Model Identification and Attitude Control Scheme for a Micromechanical Flying Insect*

Xinyan Deng, Luca Schenato and Shankar Sastry
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
{*xinyan|lusche|sastry*}*@eecs.berkeley.edu*

Abstract

This paper describes recent development on the design of the flight control system for a Micromechanical Flying Insect (MFI), a 10-25mm (wingtip-to-wingtip) device capable of sustained autonomous flight. High level attitude control is considered. Based on our previous work [1], in which the complex time-varying component of aerodynamic forces are treated as external disturbances, a nominal state-space linear time-invariant model in hover is developed through linear estimation. The identified model is validated through the Virtual Insect Flight Simulator(VIFS), and is used to design feedback controllers for the MFI. A LQG controller is designed and compared with a PD controller. The identification scheme provides a more systematic way of treating aerodynamic modeling errors, and the controllers designed based on the identified model shows better overall performance in simulation. Another advantage of this approach is that measurement of the instantaneous aerodyhamic forces is not necessary, thus simplifies the experimental setup for the real MFI.

1 Introduction

The extraordinary flight capabilities of insects have inspired the design of small UAVs with flapping wings mimicking real flying insects. 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 UAVs. This is the challenge that the Micromechanical Flying Insect project (MFI) being currently developed at UC Berkeley, has taken [2]. Figure 1 shows a mockup of the target robot fly. In this paper we will present recent improvements in the control system design for the MFI originally proposed in [3].

Similar 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 [1]. This is accomplished by the aerodynamic forces and torques generated from the wing flapping motion. However, different from aerodynamic forces exerted on helicopter blades, aerodynamic forces on insect wings are highly nonlinear and time-varying along a wingbeat due to the periodic motion of the flapping wing blades. Moreover, the total force and torques on the MFI body are the result of two wings, which can have asymmetric kinematics.

Figure 1: mockup of a robot fly

Despite the above differences, valuable lessons can still be learned from the control of helicopters. As is widely adopted by rotorcrafts based on quasi-static assumption on the rotary blades, the complicated helicopter dynamics is approximated by a linear time-invariant model, and various linear control algorithms have proved to be successful [4]. This *linearizing* idea can also be applied on the MFI, provided that the wing is flapping at high enough frequency (in our case, around $150Hz$) when the chattering of the motion is small. In this case the periodic response of the real continuous model can be approximated by a time-invariant model of averaged signals over one wingbeat, and the time varying components(residues) appears as external disturbances [1]. Moreover, the wing motion can be changed at most on a wingbeat-to-wingbeat basis, since the wings need to follow a periodic motion to generate sufficient lift to sustain the insect weight. Therefore, a continuous control modeling is not applicable.

Therefore, the first problem addressed in this work is the identification of a discrete-time linear time-invariant (LTI) model which captures the main dynamic features of the MFI near hover. The second problem considered is that of constructing controllers to stabilize hover and provide setpoint tracking.

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

2 Insect Dynamics

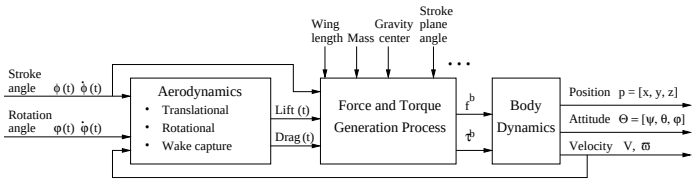

Figure 2: The model for the insect dynamics.

A complete model of an insect can be divided into three different subsystems, which are the *aerodynamics*, *force and torque generation process*, and *body dynamics* as shown in Figure 2. Stroke angles and rotation angles are defined in Figure 3, together with lift and drag aerodynamic forces generated from the wing flapping motion.

Note from Figure 2, the *actuator dynamics* is not included, which will be designed as a PWM to drive the stroke and rotation angles into periodic motion. In our present work with high level flight control, it is assumed that the stroke and rotation angles take form of trigonometric functions and their amplitude and phase can be modulated directly.

