Attitude Control for a Micromechanical Flying Insect Including Thorax and Sensor Models *

X. Deng L. Schenato S.S. Sastry

Department of EECS, University of California, Berkeley, CA 94720
{*xinyan, lusche, sastry*}*@eecs.berkeley.edu*

Abstract

This paper describes recent development on the design of the flight simulation and control system for a Micromechanical Flying Insect (MFI). High level attitude control is considered. Compared to our previous work, here we include the recently developed dynamical model for the thorax actuators, and various sensor models to close the control loop. Specifically, a new wing kinematic parameterization method was developed to generate feasible wing motions based on the available thorax model. Compared to our previous method, this parameterization schemes ensures the boundedness and smoothness of the thorax input torques while decoupling the average roll, pitch, yaw torques in the body frame. A nominal state-space LTI model in hover was identified through linear estimation and a LQG controller was designed. Sensor models such as haltere, magnetic compass, and ocelli were included inside the closed loop system and the simulations shows stable hovering and steering maneuvers.

1 Introduction

Micro aerial vehicles (MAVs) have drawn a great deal of attention in the past decade due to the quick advances in microtechnology and several groups have worked on MAVs based on fixed and rotary wings [7]. However, flapping flight provides superior maneuverability that would be beneficial in obstacle avoidance and for navigation in small spaces. Therefore, the UC Berkeley Micromechanical Flying Insect (MFI) project uses biomimetic principles to develop a flapping wing MAV that will be capable of sustained autonomous flight [4],[14]. In this paper we address the attitude control problem for an MFI, including realistic actuators and sensor models.

Based on the recently developed thorax model [1], low level wing motion parameterization and open loop control was considered. As a starting point, the analysis is based on the linear model of the thorax. The input torques are constrained only in their amplitudes, and they are assumed to take form of the sinusoidal waves and their higher harmonics.

*This work was funded by NSF KDI ECS 9873474, ONR MURI N00014-98-1-0671, and DARPA.

The original wing kinematic parameterization method in our previous works ([2], [9], [3]) is no longer feasible due to the discontinuities at the end of consecutive wingbeats, which causes the thorax input torques to demonstrate jumps and large out-of-limit amplitudes. In this work, we have to find a new wing motion parameterization, which keeps the thorax inputs smooth and bounded, while decoupling roll, pitch, yaw torques.

Based on the new parameterization method, we adopt our previous identification and control scheme [3] to find a new discrete-time linear time-invariant (LTI) model which captures the main dynamic features of the MFI near hover and design a LQG controller which stabilizes and provides setpoint tracking. In addition, the thorax model, including torque and bandwidth constraints, and various sensor models are included inside the closed loop system.

2 Insect Dynamics Model

The attitude dynamics of a flapping insect can be written as follows [9]:

$$\begin{aligned} \dot{R} &= R\hat{\omega}^b \\ \dot{\boldsymbol{\omega}}^b &= J_b^{-1}(\boldsymbol{\tau}^b - \boldsymbol{\omega}_b \times J_b\boldsymbol{\omega}_b) \\ \hat{\omega}^b &= \begin{bmatrix} 0 & \omega_z{}^b & -\omega_y{}^b \\ -\omega_z{}^b & 0 & \omega_x{}^b \\ \omega_y{}^b & -\omega_x{}^b & 0 \end{bmatrix} \end{aligned} \tag{1}$$

where $\boldsymbol{\omega}^b = [\omega_x{}^b\, \omega_y{}^b\, \omega_z{}^b]^T$ is the angular velocity vector of the insect body relative to the body frame B, $\boldsymbol{\tau}^b \in \mathbb{R}^3$ is aerodynamic torque vector relative to the body frame B attached to the center of mass of the insect body, $J_b \in \mathbb{R}^{3\times3}$ is the moment of inertia of the insect body relative to the body frame B, and $(a \times b)$ is the cross product of the vectors (a, b). To simplify the notation, we drop the superscript b from equations, implicitly assuming that all quantities are measured relative to the body frame B. The matrix $R \in SO(3) = \{R \in \mathbb{R}^{3\times3} : R^TR = I, \det R = +1\}$ is the rotation matrix representing the orientation of the insect body frame B relative to the fixed frame A. In particular, let $\mathbf{v}^b = [x_b\, y_b\, z_b]^T$ and $\mathbf{v}^a = [x_a\, y_a\, z_a]^T$ the coordinates of a vector $\mathbf{v} \in \mathbb{R}^3$ relative to the body frame B and the fixed frame A, respectively.

Then, these coordinates are linked together by the linear transformations $\mathbf{v}_a = R\mathbf{v}_b$ and $\mathbf{v}_b = R^T\mathbf{v}_a$. The above dynamics has been implemented into the Virtual Insect Flight Simulator (VIFS) [10], a software testbed for flapping flight simulations.

3 Sensors

In this section we briefly describe the three sensors, the ocelli, the magnetic compass and the halteres, used to estimate the three insect body angles (roll,pitch,yaw) relative to the fixed frame, and the corresponding angular velocities. The ocelli can be used to estimate the roll and pitch angles, the magnetic compass to estimate the yaw, and the halteres to estimate the three angular velocities. In this paper we report only the major results and details are presented in [6]. These sensors are currently being developed, and preliminary results are very promising [12].

3.1 Ocelli

The ocelli are a sensory system present in many flying insects. This system comprises of three wide angle photoreceptors placed on the head of the insect. They are oriented in such a way that they collect light from different regions of the sky. The ocelli are believed to play an important role in attitude stabilization in insect flight by comparing the intensity of light measured by the different photoreceptors [11].

