Virtual Insect Flight Simulator (VIFS): A Software Testbed for Insect Flight*

L. Schenato, X. Deng, W.C. Wu, S. Sastry
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
{*lusche*|*xinyan*|*wcwu*|*sastry*}*@robotics.eecs.berkeley.edu*

Abstract

This paper presents the design of software simulation VIFS for insect flight. In particular, it is intended to estimate flight control algorithms and performance for a Micromechanical Flying Insect (MFI), a 10-25 mm (wingtip-to-wingtip) device eventually capable of sustained autonomous flight. The VIFS is an end-to-end tool composed of several modular blocks which model the actuators dynamics, the wing aerodynamics, the body motion, the visual and inertial sensors, the environment perception, and control algorithms. A 3D virtual environment simulation is also developed as a visualization tool. We present the current state of art of its implementation and preliminary results.

Index Terms– MFI, insect flight, software simulator, insect aerodynamics, optical flow sensor, insect dynamics.

1 Introduction

The extraordinary flight capabilities of insects have captured the interest of the scientific community since ever. The fact that they can quickly change direction of motion, yet keeping perfect attitude control, as a result of millions years of evolution, has given them the appellative of perfect flying machines. They can take off or land vertically, hover for several seconds, move laterally or backward, and even fly upside-down. Moreover, the latest advances both in insect aerodynamics and in microtechnology, have suggested to explore the feasibility for the fabrication of an electromechanical device capable of autonomous flight and of mimicking real flying insects. This is the challenge that the Micromechanical Flying Insect project, MFI, being currently developed at UC Berkeley, has undertaken. Figure 1 shows a conceptual view.

*This work was funded by ONR MURI N00014-98-1-0671, ONR DURIP N00014-99-1-0720 and DARPA.

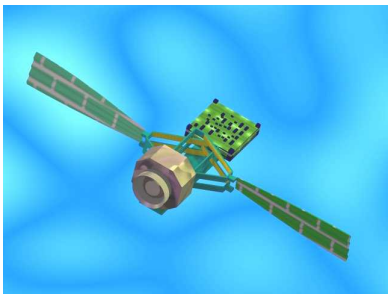

Figure 1: Graphical model for the MFI for the OpenGL 3D animation

However, very little is still known about sensors, neural processing of external information, biomechanical structure, wing aerodynamics , flight control algorithms and trajectory planning in flying insects [2] [4] [1]. Therefore, we have started developing a software tool that is meant to accurately simulate robotic flying insects by including models for the actuators dynamics, the wing aerodynamics, the sensors, the external environment and the control algorithms. This simulator, called Virtual Insect Flight Simulator, VIFS, will help to evaluate, and then improve, different electromechanical design for the wing-torax structure, different sensors design and information fusion techniques, flight control algorithms and trajectory planning. As a result, it will reduce time-to-fabrication and risk factors for the MFI.

2 MFI Overview

The design of the MFI is obviously guided by real flying insect studies, however, the challenging requirements for a feasible fabrication, such as small dimensions, low power consumption, high flapping frequency and fast robust control, have forced the development of novel approaches and new technology.

The goal of the MFI project is the fabrication an electromechanical device capable of autonomous flight and complex behaviors, mimicking a blowfly Calliphora, which has a mass of $100mg$, wing length of $11mm$, wing beat frequency of $150Hz$, and actuator power of $10mW$. The challenging requirements for the fabrication of the MFI, such as small dimensions, low power consumption, high flapping frequency and fast robust control, have forced the development of novel approaches and technologies. The fabrication of such a device requires the design of several components. In particular, it is possible to identify five main units (Figure 2), each of them responsible of a distinct task: the *locomotory unit*, the *sensory system unit* , the *power supply unit*, the *communication unit* and the *control unit*.

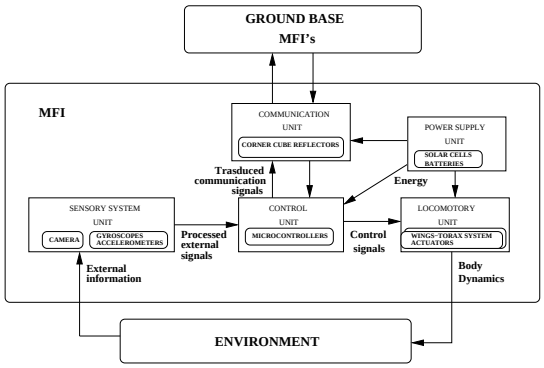

Figure 2: MFI structure

The locomotory unit, composed by the electromechanical wings complex, is responsible for generating the necessary aerodynamic forces for the flight, and thus of the MFI dynamics. One of the most challenging parts of this project is the design of the mechanical structure which must provide sufficient mobility to the wings to generate the desired kinematics. We do not enter this issue in this paper and we address the interested reader to more detailed work [15] [16] [19].

