Flight Control System for a Micromechanical Flying Insect: Architecture and Implementation[*]

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

Abstract

This paper describes recent results on the design and simulation of a flight control system for Micromechanical Flying Insect (MFI), a 10-25 mm (wingtip-to-wingtip) device eventually capable of sustained autonomous flight. The biologically inspired system architecture results in a hierarchical structure of different control methodologies, which give the possibility to plan complex missions from a sequence of simple flight modes and maneuvers. As a case study, a stabilizing hovering control scheme is presented and simulated with VIFS, a software simulator for insect flight.

Index Terms– MFI, insect flight, hierarchical control, hovering, switching control.

1 Introduction

Unmanned air vehicles, UAV, have been a very active area of research, since they are indispensable for various applications where human intervention is considered difficult or dangerous. Despite that recent remarkable achievements obtained with fixed and rotary aircrafts [8] [4], their use in several tasks is still limited by their cost or by their size. However, the latest advances in insect flight aerodynamics and in microtechnology, seem to provide the sufficient tools to fabricate flying micro-robots mimicking real flying insects. Despite flying micro-robots have limited payload capacity and require still air, their unmatched maneuverability, low fabrication cost and small size make them very attractive for cost-critical missions in environments which are unpenetrable for larger size UAV's.

This paper describes the design and the implementation of a flight control system for a Micromechanical Flying Insect being currently developed at UC Berkeley. This system is responsible for planning missions, planning flight paths, generating trajectories and sequence of flight modes, and regulating the MFI motion along simple trajectories.

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 technologies.

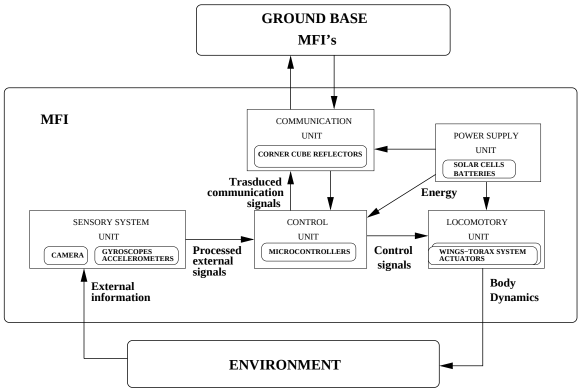

Figure 1: MFI structure

The goal of the MFI project is the fabrication of 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 demanding 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 1), 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.*

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

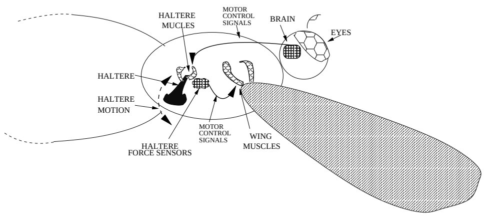

Figure 2: Neuromotor control physiology in flying insect

3 Overview of Insect Flight Control

The feasibility of fabricating a device capable to generate sufficient lift to sustain itself, does not imply the feasibility of *stable* flight. Therefore, the MFI requires a control unit which must stabilize the flight and, eventually, show complex behaviors and plan trajectories. Unfortunately, the difficulties, which have to be faced, are numerous.

First of all, little is known about the flight control in real insects [2] [1]. It seems that there are two levels of control, as shown in Figure 2. At the low level the halteres, which are biological gyros, control the wing kinematics in order to keep the insect body in hovering condition. This type of control seems to be local and passive, since it always works to maintain the insect body in hovering position, no matter what the insect is trying to do. At the high level, the brain, stimulated by visual and physiological stimuli, works like a tactical planner, since it plans a trajectory based on its ultimate goal, like finding food or fighting an enemy. Differently from the halteres, this type of controller cannot modify directly the wings kinematics, but can only force the kinematics of the hanteres themselves. Consequently, the modified kinematics of the halteres indirectly alter the wing kinematics, thus forcing different flight modes, like forward motion or steering. This control architecture is probably an effective way to safely recover from large external disturbance to the body dynamics such as sudden wind gusts, and to limit the hazards of aggressive maneuvers. In fact, it has been observed that anytime the insect body attitude exceeds some safe threshold, the low level control locks out the higher level, recovers a stable hovering orientation, and only at that point, it releases the control to the higher level.

