Flight Control for Target Seeking by 13 gram Ornithopter

Stanley S. Baek, Fernando L. Garcia Bermudez, and Ronald S. Fearing

***Abstract*— Recent advances in small-scale flapping-wing micro aerial vehicles have extended the capabilities of flight control for a number of applications, such as intelligence, surveillance, and reconnaissance activities. In this work, we demonstrate autonomous flight control of a 13 gram ornithopter capable of flying toward a target without remote assistance. For autonomous flight control, we developed 1.0 gram control electronics integrated with a microcontroller, inertial and visual sensors, communication electronics, and motor drivers. We also developed a simplified aerodynamic model of ornithopter flight to reduce the order of the control system. With the aerodynamic model and the orientation estimation from on-board inertial sensors, we present flight control of an ornithopter capable of flying toward a target using onboard sensing and computation only. To this end, we developed a dead-reckoning algorithm to recover from the temporary loss of the target which can occur with a visual sensor with a narrow field of view. As a result, the 28 cm wing-span ornithopter flying toward a target landed within a radius of 0.5 m from the target with more than 85% success (N = 20).**

I. INTRODUCTION

Flapping-wing flight in nature has unparalleled maneuverability, agility, and hovering capability. Over the last few decades, engineers have made remarkable progress toward the design of flapping-wing micro aerial vehicles (MAVs) inspired by biological systems [1], [12], [15], [16], [18], [22], [24]. Autonomous flight of such small robots is, however, still nascent. It is mainly due to the inherent high-nonlinearity of such robots and lack of system system identification. Also, small-scale flapping-wing robots usually have a minimal number of actuators due to the size constraints, and therefore the systems are generally under-actuated. Furthermore, smaller scale flapping-wing air vehicles usually require higher flapping frequencies, which produce high-frequency vibrations. The inertial force generated by flapping motion of the wings is transmitted to the body and makes the body vibrate at the flapping frequency.

Autonomous flight of flapping-wing MAVs has been studied with diverse approaches. Deng *et al*. [8], [9] designed LQR-based feedback laws for hovering control using an approximate time-invariant model of the Micromechanical Flying Insect [10] with time averaged aerodynamic forces. Khan *et al*. developed a differential flatness-based nonlinear controller for a time-averaged system that can be applied to flapping-wing MAVs [13]. Other closed-loop control for flapping-wing MAVs was demonstrated by Shigeoka [19].

The authors are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, USA {stanbaek, fgb, ronf}@eecs.berkeley.edu

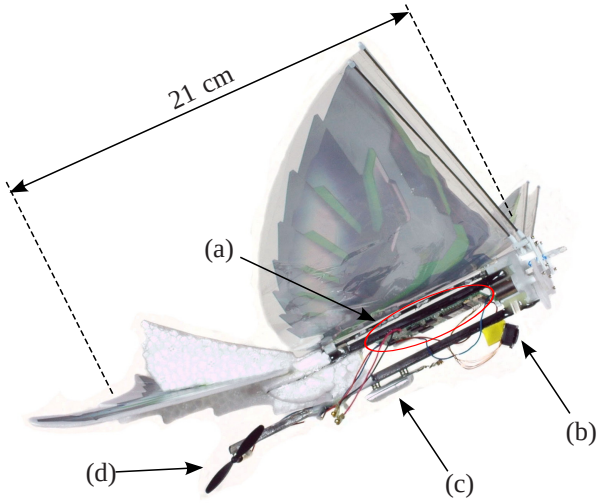
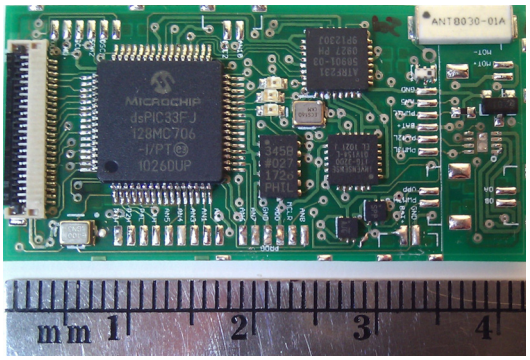


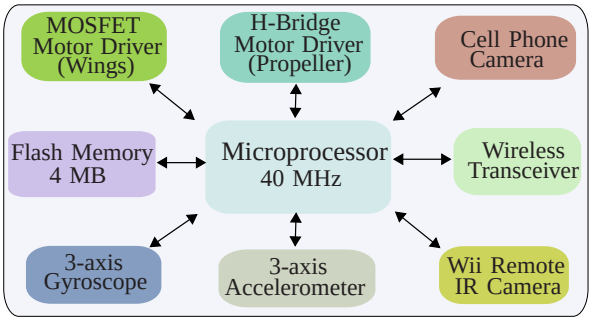
Fig. 1: The modified ornithopter equipped with (a) control electronics, (b) a visual sensor, (c) a battery, and (d) a propeller.

Acquiring acceleration and position data using an accelerometer and a Vicon motion tracking system, Shigeoka developed a simple dynamic model for a flapping-wing MAV and demonstrated simulation results using frequency analysis. These approaches performed well in computer simulations, yet they have not been implemented in a real flapping-wing robot.

Some of the significant research on autonomous flight in a real flapping-wing robot is summarized here. Tedrake *et al*. [21] demonstrated autonomous flight of a 2-meter wingspan ornithopter with enough payload to carry a 400 gram sensor and computer package. For small-scale ornithopters, engineers developed flight controllers using a ground station for computationally expensive processes such as video processing. De Croon *et al*. implemented an obstacle avoidance algorithm for the 16 gram flapping wing DelFly II with two onboard cameras [6], [7]. In their work, a ground station was used to process sensor data and compute control signals. Baek and Fearing demonstrated altitude regulation of a 12 gram ornithopter using an external camera to estimate the altitude of the ornithopter [2]. For fixed-wing small-scale aerial vehicles, Zufferey *et al*. demonstrated fully autonomous flight of a 10 gram aerial vehicle by mapping optical flow signals into control commands in an indoor environment [25].



