequency analysis. We found that the inherent nonlinearities of motion detection, and thus estimates of lateral position, prevented use of a visually derived lateral position estimate. Although not tested, numerous features of biologically realistic motion detectors were not implemented in these simulations, and might serve to improve such visually derived estimates (see Sec. II.A.1). For this reason, the small signal frequency analysis was performed using state feedback for the lateral position. We found (Fig. 19), under state feedback conditions, a gain margin of 18 dB and a phase margin of

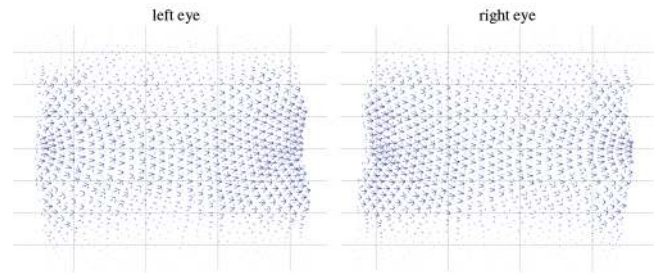


Fig. 17 The pattern of weights used by the wide-field integrator used to guide flight in a corridor under visual closed-loop control. The magnitude and sign of each elementary motion detector's contribution to the integrator's output is shown as the same cylindrical projection (as in Fig. 2), and the magnitude and direction of each weight are plotted in Cartesian coordinates.

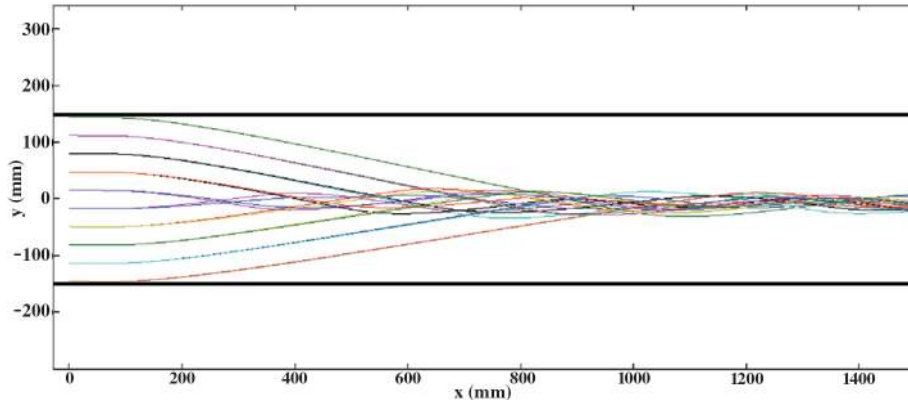
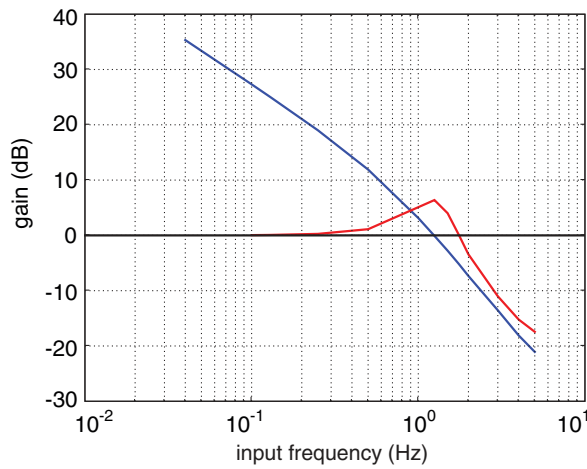
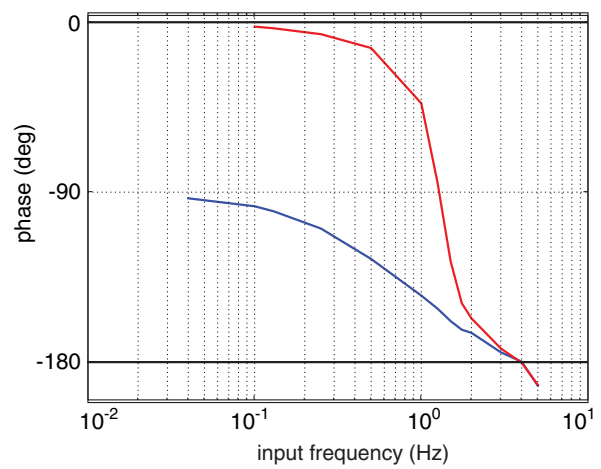