Another important difficulty for the design of a flight controller is the high uncertainty of the aerodynamic model for insect flight. Though the qualitative aspects of the aerodynamics involved are becoming clear [3] and though it is possible to safely estimate the mean forces averaged over a whole flapping cycle, no exact quantitative model is available for the instant forces, at present.

Moreover, the electromechanical model is highly nonlinear and the values of its parameters, like physical dimensions or elasticity, often present some variability from the nominal ones, due to the manufacturing [9].

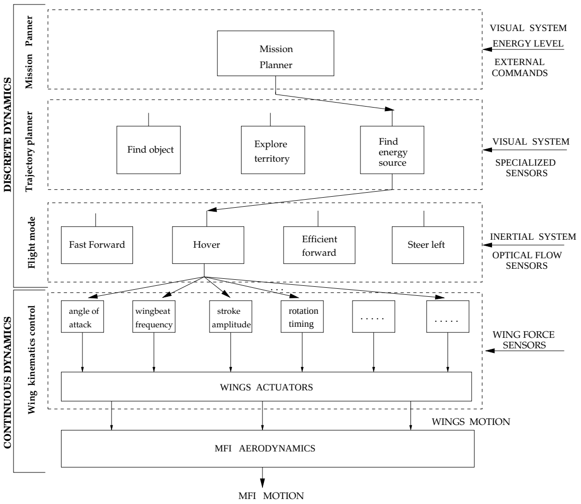

Figure 3: Design architecture for the control unit of the MFI.

Finally, the MFI is designed to show complex behaviors, such as planning articulated trajectories to accomplish a desired task. In order to solve these problems, it has been proposed an *hierarchical architecture* .

4 Control Unit Architecture

The hierarchical architecture, partially inspired by real flying insects and UAV research [4], decomposes the original global control problem into a multilevel set of simpler control problems. Moreover, thanks to this approach, the controllers on each level can be designed independently of those on higher levels, thus allowing the possibility to incrementally build more and more articulated control structures. Figure 3 shows the architecture proposed for the MFI control unit. It is possible to identify four main levels: the *mission planner*, the *trajectory planner*, the *flight mode controller* and the *wing kinematics controllers.*

Each of them has a specific task to accomplish and ,apart from the top level, is made of several controllers. Moreover, this architecture is built in a top-down structure, namely each level can interact with the lower level, but not vice versa. Finally, the two higher levels act like switchers, since they simply select one of the possible controller at the lower level. The bottom level, instead, continuously controls the wing kinematics. As a consequence the control unit presents a mixture of discrete and continuous dynamics.

4.1 Mission Planner

At the top level of the control unit there is the mission planner. Based on the sensory input from the visual system, the communication unit and the power supply unit, it selects the appropriate task. This structure not only allows the possibility to send commands to the MFI from a ground base, like "find object X" or "explore region Y", but also favors the development of autonomous decisions. For example, a low value of the energy level would force the selector to choose the behavior "find sunny spot" to recharge the MFI batteries.

4.2 Trajectory planner

The second level corresponds to a set of several complex behaviors labeled as *"explore", "find object", "find energy source", "return home", "escape from enemy"*, but only one of them can be active at a time. These are only some of the possible choices and others may be added.

Each of them is generated by a different controller, which is ultimately a trajectory planner since each behavior requires that the MFI fly from one place to another. No a priori policy is loaded into these controllers, just the ultimate goal is specified.

On the other hand, this level presents the problem to redefine those generic goals into a formal control framework.

Based on visual input information, each controller selects an appropriate sequence of flight modes from the lower level to follow the desired path. This sequence is not fixed but may be replanned to take account of noisy sensors and external disturbances in the body dynamics.

4.3 Flight mode controller

The third level contains the collections of all possible flight modes available to the MFI. In particular, they are *hovering, fast forward, power efficient forward, take off, land, steer left, steer right, move up, move down, move sideward*. As for the types of behaviors, the choice of flight modes is arbitrary, but it should be rich enough to generate any desired trajectory and motion.

Each flight mode is provided with a controller that takes the input signals from the inertial system, namely the accelerometers and gyroscopes, which provides an estimate for attitude and velocities of the MFI body. Based on this information, the controller chooses the appropriate values for the biokinematics parameters of the locomotory system, and they are updated every few wingbeat cycles, depending on the measured body dynamics.

