Flight Control of an Insect-like Tailless Flapping Wing Nano-Robot around Hover

Bart Helsen[1], Thomas Roelandt[1]*, Merijn Floren[1,2] and Dirk Vandepitte[1]
[1]KU Leuven, Celestijnenlaan 300 box 2420, 3001 Leuven, Belgium
[2] Flanders Make@KU Leuven, Leuven, Belgium

http://www.imavs.org/

ABSTRACT

This work describes a control system for a tailless insect-like flapping wing aerial vehicle. The control system is initially implemented in simulation. An experimentally validated rigid wing quasi-steady aerodynamics model represents the vehicle flight dynamics in simulation. A cascaded PID architecture is used to control vehicle flight velocity and attitude. The PID tuning process is automated through a constrained nonlinear optimization process. The attitude estimation process uses onboard accelerometer and gyroscope readings. The accelerometer readings require a low-pass filter to ignore noise from flapping wing related body pitching oscillations. The performance of a complementary filter and extended Kalman filter are considered for the attitude estimation strategy. The flight simulation controller successfully stabilizes the vehicle attitude and follows reference flight maneuvers around hover. The attitude estimation and attitude controller are verified experimentally on a single cage gimbal setup that isolates one rotational degree of freedom. The drone stabilizes itself around the upright position, and successfully maintains a reference attitude.

1 INTRODUCTION

The capability of flying insects, bats and small birds to maneuver and navigate with ease inside cluttered and confined environments has inspired the development of insect-like flapping wing aerial vehicles (FWAVs). These vehicles produce thrust by mimicking the complex wing motion of their natural counterparts. Unsteady aerodynamic phenomena that occur in low Reynolds number flapping flight produce significantly higher lift values than fixed wing or rotary wing motion [1]. Advances in lightweight miniaturized electronics have enabled the construction of insect-like FWAVs ranging in size from the 102 g Flapper Drone [2] down to the 80 mg Robobee [3]. Like insects, the wing stroke motion takes place inside the body horizontal plane, which gives the FWAV the capacity to hover in place. Because of their small size, lightweight soft material wings and relatively low wing tip speeds, insect-like flyers are at lower risk of (causing) damage in the event of a crash. Instead, these FWAVs tend to bounce off of the object they collide with, after which they continue their flight [2, 4]. Also, the flapping wing motion makes the wings unlikely to tangle on loose objects like cables and plant leaves. The combination of hovering flight and collision resilience makes insect-like FWAVs especially well suited to be deployed inside confined and cluttered environments. Possible applications are the inspection of difficult to reach industrial installations even in small cavities, crop health monitoring inside greenhouses, and the exploration of partially collapsed buildings after a natural disaster.

The test vehicle is the KUlibrie, a tailless two-winged flapping wing micro air vehicle with passive wing inclination that is in development at KU Leuven [5, 6]. The vehicle is inspired by the propulsion principles of hummingbirds and insects. The KUlibrie fits within a group of tailless, two-winged insect-like FWAVs of similar size, generally with a 10-25 g body mass and 15-20 cm total wingspan [7]. Maintaining controlled flight for this group of vehicles is an ongoing challenge. Insect FWAV flight is inherently unstable [8, 9], so a flight controller needs to continuously perform small control actions to stabilize the vehicle. Also, the small size imposes stringent weight restrictions that impede the number of actuators and onboard sensors and the amount of computing power that can be installed. Some examples of FWAV prototypes successfully maintain free flight around hover [2, 3, 4, 5, 10]. However, further improvements are required in robustness against modeling inaccuracies or uncertainties and time-dependent flight characteristics before the technology is ready for use in real-world applications.

This work employs a free flight simulation environment to evaluate attitude estimation and flight control of an insect-like FWAV around hover. Attitude estimation and control are validated on an experimental setup, with the prototype drone mounted inside a gimbal system. The control architecture in this study is based on PID, because of its common use as a relatively simple yet effective technique with low computational load. The simulation environment and the gimbal setup will later serve as building blocks for the future development and evaluation of more advanced control techniques, which can then be compared to the PID controller performance.