The sensory system unit, composed by an inertial navigation system based on MEMS accelerometers and gyros and camera for optical flow measurements, provides the control unit with the input information necessary to stabilize the flight and to plan desired trajectories. Other kinds of MEMS sensors can be installed, such as temperature and chemical sensors, which can be used for detecting particular objects or hazardous waste areas.

The power supply unit, composed by a thin sheet of solar cells at the base of the MFI body is the source of electric energy necessary to the wing actuators and to the electronics of all the units. This sheet of solar cell can generate up to $20mWcm^{-1}$.

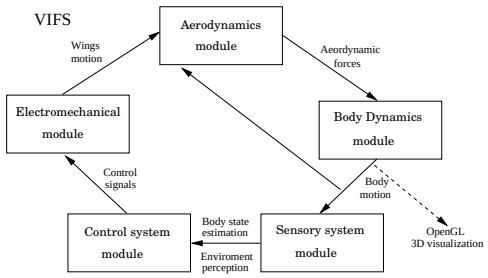

Figure 3: Virtual Insect Flight Simulator (VIFS) architecture

The communication unit, based on micro Corner Cube Reflectors(CCR) [3], a novel optoelectronic transmitter, or on ultra-low-power RF transmitters, provides a MFI with the possibility to communicate with a ground base or with others MFI's.

Finally, the control unit, embedded in the MFI computational circuitry, is responsible both for stabilizing the flight and for planning the appropriate trajectory for each desired task.

3 VIFS Architecture

In order to evaluate the performance of flight control algorithms, a Virtual Insect Flight Simulator (VIFS) is being implemented to simulate the flight of an MFI inside a virtual environment.

The VIFS is decomposed into several modular units, each of them responsible of an independent task, as shown in Figure 3.

The **Aerodynamic Module** takes as input the wings motion and the MFI body velocities, and gives as output the corresponding aerodynamic forces and torques. This module corresponds to a mathematical model for the aerodynamics. Its modeling is described in the next section.

The **Body Dynamics Module** takes the aerodynamics forces and torques generated by the wing kinematics and integrates them along with the dynamical model for the MFI body, thus computing the body's position and the attitude as a function of time.

The **Sensory System Module** models the sensors used by the MFI to recover its state and to interact with the environment. They are typically gyroscopes, accelerometers and optical flow sensors. Eventually, this module will include also a model for the environment, i.e. a description of the terrain and the objects in it. It takes as input the MFI body dynamics and generates the corresponding sensory information which is used to estimate the MFI's state, i.e. position and attitude, and displacement of objects and perception of the envi-

ronment.

The **Control System Module** takes as input the MFI body state and eventually the perception of the external world. Its task is to decide a control strategy to achieve a desired mission and to generate the control signals to the electromechanical system. Its architecture has been presented in a complementary article [13].

The **Electromechanical System Module** takes as input the electrical control signals generated by the Control System Module and generates the corresponding wing kinematics. It consists of the model of the electromechanical wing-torax architecture and the aerodynamic damping on the wings.

The traces of the motion of the wings, the corresponding aerodynamic forces acting on them, and the trajectory motion and heading of the MFI produced by the VIFS, are combined together into a 3D virtual environment simulation. This simulation, generated by OpenGL graphical tool, animates not only the MFI motion but also the flapping wings and the instantaneous aerodynamic forces, which are represented as arrow fixed on the wings center of mass (see Figure 1)

The VIFS architecture is extremely flexible since it allows readily modifications or improvements of one single module without rewriting the whole simulator. For example, different combinations of control algorithms and electromechanical structure can be tested, giving rise to the more realistic setting of flight control with limited kinematics due to electromechanical constraints. Moreover, dimensions and masses of wings and body can be modified to analyze their effects on flight stability, power efficiency and maneuverability. Finally, as soon as better aerodynamic models are available, the aerodynamic module can be updated to improve accuracy. As present, the VIFS is not fully implemented: the Electromechanical System Modules is under investigation [19] and the Sensory System Module includes only optical flow sensors. The following sections present the state of art for the VIFS. The design of the Control Module is treated thoroughly in a complementary article, and is not included here [13].