(a)



(b)

Fig. 2: (a) The 1.0 gram control board consisting of a 40 MHz dsPIC microprocessor, 4MB flash memory, 3-axis accelerometer, 3-axis gyroscope, 802.15.4 wireless transceiver, 2-channel motor driver, IR camera connector, and cell phone camera connector. (b) Functional block diagram for the control board.

In this paper, we present closed-loop onboard attitude regulation for a 13 gram ornithopter, shown in Fig. 1, equipped with inertial sensors, visual sensors, and a microprocessor for onboard processing of sensor signals as well as control input signals. We also discuss dead reckoning to the target location when the target is temporarily not exposed on the visual sensor. Based on the development of onboard attitude regulation and the dead reckoning algorithm, we demonstrate autonomous flight of the 13 gram ornithopter flying toward a target without any external resources.

II. ORNITHOPTER PLATFORM

The ornithopter used is a modified Silverlit WingMaster I-Bird. From the original 4 winged ornithopter, the styrofoam outer body was removed to reduce weight. The removal of the styrofoam, however, resulted in undesirable bending and torsional compliance of the structure that absorbs some of the power to be transferred to the environment. As a result, the aerodynamic forces generated by the wings were significantly reduced. To compensate for this, we replaced the main backbone frame with stiff carbon fiber T-beams so that the aerodynamic forces remained the same.

Another modification was to replace the rudder at the end of the I-Bird's tail with a propeller. The rudder was

TABLE I: Specifications of the Modified Ornithopter

Wing span	28 cm
Body length	21 cm
60 mAh Li-Poly Battery	1.8 grams
Motor & gear box for wings	2.8 grams
Steering motor & propeller	0.9 grams
Wings	1.8 grams
Frames & tail	2.6 grams
Control board	1.0 grams
Infrared camera	0.33 grams
Connectors & wires	2.2 grams
Total Weight	13.4 grams

driven by a magnetic actuator to the left or right direction for yaw moment. One of the problems of the magnetic actuator was that it required too much current (∼300 mA) and the ornithopter could not generate enough lift force when the magnetic actuator was activated. Another problem was that it had only tristate modes, i.e., full left, full right, and neutral, which limited the control policy to be bang-bang control. For these reasons, we replaced the rudder with a bidirectional propeller driven by a 4 mm diameter pager motor. After this modification, the ornithopter consumed less than 100 mA current at the maximum speed of the propeller while the propeller generated much higher yaw moment than the rudder.[1] It also enabled us to implement a PID controller for yaw regulation instead of a bang-bang control.

The original RC electronics on the I-Bird have also been replaced with a custom made control board shown in Fig. 2a. The 1.0 gram control board is designed for completely autonomous flight of an ornithopter, i.e., there are no remote sensor inputs or control signals during flight. The control board has 2-channel motor driver - a MOSFET and an H-Bridge. The MOSFET, driven by a PWM signal, is used to drive the main motor to generate flapping motion of the wings, and the H-Bridge is used to drive the bidirectional propeller to generate yaw moment. The infrared camera, originated from a Nintendo Wii remote controller, can detect up to 4 infrared emitters at a time and transmits the positions of the emitters at 400 Hz in an image frame of 1024×768 pixels through an I^2C communication link.[2] The 4 MB flash memory is used to save sensor outputs, state information, and the control input signals for later review. The wireless transceiver is used for communication with a PC, e.g., initializing flight and retrieving the data stored in the flash memory. The cell phone camera connector is used for an OmniVision OV7660FSL VGA camera module, which is not used in this work. The functional block diagram of the board is shown in Fig. 2b.

A 60 mAh Lithium-Polymer battery provides power for the electronics, the main DC motor that drives the flapping

[1]The turning radius with a propeller is approximately 1.0 meter whereas the turning radius with a rudder is greater than 1.5 meters.

[2]The sampling rate of the camera we used in this work was 400Hz although it could go up to 1 kHz. More details about this infrared camera can be found in [23].

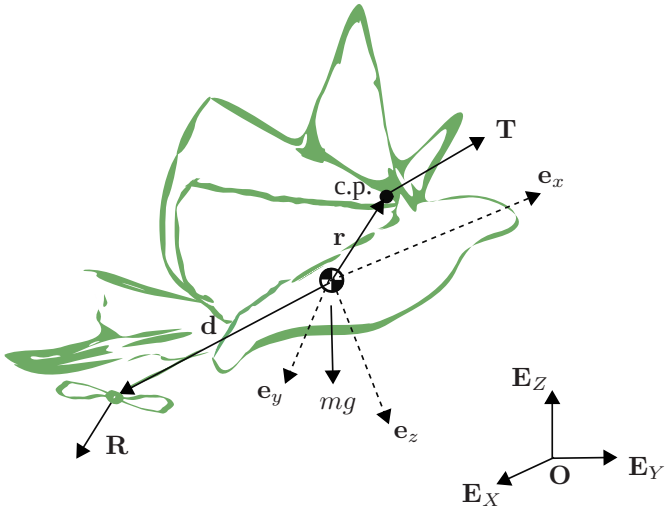


Fig. 3: The coordinate systems and the forces generated by the wings and the propeller.

wings, and the motor that drives the bidirectional propeller. The main DC motor can drive the wings at 17-20 Hz depending on battery charge. The battery has been carefully mounted on the body of the ornithopter so that we can achieve pendulum-like passive stability. The specifications of the modified ornithopter are summarized in Table I.