*Email address: thomas.roelandt@kuleuven.be

http://www.imavs.org/

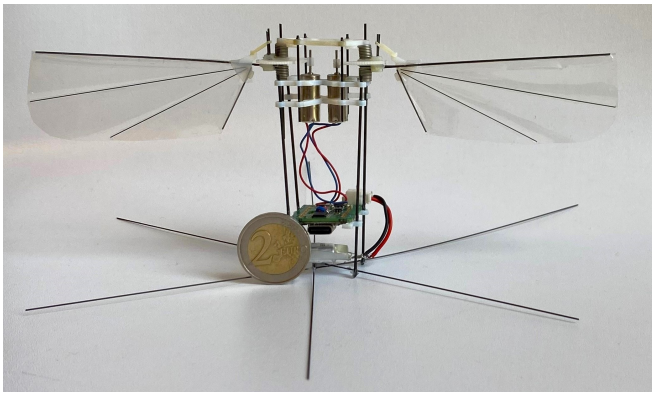

Figure 1: KUlibrie drone prototype, with a 2 euro coin added for size reference.

2 KULIBRIE FLAPPING WING SYSTEM

2.1 *Design and key parameters*

Figure 1 shows the latest KUlibrie prototype. The stroke motion of each wing is actuated independently by a brushless DC-motor, connected to the wing by a direct drive gear transmission. Two torsion springs are mounted at the root of the wing stroke axis. The spring stiffness is set such that the natural frequency of the flapping wing system is close to the stroke frequency, resulting in a resonant effect that improves driveline performance. Kinetic energy is stored in the springs as potential energy, and released again at the start of the subsequent stroke. The driveline design and performance are studied in detail in [5, 6]. The wing inclination motion, which is dominant in determining the value of the angle of attack, is passive. It is the result of the interaction between aerodynamic forces on the wing surface, counter-torque from a rubber band attached at the base of the wing, and inertial effects. This rubber band also initiates the pronation and supination at the end of upstroke and downstroke respectively, ensuring a positive angle of attack at the start of the next stroke [11].

Drone mass is 15.8 g and total wingspan is 195 mm. Table 1 lists key design parameters. The body frame is constructed with carbon fiber and 3D-printed acrylic resin components. The wings feature a 10 μm Mylar membrane, reinforced with a carbon fiber veined skeleton. The stroke frequency is set at 19 Hz. The current prototype has an onboard 'Seeed Studio XIAO nRF52840 Sense' processing unit with a built-in 3-axis gyroscope and 3-axis accelerometer IMU. A 100 mAh LiPo battery provides the system with sufficient energy for approximately 90 seconds of operation.

Compared to FWAV prototypes that use linkage mechanisms and one motor to drive the motion of both wings, the two decoupled motors on the KUlibrie provide greater control over its wing motion. The ability to alter the stroke amplitude and mean stroke angle for each wing individually theoretically provides sufficient controllability for free flight maneuvering without the need for additional drivers. Instead, flight can be realized by only control of the wing stroke motion. This pure wing stroke motion-based control strategy is similar to that of the Purdue Robotic Hummingbird [12] and the Robobee [3]. The simplicity of the direct drive stroke motion-based control strategy lends itself well for further vehicle down-scaling, and may be necessary for truly insect scale FWAV prototypes [3]. A challenge that comes with this control strategy is that the same motors responsible for providing vehicle thrust are also used to perform attitude control and velocity control. This means that some of the available payload capacity is sacrificed for flight maneuvering. Care must also be taken that the combined control signals do not saturate the motors. The control signals and prevention of motor saturation are discussed further in the next sections.

Parameter	Value	Unit
Total wingspan	195	mm
Single wing length	83.5	mm
Mean chord length	20	mm
Aspect ratio	4.175	-
y-distance CoG - CoR	10	mm
z-distance CoG - CoR	29.5	mm
Total body mass	15.8	g
Single wing mass	0.2	g

Table 1: Drone body and wing design parameters.