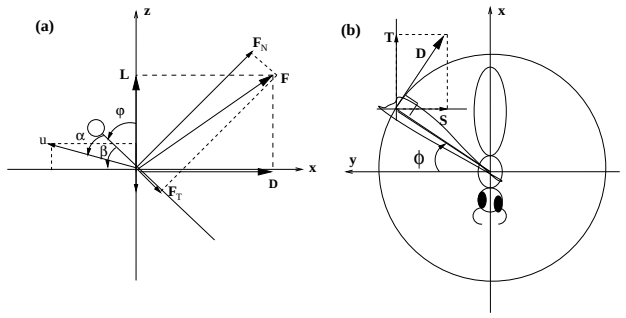

Figure 3: Aerodynamic forces decomposed into lift(L) and drag(D) forces in stroke plane; (a) lateral view; (b) top view; ϕ: stroke angle, φ: rotation angle, α: angle of attack, u: wing velocity.

Given lift, drag forces and stoke angles, the total torques in the *body frame* can be derived. As shown in [5], the attitude dynamics for a rigid body subject to an external torque τ^b applied at the center of mass and specified with respect to the body coordinate frame is given by

$$I\omega^b + \omega^b \times I\omega^b = \tau^b \tag{1}$$

where I is the inertia matrix. ω^b is the angular velocity vector in body frame.

Let R represents the rotation matrix of the body axes relative to the the spatial axes, we have $\hat{\omega}^b = R^T\dot{R}$. For $R \in SO(3)$, we parameterize R by $roll(\phi)$, $pitch(\theta)$, $yaw(\psi)$ Euler angles about x,y,z axes respectively. 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 $\Theta \in R^3$, the equations of motion of the insect is rewritten as

$$\ddot{\Theta} = (IW)^{-1}[\tau^b - W\dot{\Theta} \times IW\dot{\Theta} - I\dot{W}\dot{\Theta} \tag{2}$$

where the body torques are periodic, nonlinear and discontinuous functions of the wing kinematics. *i.e.*

$$\tau^b = \tau^b(\phi_i(t), \dot{\phi}_i(t), \varphi_i(t), \dot{\varphi}_i(t)) \tag{3}$$

where $i \in \{l, r\}$ represents the left and right wing, respectively. The aerodynamic force and torque calculations are very complicated and highly nonlinear, which is described in our previous work [6].

3 Wing Kinematic Parameterization

Since the relation between wing motions and corresponding torques are highly nonlinear and not easily invertible, as found from Equation (3), we simplify the problem by representing the wing motions $(\phi_i(t), \dot{\phi}_i(t), \varphi_i(t), \dot{\varphi}_i(t))$ within one wingbeat by a set of three parameters. These parameters, if properly chosen, can decouple the averaged roll, pitch, yaw torques generated during one wingbeat, thus simplifying the design of hovering controllers.

The parameterization is based on recent work [7] that have evidenced how the modulation of the mean angle of attack and the phase of rotation between the two wings can generate asymmetric 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.

Following these observations, we parameterize the motion of the wings with only three parameters as follows:

$$\begin{aligned}
\phi_r(t) &= \phi_l(t) = \Phi\sin(2\pi f\, t) \\
\varphi_r(t) &= \Upsilon_r\,[\sin(2\pi f\, t) + \alpha_r \sin(4\pi f\, t)] \\
\varphi_l(t) &= \Upsilon_l\,[\sin(2\pi f\, t) + \alpha_l \sin(4\pi f\, t)] \\
\Upsilon_l &= \frac{\pi}{4} + \frac{\pi}{8} ramp(\gamma) \\
\Upsilon_r &= \frac{\pi}{4} + \frac{\pi}{8} ramp(-\gamma)
\end{aligned} \tag{4}$$

where ϕ is the stroke angle, f is the wingbeat frequency, Φ is the maximal stroke amplitude, φ is the rotation angle, Υ is the maximal rotation angle and the subscript r and l stand for right and left wing, respectively. The function $ramp()$ is defined as follows:

$$ramp(\gamma) = \begin{cases} 0 & : \quad \gamma < 0 \\ \gamma & : \quad \gamma \geq 0 \end{cases} \tag{5}$$