Fig. 18 Top view showing trajectories of visually guided, closed-loop simulations beginning at several positions within a corridor environment using the control law given by Eq. (37). Each starting point was tried with random checkerboard patterns of two different checker size to investigate sensitivity to visual conditions.



a)



b)

Fig. 19 Open-loop (blue) and closed-loop (red) small signal frequency responses of the lateral position controller.

30 deg. This suggests that, near midtunnel and with perfect sensory estimates, the countersteering control law is stable.

When tested with visually derived lateral position estimates, the countersteering control law, as shown above, did appear successful in centering from a large range of starting conditions and with two visual patterns. Our observations are that visually derived estimates of lateral position (as calculated above) are less accurate during periods of rapid rotation, despite correction for the azimuth rate ψ given in Eq. (36). The successful countersteering control law has an output that is frequently saturated. This output, the azimuth set point, therefore results in periods of relatively straight flight, allowing usable visual estimates of lateral position.

Other control laws tested under similar conditions resulted in the simulated fly frequently crashing into the tunnel walls. This could be due to the control law itself being unstable or to inaccurate visual estimates of lateral position. The observation that visual estimates of lateral position are only accurate during straight flight, combined with the tendency of these other control laws to turn frequently, suggests that inaccurate visual estimates of lateral position resulted in the poor performance. Nevertheless, a rigorous characterization of these other controllers, such as a small signal analysis using state feedback, was not performed.

Given that accurate visual estimates of lateral position appear to require periods of relatively straight flight, one feature of a successful controller within our framework is that it generates such trajectories. Although it is unclear whether our model controller has any fundamental similarity to the control laws employed by real flies, it is interesting to note that flight trajectories of real flies are segmented into periods of relatively straight flight punctuated by rapid turns

[66]. Furthermore, it has recently been proposed that this pattern of behavior creates sensory input particularly suitable for analysis by a well-studied LPTC [67].

III. Conclusions

In this paper, we have presented an integrated framework for simulating the flight dynamics and control strategies of the fruit fly. By taking a bottom-up approach based on physically and biologically realistic components, we are able to formulate and test explicit hypotheses regarding flight control at levels ranging from stroke-by-stroke stabilization of pitch to long-range flight through a corridor. Special emphasis is placed on the sensory feedback components of the model, which limit the potential information available to controllers in our model, just as real sensory systems must do in real flies. The availability of the modeling environment enables inquiry and analysis into the principles underlying insect flight control in a closed-loop, feedback driven system. This approach is necessary due to tight coupling between motor output and multimodal sensory input, making flight control difficult to study with traditional reductionist approaches that elucidate feed-forward mechanisms. By adopting this integrative modeling approach in conjunction with further biological experimentation, we hope we will be able to provide some insight into the nature of the solutions that endow flies with their remarkable flight and goal directed behaviors. Such an endeavor seems worthwhile, because, as anyone who has tried to rid a kitchen of fruit flies knows, these animals are robust performers in a wide range of environments and in the face of severe environmental perturbations.

Acknowledgments

This work was supported by grants from the Packard Foundation, the Air Force Office of Scientific Research (AFOSR) (F49620-03-1-0171), National Science Foundation (NSF) (IBN-0217229), and Army Research Office (ARO) (DAAD 19-03-D-0004).