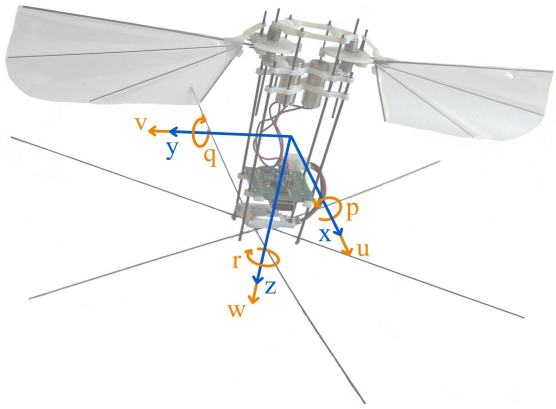

Figure 2: Body translation components $[u, v, w]$ and rotation components $[p, q, r]$, defined with respect to the body-fixed coordinate system xyz.

2.2 *Flight simulation model*

A discrete-time flight simulation model in MATLAB Simulink emulates the flapping wing drone prototype. Model input parameters include the drone dimensions, inertia (cfr. Table 2) and motor characteristics. Wing stroke motion is calculated based on the motor voltage inputs and wing dimensions [6]. Wing inclination and deviation motion are

http://www.imavs.org/

described based on earlier experimental results [6]. Aerodynamic force production is computed using an experimentally validated rigid wing quasi-steady aerodynamics model [6, 13]. The discrete time step of the simulation is set to 1% of the stroke cycle period to capture the fast body oscillations that are present in two-winged FWAV systems, occurring at stroke frequency [7, 14]. Finally, body motion is computed using rigid body equations of motion. The drone body has three translation (u, v, w) and three rotation (p, q, r) degrees of freedom (DOFs), resulting in the dynamics model:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{V}(t)), \tag{1}$$

with $\mathbf{x}(t) = \begin{bmatrix} u(t) & v(t) & w(t) & p(t) & q(t) & r(t) \end{bmatrix}^\top$, as indicated in Figure 2, and $\mathbf{V}(t) = \begin{bmatrix} V_{\text{left}}(t) & V_{\text{right}}(t) \end{bmatrix}^\top$ the voltages applied to the left and right wing, respectively. The over-dot indicates a derivative with respect to time variable t. The explicit dependence on t is further omitted where it is implicitly clear. The reader is referred to [6, 14] for a detailed derivation of the dynamics model $\mathbf{f}$ in Equation 1.

2.3 *Control actions*

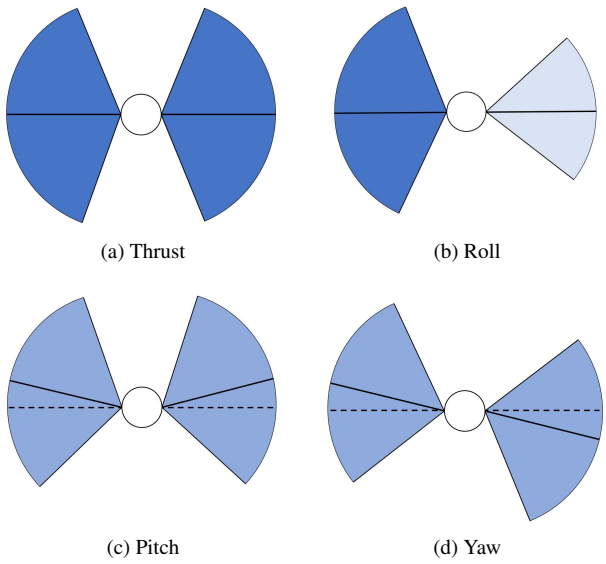

Figure 3: Top view of control action effects on stroke amplitude and mean stroke angle (solid black line). Stroke amplitude determines the mean lift force. A darker shade indicates a higher lift force.