The parameters α_l and α_l are strongly related to wing flip timing: a positive value corresponds to advancing

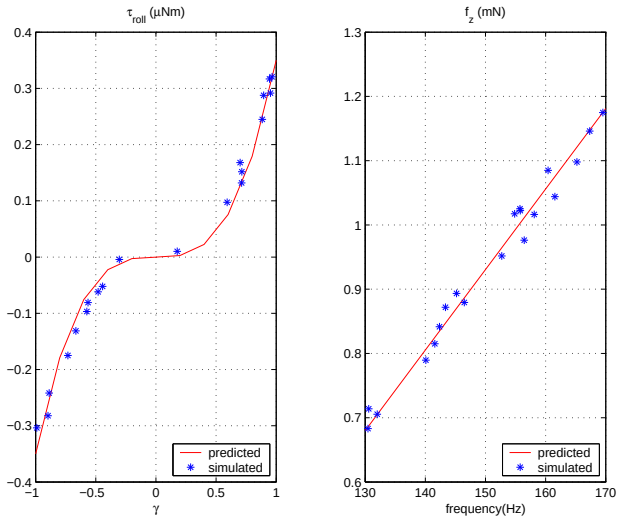

Figure 4: Average roll torque, τ_η, map (left) as a function of the parameter γ and different values for the other two parameters (dotted lines). The solid line corresponds to the approximate function $\tau_\eta = c^{-1}\gamma^3$. Mean lift, f_z, calculated at different frequencies (right).

the wing rotation on the downstroke and delaying on the upstroke, a negative value does the opposite, a null value results in a symmetric wing rotation at both the half-strokes. The parameter γ modifies the mean angle of attack of the wings: a negative value corresponds to a smaller mean angle of attack on the right wing, a positive value to the opposite, and a zero value to equal mean angle of attack.

We obtained an empirical map from wing kinematic parameters to the average torques generated over one wingbeat through the VIFS with the morphology of a honey bee. Figure 4 and Figure 5 show the simulation results. The empirical map can be written as follows.

$$\begin{aligned} \bar{\tau}_\eta &= f_1(\alpha_r, \alpha_l, \gamma) = c\gamma^3 + \delta_\eta \\ \bar{\tau}_\theta &= f_2(\alpha_r, \alpha_l, \gamma) = a_{11}\alpha_l + a_{12}\alpha_r + \delta_\theta \\ \bar{\tau}_\psi &= f_3(\alpha_r, \alpha_l, \gamma) = a_{21}\alpha_l + a_{22}\alpha_r + \delta_\psi \end{aligned} \quad (6)$$

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 f_2 and f_3 can be approximated with linear functions of α_l and α_r, while f_1 with a simple nonlinear function of γ only. Therefore the mean torques are decoupled for the identification and control purposes.

4 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). With respect to the original continuous dynamics as in Equation (2), our goal is to approximate the averaged attitude model in hover by a discrete-time LTI model in the following form

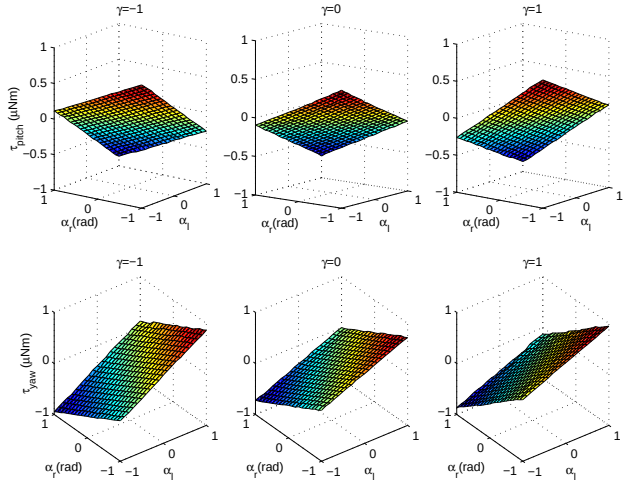

Figure 5: Average pitch and yaw torque maps.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k) \\ y(k) &= Cx(k) + Du(k) + v(k) \end{aligned} \quad (7)$$