References

- [1] Strausfeld, N., *Atlas of an Insect Brain*, Springer-Verlag, New York, 1976.
- [2] Dickinson, M., "The Initiation and Control of Rapid Flight Maneuvers in Fruit Flies," *Integrative and Comparative Biology*, Vol. 45, No. 2, 2005, pp. 274–281.
doi:10.1093/icb/45.2.274
- [3] Dickinson, M., Farley, C., Full, R., Koehl, M., Kram, R., and Lehman, S., "How Animals Move: An Integrative View," *Science*, Vol. 288, No. 5463, 2000, pp. 100–106.
doi:10.1126/science.288.5463.100
- [4] Taylor, G., "Mechanics and Aerodynamics of Insect Flight Control," *Biological Reviews*, Vol. 76, No. 4, 2001, pp. 449–471.
- [5] Reiser, M., Humbert, S., Dunlop, M., Vecchio, D., Murray, M., and Dickinson, M., "Vision as a Compensatory Mechanism for Disturbance Rejection in Upwind Flight," *Proceedings of the 2004 American Control Conference*, IEEE, Piscataway, NJ, 2003, pp. 311–316.
- [6] Deng, X., Schenato, L., Wu, W., and Sastry, S., "Flapping Flight for Biomimetic Robotic Insects: Part 1: System Modeling," *IEEE Transactions on Robotics and Automation*, Vol. 22, No. 4, 2006, pp. 776–788.
- [7] Deng, X., Schenato, L., and Sastry, S., "Flapping Flight for Biomimetic Robotic Insects: Part 2: Flight Control Design," *IEEE Transactions on Robotics and Automation*, Vol. 22, No. 4, 2006, pp. 789–803.
- [8] Beyeler, A., Mattiussi, C., Zufferey, J., and Floreano, D., "Vision-Based Altitude and Pitch Estimation for Ultra-Light Indoor Aircraft," *IEEE International Conference on Robotics and Automation (ICRA '2006)*, IEEE, Piscataway, NJ, 2006.
- [9] Serres, J., Ruffier, F., Viollet, S., and Franceschini, N., "Toward Optic Flow Regulation for Wall-Following and Centring Behaviours," *International Journal of Advanced Robotic Systems*, Vol. 3, No. 2, 2006, pp. 147–154.
- [10] Ellington, C., "The Aerodynamics of Hovering Insect Flight 1. Lift and Power Requirements," *Philosophical Transactions of the Royal Society of London, Series B: Biological Sciences*, Vol. 305, No. 1122, 1984, pp. 1–15.
doi:10.1098/rstb.1984.0049
- [11] Ellington, C., "The Aerodynamics of Hovering Insect Flight 2. Morphological Parameters," *Philosophical Transactions of the Royal Society of London, Series B: Biological Sciences*, Vol. 305, No. 1122, 1984, pp. 17–40.
doi:10.1098/rstb.1984.0050
- [12] Ellington, C., "The Aerodynamics of Hovering Insect Flight 3. Kinematics," *Philosophical Transactions of the Royal Society of London, Series B: Biological Sciences*, Vol. 305, No. 1122, 1984, pp. 41–78.
- [13] Ellington, C., "The Aerodynamics of Hovering Insect Flight 4. Aerodynamic Mechanisms," *Philosophical Transactions of the Royal Society of London, Series B: Biological Sciences*, Vol. 305, No. 1122, 1984, pp. 79–115.
doi:10.1098/rstb.1984.0052
- [14] Ellington, C., "The Aerodynamics of Hovering Insect Flight. 5. A Vortex Theory," *Philosophical Transactions of the Royal Society of London, Series B: Biological Sciences*, Vol. 305, No. 1122, 1984, pp. 115–144.
doi:10.1098/rstb.1984.0053
- [15] Ellington, C., "The Aerodynamics of Hovering Insect Flight. 6. Lift and Power Requirements," *Philosophical Transactions of the Royal Society of London, Series B: Biological Sciences*, Vol. 305, No. 1122, 1984, pp. 145–181.
doi:10.1098/rstb.1984.0054
- [16] Borst, A., and Egelhaaf, M., "Principles of Visual Motion Detection," *Trends in Neurosciences*, Vol. 12, No. 8, 1989, pp. 297–306.
doi:10.1016/0166-2236(89)90010-6
- [17] Dickinson, M., and Gotz, K., "Unsteady Aerodynamic Performance of Model Wings at Low Reynolds Numbers," *Journal of Experimental Biology*, Vol. 174, No. 1, 1993, pp. 45–64.
- [18] Dickinson, M., "The Effects of Wing Rotation on Unsteady Aerodynamic Performance at Low Reynolds Numbers," *Journal of Experimental Biology*, Vol. 192, No. 1, 1994, pp. 179–206.
- [19] Dickinson, M., Lehmann, F., and Sane, S., "Wing Rotation and the Aerodynamic Basis of Insect Flight," *Science*, Vol. 284, No. 5422, 1999, pp. 1954–1960.
doi:10.1126/science.284.5422.1954
- [20] Birch, J., and Dickinson, M., "The Influence of Wing-Wake Interactions on the Production of Aerodynamic Forces in Flapping Flight," *Journal of Experimental Biology*, Vol. 206, No. 13, 2003, pp. 2257–2272.