Inspired by real insects, we describe a biomimetic ocelli-like system composed by four photoreceptors rather than three, as observed in nature. Although all the results in this paper can be extended to a three-photoreceptor ocelli, we prefer to present them relative to a four-photoreceptor configuration since the design is simplified and results are more intuitive. All the results are based on the assumption that the light intensity is a monotonic decreasing function of the angle between the light source and the photoreceptor.

The ocelli sensory system is modeled as four ideal photoreceptors, called P_1, P_2, P_3, P_4, fixed with respect to the body frame B. These photoreceptors collect the light intensity from a different region of the sky. They are oriented symmetrically such that they have the same latitude and their axes intersect the sky sphere forming an imaginary pyramid, whose vertex is placed at the center of the insect head. Every photoreceptor collects light from a conic region A_i around its ideal orientation P_i as shown in Figure 1a. The measurements from the photoreceptors are simply subtracted pairwise and these two signals are the output from the ocelli:

$$y_1^o = I(P_1) - I(P_2), \qquad y_2^o = I(P_3) - I(P_4) \tag{2}$$

where $I(P_i)$ is the output from the i-th photodiode. Without loss of generality we assume that the light source is placed at position $X = (0, 0, 1)$ relative to the fixed frame. If the output of a photodiode is a *monotonic decreasing* function of its latitude relative to the light source, we showed that the following proposition holds true [6]:

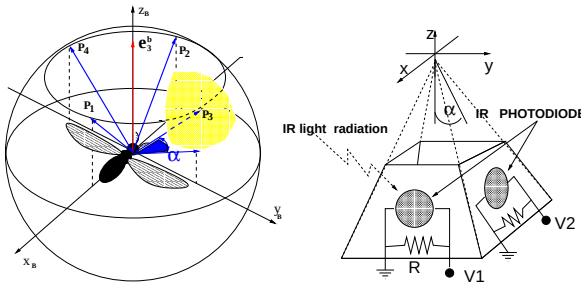

Figure 1: (a) Graphical rendering of ocelli present on flying insects. Four photoreceptors, referenced as P_1, P_2, P_3, P_4, collect light from different regions of the sky. The shadowed area represents this region for photoreceptor P_3. (b) Schematic of ocelli biomimetic sensor.

Proposition 1. *If the photoreceptor output is a monotonic decreasing function of its latitude θ_i relative to the light source, then the signals y_1^o and y_2^o defined in Equations (2) always satisfy the following conditions:*

$$k_{min} r_{32} \le y_1^o \le k_{max} r_{32} \tag{3}$$

$$k_{min} r_{31} \le y_2^o \le k_{max} r_{31} \tag{4}$$

$$r_{33} \to 1 \Longrightarrow y_1^o \to k_o r_{32}; y_2^o \to k_o r_{31} \tag{5}$$

where $0 < k_{min} < k_o < k_{max}$ are constant, and r_{ij} is the $i-j$ entry of the rotation matrix R.

Therefore, it is evident that the outputs from the ocelli can be used as an estimate of the position of the ocelli reference frame relative to the light source, since for small deviations from the equilibrium, *i.e.* $R \approx I_{3\times3}$, r_{31} and r_{32} correspond to the roll and pitch angles, respectively. More general results for attitude stabilization are given in [6]. A prototype for the ocelli system, which is currently being developed seems to confirm the mathematical model results [12].

3.2 MEMS Magnetic Compass

Attitude control of MFI body requires a set of sensors that can estimate its orientation relative to a desired frame. The ocelli sensory system provides a means to reorient the insect body towards a specific direction, however its heading remains arbitrary. Since heading it is fundamental for forward flight and maneuvering, we propose to solve this problem by adding a MEMS magnetic compass. This magnetic sensor can estimate MFI heading based on the terrestrial geomagnetic field. The MEMS compass is a "U-shaped" suspended structure as shown in Figure 2. Electric current flows across this structure, interacting with the terrestrial geomagnetic field. The Lorentz forces acting on the electric currents generates a force given by $\mathbf{F}_l = L\mathbf{i} \times \mathbf{B}$, where $\mathbf{F}_l$ is the total force at the tip of

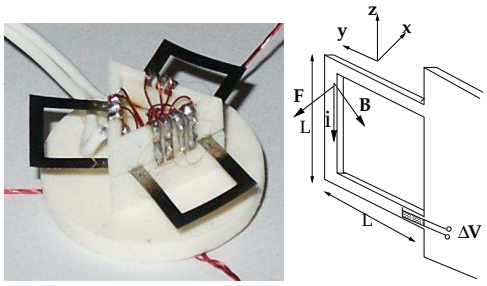

Figure 2: (a) Photo of a MEMS magnetic sensor prototype (b) Schematic of a magnetic compass.

the cantilever, L is the length of one side of the cantilever, $\mathbf{i}$ is the total current, and $\mathbf{B}$ is the terrestrial electromagnetic field. The deflection of the cantilever is sensed at the base by a strain gage. The deflection is proportional to the force perpendicular to the cantilever,*i.e.* $F_c = \mathbf{F}_l \cdot \mathbf{n}$, where $\mathbf{n}$ is the sensing direction of the strain gage. Therefore, the output from the strain gage can be used to estimate the heading of the MFI when the magnetic sensor is attached to the MFI body. In fact, without loss of generality, assume that the earth magnetic field is oriented along the x-axis of the fixed frame, *i.e.* $\mathbf{B} = [0\ 0\ B]^T$ relative to the fixed frame. Also, we orient the compass such that the current $\mathbf{i} = [0\ 0\ \ -i]^T$ and the sensing direction $\mathbf{n} = [1\ 0\ 0]^T$ relative to the body frame (see Figure 2). According to these definitions, the signal measured by the staring gage is given by:

$$\begin{aligned} y^c &= F_c = L\left([0\ 0\ \ -i] \times (R^T[0\ 0\ B]^T)\right)[1\ 0\ 0]^T \\ &= LiB\ r_{12} = k_c\ r_{12} \end{aligned} \tag{6}$$

where we use the fact that the coordinates of the earth magnetic field relative to the body frame is given by $v^b = R^T[0\ 0\ B]^T$. For small deviations from the equilibrium, *i.e.* $R \approx I_{3\times 3}$, r_{12} correspond to the yaw angle, thus providing an estimate for the heading.