The voltages $\mathbf{V}$ are applied to the two motors in the form of block waves. The block waves for the left and right wing are set independently. A non-symmetric setting generates torque on the drone body. Specifically, the direct drive configuration controls both the stroke amplitude and the mean stroke angle for each wing separately through a square wave with variable amplitude and mean value:

$$V_{\text{left}}(t) = \begin{cases} V_{I0} + dV + V_0 & \tau < 0.5 + ds \\ -(V_{I0} + dV - V_0) & 0.5 + ds < \tau < 1 \end{cases} \tag{2a}$$

$$V_{\text{right}}(t) = \begin{cases} V_{I0} - dV + V_0 & \tau < 0.5 - ds \\ -(V_{I0} - dV - V_0) & 0.5 - ds < \tau < 1 \end{cases} \tag{2b}$$

$\tau = ft$ represents normalized time, ranging from $\tau = 0$ (start of upstroke) to $\tau = 1$ over one full stroke cycle. ds marks a deviation from the center of the time interval of a full stroke. The individual components of Equation 2 affect the motion of the vehicle in the following way (see also Figure 3):

Thrust: An increase/decrease in motor voltage amplitude V_{I0} results in an increase/decrease in lift production. If the increment in stroke amplitude for the two wings is equal, net torque is zero and only total lift production changes.

Roll: An asymmetric increment dV of the motor voltage amplitudes increases lift produced on one wing, and decreases lift on the other wing. This difference in lift production results in a net rolling torque.

Pitch: A shift in mean motor voltage V_0 results in a symmetric change in the mean stroke angle. This produces a longitudinal shift of the mean lift vector for each wing. The result is a net pitching torque.

Yaw: A shift ds in duration of upstroke versus downstroke also shifts the mean stroke angle. Because the wing stroke axes do not coincide with the drone center of gravity (CoG), the length of the moment arm between the aerodynamic drag vector on each wing and the drone CoG depends on the stroke angle. When an asymmetric shift in mean stroke angle is applied, this time-varying moment arm for the two wings results in a net yawing torque. For small alterations, the yawing torque scales approximately linearly with the shift in mean stroke angle [15].

The four control actions result in an underactuated flight system. The rolling motion is coupled with lateral translation, and the pitching motion is coupled with longitudinal translation. The driveline configuration also allows for control actions based on stroke frequency modulation. However, the principle of resonance support sets a number of constraints on system control. Deviation of the actuation frequency from the set resonance frequency affects both system efficiency and system response [6].

3 CONTROL METHODOLOGY

3.1 *Cascaded flight controller layout*

Figure 4 shows the controller layout incorporated into the free flight simulation. An outer-loop velocity controller computes the reference input for an inner-loop attitude controller, based on the desired flight velocity. The attitude controller

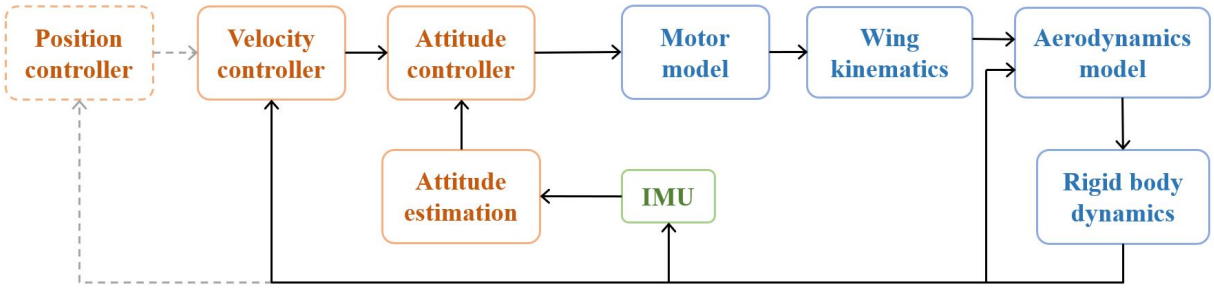

Figure 4: Schematic overview of the closed-loop control architecture in the flight simulation environment.

uses the output of an attitude estimation algorithm that is based on onboard IMU readings. The attitude estimation and control methods are discussed in the next sections. The cascaded flight controller may be expanded to be preceded by an additional outer-loop position controller that computes velocity controller inputs based on a reference flight trajectory. The expansion to include position control is highlighted in dashed lines in Figure 4. Since the onboard IMU does not measure the vehicle velocity, the velocity controller directly uses the actual velocity components $[u, v, w]$ of the vehicle. In future work, velocity estimation may be based on measurements from an external multi-camera configuration [3], or onboard camera system [16].