doi:10.1242/jeb.00381
- [21] Sane, S., and Dickinson, M., "The Control of Flight Force by a Flapping Wing: Lift and Drag Production," *Journal of Experimental Biology*, Vol. 204, No. 15, 2001, pp. 2607–2626.
- [22] Sane, S., and Dickinson, M., "The Aerodynamic Effects of Wing Rotation and a Revised Quasi-Steady Model of Flapping Flight," *Journal of Experimental Biology*, Vol. 205, No. 8, 2002, pp. 1087–1096.
- [23] Fry, S., Sayaman, R., and Dickinson, M., "The Aerodynamics of Free-Flight Maneuvers in *Drosophila*," *Science*, Vol. 300, No. 5618, 2003, pp. 495–498.
doi:10.1126/science.1081944
- [24] Birch, J., Dickson, W., and Dickinson, M., "Force Production and Flow Structure of the Leading Edge Vortex on Flapping Wings at High and Low Reynolds Numbers," *Journal of Experimental Biology*, Vol. 207, No. 7, 2004, pp. 1063–1072.
doi:10.1242/jeb.00848
- [25] Dudley, R., *The Biomechanics of Insect Flight*, Princeton Univ. Press, Princeton, NJ, 2000.
- [26] Srinivasan, M., Poteser, M., and Kral, K., "Motion Detection in Insect Orientation and Navigation," *Vision Research*, Vol. 39, No. 16, 1999, pp. 2749–2766.
doi:10.1016/S0042-6989(99)00002-4
- [27] Krapp, H., and Hengstenberg, R., "Estimation of Self-Motion by Optic Flow Processing in Single Visual Interneurons," *Nature (London)*, Vol. 384, No. 6608, 1996, pp. 463–466.
doi:10.1038/384463a0
- [28] Egelhaaf, M., "Visual References to Flight Steering Muscles Controlling Optomotor Responses of the Fly," *Journal of Comparative Physiology A, Sensory, Neural, and Behavioral Physiology*, Vol. 165, No. 6, 1989, pp. 719–730.
doi:10.1007/BF00610871
- [29] Higgins, C., Douglass, J., and Strausfeld, N., "The Computational Basis of an Identified Neuronal Circuit for Elementary Motion Detection in Dipterous Insects," *Visual Neuroscience*, Vol. 21, 2004, p. 567.
doi:10.1017/s0952523804214079
- [30] Shoemaker, P., O'Carroll, D., and Straw, A., "Velocity Constancy and Model Visual Motion Detection in Insects," *Biological Cybernetics*, Vol. 93, No. 4, 2005, pp. 275–287.
doi:10.1007/s00422-005-0007-y
- [31] Lindemann, J., Kern, R., van Hateren, J., Ritter, H., and Egelhaaf, M., "On the Computations Analyzing Natural Optic Flow: Quantitative Model Analysis of the Blowfly Motion Vision Pathway," *Journal of Neuroscience*, Vol. 25, No. 27, 2005, pp. 6435–6448.
doi:10.1523/JNEUROSCI.1132-05.2005
- [32] Humbert, J., "Bio-Inspired Visuomotor Convergence in Navigation and Flight Control Systems," Ph.D. Thesis, California Institute of Technology, Pasadena, CA, 2005.
- [33] Hassenstein, B., and Reichardt, W., "Systemtheoretische Analyse der Zeit-, Reihenfolgen- und Vorzeichenbewertung bei der Bewegungserkennung des Rüsselkäfers *Chlorophanus*," *Zeitschrift Fur Naturforschung*, Vol. 11b, 1956, pp. 513–524 ("System Theoretical Analysis of the Time-, Order-, and Sign-Evaluation in Movement Perception by the Weevil *Chlorophanus*," *Journal of Natural Research*) (English translation).
- [34] Buchner, E., "Dunkelanregung des stationären Flugs der Fruchtfliege *Drosophila*. Dipl. Thesis," Ph.D. Thesis, Univ. Tuebingen, Tuebingen, Germany, 1971.
- [35] Heisenberg, M., and Wolf, R., "Vision in *Drosophila*: Genetics of Microbehavior," *Studies of Brain Function*, edited by V. Braitenberg, Springer-Verlag, Berlin, 1984, Vol. 12, p. 11.
- [36] Snyder, A., "Physics of Vision in Compound Eyes," *Handbook of Sensory Physiology*, edited by H. Autrum, Springer-Verlag, Berlin, 1979, Vol. 8, pp. 255–313.
- [37] Buchner, E., "Behavioral Analysis of Spatial Vision in Insects," *Photoreception and Vision in Invertebrates*, edited by M. Ali, Plenum Press, New York, 1984, pp. 561–621.
- [38] Neumann, T., "Modeling Insect Compound Eyes: Space-Variant Spherical Vision," *Biologically Motivated Computer Vision, Proceedings*, Springer-Verlag, Berlin, 2002, Vol. 2525, pp. 360–367.
- [39] Howard, J., Dubs, A., and Payne, R., "The Dynamics of Phototransduction in Insects," *Journal of Comparative Physiology A*,