3.3 Halteres

Biomechanical studies on insect flight revealed that insects use structures, called halteres, to measure body rotations via gyroscopic forces [5]. The halteres of a fly resemble small balls at the end of thin sticks. During flight the two halteres beat up and down in non-coplanar planes through an angle of nearly 180^o anti-phase to the wings at the wingbeat frequency. This non-coplanarity of the two halteres is essential for a fly to detect rotations about all three turning axes [8]. As a result of insect motion and haltere kinematics, a complex set of forces acts on the halteres during flight: inertial, angular acceleration, centrifugal, Coriolis, and gravitational forces. However, by taking advantage the peculiar characteristics (frequency, modulation, and phase) of the Coriolis signals on the left and right halteres, a demodulation scheme has been proposed to decipher roll, pitch, and yaw angular velocity

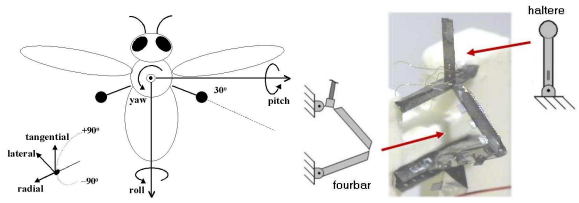

Figure 3: (a) Schematic of enlarged halteres of a fly. (b) Photo of the completed haltere.

[13]. It is shown in [6] that the halteres outputs are almost equivalent to the following smoothed version of the insect angular velocities:

$$\begin{aligned} y_1^h(t) &= a\int_{t-T}^{t}\omega_x(\tau)\mathrm{d}\tau = k_{o1}\,\bar{\omega}_x(t) \\ y_2^h(t) &= b\int_{t-T}^{t}\omega(\tau)\mathrm{d}\tau = k_{o2}\,\bar{\omega}_y(t) \\ y_3^h(t) &= c\int_{t-T}^{t}\omega(\tau)\mathrm{d}\tau = k_{o3}\,\bar{\omega}_z(t) \end{aligned} \tag{7}$$

where T is the period of oscillation of the halteres, $a, b, c, k_{o1}, k_{o2}, k_{o3}$ are constants and $\bar{\omega}_i$ is the mean angular velocity of the insect over a period of oscillation of the halteres.

4 Linear Thorax Model

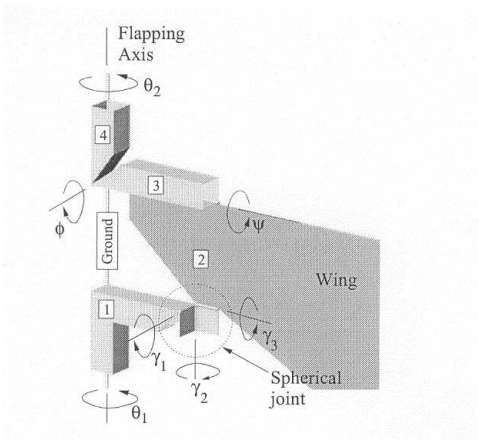

Figure 4: Wing-Thorax structure. Courtesy of [1]

Each wing is controlled by the thorax, a complex trapezoidal structure actuated by two piezoelectric actuators at its base, as shown in Figure 4. A complete nonlinear model for the thorax, developed in [1], can be written as follows

$$M\begin{bmatrix}\ddot{\theta}_2\\ \ddot{\alpha}\end{bmatrix} + B\begin{bmatrix}\dot{\theta}_2\\ \dot{\alpha}\end{bmatrix} + K\begin{bmatrix}\theta_2\\ \alpha\end{bmatrix} + \begin{bmatrix}0\\ f(\dot{\alpha})\end{bmatrix} = T\begin{bmatrix}u_1\\ u_2\end{bmatrix} \tag{8}$$

where $f(\dot{\alpha}) = \frac{1}{2}m'_{\omega,2}(\dot{\alpha})^2$, θ_2 is the leading edge flapping angle from the four bar mechanism, $\alpha = \theta_1 - \theta_2$ is the phase difference between the four bar output angles, u_1 and u_2 are the control input torques to the

actuators, M and B are the inertia and damping matrices, which are assumed to be constant. However, parameters in K and T matrices includes some slowly time varying terms, and the control inputs (u_1, u_2) are limited to $10\mu Nm$ by physical constraints.

The relation between the state variables in Equation (8) and the wing motion variables (stroke angle ϕ, rotation angle φ) can be approximated as $\phi = \theta_2$ and $\varphi = 2\alpha$. Based on Equation (8), with a change of variables, neglecting the nonlinear components, we can derive the linear actuator model as

$$M_0 \begin{bmatrix} \ddot{\phi} \\ \ddot{\varphi} \end{bmatrix} + B_0 \begin{bmatrix} \dot{\phi} \\ \dot{\varphi} \end{bmatrix} + K_0 \begin{bmatrix} \phi \\ \varphi \end{bmatrix} = T_0 \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (9)$$

where M_0, B_0, K_0, and T_0 are constant matrices calculated from the data provided in [1].