3.1 Aerodynamic Module

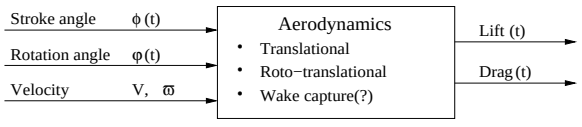

Figure 4: Block diagram of the Aeodynamical Module

Insect flight aerodynamics has been a very active area of research in the past decades after the seminal work of Ellington [7]. Although, at present, few numerical simulations of unsteady insect flight aerodynamics give sensible results for aerodynamics forces [18], several advances have been achieved in comprehending qualitatively and quantitatively unsteady state aerodynamics mechanisms thanks to scaled model of flapping wing [4]. This apparatus, known as Robofly, consists of two-winged system mimicking the wing motion of flying insects. It is provided with force sensors at the wing base, which can measure instantaneous wing forces along a wingbeat.

Results obtained with this apparatus have identified 3 main aerodynamics mechanisms: *delayed stall*,*rotational circulation* and *wake capture*. The delayed stall is the result of the translational motion of the wing and it depends only on the wing translational velocity and angle of attack. This mechanism is similar to one present on fixed wing aircrafts, but the former is stable also at high angle of attack. Rotational circulation results from the interaction of translational and rotational velocity of the wing. It is analogous to the effect of back or top spin on a tennis ball, which can bend the trajectory. However, the fact that the wing profile is flat and not spherical, is an important difference, since the force direction depends on the angle of attack. Finally, the wake capture is the result of the interaction of the wing, when it inverts its motion, with the fluid wake generated in the previous stroke. This mechanism increases the pressure force on the wing and therefore enhances lift generation. The main key finding of Dickinson's work [4] was that modulating a few wing kinematics parameters, such as wing timing rotation at the stroke reversals and angle of attack, an insect can readily apply torques on the body and therefore control its attitude and motion.

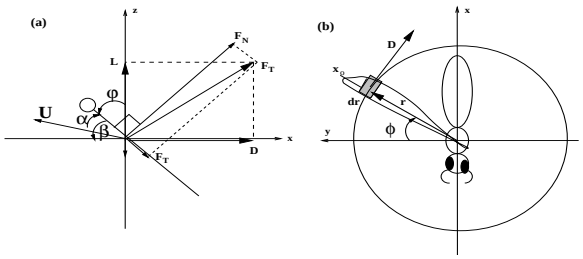

Figure 5: Force Decomposition for Horizontal Stroke Plane: (a) lateral view; (b) top view

The aerodynamic module is a combination of an analytical model, based on quasi-steady state equations for the delayed stall and rotational circula-

tion, and an empirically matched model based on Robofly data. Wake capture is very complicated to treat analytically and it is not considered in this work. However, this mechanism seems to have a small contribution for sinusoidal-like motion for the wings, motion that it is widely used in our simulations.

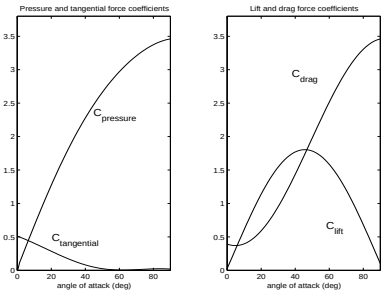

Figure 6: Aerodynamic force coefficients empirically obtained from RoboFly data

A quasi-state state aerodynamic model assumes that force equations derived for 2D thin aerofoils translating with constant velocity and constant angle of attack, hold also for time varying 3D flapping wings. In steady state condition the aerodynamic force per unit length exerted on a aerofoil due to delayed stall is given [10]:

$$F'_{tr,N} = \frac{1}{2} C_N(\alpha) \rho\, c\, U^2 \quad (1)$$

$$F'_{tr,T} = \frac{1}{2} C_T(\alpha) \rho\, c\, U^2 \quad (2)$$

where $F'_{tr,N}$ and $F'_{tr,N}$ are, respectively, the normal and tangential components of the force with respects to the aerofoil profile, c is the cord width of the aerofoil, ρ is the density of air, α is the angle of attack defined as the angle between the wing profile and the wing velocity relative to the fluid, U, and C_N and C_T are the dimensionless force coefficients. Figure 5 shows a graphical representation of these forces. In the aerodynamics literature, it is more common to find the lift and drag force coefficients, C_L and C_D. Lift, L and drag, D are defined, respectively, as the normal and tangential components of the force with respect to the stroke plane, i.e. the plane of motion of the wings with respect to the body. However, our decomposition is more intuitive, since aerodynamic forces are in general pressure force which act perpendicularly to the surface. Figure 6 shows the empirical values for the force coefficients obtained with RoboFly. It is clear how, for high angle of attack, the tangential component, mainly due to skin friction, gives only a minor contribution. Nevertheless, lift and drag can be readily computed as:

$$C_L = C_N \cos(\beta) - C_T \sin(\beta) \quad (3)$$

$$C_D = C_N \sin(\beta) + C_T \cos(\beta) \quad (4)$$

where β is the angle between the stroke plane and the wing profile. The angles α and β coincides only if the insect body is moving parallel to the stroke plane.