The biokinematics parameters correspond to those particular features of wing motion that are responsible for flight. They are the wing beat frequency, f, the mean angle of attack, $\bar{\alpha}$, the stroke amplitude, Φ, the mean stroke angle, $\bar{\phi}$, the rotation timings at pronation and supination, t_{start} and t_{end}. Note that each wing have its own parameters which may differ from the other wing. Other biokinematics parameters can be considered, but those mentioned here, are sufficient to generate any flight mode [3].

The visual information is not used at this level for two main reasons, one related to real time control issues and the other to noise in the sensors signal.

First, the time required to process visual signals is longer than the time for inertial signal, and may be not fast enough to provide effective feedback, since the MFI is designed to have a wingbeat frequency of $100 - 200Hz$.

Secondly, the dynamics of any mechanical device is given, and controlled, by its angular and linear velocities. As a consequence, in almost any kind of feedback approach, it is necessary to estimate these velocities. Since the visual system can measure only displacements, their first derivative must be taken to obtain the corresponding velocities, thus giving rise to large estimation errors if the data are noisy. The inertial sensors, instead, measure directly the forces which are simply proportional to accelerations and velocities, thus providing a more robust estimation.

4.4 Wing kinematics controller

The bottom level consists on a single controller that generates the electrical signals for the actuators in order to generate the wings motion corresponding to the biokinematic parameters given by the flight mode controller. This controller receives input information from force sensors placed at the wings base. This input information can be directly used to estimate instantaneously the position and velocity of the wings, thus improving wing motion control through feedback.

5 A case study: the hovering flight mode

In this section, we focus on the design and implementation of the hovering flight mode, i.e. we want the MFI to stabilize about a fixed point in the space $[x_d, y_d, z_d]$, regardless of its orientation. To simplify the analysis we assume perfect state information, i.e. position, attitude and velocity of the MFI are accessible, and direct control of wings kinematics, i.e. we neglect the actuators dynamics. Although these are strong assumptions, the problem of stable hovering is still a difficult problem, since very little is known on real insect flight control algorithms.

5.1 Switching control approach

Similarly to aerial vehicles based on rotary wings, such as helicopter, flying insects control their flight by controlling their attitude and the magnitude of the vertical thrust [8]. However, differently from aerodynamic forces exerted on helicopter blades, aerodynamic forces on insect wings are highly time-varying along a wingbeat and are complex to model analytically. Moreover, the total force and torques on the MFI body are the result of the forces generated by the two wings. Therefore, techniques like feedback linearization [6] and robust linear control [5] are likely to fail, unless a better understanding of insect flight dynamics is available. Instead, we derive a very crude model for the aerodynamics and we implement a switching controller that, based on feedback error, selects among a set of wings motions. Each of this wings motion, can generate positive or negative net torques along one of the three main body axes.

As a first approximation, neglecting nonlinearity and coupling among variables, the dynamics of the MFI is given by:

$$\begin{aligned}
\ddot{\theta} &= I_p^{-1}\tau_p(t) \\
\ddot{\eta} &= I_r^{-1}\tau_r(t) \\
\ddot{\psi} &= I_y^{-1}\tau_y(t) \\
\ddot{x} &= m^{-1}F_v(t)\sin(\theta) \\
\ddot{y} &= m^{-1}F_v(t)\sin(\eta) \\
\ddot{z} &= m^{-1}(F_v(t)-g) \qquad (1)
\end{aligned}$$

where , $[I_p, I_r, I_y]$, are the moment of inertia of the pitch, roll and yaw axes, respectively, $[\tau_p, \tau_r, \tau_y]$, are the corresponding torques generated by the wings, m is the total mass of the insect, F_v is the mean aerodynamic vertical thrust, and g is the gravitational acceleration. If the orientation angles of MFI are small the position dynamics can be simplified as follows:

$$\begin{aligned}
x^{(4)} &= m^{-1}I_p^{-1}F_v(t)\tau_p(t) \\
y^{(4)} &= m^{-1}I_r^{-1}F_v(t)\tau_r(t) \\
z^{(2)} &= m^{-1}(F_v(t)-g) \qquad (2)
\end{aligned}$$