Equation (9) is a stable linear MIMO system, its steady state solution at a particular frequency can be calculated through complex matrix operations. To generate sufficient lift to sustain the insect, the stroke and rotation angles must be follow a trajectory which mimic insect wings flapping motions, such as $\phi = \frac{\pi}{3}\cos(wt)$ and $\varphi = \frac{\pi}{4}\sin(wt)$, where $w = 2\pi f$ and $f = 150Hz$ is the wingbeat frequency. The desired input torques to the actuators that, in *steady state*, give rise to the wing trajectory above can be calculated from

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} G_{11}(jw)G_{12}(jw) \\ G_{21}(jw)G_{22}(jw) \end{bmatrix}^{-1}_{w=150*2\pi} \begin{bmatrix} \frac{\pi}{3}e^{j\pi/2} \\ \frac{\pi}{4} \end{bmatrix} \quad (10)$$

where $G(jw)$ is the frequency domain system transfer function matrix, the resulted steady state inputs are $u_1 = 5.64\sin(wt - 2.67)$, $u_2 = 6.48\sin(wt - 2.47)$. These inputs drive the wings to their steady state trajectory within 2-3 wingbeats, when the wings are started from rest.

5 Wing Motion Parameterization

In order to decouple roll, pitch, yaw toques, we need to generate different wing motions through proper kinematic parameterization schemes, while still keeping the input torques within limits. Due to discontinuities at consecutive wingbeats, our previous parameterization scheme presented in [3] is no longer feasible, since the corresponding control inputs demonstrate large amplitude jumps between wingbeats. In order to keep the control inputs smooth and bounded, proper wing motions need to be designed to ensure continuity at the end of wingbeats, while still preserving maneuverability during each wingbeat. Based on qualitative analysis of wing aerodynamics and thorax dynamics, one feasible parameterization was found by adding one additional term, $g(t)$ in the stroke angle profile to change its mean amplitude, and in the rotation angle profile to change the timing of rotation at the end of the first half-stroke. Since we want to keep the input torque continuous at the beginning and end of each wingbeat, the additional term with its first

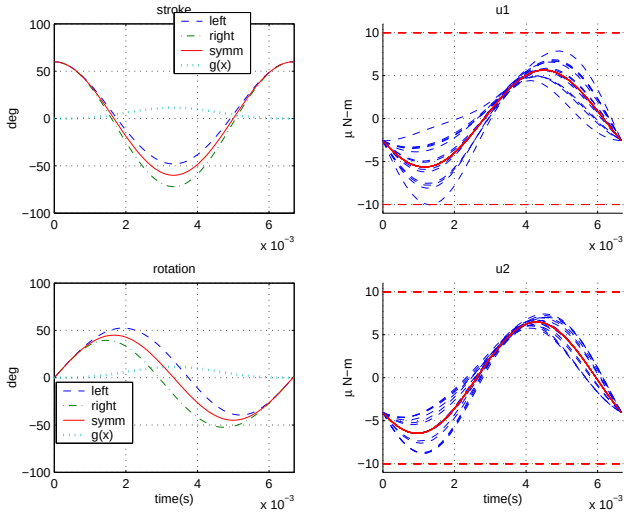

Figure 5: (a) Left plots: Wing motions results from $(\gamma, \alpha_l, \alpha_r) = (1, 1, -1)$. The solid line traces correspond to the symmetric motion $(\gamma, \alpha_l, \alpha_r) = (0, 0, 0)$. (b) Right plots: Input torques family for all 9 possible permutations of the parameters $(\gamma, \alpha_l, \alpha_r) \in \{-1, 1\}$. The solid line is obtained by setting these parameters to zero. The horizontal lines correspond to the input saturation limits.

and second derivative should be zero at the beginning and end of wingbeats, *i.e.* $g(0) = g(T) = \dot{g}(0) = \dot{g}(T) = \ddot{g}(0) = \ddot{g}(T) = 0$, where T is the wingbeat period. A possible parameterization that satisfies such constraints is the following:

$$\begin{aligned} g(t) &= \frac{3}{10}\sin(\frac{1}{2}wt) - \frac{1}{10}\sin(\frac{3}{2}wt) \\ \phi_l(t) &= \frac{\pi}{3}\left(\cos(wt) + \frac{1}{2}\gamma g(t)\right) \\ \phi_r(t) &= \frac{\pi}{3}\left(\cos(wt) - \frac{1}{2}\gamma g(t)\right) \\ \varphi_l(t) &= \frac{\pi}{4}\left(\sin(wt) + \alpha_l g(t)\right) \\ \varphi_r(t) &= \frac{\pi}{4}\left(\sin(wt) + \alpha_r g(t)\right) \end{aligned} \quad (11)$$

where $(\gamma, \alpha_l, \alpha_r) \in [-1, 1]$ are the tunable kinematic parameters, and the subscript r and l stand for right and left wing, respectively. The parameters α_l and α_r are strongly related to wing flip timing at the end of first half stroke: a positive value corresponds to delaying the wing rotation at the upstroke and a negative value advancing the rotation, a null value results in a symmetric wing rotation at both the half-strokes. The parameter γ modifies the mean stroke angle amplitudes of the wing: a positive value corresponds to a larger mean stroke angle amplitude on the left wing, a negative value to the opposite, and a zero value to equal stroke angle amplitudes.