III. Dynamic Model

The coordinate systems and the forces generated by the wings and the propeller are depicted in Fig. 3. The body coordinates ($\mathbf{e}_x$, $\mathbf{e}_y$, $\mathbf{e}_z$), attached to the center of mass, are chosen to be the principal axes of inertia of the ornithopter. The dynamic equations of motion can be written as

$$m\dot{\mathbf{V}}^b + \boldsymbol{\omega}^b \times m\mathbf{V}^b = m\mathbf{g} + \mathbf{F}_i + \mathbf{F}_a, \quad (1)$$

$$\dot{\mathbf{H}}^b + \boldsymbol{\omega}^b \times \mathbf{H}^b = \mathbf{M}_i + \mathbf{M}_a, \quad (2)$$

where

$\mathbf{V}^b = (u, v, w)^\mathsf{T}$ = velocity of center of mass in body axes,
$\boldsymbol{\omega}^b = (p, q, r)^\mathsf{T}$ = angular rate in body axes,
$\mathbf{H}^b = \mathbf{I} \cdot \boldsymbol{\omega}^b$ = angular momentum in body axes,
$\mathbf{I}$ = moment of inertia of the ornithopter in body axes,
$\mathbf{F}_i = (F_u, F_v, F_w)^\mathsf{T}$ = induced aerodynamic force,
$\mathbf{F}_a$ = propulsive force,
$\mathbf{M}_i = (M_p, M_q, M_r)^\mathsf{T}$ = induced aerodynamic torque,
$\mathbf{M}_a$ = propulsive torque,
m = mass of the ornithopter,
$\mathbf{g}$ = gravitational acceleration.

$\mathbf{T} = (T\cos\epsilon)\mathbf{e}_x - (T\sin\epsilon)\mathbf{e}_z$, depicted in Fig. 3, represents the thrust force on the average center of pressure generated by the flapping motion of the wings. The the average center of pressure is denoted by c.p. in Fig. 3 and ϵ is the angle between $\mathbf{T}$ and $\mathbf{e}_x$. The steering force, $\mathbf{R} = R\mathbf{e}_y$, produced by the propeller, generates the moment about the $\mathbf{e}_z$ axis. The distance vector, $\mathbf{r} = d_1\mathbf{e}_x - d_2\mathbf{e}_z$, denotes the location of the average center of pressure from the center of mass, and $\mathbf{d} = -d_3\mathbf{e}_x + d_4\mathbf{e}_z$ denotes the location of the propeller from the center of mass. Using the thrust and steering force, the propulsive force, $\mathbf{F}_a$, and propulsive torque, $\mathbf{M}_a$, can be written

$$\begin{aligned}\mathbf{F}_a &= \mathbf{T} + \mathbf{R} \\ &= (T\cos\epsilon)\mathbf{e}_x + R\mathbf{e}_y - (T\sin\epsilon)\mathbf{e}_z, \\ \mathbf{M}_a &= \mathbf{r}\times\mathbf{T} + \mathbf{d}\times\mathbf{R} \\ &= -d_4R\mathbf{e}_x + (d_1T\sin\epsilon - d_2T\cos\epsilon)\mathbf{e}_y - d_3R\mathbf{e}_z.\end{aligned}$$

Whereas the propulsive force, $\mathbf{F}_a$, and the propulsive torque, $\mathbf{M}_a$, are solely generated by the actuators of the ornithopter, the aerodynamic force, $\mathbf{F}_i$, and torque, $\mathbf{M}_i$, are *induced* by the motion of the body. Hence, $\mathbf{F}_i = \mathbf{M}_i = 0$ if the body of the ornithopter is stationary, i.e., $\mathbf{V}^b = \boldsymbol{\omega}^b = 0$.

The kinematic equations of angular motion of the ornithopter can be written in 3-2-1 (or Z-Y-X) Euler angles [17],

$$\begin{pmatrix}\dot\phi\\ \dot\theta\\ \dot\psi\end{pmatrix} = \begin{pmatrix} p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \\ q\cos\phi - r\sin\phi \\ q\sin\phi\sec\theta + r\cos\phi\sec\theta\end{pmatrix}, \quad (3)$$

where p, q, and r are angular rates in the $\mathbf{e}_x$, $\mathbf{e}_y$, and $\mathbf{e}_z$ axes, respectively, and ϕ, θ, and ψ are the Euler angle representations for roll, pitch, and yaw, respectively.

Assuming that

$$\begin{aligned}\phi^2 &\ll 1, \\ \theta &\approx \theta_\circ, \\ (v^2, w^2) &\ll u^2, \\ (p^2, q^2, r^2) &\ll u^2/b^2,\end{aligned} \quad (4)$$

where b = wing span, the quadratic terms in (1) - (3) can be neglected [4]. Now, (1) - (3) can be simplified by

$$\begin{bmatrix} m\dot u \\ m\dot v \\ m\dot w \\ I_x\dot p \\ I_y\dot q \\ I_z\dot r \\ \dot\phi \\ \dot\theta \\ \dot\psi\end{bmatrix} = \begin{bmatrix} F_u - mg\sin\theta_\circ + T\cos\epsilon \\ F_v + mg\cos\theta_\circ\phi + R \\ F_w + mg\cos\theta_\circ - T\sin\epsilon \\ M_p - d_4R \\ M_q + d_1T\sin\epsilon - d_2T\cos\epsilon \\ M_r - d_3R \\ p + r\tan\theta_\circ \\ q \\ r\sec\theta_\circ\end{bmatrix}. \quad (5)$$

Since the shape of the ornithopter is symmetric about the $\mathbf{e}_x$ - $\mathbf{e}_z$ plane, (u, w, q, θ) do not produce any (F_v, M_p, M_r), and (v, p, r, ϕ, ψ) do not produce any (F_u, F_w, M_q), at least to first order when $p^2 \ll u^2/b^2$ [4].