where $x = [\bar{\phi}, \bar{\theta}, \bar{\psi}, \bar{p}, \bar{q}, \bar{r}]^T$ is the vector of average roll, pitch, yaw angles and angular rates over one wingbeat, and w represents the time varying component which appears as an external disturbance to the linear model [1]. y is the vector of measured outputs, with additional measurement noise v. C and D matrices are set to be identity and zero matrices, respectively. $u = [u_1, u_2, u_3]^T = [\gamma^3, \alpha_l, \alpha_r]^T$ are the control inputs, *i.e.*, the wing kinematic parameters.

The model identification problem can be recast into a least square solution to an overdetermined set of linear equations as $Ez = b$, where $z = [A_{11}, ..., A_{66}, B_{11}, ..., B_{63}]^T$ is the vector of system parameters to be estimated, E and b are matrices whose elements consists of the experiment data. The least square solution which minimizes the norm of the error $\| e \|^2 = \| b - Az \|^2$ is given by $z = E(E^TE)^{-1}E^Tb$.

The experiments were performed on the Virtual Insect Flight Simulator (VIFS), developed by the authors to provide a software testbed for insect flight [**?**]. 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 (7) 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 sampling period, namely the wingbeat period. The parameters of the state-space realization in Equation (7) consist of the elements of the A_{21}, A_{22} and B_{22} matrices. It was found that the diagonal elements

of A_{22} and the block diagonal elements in B_{22} were very repeatable on different experiments. The final values of the estimated parameters are given in Equation (8).

$$
\begin{aligned}
T &= 6.8139 \\
A_{21} &= 0.0001 \begin{bmatrix} -0.1555 & 0.0453 & -0.1317 \\ 0.2282 & 0.0571 & -0.0242 \\ -0.0501 & -0.0683 & 0.0099 \end{bmatrix} \\
A_{22} &= \begin{bmatrix} 1.0049 & -0.0090 & 0.0338 \\ 0.0059 & 1.0013 & 0.0021 \\ 0.0160 & 0.0010 & 1.0006 \end{bmatrix} \\
B_{22} &= 0.00001 \begin{bmatrix} 210 & -4.1765 & 2.8645 \\ -2.8950 & -70 & -80 \\ 0.2153 & -110 & 120 \end{bmatrix} \quad (8)
\end{aligned}
$$

It can be seen from Equation (8) that A_{21} is close to an zero matrix, and 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 (6).

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 6 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.

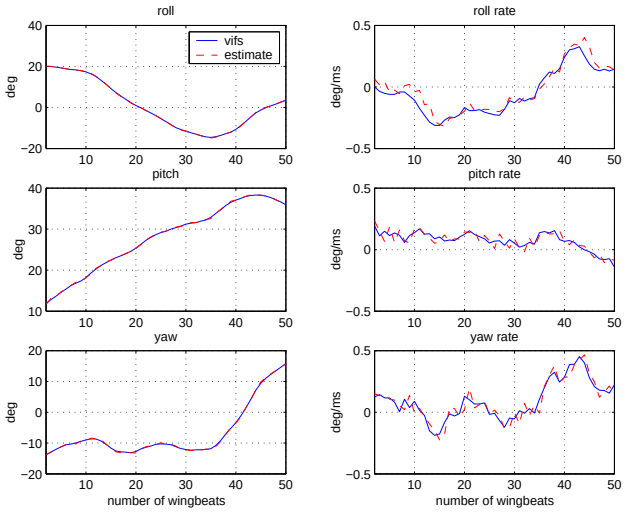

Figure 6: Comparison of the predicted mean angles and angular velocities from the nominal LTI model(dashed line) and those simulated from VIFS (solid line) over a consequetive 50 wingbeats; γ, α_l, and α_r are chosen randomly.

5 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 implemented on the MFI. This approach provides a more systematic way of design feedback controllers compared to our previous work [3]. Non-linearities are negligible and can be treated as additional disturbance into the nominal LTI model, which captures the main dynamic behaviour of the system near hover, as shown in Figure (6). As our first approach, we employ a classical PD controller through multi-input multi-output pole placement method. To account for model uncertainties and control input saturation, a LQG controller were designed and compared to the PD controller.