where the index in the parenthesis stands for the order of the derivative. Though this is a very crude approximation, it clearly evidences how position control can be achieved by controlling only three parameters, the pitch torque, τ_p, the roll torque,τ_r, and the vertical thrust, F_v. However, analytical expression for those parameters as a function of the wings motions is currently hopeless. Nevertheless, if we can find wing motions which are guaranteed to generate *on average* positive and negative roll and pitch torque, and sufficient vertical thrust to balance the gravitation force, we can further simplify Equations 2 as follows:

$$\begin{aligned}
x^{(4)} &= a\,u_x \\
y^{(4)} &= b\,u_y \\
z^{(2)} &= c\,u_z \qquad (3)
\end{aligned}$$

where $[a, b, c]$ are appropriate constants and the control inputs, $[u_x, u_y, u_z]$, can assume only two discrete values, $\{-1, +1\}$.

Before proceeding further, we want to remark two facts. The first fact is that the constant $[a, b, c]$ are in reality time-varying parameters that depend on the insect orientation and wing kinematics. The second, and most important, fact is that the control inputs $[u_x, u_y, u_z]$ cannot be chosen *continuously* and *independently.* In fact, the inputs can be changed only every wingbeat and they take positive or negative values only on average inside a wingbeat. As a consequence, it is hazardous,to design, based on the Equations 3 an aggressive controller , that asymptotically stabilizes the MFI position.

5.2 Wing kinematics for attitude control

Before presenting the control scheme, we need to generate a set of wings kinematics that provide the desired control input. Although little is still known about optimal wings motion in terms of power consumption and maneuverability, recent work done by Dickinson's group [3] have evidenced how a different mean angle of attack and the phase of rotation between the two wings, can generate asymmetrical instantaneous forces along a wingbeat, thus giving rise to positive or negative mean torque and forces. Intuitively, the mean angle of attack can modulate the magnitude of the aerodynamic forces on the wing: lift is maximal at an angle of attack of 45^o and decreases for different angles. The advanced or delayed phase of rotation respectively increases or decreases both lift and drag at the stroke reversals.

These findings suggest how to select wings kinematics that generate desired torque. Figure 4 shows *only some* of such kinematics. In the scenario (A) the wings have the same motion stroke angle motion and the phase of wings rotation, $\varphi(t)$, is advanced on the back of the insect body and delayed on the front, giving rise to a net pitch down torque. In the scenario (B) the wings have the same stroke angle motion, $\phi(t)$, but the phase of rotation for the left wing is advanced on the back of the insect body and delayed on the front, and it is opposite on the right wing, giving rise to a net clockwise yaw torque. In the scenario (C) the wings have the same stroke angle motion, $\phi(t)$, and phase of rotation, but the right wing has a smaller mean angle of attack, giving rise to a net right roll torque.

In particular, we parameterize the wing kinematics as follow:

$$\begin{aligned}
\phi_r(t) &= \phi_l(t) = \Phi\sin(2\pi f\,t) \\
\varphi_r(t) &= \Upsilon_r\left[\sin(2\pi f\,t) + \nu_r\sin(4\pi f\,t)\right] \\
\varphi_l(t) &= \Upsilon_l\left[\sin(2\pi f\,t) + \nu_l\sin(4\pi f\,t)\right] \qquad (4)
\end{aligned}$$

where f is the wingbeat frequency, Φ is the maximal stroke amplitude, $\varphi = \frac{\pi}{2} - \alpha$, is the rotation angle, defined as the angle between the wing profile

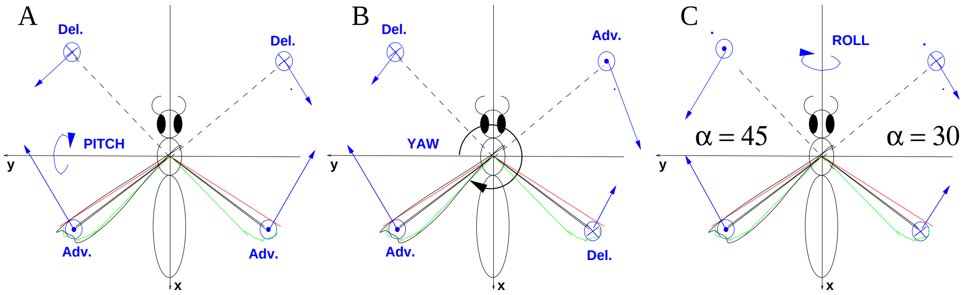

Figure 4: Open loop wing motions that can generate: (A) pitch, (B) yaw, (C) roll torques. The arrows represent the instantaneous aerodynamic forces acting on the wings. The circles with a cross or with a dot correspond, respectively to the perpendicular component of the force entering or exiting the stroke plane. `Adv.` and `Del.` stand for advanced and delayed rotation, respectively.