A. Longitudinal Equations of Motion

The *longitudinal* equations of motion can be written as

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \delta\dot{\theta} \end{bmatrix} = \begin{bmatrix} F_{u|u} & F_{u|w} & 0 & -g\cos\theta_\circ \\ F_{w|u} & F_{w|w} & 0 & -g\sin\theta_\circ \\ M_{q|u} & M_{q|w} & M_{q|q} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ q \\ \delta\theta \end{bmatrix} + \begin{bmatrix} \cos\epsilon/m \\ -\sin\epsilon/m \\ (d_1\sin\epsilon - d_2\cos\epsilon)/m \\ 0 \end{bmatrix} T, \quad (6)$$

where $\delta\theta$ is the deviation from the steady equilibrium value of θ, $F_{i|j} \equiv (1/m)(\partial F_i/\partial j)$, and $M_{i|j} \equiv (1/I_i)(\partial M_i/\partial j)$. The characteristic polynomial of this subsystem is $s(s - F_{u|u})(s - F_{w|w})(s - M_{q|q}) - g\cos\theta_\circ F_{w|u} M_{q|w}$. If $g\cos\theta_\circ F_{w|u} M_{q|w} = 0$, the poles of the subsystem are located at ($F_{u|u}$, $F_{w|w}$, $M_{q|q}$, 0). Since $F_{u|u}$, $F_{w|w}$, and $M_{q|q}$ are the coefficients of aerodynamic damping forces/torques with negative signs, the poles are located in the left half plane. Assuming that $g\cos\theta_\circ F_{w|u} M_{q|w}$ is small, the poles are located near ($F_{u|u}$, $F_{w|w}$, $M_{q|q}$, 0) and we can stabilize the subsystem with a small feedback gain.

B. Lateral Equations of Motion

The *lateral* equations of motion can be written as

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} F_{v|v} & 0 & 0 & g\cos\theta_\circ & 0 \\ M_{p|v} & M_{p|p} & M_{p|r} & 0 & 0 \\ M_{r|v} & M_{r|p} & M_{r|r} & 0 & 0 \\ 0 & 1 & \tan\theta_\circ & 0 & 0 \\ 0 & 0 & \sec\theta_\circ & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \\ \psi \end{bmatrix} + \begin{bmatrix} 1/m \\ -d_4/I_p \\ -d_3/I_r \\ 0 \\ 0 \end{bmatrix} R. \quad (7)$$

The poles of this subsystem are located at ($F_{v|v}$, $M_{p|p}$, $M_{r|r}$, 0, 0). Since there are two poles at zero and three real poles in the left half plane, the subsystem cannot be stabilized with a simple feedback gain. For a stable system, therefore, we need to relocate the poles. One way to relocate the poles is to implement a state-space based controller, such as a linear quadratic regulator (LQR), which usually requires an accurate estimation of the system parameters. However, the induced aerodynamic forces and torques in this work have not been identified yet. Another way to relocate the poles is to design a PID controller with derivative action acting on the process output only (without reference input), which will be discussed in Section VI.

IV. ATTITUDE ESTIMATION

To estimate the orientation of the ornithopter, we calibrate the gyroscope before each flight in order to remove constant biased measurement errors. During flight we obtain the angular velocities of the ornithopter in the body axes (p, q, r) using the 3-axis gyroscope and compute ($\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$) at 400 Hz using (3). Then, we integrate ($\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$) to estimate the orientation of the ornithopter in Euler angles, (ϕ, θ, ψ).

The integration-based attitude estimation used in this work requires much less computational power than other estimators based on Bayesian statistics such as [5], [11], [14], [20]. However, it has a serious drawback; integration drift gradually accumulates biased measurements and generates a large amount of error as time elapses. The integration drift for the gyroscopes used in this work is approximately 0.1 degrees per second. A bias estimation of a solid-state gyroscope has been studied by Barshan and Durrant-Whyte [3] using a Kalman filter to improve the orientation estimation. In their work, the thermal drift of the gyroscope has been measured for a long period of time (∼12 hours) to model the error dynamics of the gyroscope. For a short period of time ($<$ 1 minute), however, the gyroscope calibration to remove constant biased measurement error used in this work has the same effect. Furthermore, the deviation angles between the ornithopter's orientation angles and the target location is not affected by the drift when we determine the desired orientation angles, which will be discussed in the following section.

V. PREDICTION OF TARGET LOCATION

The desired yaw angle, $r_\psi(k)$, and desired pitch angle, $r_\theta(k)$, at time k, are computed by

$$r_\psi(k) = \psi(k) + \epsilon_\psi(k), \quad (8)$$

$$r_\theta(k) = \theta(k) + \epsilon_\theta(k), \quad (9)$$

where ϵ_ψ and ϵ_θ are, respectively, the yaw deviation angle and the pitch deviation angle between the ornithopter's heading and the target. We can rewrite (8) and (9) in vector form,

$$\mathbf{r}(k) = \mathbf{\Psi}(k) + \boldsymbol{\epsilon}(k), \quad (10)$$

where $\mathbf{r} = (r_\psi, r_\theta)^\mathsf{T}$, $\mathbf{\Psi} = (\psi, \theta)^\mathsf{T}$, and $\boldsymbol{\epsilon} = (\epsilon_\psi, \epsilon_\theta)^\mathsf{T}$.

The deviation angle, $\boldsymbol{\epsilon}$, is computed by

$$\boldsymbol{\epsilon}(k) = \mathbf{\Omega}(k)\mathbf{p}(k),$$

where $\mathbf{p}$ is a vector of pixel values of the target location directly obtained from the infrared camera, defined by $(p_x, p_y)^\mathsf{T}$, and $\mathbf{\Omega}$ is the combination of scaling matrix and rotational transformation defined by

$$\mathbf{\Omega}(k) = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} \cos(\phi(k)) & \sin(\phi(k)) \\ -\sin(\phi(k)) & \cos(\phi(k)) \end{pmatrix}.$$

Here, s_x = 20°/1024 and s_y = 15°/768 are the scale factors to convert pixel locations to angles, and the roll angle, ϕ, is the rotation of the infrared camera about the $\mathbf{e}_x$ axis.

As the ornithopter travels, the orientation angle, $\mathbf{\Psi}$, as well as the deviation angle, $\boldsymbol{\epsilon}$, change in time as shown in Fig. 4. Therefore, the attitude regulation using the desired orientation angle computed by (10) can be improved with the prediction of the orientation angle and the deviation angle at the subsequent state. This prediction can also be used in dead reckoning of the target location when the target is temporarily lost.