- Sensory, Neural, and Behavioral Physiology*, Vol. 154, No. 5, 1984, pp. 707–718.
doi:10.1007/BF01350224
- [40] Juusola, M., and Hardie, R., “Light Adaptation in *Drosophila* Photoreceptors: 1. Response Dynamics and Signaling Efficiency at 25 degrees C,” *Journal of General Physiology*, Vol. 117, No. 1, 2001, pp. 3–25.
- [41] Straw, A., Warrant, E., and O’Carroll, D., “A ‘Bright Zone’ in Male Hoverfly (*Eristalis Tenax*) Eyes and Associated Faster Motion Detection and Increased Contrast Sensitivity,” *Journal of Experimental Biology*, Vol. 209, No. 21, 2006, pp. 4339–4354.
doi:10.1242/jeb.025117
- [42] Derham, W., “Boyle Lecture for 1711,” *Physico-Theology*, 1713.
- [43] Pflugsteadt, H., “Die Halteren der Dipteren,” *Zeitschrift für Wissenschaftliche Zoologie* (Journal for Scientific Zoology), Vol. 100, No. 1, 1912, pp. 1–59.
- [44] Pringle, J., “The Gyroscopic Mechanism of the Halteres of Diptera,” *Philosophical Transactions of the Royal Society of London, Series B: Biological Sciences*, Vol. 233, No. 602, 1948, pp. 347–384.
doi:10.1098/rstb.1948.0007
- [45] Gnatzym, W., Grünert, U., and Bender, M., “Companiform Sensilla of *Calliphora vicina* (Insecta Diptera) 1. Topography,” *Zoomorphology*, Vol. 106, No. 5, 1987, pp. 312–319.
doi:10.1007/BF00312005
- [46] Hengstenberg, R., “Mechanosensory Control of Compensatory Head Rolling During Flight in the Blowfly *Calliphora erythrocephala*,” *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, Vol. 163, No. 2, 1988, pp. 151–165.
- [47] Nalbach, G., “The Halteres of the Blowfly *Calliphora* 1. Kinematics and Dynamics,” *Journal of Comparative Physiology A, Sensory, Neural, and Behavioral Physiology*, Vol. 173, No. 3, 1993, pp. 293–300.
doi:10.1007/BF00212693
- [48] Fayyazuddin, A., and Dickinson, M., “Haltere Afferents Provide Direct, Electrotonic Input to a Steering Motor Neuron in the Blowfly *Calliphora*,” *Journal of Neuroscience*, Vol. 16, No. 16, 1996, pp. 5225–5232.
- [49] Faust, R., “Untersuchungen zum Halterenproblem,” *Zoologische Jahrbucher für Abteilung für Allgemeine Zoologie und Physiologie der Tiere* (Zoological Yearbook Section for Common Zoology and Physiology of Animals), Vol. 63, 1952, pp. 325–366.
- [50] Dickinson, M., “Haltere-Mediated Equilibrium Reflexes of the Fruit Fly, *Drosophila melanogaster*,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, Vol. 354, No. 1385, 1999, pp. 903–916.
- [51] Sherman, A., “The Integration of Visual and Haltere Feedback in *Drosophila* Flight Control,” Ph.D. Thesis, University of California, Berkeley, CA, 2003.
- [52] Shabana, A., *Computational Dynamics*, Wiley, New York, 2001.
- [53] Coutinho, M., *Dynamic Simulations of Multibody Systems*, Springer-Verlag, New York, 2001.