and the vertical, Υ is the maximal rotation angle and the subscript r and l stand for right and left wing, respectively. According to this parameterization, scenario (A) is given by, $\Upsilon_r = \Upsilon_l = \frac{\pi}{4}$, and $\nu_r = \nu_l = 0.4$; scenario (B) is given by, $\Upsilon_r = \Upsilon_l = \frac{\pi}{4}$, $\nu_r = 0.4$ and $\nu_l = -0.4$; scenario (C) is given by, $\Upsilon_l = \frac{\pi}{3}$, $\Upsilon_r = \frac{\pi}{4}$, and $\nu_r = \nu_l = 0$.

Table 1 reports the mean wrench over a complete wingbeat, due to wing aerodynamic forces for several scenarii, and it evidences the components that are most largely affected. As expected, the parameterization of the wing kinematics given in Equations (4), generates wing motions that can control body attitude and vertical thrust. This set of wing kinematics, although arbitrary and limited, is sufficient to control the MFI attitude and vertical thrust and to stabilize hovering or other flight modes.

5.3 Feedback control and simulations

The feedback law is based on global errors, calculated at the end of every wingbeat, which are defined as a linear combination of position and velocity errors averaged over a wingbeat:

$$\begin{aligned} e_x &= K_1\bar{\dot{\theta}} + K_2\bar{\theta} + K_3\bar{\dot{x}} + K_4(\bar{x} - x_d) \\ e_y &= K_1\bar{\dot{\eta}} + K_2\bar{\eta} + K_3\bar{\dot{y}} + K_4(\bar{y} - y_d) \\ e_z &= K_5\bar{\dot{z}} + K_6(\bar{z} - z_d) \qquad (5) \end{aligned}$$

where the constants $\{K_i\}$ are empirically chosen. According to these arrows the control inputs are

$$\begin{aligned} u_x &= -sign(e_x) \\ u_y &= -sign(e_y) \\ u_z &= -sign(e_z) \qquad (6) \end{aligned}$$

where the function $sign(s)$ returns 1 if s is positive and -1 otherwise.

To stabilize the hovering flight mode, we have designed a fixed scheduler that selects, at the end of each wingbeat, one of the possible wing kinematics presented in Table 1 as a function of the control inputs $[u_x, u_y, u_z]$. The control algorithm can be summarized as followed:

$$\begin{aligned} \nu_r &= \nu_l = 0.4\,u_x \\ \gamma &= 1.1 - 0.1\,u_z \\ \Upsilon_r &= \gamma\left(\frac{7}{12} + \frac{1}{12}u_y\right)\pi \\ \Upsilon_l &= \gamma\left(\frac{7}{12} - \frac{1}{12}u_y\right)\pi \qquad (7) \end{aligned}$$

The proposed control algorithm is simulated with VIFS [7] and the results are shown in Figure 5. It succeeds in stabilizing hovering, hoverer, not surprisingly, the MFI evidences a chattering motion about the desired fixed position. This effect is mainly due to the fact that nonlinearity and coupling among dynamic variables have been neglected. Moreover, since attitude stabilization was not included, we notice a drift in the yaw orientation, which, however, does not affect stability.

In order to simplify the model, we do not take into account external disturbances such as wind gusts and rain. However, our goal is to design a controller with a large basin of stability, such that the MFI is able to recover the hovering flight mode even from an upside-down position. As a consequence, albeit wind gusts and rain may degrade flight performance, they should not compromise the overall behavior of the MFI. We will address this issue in future work.