The aerodynamic force per unit length exerted on a aerofoil due to rotational circulation is given by [8]:

$$F'_{rot,N} = C_{rot} \rho\, c^2\, U\, \omega \quad (5)$$

where $C_{rot} = \pi \left(\frac{3}{4} - \hat{x}_o\right)$ is the rotational force coefficients, approximately independent of the angle of attack, x_o is the dimensionless distance of the rotation axis from the leading edge, and ω is the angular velocity of the wing with respects to that axis. This is a pure pressure force and therefore acts perpendicularly to the wing profile.

According to the quasi-steady state approach, the total force on a wing is computed by dividing the wing into infinitesimal blades as shown in Figure 5. First, we calculate the total force on each blade:

$$\begin{aligned} dF_{tr,N}(t,r) &= \frac{1}{2} C_N(\alpha(t)) \rho\, c(r)\, U^2(t,r)\, dr \\ dF_{tr,T}(t,r) &= \frac{1}{2} C_T(\alpha(t)) \rho\, c(r)\, U^2(t,r)\, dr \\ dF_{rot,N} &= C_{rot} \rho\, c(r)^2\, U(t,r)\, \dot{\alpha}(t)\, dr \\ U(t,r) &= \dot{\phi}(t) r + U_{body}(t) \end{aligned} \quad (6)$$

where ϕ is the stroke angle and U_{body} is the velocity of the insect body w.r.t. an inertial system, and the wing angular velocity, ω is approximately $\dot{\alpha}$. Then we integrate the forces in Equations (6) along the wing, assuming that superposition of effects holds, to get:

$$\begin{aligned} F_{tr,N}(t) &= A\, C_N(\alpha(t)) \left(d\, \dot{\phi}(t) + U_{body}(t)\right)^2 \\ F_{tr,T}(t) &= A\, C_T(\alpha(t)) \left(d\, \dot{\phi}(t) + U_{body}(t)\right)^2 \\ F_{rot,N} &= B\, C_{rot}\, \dot{\alpha}(t) \left(d\, \dot{\phi}(t) + U_{body}(t)\right) \end{aligned} \quad (7)$$

where, $A = \frac{1}{2} R\, \bar{c}$, $B = \frac{1}{2} \left(\frac{3}{4} - \hat{x}_o\right) \hat{v}\, R\, \bar{c}^2$,and , $d = R\, \hat{r}_2$, are constants that depend on the wing morphology. As a result of this approach, the wing forces can be assumed to be applied at a distance, d, from the wing base .

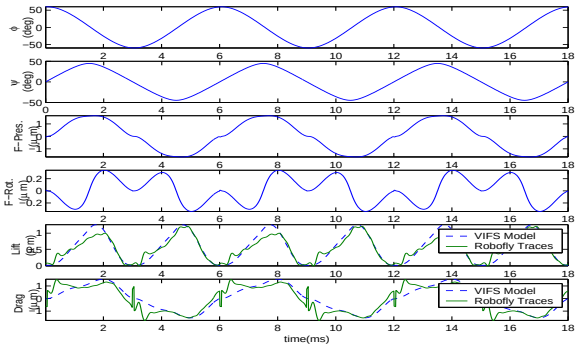

Figure 7: Lift and drag forces simulation and measured Robofly data

The total lift and drag forces can be derived from Equations (7) through a trigonometric transformation analogous to the one used in Equations (4). The Aerodynamic Module, whose block diagram shown in Figure 4, is based on Equations (7). Figure 7 shows the simulated aerodynamic forces for a typical wings motion.

3.2 Body Dynamics Module

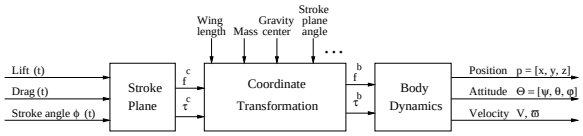

Figure 8: Body Dynamics Block Diagram

Given the aerodynamic forces generated by the wing kinematics, the body Dynamics Module integrates the rigid body equations of motion, and gives the body position and attitude trajectories w.r.t. the inertia frame. The input to the body dynamics module is the stroke angle, lift and drag forces. Since the lift and drag forces are generated and described in the the *stroke plane frame*, we need to do a coordinate transformation before we obtain the forces and torques acting on the *body frame*. Before entering the discussion, three coordinate systems and their corresponding parameters need to be defined.