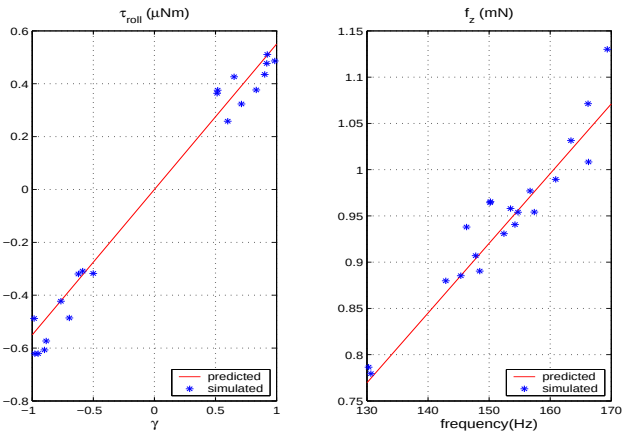

Figure 6: Average roll torque, τ_{roll} (left) as a function of the parameter γ. Mean lift, f_z (right) as a function of the wingbeat frequency. The simulated points are obtained by randomly select the values for the parameters $(\gamma, \alpha_l, \alpha_r)$.

As an example, Figure 5a shows both left and right wing motions resulted from choosing the parameters as $\gamma = 1$, $\alpha_l = 1$, and $\alpha_r = -1$. Also plotted is the additional term, $g(t)$, which serves as amplitude modulation for the stroke and rotation angles. We can see that with this parameterization scheme, continuity is ensured at the end of wingbeat, while different wing motion can be manipulated in the middle of each wingbeat by varying the three parameters $(\gamma, \alpha_l, \alpha_r)$.

With the designed wing motion trajectory as the output from the thorax model, we calculate corresponding desired input torques by substitution :

$$\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = T_0^{-1}\left(M_0 \begin{bmatrix} \ddot{\phi}(t) \\ \ddot{\varphi}(t) \end{bmatrix} + B_0 \begin{bmatrix} \dot{\phi}(t) \\ \dot{\varphi}(t) \end{bmatrix} + K_0 \begin{bmatrix} \phi(t) \\ \varphi(t) \end{bmatrix} \right) \tag{12}$$

This approach is equivalent to feed-forward control of wings trajectory within a single wingbeat. Figure 5b plots in detail the family of all control inputs that we can obtaining by varying the parameters $(\gamma, \alpha_l, \alpha_r)$ between $(-1, 1)$. We can see that the inputs are always bounded by $\pm 10 \mu Nm$. To test weather the above parameterization scheme can be used to generate desired aerodynamic torques to steer the insect body, while producing sufficient lift to sustain the insect, we found the empirical map from wing kinematic parameters to the average body torques generated over one wingbeat through VIFS. Figure 6 and Figure 7 show the simulation results. The empirical map can be written as follows.

$$\begin{aligned} \bar{\tau}_\varphi &= f_1(\alpha_r, \alpha_l, \gamma) = c\gamma + \delta_\varphi \\ \bar{\tau}_\theta &= f_2(\alpha_r, \alpha_l, \gamma) = a_{11}\alpha_l + a_{12}\alpha_r + \delta_\theta \\ \bar{\tau}_\psi &= f_3(\alpha_r, \alpha_l, \gamma) = a_{21}\alpha_l + a_{22}\alpha_r + \delta_\psi \end{aligned}$$

where the coefficients $a_{11}, a_{12}, a_{21}, a_{22}, c$ are constants, and the errors $\delta_\eta, \delta_\theta, \delta_\psi$ are bounded. It is seen that

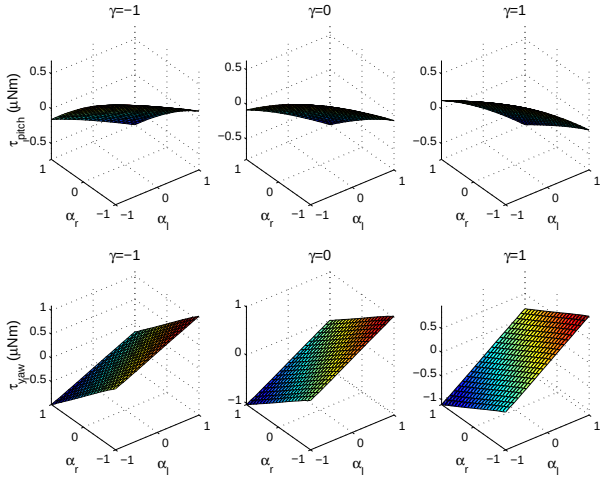

Figure 7: Average pitch and yaw torques as a linear map of the parameters α_l and α_r.

f_2 and f_3 can be approximated with linear functions of α_l and α_r, while f_1 with a linear function of γ only. Therefore the mean torques are decoupled for the identification and control purposes. One more advantage of this scheme is the linearity of all the functions. Consequently, given the desired mean torques for the next wingbeat, we can obtain the desired wing kinematic parameters through the linear inverse maps. Once the kinematic parameters are known, the wings trajectory is obtained by substituting their value into Equations (11), and the corresponding control inputs by substituting the wings trajectory into Equations (12). In other words, this means that we know the desired input torques that we need to apply to the actuators to generate the desired mean body torques in the next wingbeat.

As with our previous work, in order to tune the average vertical thrust to balance the insect's weight, it is sufficient to modulate the wing flapping frequency. Simulation results in Figure 6 shows the average mean lift as an approximate linear function of frequency.