3.2 *Attitude estimation*

The attitude is estimated based on accelerometer and gyroscope readings. Assuming body accelerations $a_{x,y,z}$ are small in comparison to gravitational acceleration g, the accelerometer readings can be used to estimate the body roll and pitch angles:

$$\begin{cases} \Phi_{ac} = \arctan\left(\frac{a_y}{a_z}\right) \\ \Theta_{ac} = \arcsin\left(\frac{a_x}{g}\right) \end{cases} . \tag{3}$$

The gyroscope measures the velocities of the rotational DOFs, which are transformed from the body reference frame to a global reference frame according to:

$$\begin{cases} \dot{\Phi}_{gy} = p + (q \sin \Phi_{gy} + r \cos \Phi_{gy}) \tan \Theta_{gy} \\ \dot{\Theta}_{gy} = q \cos \Phi_{gy} - r \sin \Phi_{gy} \\ \dot{\Psi}_{gy} = (q \sin \Phi_{gy} + r \cos \Phi_{gy}) \sec \Theta_{gy} \end{cases} . \tag{4}$$

$[\Phi, \Theta, \Psi]$ respectively represent body roll, pitch, and yaw. Biases in the gyroscope measurements make its attitude estimation sensitive to drift. Gyroscope readings are modeled with additional Gaussian noise of variance 10^{-4} rad^2/s^2 and mean value (i.e. bias) of 0.3 rad/s. To limit the effect of bias on the gyroscope attitude estimation, the measurements are passed through a High-Pass Filter (HPF). The HPF excludes the DC biases from the gyroscope readings.

FWAVs are known to exhibit fast body pitching oscillations that occur at the stroke frequency [17, 10, 12]. These oscillations result in noisy accelerometer readings, that impede pitch angle estimation. A Low-Pass Filter (LPF) removes the effect of body pitching oscillations on the accelerometer readings. The LPF cutoff frequency is set to 1 Hz.

Figure 5 shows the attitude estimation strategies that are considered. The first strategy is a complementary filter, for which the attitude estimation is a weighted sum of the estimations based on the accelerometer and gyroscope readings. By varying the weighting factor α between 0 and 1, the estimation relies more heavily on the gyroscope or on the accelerometer. The complementary filter is considered because of its simplicity, resulting in a low computational load.

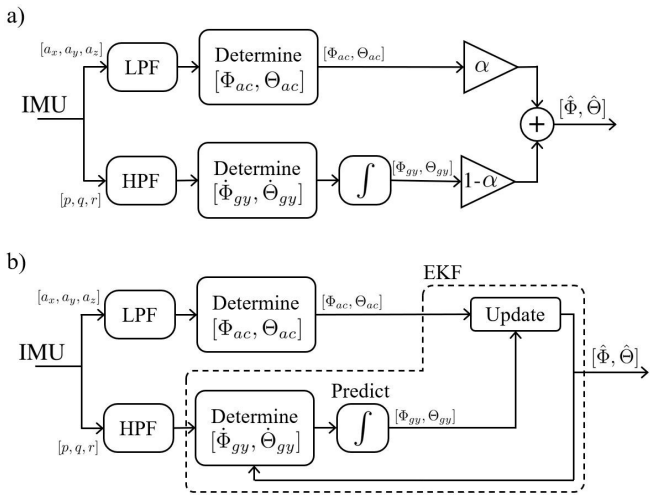

Figure 5: Schematic representation of attitude estimation methods: a) Complementary filter, b) Extended Kalman filter.

The second attitude estimation strategy uses an extended Kalman filter (EKF). The EKF is similar to the complementary filter, as its attitude estimation is based on a weighted average of the accelerometer and gyroscope readings. In this setting, the EKF uses the accelerometer estimates (Equation 3) to update its integrated gyroscope predictions (Equation 4). The main difference between the EKF and complementary filter, is that the EKF continuously updates the weight it attributes to each sensor reading based on the perceived process noise and measurement noise. Compared to

http://www.imavs.org/

http://www.imavs.org/

the complementary filter, the EKF is expected to be more robust to measurement noise and uncertainty, at the cost of a higher computational load.