As seen from Figure 9, the *fixed frame* is identified by the axis (x, y, z). The *insect body frame* is described by the coordinate system (x', y', z') attached to the body center of gravity. The *stroke plane frame* is identified by the new axis $(\bar{x}, \bar{y}, \bar{z})$, and is attached to the wing base. The *stroke plane frame* is defined as the plane to which the wing motion is approximately confined during flapping flight.

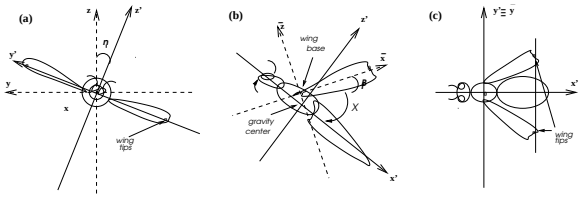

Figure 9: Coordinate Systems: (a) Front View; (b) Lateral View; (c) Top View

As shown in [11], the rigid body motion equations subject to an external wrench $F^b = [f^b, \tau^b]^T$ applied at the center of mass and specified with respect to the body coordinate frame, are given:

$$\begin{bmatrix} \mathrm{mI} & 0 \\ 0 & \mathcal{I} \end{bmatrix} \begin{bmatrix} \dot{v}^b \\ \dot{\omega}^b \end{bmatrix} + \begin{bmatrix} \omega^b \times mv^b \\ \omega^b \times \mathcal{I}\omega^b \end{bmatrix} = \begin{bmatrix} f^b \\ \tau^b \end{bmatrix}$$

where $\mathcal{I}$ is the inertia matrix. v^b is the velocity vector of the center of mass in spatial coordinates, and ω^b is the angular velocity vector in body frame.

Given the lift and drag generated by aerodynamics, together with the stroke angle, the forces and torques in *stroke plane* can be calculated as

$$f_a^c = \begin{bmatrix} (f_d^l + f_d^r)\cos\phi \\ (f_d^l - f_d^r)\sin\phi \\ f_l^l + f_l^r \end{bmatrix} \quad \tau_a^c = \begin{bmatrix} (f_l^r - f_l^1)r\cos\phi \\ -(f_l^l + f_l^r)r\sin\phi \\ (f_d^l - f_d^r)r \end{bmatrix}$$

To obtain the aerodynamics forces and torques in the *body frame*, we do a coordinate transformation as

$$\begin{bmatrix} f_a^b \\ \tau_a^b \end{bmatrix} = \begin{bmatrix} R_{cb}^T & 0 \\ -R_{cb}^T \hat{p_{cb}} & R_{cb}^T \end{bmatrix} \begin{bmatrix} f_a^c \\ \tau_a^c \end{bmatrix}$$

where R_{cb} is the rotation matrix of the body frame relative to the stroke plane, and p_{cb} represents the translation of the origin of the body frame from the stroke plane.

The gravitational forces and torques in *body frame*, are given by:

$$\begin{bmatrix} f_g^b \\ \tau_g^b \end{bmatrix} = \begin{bmatrix} R^T \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \\ 0 \end{bmatrix}$$

where R is the rotational matrix of the body frame relative to the spatial ftame, and g is the gravitational acceleration.

The viscous damping exerted by the air on the insect body are approximately given by:

$$\begin{bmatrix} f_d^b \\ \tau_d^b \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{2}C_d\rho 2rlv_y^2 \\ \frac{1}{2}C_d\rho 2rlv_z^2 \\ 0 \\ \frac{1}{4}C_d\rho r[l_1^4 + (l-l_1)^4]\omega_{pitch}^2 \\ \frac{1}{4}C_d\rho r[l_1^4 + (l-l_1)^4]\omega_{yaw}^2 \end{bmatrix}$$

where C_d is the viscous damping coefficient, ρ is the density of the air, l_1 is the gravity acting point, l is the body length, r is the body radius. The total forces and torques in the body frame are given by the sum of the previous three forces, i.e $f^b = f_a^b + f_g^b + f_d^b$ and $\tau^b = \tau_a^b + \tau_g^b + \tau_d^b$. The values for the body and wing morphological parameters, such as lengths and masses, are not fixed, thus allowing the simulation of different insects. The block diagram of the Body Dynamics Module is shown in Figure 8.