6 Model Identification

The analysis in the previous section provide us with a torque decoupling scheme, together with a set of feasible control inputs (wing kinematic parameters). Since we are interested in the insect dynamics close to the hovering regime where angular deviations and angular velocities are small, we linearize and average the dynamics of Equations (1) within a single wingbeat. Therefore, for the purpose of designing a simple feedback controller, we approximate the continuous-time nonlinear system, with a discrete-time LTI model in the following form:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k) \\ y(k) &= x(k) + v(k) \end{aligned} \tag{13}$$

where $x = [\bar{\phi}\ \bar{\theta}\ \bar{\psi}\ \dot{\bar{\phi}}\ \dot{\bar{\theta}}\ \dot{\bar{\psi}}]^T$ is the vector of average roll, pitch, yaw angles and angular rates over one wingbeat; w represents the time varying component which appears as an external disturbance to the linear model [9]; $y = [\bar{y}_2^o\ \bar{y}_1^o\ \bar{y}^c\ \bar{y}_1^h\ \bar{y}_2^h\ \bar{y}_3^h]^T$ is the vector of measured outputs, with additional measurement noise v; C and D matrices are set to be identity and zero matrices, respectively; and $u = [u_1\, u_2\, u_3]^T = [\gamma\, \alpha_l\, \alpha_r]^T$ are the control inputs, *i.e.*, the wing kinematic parameters. For small angles and angular velocities, we have $x = [\bar{r}_{32}\ \bar{r}_{31}\ \bar{r}_{21}\ \bar{\omega}_x\ \bar{\omega}_y\ \bar{\omega}_y]^T$, where r_{ij} is the $i-j$ entry of the rotation matrix R.

The matrices $[A, B]$ can be obtained directly from MFI morphological parameters such as mass, moment of inertia, center of mass, etc. However, these parameters are difficult to obtain in practice. Moreover, this approach cannot model the effect of the time varying part of the aerodynamic forces. An alternative approach is to run a large number of experiments and record the pair $[y(k), u(k)]$, and then find the matrices $[A, B]$ that best fit the data. The model identification problem can be recasted into a least square solution to an overdetermined set of linear equations as $Ez = d$, where $z = [a_{11}, ..., a_{66}, b_{11}, ..., b_{63}]^T$ is the vector of system parameters to be estimated, a_{ij} and b_{ij} are the $i-j$ entries of the matrices A and B respectively, and $E = E\,(y(\cdot), u(\cdot))$ and $d = d\,(y(\cdot))$ are matrices whose elements consists of the experiment data. The least square solution which minimizes the norm of the error $\| e \|^2 = \| d - Az \|^2$ is given by $z = E(E^TE)^{-1}E^Td$.

The experiments were performed on the Virtual Insect Flight Simulator (VIFS), developed by the authors to provide a software testbed for insect flight [10]. The experimental data were generated with random inputs and initial conditions near the equilibrium.

Estimation of the system parameters and further investigation into the system dynamics in Equation (13) results in the following approximate parameter structures:

$$A = \begin{bmatrix} I_{3\times3} & TI_{3\times3} \\ A_{21} & A_{22} \end{bmatrix} \quad B = \begin{bmatrix} 0_{3\times3} \\ B_{22} \end{bmatrix}$$

where T is the wingbeat period. The parameters of the state-space realization in Equation (13) consist of the elements of the A_{21}, A_{22} and B_{22} matrices. As expected, it was found that A_{22} matrix is close to an identity matrix. The structure of the B_{22} matrix also reflects our previous torque decoupling scheme through wing kinematic parameterization of Equation (13).

To check the ability of the identified model to predict the behavior of the MFI in hover, the model was simulated for a consecutive 50 wingbeats, and is compared to the results from the simulator. Figure 8 plots the mean angle and angular rates predicted by the LTI model together the simulation results from VIFS. It can be seen that the predicted values match the simulated ones very well.

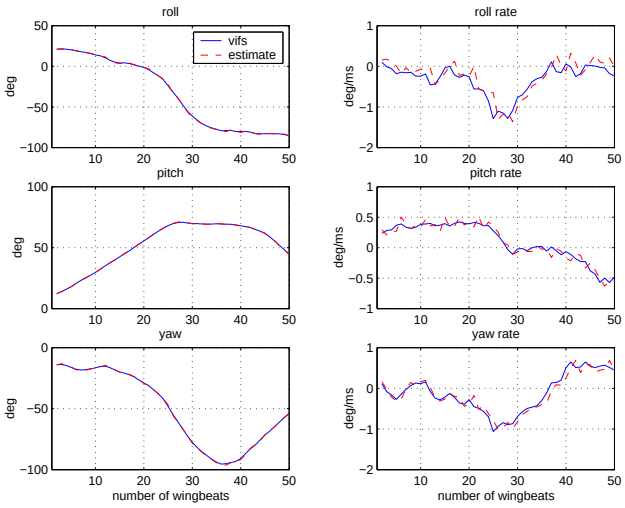

Figure 8: Comparison of the predicted mean angles and angular velocities from the nominal LTI model(dashed line) and those simulated from VIFS (solid line) over 50 consecutive wingbeats; γ, α_l, and α_r are chosen randomly.

7 LQR Controller Design

Based on the identified model found above, stabilizing state feedback control laws are designed and tuned first on the nominal LTI model, then tested on the fully nonlinear continuous time of Equations (1) . This approach provides a more systematic and robust way of design feedback controllers compared to our previous work [2]. In order to address the trade off between regulation performance and control effort to avoid control input saturation, and also to take into account process disturbances and measurement noise in Equation (13), we employed a linear quadratic Gaussian (LQG) optimal controller.