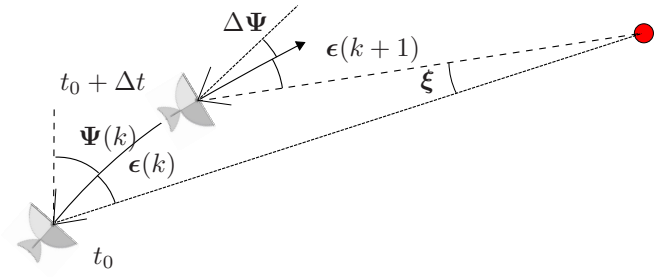


Fig. 4: Prediction of target location at time $k+1$ based on the current orientation and its rate at time k.

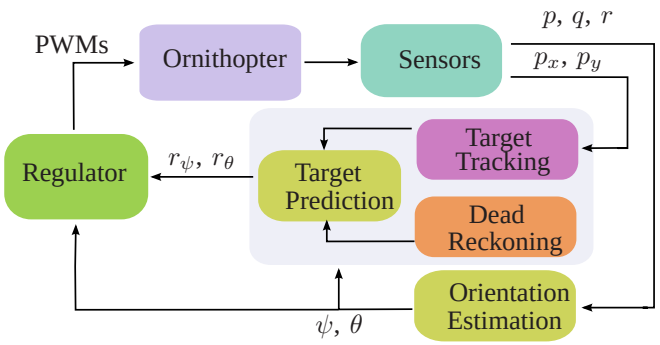


Fig. 5: Signal flow for closed-loop onboard controller

At $t = t_0$, the deviation angle is $\epsilon(k)$. For Δt, the orientation of the ornithopter is changed by $\Delta\Psi(k+1) \triangleq \Psi(k+1)$ - $\Psi(k)$, and the deviation angle at $k+1$ becomes $\epsilon(k+1)$. From Fig. 4, we can find

$$\Delta\Psi(k+1) + \epsilon(k+1) \approx \epsilon(k) + \xi, \tag{11}$$

where ξ is the angle between the position of the ornithopter at t_0 and the position at $t_0 + \Delta t$ with respect to the target. Since the distance between the ornithopter and the target is unknown, ξ is also unknown. Then, using (10) and (11), the prediction of the desired orientation can be written by

$$\begin{aligned} \mathbf{r}(k+1|k) &= \Psi(k) + \epsilon(k+1) \\ &\approx \Psi(k) + \epsilon(k) + \xi(k) - \Delta\Psi(k+1) \\ &\approx \mathbf{r}(k) + \xi(k) - \Delta\Psi(k). \end{aligned}$$

Now, we can formulate a Kalman filter to predict the desired orientation angle as follows

$$\begin{aligned} \mathbf{r}(k+1|k) &= \mathbf{r}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{w}(k), \\ \epsilon(k) &= \mathbf{r}(k) + \mathbf{D}\mathbf{u}(k) + \mathbf{v}(k). \end{aligned}$$

Here, $\mathbf{u}(k) = (\Psi(k), \Psi(k-1))^{\mathsf{T}}$ is the input vector to the Kalman filter, the process noise $\mathbf{w}(k)$ and the measurement noise $\mathbf{v}(k)$ are independent zero mean Gaussian noises, $\mathbf{B}$ is the process weighting factor to compensate the unknown value of ξ[3] and defined by

$$\mathbf{B} = \begin{pmatrix} -\alpha_\psi & 0 & \alpha_\psi & 0 \\ 0 & -\alpha_\theta & 0 & \alpha_\theta \end{pmatrix},$$

where α_ψ is the process weighting factor for the yaw angle and α_θ is the process weighting factor for the pitch angle, and D is defined by

$$\mathbf{D} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}.$$

The value of α is determined by the state of target tracking. We have experimentally found that (α_ψ, α_θ) = (1.2, 1.2) is sufficient if the target is exposed on the infrared camera. The target, however, is occasionally not exposed on the camera after large external disturbances or large overshoots from the desired trajectory.[4] For this case, we need to have a dead reckoning algorithm to estimate the target location without any sensor input.

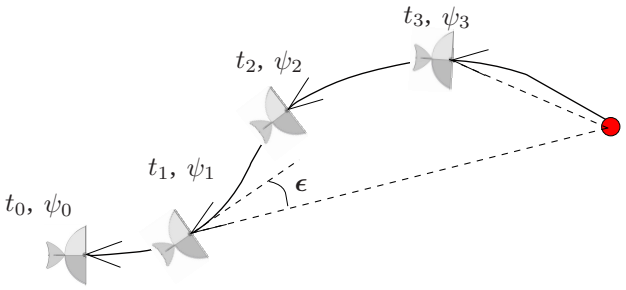


Fig. 6: Illustration of dead reckoning of target location. The two lines at the head of the ornithopter represent the field of view of the infrared camera, and the dot represents the location of the target.

When the target is not exposed on the camera, the innovation term in the Kalman filter is ignored since the measurement from the visual sensor is meaningless. Then, an algorithm for dead reckoning takes the place of target tracking as depicted in Fig. 5 to predict the target location.

Figure 6 illustrates an example of the ornithopter's trajectory when the ornithopter momentarily loses the target. At t = t_1, the target is lost, and for $t_1 \le t \le t_2$, the ornithopter flies away from the target, i.e., $\text{sign}(\epsilon_\psi) \neq \text{sign}(\Delta\psi)$. For $t_2 \le t \le t_3$, the target is still lost, but the ornithopter flies toward the target, i.e., $\text{sign}(\epsilon_\psi) = \text{sign}(\Delta\psi)$. The target is rediscovered at t = t_3.