5.1 PD control via Pole Placement

Given the LTI system 7 be controllable, the closed loop system poles (*i.e.*, eigenvalues of $A - BK$) can be arbitrarily assigned through a state feedback control law $u(k) = -Kx(k)$. The controller gain K was found by classical multi-input multi-output pole placement method such that the closed loop poles are assigned at desired locations inside the unit circle [8]. The controller was designed with standard discrete time pole placement software and through iteratively tuning and experimenting on the nominal LTI model, a set of optimal poles are found to be $p = (0.4, 0.5, 0.55, 0.6, 0.65, 0.7)$.

As an alternative to sensor measurements, we implemented a full state observer and substitute the states with their estimates in the above control law design.

Since the LTI system (7) is observable, the state estimator of the form $\xi(k+1) = A\xi(k) + Bu(k) + L(y - C\xi - Du(k))$ can be constructed by proper selection of the estimator gain matrix L, such that the estimator poles (*i.e.*, eigenvalues of $A-LC$) are assigned at proper locations. Replacing the control law by $u = -K\xi$ yields the closed loop dynamic output feedback compensator with both states x and estimation error $e = x - \xi$ converge to zero [9].

As a rule of thumb, the estimator dynamics should be faster than the controller dynamics (eigenvalues of $A - BK$). Here we design the estimator poles to be $q = 0.5p$, and again the estimator gain matrix L is calculated by available pole placement algorithm. Figure 7 shows the simulation results of the observer-based state feedback controller through VIFS. As can be seen, the angles are recovered from $[35^o, -30^o, 25^o]$ to zeros in less than 50 wingbeats. However, there is serious saturation of the control inputs which is not desirable in our design. Furthermore, the simulations show chattering in both states and control inputs due to the external disturbances resulting from the periodic motion and also measurement noise.

5.2 LQG control

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 (7), 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

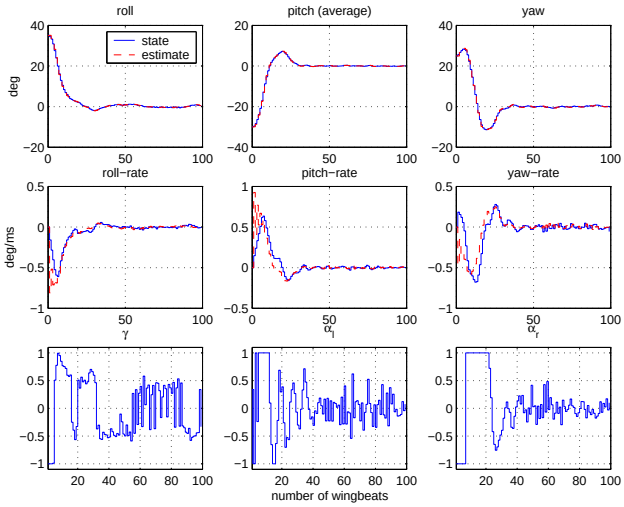

Figure 7: PD control in hover: both states(solid line) and their estimates (dashed line) from the observer are plotted. The last row are the control inputs.

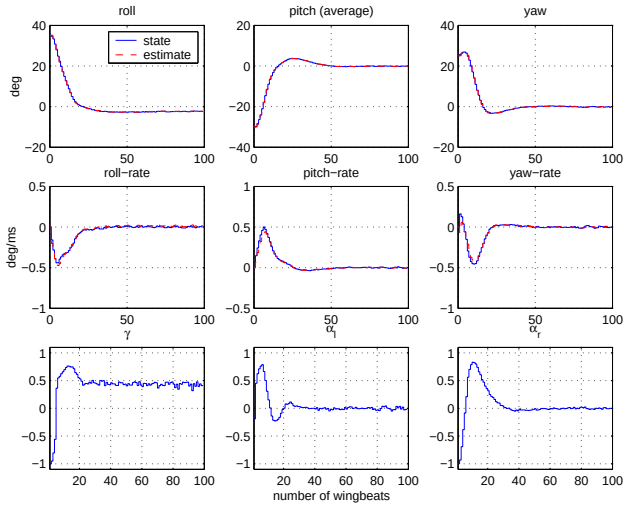

Figure 8: LQG control in hover: both states(solid line) and their estimates (dashed line) from the Kalman filter are plotted. The last row are the control inputs.