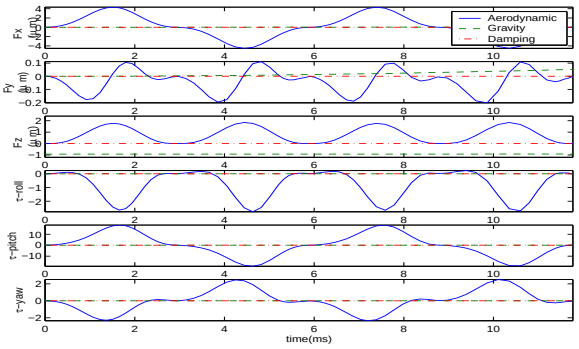

Figure 10: Forces and Torques in Body Frame

For a simpler numerical implementation, we define the new variables $\dot{P} = v^p = Rv^b$ and $\hat{\omega}^b = R^T\dot{R}$.

For $R \in SO(3)$, we parametrize R by ZYX euler angles with ϕ, θ, and ψ about x,y,z axes respectively, and hence $R = e^{\hat{z}\psi}e^{\hat{y}\theta}e^{\hat{x}\phi}$ with $x = [1\ 0\ 0]^T$, $y = [1\ 0\ 0]^T$,$z = [0\ 0\ 1]^T$ and $\hat{x}, \hat{y}, \hat{z} \in so(3)$. By differentiating R with respect to time, we have the state equations of the Euler angles, $\Theta = [\phi\ \theta\ \psi]^T$, which can be defined as $\dot{\Theta} = W\omega^b$. By defining the state vector $[P, \Theta] \in R^3 \times R^3$ where P is the position of the center of mass w.r.t. the inertia frame, and Θ are the euler angles which we use to parametrize the rotation matrix R, we can rewrite the equations of motion of a rigid body as:

$$\begin{aligned} \ddot{\Theta} &= (\mathcal{I}W)^{-1}[\tau^b - W\dot{\Theta} \times \mathcal{I}W\dot{\Theta} - \mathcal{I}\dot{W}\dot{\Theta}] \\ \ddot{P} &= \frac{1}{m}Rf^b \end{aligned} \tag{8}$$

The output from the body dynamic module is the position and body orientation of the insect. Figure 11 shows the simulation results using the lift and drag forces wings kinematics shown in Figure 7. The dynamics of the insect shows an oscillatory motion superimposed to a drifting term. The drifting term is a result of a mean non-zero force along a wingbeat, while the oscillatory motion is the result of the time-varying nature of aerodynamic forces for insect flight.

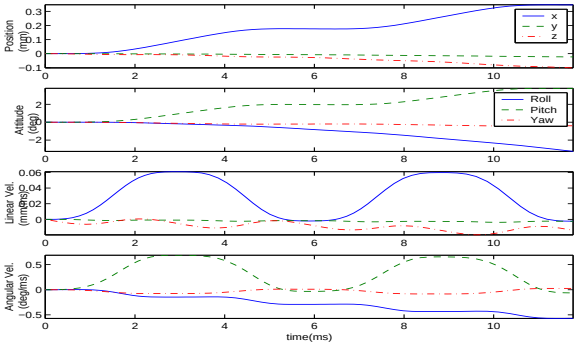

Figure 11: Insect dynamics simulation resulting from wings kinematics shown in Figure 7

3.3 Sensory Module: Optical flow sensor

Our visual sensor model is based on the implementation of the Reichardt motion detectors in VLSI hardware developed at California Institute of Technology [9]. The Reichardt motion detector is a correlation-based algorithm proposed by Reichardt and Hassenstein in 1956. The architecture of the Reichardt motion detector is shown in Figure 12. Since the two adjacent photoreceptors are close to each other (1.5^o apart in the VLSI implementation), they will sense the same light intensity. However, if the scene is moving, the signal from one photoreceptor will lead or lag behind the signal from the other depending on the direction of the scene movement. If the signal from the left photoreceptor leads that from the right receptor, then we know that the scene is moving from left to right. On the other hand, if the signal from the left receptor lags behind the signal from the right signal, the scene is moving from right to left. One method to determine the direction of the scene movement is to delay the signal from one receptor and compare this delayed signal with the signal from the adjacent receptor. If the delayed signal from the left receptor is more strongly correlated with the signal from the right receptor than the delayed signal from the right receptor is with the signal from the left receptor, then the scene is moving from left to right. Similarly, if the opposite situation holds, the scene is moving from right to left.