The dead reckoning algorithm keeps the desired pitch angle unchanged while the target is lost, i.e., $r_\theta(k+1) = r_\theta(k)$. For the desired yaw angle, we have experimentally found that α_ψ = 2 is sufficient most of time when the ornithopter is flying away from the target.[5] When the ornithopter is turning toward the target while the target is still lost, we keep the desired yaw angle unchanged, i.e., $r_\psi(k+1) = r_\psi(k)$. The

[3]If the target is located infinitely far away from the ornithopter, the practical value of ξ would be zero. In this case, the process weighting factor, α, will be 1.

[4]The infrared camera used in this work has a narrow field of view, which is $\pm 20^\circ$ horizontally and $\pm 15^\circ$ vertically.

[5]With a constant value of α_ψ, the ornithopter cannot rediscover the target if the ornithopter flies past the target, or it loses the target at a close distance.

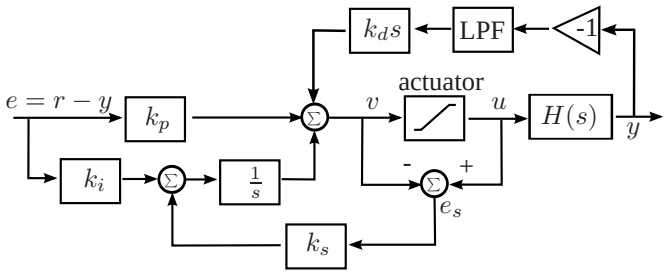


Fig. 7: Block diagram for the PID controllers

dead reckoning discussed so far is summarized as follows:

$$r_\theta(k+1) = r_\theta(k),$$
$$r_\psi(k+1) = \begin{cases} r_\psi(k) & \text{if } \mathrm{sign}(\epsilon_\psi) = \mathrm{sign}(\Delta\psi), \\ r_\psi(k) - 2\Delta\Psi & \text{otherwise.} \end{cases}$$

VI. CONTROL METHOD

Autonomous flight control of the ornithopter is implemented with the onboard sensors and microprocessor described in Section II. The desired angle and the orientation angle are fed into the regulator, shown in Fig. 5, consisting of two PID controllers running at 400 Hz. The PID controllers compute the input signals (two PWM signals) to drive the wings and the propeller. For the PID controllers, we assume that the propeller mainly contributes the yaw moment and the wings mainly contribute pitch moment and thrust.

The input signals for the ornithopter for thrust from the wings and steering moment from the propeller are computed based on the PID controller given by

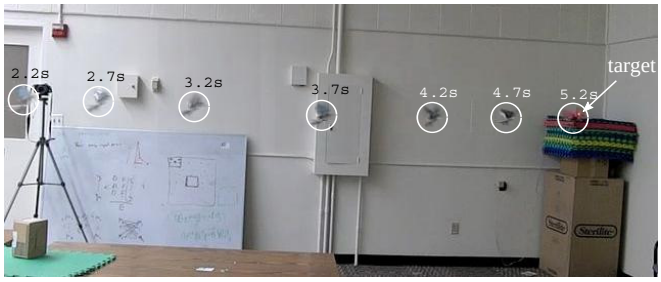
$$u = k_p(r - y) + k_i \int_0^t (r(\tau) - y(\tau))d\tau + k_d \left(\gamma\dot{r} - \dot{y}\right),$$

where u is the duty cycle for a motor, k_p, k_i, and k_d are, respectively, proportional, integral, and derivative gains for the PID controller, r is the desired angle, y is the measured orientation angle, and γ is the reference weight for the derivative controller. We set γ = 0 to avoid large transients in the control signal when the reference is changed. Setting γ = 0 also allows us to relocate the poles at zero to the left half plane. There are two such PID controllers running simultaneously for yaw and pitch regulation. A third order Butterworth low pass filter is implemented to reduce high frequency noise for $\dot{y}$. A discrete time anti-windup algorithm is also implemented by testing if the actuator is saturated; the PWM duty cycle to the motors, denoted by u, should stay between v_{min} and v_{max}, where v is the input to the plant. If the actuator is not saturated, i.e., $v_{min} < u < v_{max}$, we accumulate the errors for the integral controller. If the actuator is saturated, the accumulated errors remain unchanged. The controller algorithm discussed here is summarized in Fig. 7.

VII. EXPERIMENTAL RESULTS AND DISCUSSION

We conducted a number of experiments to verify the control method described in Section VI. First, we placed

(a)



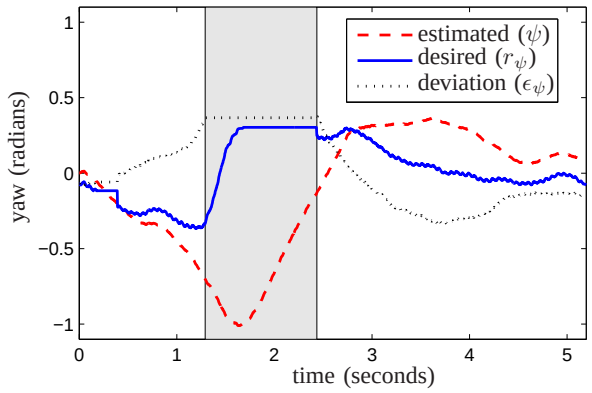
(b)

Fig. 8: Mosaics of the ornithopter flying toward the target. The numbers denote the time stamp (a) longitudinal view (b) lateral view.

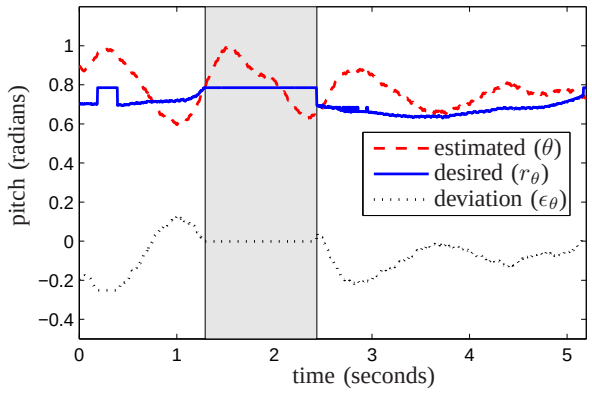
an infrared emitting target approximately 10 meters away from the initial location of the ornithopter. To test loss of the target, the ornithopter was released with an initial nonzero roll angle. Subsequently, the ornithopter could turn away from the target at the beginning of its flights and lost the target in one or two seconds. To inspect the trajectories of the ornithopter flights, we took movies of the ornithopter while it was flying toward the target. The mosaics of a flight toward the target in the longitudinal and lateral views are shown in Fig. 8.