cost function

$$J = \lim_{N\to\infty} E(\sum_{k=1}^{N} x(k)^T Q x(k) + u(k)^T R u(K)) \qquad (9)$$

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 $R = diag(1, 2, 5)$.

As in the case of pole placement, the above LQR optimal state feedback $u = -Kx$ is not implementable without full state measurement. As the second step for the output feedback problem, a state estimator was generated by the Kalman filter $\hat{x}(k+1) = A\hat{x} + Bu + L(y - C\hat{x} - Du)$, with inputs u (controls) and y (measurements). The noise covariance data $E(ww^T) = Q_n$ and $E(vv^T) = R_n$ determines the Kalman gain L through an algebraic Riccati equation [9].

The input disturbance noise covariance matrix $Q_n = E(ww^T)$ was calculated from the $w(k)$ of the identified LTI model in (7), where $w(k)$ represents the periodic component of the real system which enters into the LTI model as a pseudo-disturbance, and was obtained as the residue of the identified model from the real process. The measurement noise $v(k)$ was set to be random noise of the same covariance matrix $R_n = Q_n$.

The Kalman filter was also designed through standard software, and the resulting LQG controller $u = -K\hat{x}$ was implemented on the MFI and simulated for a consecutive 100 wingbeats. Figure 8 shows the simulation results from VIFS. As is with the results of the previous PD controller, the angles are recovered from $[35^o, -30^o, 25^o]$ to zeros in less than 50 wingbeats. However, comparing with Figure 7, there are several notable improvements of the LQG controller over the PD controller: the response is smoother and with less chattering near equilibrium; the Kalman filter provides a much faster and closer estimate of the system states; and most importantly, the control inputs are not saturated, they are also smoother and shows much less chattering.

Figure 9 shows the actual response in the continuous time (with respect to the mean signals in Figure 8). Note that in the continuous time, the oscillations (especially in the pitch axis) are due to the back and forth periodic wing flapping motion.