In the hardware implementation, the photoreceptor is modeled as a second order band-pass filter. The delay component is realized as a low-pass filter. The correlation is achieved by multiplying the delayed signal in one half-detector (i.e. one photoreceptor) with the signal in the neighboring half-detector. The directional selectivity is accomplished by subtracting the correlated signals in the

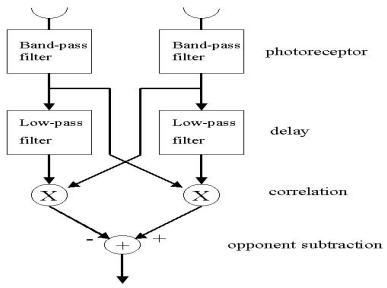

Figure 12: Reichardt motion detector architecture. The photoreceptor senses the incoming light intensity. The filtered signal in one leg of the detector is multiplied by the delayed signal in an adjacent leg. The responses of the two legs are subtracted in opponency to achieve directional selectivity.

two adjacent half-detectors in opponency. In our simulation, we used an array of 20 motion detectors and summing their outputs to give an overall sensor response. This spatial summation has the effect of integrating over different phases of the stimulus and hence eliminating the pattern dependent oscillations observed in a single motion detector [12] [14].

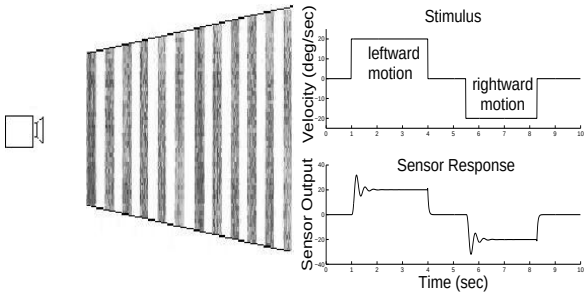

Figure 13: The directional selectivity of the sensor to a sinusoidal grating. The leftward motion of the stimulus produces a positive output, while the rightward motion of the stimulus produces a negative output. Transient reponses are observed at the stimulus motion onsets.

In the following experiments, we test our visual sensor model as an attempt for the state estimation for the insect flight simulator. First, we expose the sensor to a sinusoidal grating moving along the sensor axis. This is shown in Figure 13. When the stimulus is moving to the left, the sensor outputs a positive response. When the stimulus is moving to the right, the response is negative. In addition, a large transient response at the onset of stimulus motion oscillates with the temporal frequency of the stimulus pattern. This oscillatory behavior at the stimulus motion onset is also observed in the fly's visual system [5].

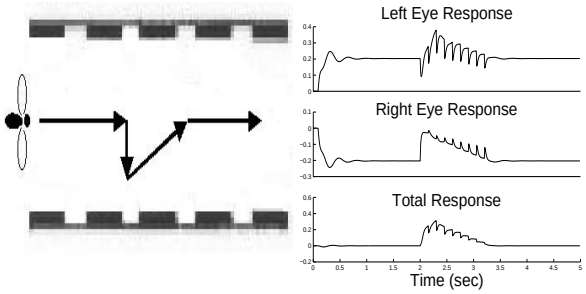

Figure 14: Insect flying through a tunnel. (a) The insect initially flies in the center of the tunnel. However, the insect is shifted towards right side of the tunnel during its flight. (b) When the insect is closer to the right wall, its right eye sees a greater speed of the image motion than its left eye does. Therefore, the insect returns to the middle line in order to restore the balance in the image speeds in its two eyes.

Next, we use two sensors for our experiment. We placed one sensor on the left hand side of an insect and the other on the right. The insect was assumed to fly through a tunnel and each wall of the tunnel was painted with a pattern consisting of black and white strips. Furthermore, we require that the patterns on the two walls have the same spatial frequency (see Figure 14 (a)). It has been observed that when an insect flies through an opening, it tends to fly through the center. This centering behavior is primarily due to the fact that the insect tries to maintain equidistance from the sides by balancing the speeds of the image motion in its eyes [17]. As shown in Figure 14, the insect initially followed the middle path of the tunnel. The responses of the left and the right sensors balanced and hence the insect would maintain a stable flight. However, when the insect's trajectory was shifted (by a gust of wind) towards one side of the tunnel during its flight, there would be an unequal speeds of the image motion in its two eyes. If the insect was closer to the left wall, its left eye saw a greater speed of the image motion than its right eye did, and vice versa. In our experiment, the two sensors produced a non-zero total response when the insect's trajectory was disturbed from the middle line. As a consequence, the insect would return to the center of the tunnel by trying to restore the balance in the image speeds (balancing the sensor outputs) in its two eyes.