- [54] Balint, C., and Dickinson, M., “The Correlation Between Wing Kinematics and Steering Muscle Activity in the Blowfly *Calliphora vicina*,” *Journal of Experimental Biology*, Vol. 204, No. 24, 2001, pp. 4213–4226.
- [55] Durikovic, R., and Numata, K., “Human Hand Model Based on Rigid Body Dynamics,” *Proceedings of the (IV’04) Information Visualization, Eighth International Conference*, IEEE, Washington, D.C., 2004, pp. 853–857.
- [56] Pollard, N., and Zordan, V., “Physically Based Grasping and Control from Example,” *ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2005*, Association for Computing Machinery, Inc., New York, 2005, pp. 311–358.
- [57] Go, J., Browning, B., and Veloso, M., “Accurate and Flexible Simulation for Dynamic, Vision Centric Robots,” *Autonomous Agents and Multiagent Systems (AAMAS 2004)*, IEEE, New York, 2004, pp. 1386–1387.
- [58] Stewart, D., and Trinkle, J., “An Implicit Time-Stepping Scheme for Rigid Body Dynamics with Inelastic Collisions and Coulomb Friction,” *International Journal of Numerical Methods in Engineering*, Vol. 39, No. 15, 1996, pp. 2673–2691.
doi:10.1002/(SICI)1097-0207(19960815)39:15<2673::AID-NME972>3.0.CO;2-I
- [59] Mirtich, B., “Fast and Accurate Computation of Polyhedral Mass Properties,” *Journal of Graphics Tools*, Vol. 1, No. 2, 1996, pp. 31–50.
- [60] Birch, J., and Dickinson, M., “Spanwise Flow and the Attachment of the Leading-Edge Vortex on Insect Wings,” *Nature (London)*, Vol. 412, No. 6848, 2001, pp. 729–733.
doi:10.1038/35089071
- [61] Fung, Y., *An Introduction to the Theory of Aeroelasticity*, Dover, New York, 1969.
- [62] Sedov, L., *Two-Dimensional Problems in Hydrodynamics and Aerodynamics*, Interscience Publishers, New York, 1965.
- [63] David, C., “Relationship Between Body Angle and Flight Speed in Free-Flying *Drosophila*,” *Physiological Entomology*, Vol. 3, No. 3, 1978, pp. 191–195.
- [64] Dickson, W., and Dickinson, M., “The Effect of Advance Ratio on the Aerodynamics of Revolving Wings,” *Journal of Experimental Biology*, Vol. 207, No. 24, 2004, pp. 4269–4281.
doi:10.1242/jeb.01266
- [65] Sherman, A., and Dickinson, M., “Summation of Visual and Mechanosensory Feedback in *Drosophila* Flight Control,” *Journal of Experimental Biology*, Vol. 207, No. 1, 2004, pp. 133–142.
doi:10.1242/jeb.00731
- [66] Collett, T., and Land, M., “Visual Control of Flight Behavior in the Hoverfly, *Syriffa pipiens* L.,” *Journal of Comparative Physiology*, Vol. 99, No. 1, 1975, pp. 1–66.
doi:10.1007/BF01464710
- [67] Kern, R., van Hateren, J., Michaelis, C., Lindemann, J., and Egelhaaf, M., “Function of a Fly Motion-Sensitive Neuron Matches Eye Movements During Free Flight,” *PLoS Biology*, Vol. 3, No. 6, 2005, p. e171.
doi:10.1371/journal.pbio.0030171

G. Abate
Guest Editor