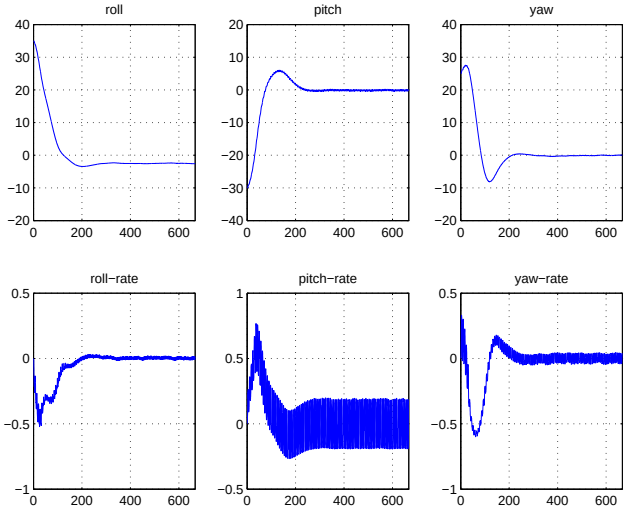

Figure 9: LQG control in hover: detail in continuous time

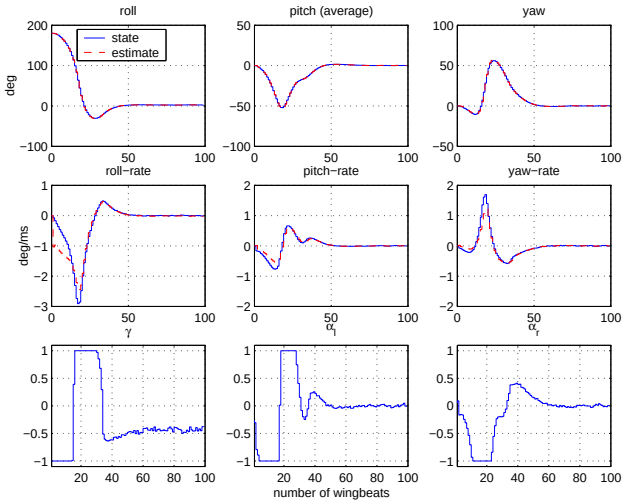

Figure 10: LQG control in recovering from upside down

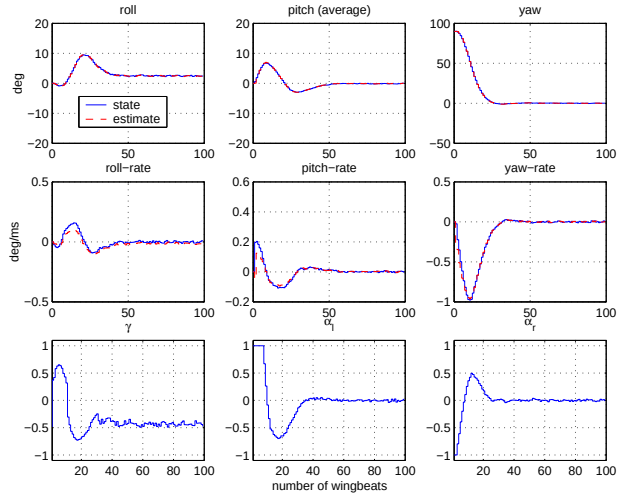

Figure 11: LQG control in steering

Additional tests were performed on various initial conditions which all yield good results. For example Figure 10 shows the response in recovering from upside down and Figure 11 shows the response of steering 90 degrees in the yaw axis. It can be seen that the angles are recovered less than 50 wingbeat with some saturation of the control input, but it can be improved by adjusting the weights in 9. Therefore we can conclude that the nominal LTI model captures the main dynamics of the insect very well, and the LQG controller is well suited for our control design.

6 Conclusion

In this work, high level attitude control of the MFI was considered. Based on proper wing kinematic parameterization, a nominal discrete-time linear time-invariant (LTI) model which captures the main dynamic features of the MFI near hover was identified. Feedback controllers were designed and tuned on the nominal LTI model before they are implemented on the MFI. An observer based dynamic output feedback control was designed and compared to a LQG controller. The controller are simulated through VIFS and both PD controller and LQG controller shows good results. In addition, the LQG controller demonstrate better overall performance, and is preferred to the PD control in its ability to avoid control input saturation. It is also shown that the under LQG control the MFI is able to recover from large angular displacement such as recovering from upside down and steering 90 degrees in the yaw axis with fast transient response.

Recent development involves adding the thorax(actuator) model and various sensor models such as halteres(gyro), ocelli(light sensor), and magnetic campuss and simulations through VIFS showed satisfactory overall performances. Future work involves quantification of the parameter uncertainties in our nominal model, and apply robust control methods such as H_∞ control and μ analysis to address model uncertainty, external disturbance, and measurement noise. Extension to the complete 6 DOF system dynamics including position control need to be investigated in hovering and forward flight. Implementation of an atmaspheric turbulence model will also need be addressed to the test the MFI's ability for wind gust disturbance rejection.

References

[1] L. Schenato, X. Deng, and S.S. Sastry, "Hovering flight for a micromechanical flying insect: Modeling and robust control synthesis," in *Proc. of IFAC*, 2002.

[2] 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.

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

[4] R.W. Prouty, *Helicopter Performance, Stability, and Control*, Krieger Publishing Company, 1995.

[5] R. M. Murray, Z. Li, and S.S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1994.

[6] 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.

[7] M.H. Dickinson, F.O. Lehnmann, and S.S. Sane, "Wing rotation and the aerodynamic basis of insect flight," *Science*, vol. 284, no. 5422, pp. 1954–1960, 1999.

[8] C. Chen, *Linear System Theory and Design*, Oxford University Press, 1998.

[9] K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice Hall, 1996.