3.4 Implementation

All the models presented above have been implemented using Matlab. In particular, the differential Equations (8) have been simulated in Matlab using the built-in routine `ode45`, an adaptive step solver

for non-stiff differential equations. At present, the wing motion is specified off-line and then simulations are performed. We are currently extending our software to include wing motions that are chosen depending on the state of the MFI body motion. For the insect body model, we have adopted the morphological parameters of a *drosophila* [6]. The flow sensors have been implemented as a set of linear systems, one for each photodetector, connected together as shown in Figure 12.

4 Conclusion

In this paper we have proposed the design of a accurate software simulation for insect flight that includes all major components involved: aerodynamics, MFI elechromechanics, sensors and external environment. We have modeled and implemented several of those components and we have obtained simulation results that are consistent with observations from real flying insects. Finally, we have implemented a 3D graphical visualization tool which can animate the motion of the simulated MFI in a 3D environment. Current research is directed in improving some of the models considered, such as the sensors and the actuators, and to take advantage of this simulator to evaluate flight control schemes.

5 Acknowledgments

The authors thank R. Fearing, M.Dickinson, S.Sane, M.Cavusoglu, L. Tammero for helpful discussions and insights, and C. Ma for helping us with OpenGL tool.

References

[1] A.Fayyazuddin and M.H. Dickinson. Haltere afferents provide direct, electrotonic input to a steering motor neuron of the blowfly. *J. Neurosci.*, 16:5225–5232, 1996.

[2] W.P. Chan, F. Prete, and M.H. Dickinson. Visual input to the efferent control system of a fly's 'gyroscope'. *Science*, 289:289–292, 1998.

[3] P.B. Chu, N.R. Lo, E.C. Berg, and K.S.J. Pister. Optical communication using micro corner cube reflectors. In *Proceedings of the IEEE on Micro Electro Mechanical Systems*, pages 350–355, 1997.

[4] M.H. Dickinson, F.-O Lehmann, and S.S. Sane. Wing rotation and the aerodynamic basis of insect flight. *Science*, 284(5422), 1999.

[5] M. Egelhaaf and A. Borst. Transient and steady-state response properties of movement detectors. *Journal of the Optical Society of America A*, 6(1):116–127, 1989.

[6] C. P. Ellington. The aerodynamics of hovering insect flight. ii. morphological parameters. *Phil. Trans. R. Soc. London B*, 305:17–40, 1984.

[7] C.P Ellington. The aerodynamics of hovering insect flight.I-VI. *Phil. Trans. R. Soc. London B*, 305:1–181, 1984.

[8] Y. C. Fung. *An introduction to the theory of aeroelasticity*. New York, Dover, 1969.

[9] R. R. Harrison. *An Analog VLSI Motion Sensor Based on the Fly Visual System*. Computation and neural systems program, California Institute of California, 2000.

[10] A. M. Kuethe and C.-Y. Chow. *Foundations of aerodynamics*. John Wiley & Sons, New York, 1986.

[11] R.M. Murray, Z. Li, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. RCR Press, New York, 1993.

[12] W. Reichardt and M. Egelhaaf. Properties of individual movement detectors as derived from behavioural experiments on the visual system of the fly. *Biological Cybernetics*, 58(5):287–294, 1988.

[13] L. Schenato, X. Deng, and S.S. Sastry. Flight control system for a micromechanical flying insect: Architecture and implementation. In *IEEE International Conference on Robotics and Automation*, 2001.

[14] S. Single and A. Borst. Dendritic integration and its role in computing image velocity. *Science*, 281(5384):1848–50, 1998.

[15] M. Sitti. PZT actuated four-bar mechanism with two flexible links for micromechanical flying insect thorax. In *IEEE International Conference on Robotics and Automation Conf.*, 2001.

[16] M. Sitti, D. Campolo, J. Yan, R.S. Fearing, T. Su, D. Taylor, and T. Sands. Development of pzt/pznpt unimorph actuators for micromechanical flapping mechanisms. In *IEEE International Conference on Robotics and Automation*, 2001.

[17] M. V. Srinivasan, J. S. Chahl, K. Weber, S. Venkatesh, M. G. Nagle, and S. W. Zhang. Robot navigation inspired by principles of insect vision. *Robotics and Autonomous Systems*, 26(2-3):203–216, 1999.

[18] Z. Jane Wang. Two dimensional mechanism for insect hovering. *Phys. Rev. Lett*, 85, 2000.

[19] J. Yan, R. Wood, S. Avandhanula, M. Sitti, and R. Fearing. Towards flapping wing control for a micromechanical flying insect. In *IEEE International Conference on Robotics and Automation*, 2001.