As a first step, a state feedback LQR regulator $u = -Kx$ was designed to minimizes the following quadratic cost function

$$J = \lim_{N\to\infty} E(\sum_{k=1}^{N} x(k)^T Q x(k) + u(k)^T R u(K)) \quad (14)$$

where $Q \geq 0$ and $R > 0$ are the weighting matrices to define the trade-off between regulation performance and control effort. The controller was designed with standard discrete-time LQG software, and the diagonal entries in the weighting matrices are iteratively tuned to ensure a good transient response without saturating the control inputs. The final choice of the the weighting matrices Q and R for the regulator are $Q = diag(10, 20, 20, 1, 1, 1)$ and $R = diag(1, 2, 5)$.

The above LQR optimal state feedback $u = -Kx$ can be substituted with a more realistic output feedback $u = -Ky$, where the output y is given by the ocelli, MEMS compass and halteres.

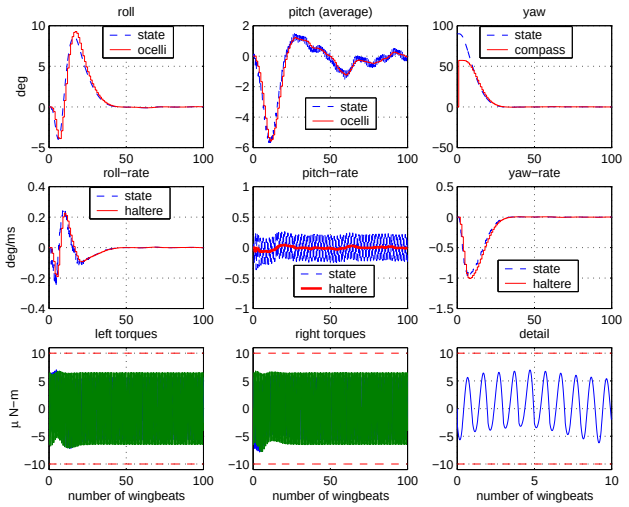

Figure 9: Simulation of MFI evolution and sensor outputs during 90^o steering. (Top plots): Roll, η, pitch, θ, and yaw, ψ angles. (Middle plots): Roll, $\dot{\eta}$, pitch, $\dot{\theta}$, and yaw, $\dot{\psi}$, angular velocities. (Bottom plots): Actuators control inputs (u_1, u_2) for left (left plot) and right (center plot) wing. The bottom right plot shows a detail of the torque u_1 of the left wing for the first 10 wingbeats. Time on abscissae is expressed in wingbeats, *i.e.* 1 wingbeat = $6.7ms$

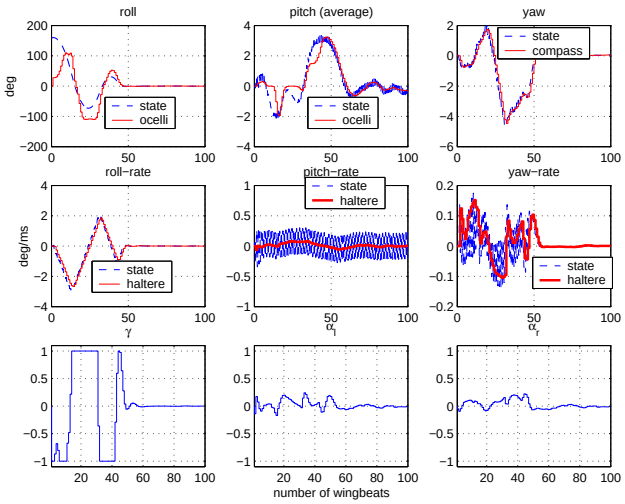

Figure 10: Simulation of MFI evolution and sensor outputs during recovering from upsidedown orientation. (Bottom plots): Wing kinematic parameters $(\gamma, \alpha_r, \alpha_l)$.

The LQR controller was finally tested on the fully nonlinear continuous time model which includes the MFI dynamics of Equation (1), the thorax dynamics of Equation (9), and the sensors models described in Section 3. The simulations are based on an MFI of $100mg$ and $2cm$ tip-to-tip wingspan with wingbeat frequancy $f = 150Hz$.

In the first simulation, shown in Figure 9, the MFI is started from initial condition $x_0 = (\eta, \theta, \psi, \dot{\eta}, \dot{\theta}, \dot{\psi}) = (0, 0, 90^o, 0, 0, 0)$, and it rotates the heading to the desired hovering condition $x^* = (\eta, \theta, \psi, \dot{\eta}, \dot{\theta}, \dot{\psi}) = (0, 0, 0, 0, 0, 0)$. The LQR controller drives smoothly the MFI to the desired final position in approximately 50 wingbeats, which means in approximately a third of the second. Note that the pitch velocity, shown in Figure 9 (center middle plot), exhibits a highly oscillatory behavior. This is not the result of a poor controller, but the result of the unavoidable periodic pitch torque due to the flapping of the wings. In the Figure 9, we overimposed the output of the sensors to the evolution of the MFI angles and angular velocities. The halteres estimates remarkably well the mean of the angular velocities by filtering out the high time varying disturbance due to the flapping wing. The ocelli track the roll and pitch angles correctly. The MEMS compass initially underestimates the yaw angle, but provides the correct error sign, and eventually it tracks the yaw correctly when the the angle becomes small. It is possible to show that the MEMS compass always estimates correctly the sign of the yaw angle, thus driving the MFI to the desired position. The bottom plots of Figure 9 present the corresponding control inputs to the thorax actuators. As expected, the control inputs never exceed the torque limits and they are continuous at the end of every wingbeat.