6 Conclusion

In this paper we have presented a hierarchical control system architecture for flight control and management. This scheme takes inspiration from real flying insects and greatly simplifies implementation and performance analysis of control algorithms. Finally we have proposed and simulated a stabilizing controller for the hovering flight mode. Although

N	Kinematic Parameters				Wrench					
	r_l	r_r	v_l	v_r	F_x	F_y	F_z	τ_{roll}	τ_{pitch}	τ_{yaw}
1	45	45	0.4	-0.4	-0.0215	-0.0015	1.2072	0.0208	-0.0939	**-0.7202**
2	45	45	-0.4	0.4	-0.0215	0.0015	1.2072	-0.0208	-0.0939	**0.7202**
3	45	45	0.4	0.4	-0.1384	0	1.2049	0	**-0.7186**	0
4	45	45	-0.4	-0.4	-0.0965	0	1.2117	0	**0.5292**	0
5	45	45	0	0	-0.0226	0	**1.2324**	0	-0.1009	0
6	60	60	0	0	-0.0479	0	**0.7939**	0	-0.195	0
7	45	60	0	0	-0.035	-0.0403	1.0135	**-0.7452**	-0.147	0.0652
8	60	45	0	0	-0.035	0.0403	1.0135	**0.7452**	-0.147	0.0652

Table 1: Mean Wrench Map over a complete wingbeat: forces are expressed in (ηN) and torque in $(\eta N \cdot mm)$

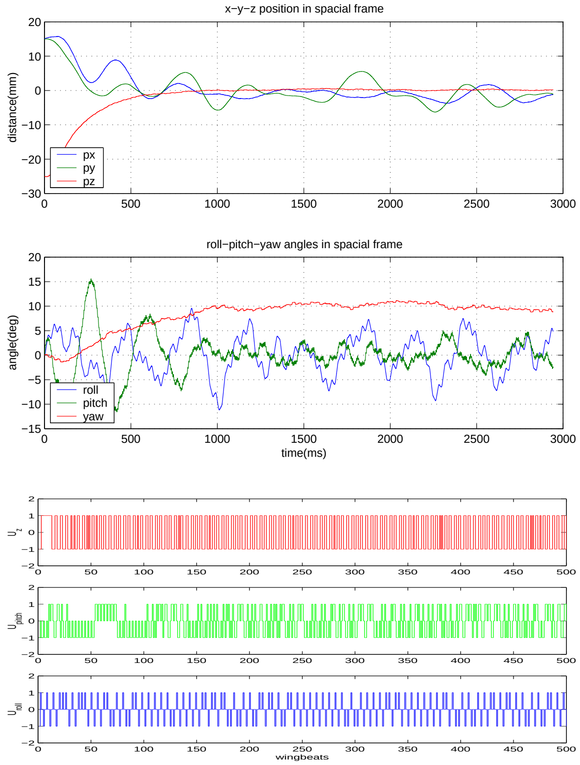

Figure 5: Simulated MFI dynamics resulting from hovering controller. From top to bottom: position, orientation and control inputs $[u_x, u_y, u_z]$

the hovering controller is far from been optimal, the main point that emerges from this control approach is that a small set of wing kinematics might be sufficient to generate all possible flight modes. In fact, the key point for designing any of those modes, is the capability to control the MFI attitude, as it is done for hovering. Future research must address the problem of wing kinematics optimization in terms of force generation and power consumption, and the design of controllers that take into account actuators and sensors dynamics. Finally, we will verify our control model on the MFI prototype, which is currently being developed. Initially, the controller will be implemented using off-line sensors and processing unit, but, eventually, controller unit and sensors will be integrated into a single chip installed in the MFI device.

7 Acknowledgments

The authors thank R. Fearing, M.Dickinson, S.Sane for helpful discussions and insights.

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

[4] T. J. Koo, D. H. Shim, O. Shakernia, B. Sinopoli, Y. Ma, F. Hoffmann, and S. Sastry. Hierarchical hybrid system design on berkeley uav. In *Submitted to International Aerial Robotics Competition, Richland, Washington, USA*, 1998.

[5] R. Sanchez-Pena and M. Sznaier. *Robust Systems: Theory and Applications*. Wiley, 1998.

[6] S.S. Sastry. *Nonlinear Systems; Analysis, Stability, and Control*. Springer Verlag, 2000.

[7] L. Schenato, X.Deng, W.C. Wu, and S.S. Sastry. Virtual insect flight simulator (VIFS): A software testbed for insect flight. In *IEEE International Conference on Robotics and Automation*, 2001.

[8] D. H. Shim, H. J. Kim, and S. Sastry. Control system design for rotorcraft-based unmanned aerial vehicles using time-domain system identification. In *IEEE International Conference on Control Applications*, Anchorage, 2000.

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