The orientation angles (ψ and θ), the deviation angles (ϵ_ψ and ϵ_θ), the desired angles (r_ψ and r_θ), and the trajectory of the flight in Fig. 8 are plotted in Fig. 9. The ornithopter flew away from the desired yaw angle at the beginning of its flight as shown in Fig. 9a because of the initial nonzero roll angle. Subsequently, it lost the target as the target was no longer exposed on the camera. The loss of the target is represented by regions covered by gray shadows in Fig. 9. The constant values of the deviation angles in the gray area shown in Fig. 9a and Fig. 9b indicate that the deviation angles were not updated due to the loss of the target. As the ornithopter flew away from the target while the target was lost, the desired angle rapidly increased for 1.2 s $\leq t \leq$ 1.7 s because we set α = 2. When t was approximately 1.7 s, the ornithopter started turning back toward the target and the desired yaw angle stayed constant for $r_\psi(k+1) = r_\psi(k)$. As the yaw angle approached close to the desired yaw angle, the target was discovered by the camera at $t \approx$ 2.4 s. After a small overshoot, the yaw angle stayed close to the desired value as shown in Fig. 9a.

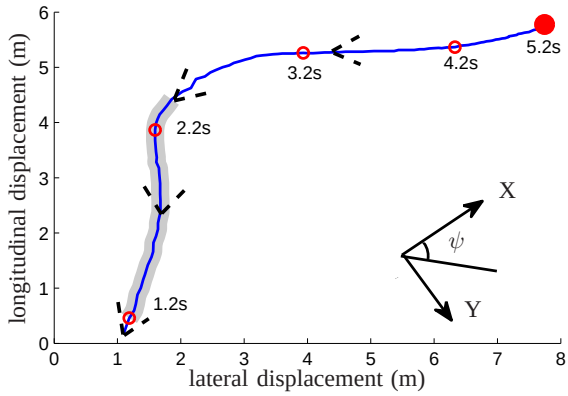
The desired pitch angle remained unchanged due to $r_\theta(k+$



(a)



(b)



(c)

Fig. 9: An experimental result of flight control of target seeking. The regions covered by gray shadows indicate that the target was lost. (a) yaw angle regulation (b) pitch angle regulation (c) the trajectory of the ornithopter flying toward the target. The dashed lines in (c) represent the field of view of the infrared camera, the numbers next to circles denote the time stamps, and the dot on the upper right corner represents the target.

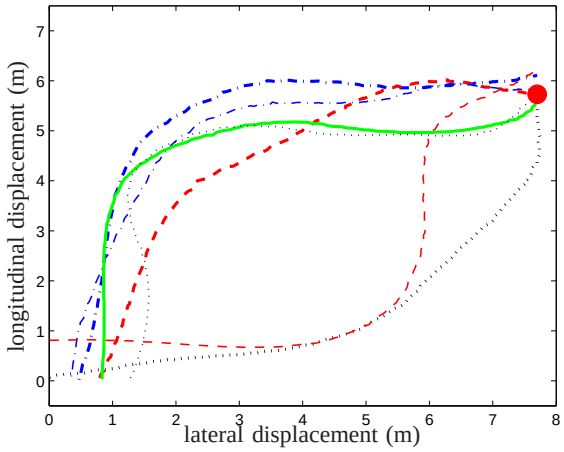


Fig. 10: A set of trajectories of ornithopter flights toward the target. The dot on the upper right corner represents the target.

$1) = r_\theta(k)$ while the target was lost. After the target was discovered at $t \approx 2.4$ s, the desired pitch angle was updated with the deviation angle. Although the pitch angle was oscillating with approximately 0.7 Hz because of the inherent dynamics of the ornithopter, it stayed close to the desired pitch angle as time elapsed as shown in Fig. 9b.

From the movies, we were able to extract the last 4 seconds of the path data of the ornithopter as shown in Fig. 9c.[6] The dot on the upper right corner in Fig. 9c represents the target location. The dashed lines represent the field of view of the infrared camera. It is evident from the trajectory that the target was not exposed on the camera momentarily, but the ornithopter still managed to turn toward the target and reached it at last.

We performed more experiments to investigate the accuracy of the flight control for target seeking as shown in Fig. 10. The set of ornithopter trajectories shows that the ornithopter successfully rediscovered the target after the target was lost, and the ornithopter finally reached the target for most trials. With 20 trials with different initial roll angles, the ornithopter reached 17 times within a radius of 0.5 m from the target.

VIII. CONCLUSIONS AND FUTURE WORK

In this paper, we have discussed the implementation of closed-loop attitude control of a small-scale flapping-wing robot using onboard sensors and a microcontroller without any external assistance. We have also discussed an algorithm to predict the desired orientation angle when the target is temporarily not observable. With these developments, we have presented the ornithopter flying toward an infrared emitting target using an onboard infrared camera. The 28 cm wing-span ornithopter was able to land within a radius of 0.5 m from the target with more than 85% success (N = 20).

[6]Due to the narrow field of view of the external cameras, we were not able to take movies for the beginning of the flight.

We are currently working on system identification of the ornithopter using vision based 3D motion capturing systems to simultaneously estimate the linear motions and angular motions of the ornithopter. Also, the direct measurement of acceleration, already integrated in the control board, will enable us to measure free flight aerodynamic forces. With the identification of system parameters, we believe we can implement a better control policy for higher level behaviors, such as multiple coordinated flight or obstacle avoidance.