In the second simulation in Figure 10, the MFI is started from initial condition $x_0 = (\eta, \theta, \psi, \dot{\eta}, \dot{\theta}, \dot{\psi}) = (160^o, 0, 0, 0, 0, 0)$, which correspond to an almost upsidedown orientation, and it rotates the heading to the desired hovering condition $x^* = (0, 0, 0, 0, 0, 0)$ as above. Again, the LQR controller drives smoothly the MFI to the desired final position in approximately 75 wingbeats, which correspond to a half of the second. In this situation, when the roll and pitch angles are very large, the ocelli fail to estimate them exactly, however they still drive the system to the desired position. This is not accidental and it is shown in [6] that they always provide a signal that drives the MFI to hovering. This implies that the LQR controller, albeit designed for small angles and angular velocities, always drives the system to the desired position. The bottom plots the Figure 10 present the corresponding wing kinematic parameters that specify the wings motion. They are strongly related to the control effort required by the LQR controller. The parameter γ, which is directly related to roll body torque, saturates, thus implying an aggressive maneuvering, and infact the MFI roll angle exhibit some overshooting as it in generally common to controller with high gains. On the other hand, the LQG gains were designed for small angle maneuvers, and anti wind-up techniques for large angles maneuvers will be explored in the future. Additional tests were performed on various initial conditions which all yield good results.

8 Conclusion

In this work, high level attitude control of the MFI was considered. Based on recently developed thorax model, a new wing kinematic parameterization method was developed to generate feasible wing motions. Compare to our previous method, this parameterization schemes ensures the boundedness and smoothness of the thorax input torques while at the same time decouples the average roll, pitch, yaw torques in the body frame. A nominal state-space LTI model in hover was identified through linear estimation and a LQR controller was designed. Sensor models such as haltere, magnetic compass, and ocelli were included inside the closed loop system and the simulations show a performance comparable to that of real insect that can complete a full maneuver in approximately $30 - 50$ wingbeats. It is also shown that under LQR control the MFI is able to recover from large angular displacements such as recovering from an upside down orientation and steering 90^o degrees in the yaw axis with fast transient response, despite the fact the LQR controller was designed for small angular errors and the sensor outputs are nonlinear for large angles.

Future work involves quantification of the parameter uncertainties in our nominal model, such as the sensors noise and the atmospheric turbulence, from experimental data. Also, given the limited computational resources of the MFI, we will address the problem of reducing complexity of the controller, for example by controlling the wing motions every N-wingbeats. Extension to the complete 6 DOF system dynamics including position control need to be investigated in hovering and forward flight. Alternative wings motion parameterizations will be evaluated to minimize energy consumption and generate large body torques.

References

[1] S. Avadhanula, R. J. Wood, D. Campolo, and R. S. Fearing. Dynamically tuned design of the mfi thorax. In *Proc. of ICRA*, 2002.

[2] X. Deng, L. Schenato, and S.S. Sastry. Hovering flight control of a micromechanical flying insect. In *Proc. of the 40th CDC*, 2001.

[3] X. Deng, L. Schenato, and S.S. Sastry. Model identification and attitude control scheme for a micromechanical flying insect. In *Proc. of the 7th ICARCV*, 2002.

[4] R.S. Fearing, K.H. Chiang, M.H. Dickinson, D.L. Pick, M. Sitti, and J. Yan. Transmission mechanism for a micromechanical flying insect. In *Proc. of ICRA*, 2000.

[5] R. Hengstenberg. Mechanosensory control of compensatory head roll during flight in the blowfly Calliphora erythrocephala Meig. *Journal of Comparative Physiology A*, 163:151–165, 1988.

[6] L.Schenato, W.C. Wu, and S.S. Sastry. Attitude control for a micromechanical flying insect via sensor output feedback. *Submitted to IEEE Journal of Robotics and Automation.*

[7] B. Motazed, D. Vos, and M. Drela. Aerodymatics and flight control design for hovering MAVs. In *Proc of Amer Control Conference*, Philadelphia, PA, June 1998.

[8] G. Nalbach. The halteres of the blowfly Calliphora: I. kinematics and dynamics. *Journal of Comparative Physiology A*, 173:293–300, 1993.

[9] L. Schenato, X. Deng, and S.S. Sastry. Hovering flight for a micromechanical flying insect: Modeling and robust control synthesis. In *Proc. of IFAC World Congress on Automation and Control*, 2002.

[10] L. Schenato, X. Deng, W.C. Wu, and S.S. Sastry. Virtual insect flight simulator (vifs): A software testbed for insect flight. In *Proc. of ICRA*, 2001.

[11] H. Schuppe and R. Hengstenberg. Optical properties of the ocelli of calliphora erythrocephala and their role in the dorsal light response. *Journal of Comparative Biology A*, 173:143–149, 1993.

[12] W.C. Wu, L. Schenato, R.J. Wood, and R.F. Fearing. Biomimetic sensor suite for flight control of a micromechanical flight insect: Design and experimental results. In *Submitted to ICRA*, Taipei, Taiwan, 2003.

[13] W.C. Wu, R.J. Wood, and R.S. Fearing. Halteres for the micromechanical flying insect. In *Proc of the IEEE Int'l Conf on Robotics and Automation*, Washington, DC, May 2002.

[14] J. Yan, R.J. Wood, S. Avadhanula, R.S. Fearing, and M. Sitti. Towards flapping wing control for a micromechanical flying insect. In *Proc of the IEEE International Conference on Robotics and Automation.*