IX. ACKNOWLEDGMENTS

The authors thank Paul Birkmeyer and Andrew Pullin for their help with editing pictures, Kevin Peterson for his Matlab script for video processing, and the members of the Biomimetic Millisystems Lab for their insightful comments on this work. This material is based upon work supported by the National Science Foundation under Grant No. IIS-0705429 and IIS-0931463. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF).

REFERENCES

[1] S. Avadhanula, R. Wood, E. Steltz, J. Yan, and R. Fearing, "Lift force improvements for the micromechanical flying insect," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, October 2003, pp. 1350–1356.

[2] S. Baek and R. Fearing, "Flight forces and altitude regulation of 12 gram I-Bird," in *IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, Tokyo, Japan, September 2010, pp. 454–460.

[3] B. Barshan and H. Durrant-Whyte, "Orientation estimate for mobile robots using gyroscopic information," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, vol. 3, Sep 1994, pp. 1867 –1874.

[4] A. E. Bryson, *Control of Spacecraft and Aircraft*. Princeton, NJ: Princeton University Press, 1994.

[5] J. L. Crassidis and F. L. Markely, "Unscented filtering for spacecraft attitude estimation," *Journal of Guidance, Control, and Dynamics*, vol. 26, pp. 536–542, August 2003.

[6] G. de Croon, K. de Clerq, R. Ruijsink, B. Remes, and C. de Wagter, "Design, aerodynamics, and vision-based control of the DelFly," *International Journal of Micro Air Vehicles*, vol. 1, no. 2, pp. 71 – 97, 2009.

[7] G. de Croon, E. de Weerdt, C. de Wagter, and B. Remes, "The appearance variation cue for obstacle avoidance," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2010, pp. 1606–1611.

[8] X. Deng, L. Schenato, and S. S. Sastry, "Flapping flight for biomimetic robotic insects: part II - flight control design," *IEEE Trans. on Robotics*, vol. 22, no. 4, pp. 789–803, 2006.

[9] X. Deng, L. Schenato, W. C. Wu, and S. S. Sastry, "Flapping flight for biomimetic robotic insects: part I - system modeling," *IEEE Trans. on Robotics*, vol. 22, no. 4, pp. 776–788, 2006.

[10] R. Fearing, K. Chiang, M. Dickinson, D. Pick, M. Sitti, and J. Yan, "Wing transmission for a micromechanical flying insect," in *IEEE International Conference on Robotics and Automation*, vol. 2, 2000, pp. 1509–1516.

[11] C. W. Kang and C. G. Park, "Attitude estimation with accelerometers and gyros using fuzzy tuned Kalman filter," in *the European Control Conference*, Budapest, Hungary, August 2009, pp. 3713–3718.

[12] M. Keennon, K. Klingebiel, A. Andryukov, B. Hibbs, and J. Zwaan, "Air vehicle flight mechanism and control method," Patent 20 100 308 160, December, 2010.

[13] Z. A. Khan and S. K. Agrawal, "Design and optimization of a biologically inspired flapping mechanism for flapping wing micro air vehicles," in *IEEE International Conference on Robotics and Automation*, Rome, Italy, April 2007, pp. 373–378.

[14] D. B. Kingston and A. W. Beard, "Real-time attitude and position estimation for small UAVs using low-cost sensors," in *AIAA 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit*, 2004, pp. 2004–6488.

[15] D. Lentink, "Exploring the biofluiddynamics of swimming and flight," Ph.D. dissertation, Wageningen University, 2008.

[16] D. Lentink, S. R. Jongerius, and N. L. Bradshaw, *Flying Insects and Robots*. Edited by Dario Floreano, Jean-Christophe Zuffery, Mandyam V. Srinavasan, and Charlie Ellington, Springer, 2009, ch. 14: The Scalable Design of Flapping Micro-Air Vehicles Inspired by Insect Flight, pp. 185–205.

[17] O. M. O'Reilly, *Intermediate Dynamics for Engineers*. New York, NY: Cambridge University Press, 2008.

[18] J. H. Park and K.-J. Yoon, "Designing a biomimetic ornithopter capable of sustained and controlled flight," *Journal of Bionic Engineering*, vol. 5, no. 1, pp. 39–47, March 2008.

[19] K. S. Shigeoka, "Velocity and altitude control of an ornithopter micro aerial vehicle," Master's thesis, The University of Utah, 2007.

[20] A. Tayebi, S. McGilvray, A. Roberts, and M. Moallem, "Attitude estimation and stabilization of a rigid body using low-cost sensors," in *IEEE Conference on Decision and Control*, New Orleans, LA, December 2007, pp. 6424–6429.

[21] R. Tedrake, Z. Jackowski, R. Cory, J. W. Roberts, and W. Hoburg, "Learning to fly like a bird," Massachusetts Institute of Technology Computer Science and Artificial Intelligence Lab, Tech. Rep., 2006.

[22] F. van Breugel, Z. E. Teoh, and H. Lipson, *Flying Insects and Robots*. Edited by Dario Floreano, Jean-Christophe Zuffery, Mandyam V. Srinavasan, and Charlie Ellington, Springer, 2009, ch. 13: A Passively Stable Hovering Flapping Micro-Air Vehicle, pp. 171–184.

[23] WiiBrew, "Wiimote," 2011, (accessed on 20-March-2011). [Online]. Available: http://wiibrew.org/wiki/Wiimote

[24] R. Wood, "Liftoff of a 60mg flapping-wing MAV," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, November 2007, pp. 1889 –1894.

[25] J.-C. Zufferey, A. Beyeler, and D. Floreano, *Flying Insects and Robots*. Edited by Dario Floreano, Jean-Christophe Zuffery, Mandyam V. Srinavasan, and Charlie Ellington, Springer, 2009, ch. 6: Optic Flow to Steer and Avoid Collision in 3D, pp. 73–86.