Note that the attitude estimator only determines the estimated roll and pitch angle. Since the accelerometer readings are unaffected by the yaw angle, only the gyroscope readings provide information on the yaw rate. This means that the yaw rate measurement remains sensitive to sensing errors. Therefore, the real yaw angle is used as input for the yaw controller during the flight controller simulations. In future work, yaw sensing may be improved by the addition of an onboard magnetometer or camera.

3.3 Attitude control

The attitude controller has a PID architecture. PID is a popular control method in FWAV flight control [16, 10, 12] because of its simplicity and low computational load, which result in lower onboard memory requirements and allow for high sampling rates. Roll, pitch and yaw are controlled separately by three individual PID controllers, with the controller outputs corresponding to the components of Equation 2:

$$dV(t) = K_{P_1} e_\Phi(t) + K_{I_1} \int e_\Phi(t)\mathrm{d}t + K_{D_1} \frac{\mathrm{d}e_\Phi(t)}{\mathrm{d}t}, \quad (5a)$$

$$V_0(t) = K_{P_2} e_\Theta(t) + K_{I_2} \int e_\Theta(t)\mathrm{d}t + K_{D_2} \frac{\mathrm{d}e_\Theta(t)}{\mathrm{d}t}, \quad (5b)$$

$$ds(t) = K_{P_3} e_\Psi(t) + K_{I_3} \int e_\Psi(t)\mathrm{d}t + K_{D_3} \frac{\mathrm{d}e_\Psi(t)}{\mathrm{d}t}, \quad (5c)$$

where K_{P_i}, K_{I_i} and K_{D_i}, with $i = \{1, 2, 3\}$, are the respective PID gains. Moreover, the errors in the roll, pitch, and yaw angles are represented as $e_\Phi(t) = r_\Phi(t) - \Phi(t)$, $e_\Theta(t) = r_\Theta(t) - \Theta(t)$ and $e_\Psi(t) = r_\Psi(t) - \Psi(t)$, respectively. The reference signals r_Φ, r_Θ and r_Ψ are typically provided by the outer-loop velocity controller (see Figure 4). Finally, note that V_{I0} is not included in the attitude controller since it does not affect the orientation of the vehicle.

3.4 Determination of PID parameters

This section focuses on the setting of the PID gains of Equation 5. The gains are first obtained in an automated manner inside the flight simulation environment. The PID parameters are subsequently used as initial values for the experimental attitude controller, with further tuning being carried out manually.

A constrained nonlinear optimization problem is formulated to automate the tuning of the PID parameters. Specifically, the PID gains are iteratively adjusted based on a mean squared error objective over time span T_s. The gain tuning optimization problem is formulated around hover, i.e., $r_\Phi(t) = r_\Theta(t) = r_\Psi(t) = 0$, and is initialized with a roll and yaw angle of $\Phi = \Psi = 5°$ that represent an initial disturbance away from hovering conditions. The optimization problem is formally defined as:

$$\begin{aligned} \underset{K_{P_i}, K_{I_i}, K_{D_i}}{\text{minimize}} \quad & \frac{1}{T_s} \int_0^{T_s} (e_\Phi^2(t) + e_\Theta^2(t) + e_\Psi^2(t))\mathrm{d}t \\ \text{subject to} \quad & \\ \text{Eqs. } 1, 2, 3, 4, 5 \quad & \forall t \in [0, T_s], \\ K_{P_i}, K_{I_i}, K_{D_i} > 0 \quad & i = \{1, 2, 3\}. \end{aligned} \quad (6)$$

This formulation tunes all controller gains simultaneously, and easily re-tunes the controller in case a design alteration is made to the FWAV. The constraints in Equation 6 ensure that the PID gains are positive, and that the combined contribution of the PID outputs (according to Equation 2) does not exceed the maximum motor voltage $V_{max} = 3.4$ V. If at any time during simulation motor voltages V_{left} or V_{right} would exceed V_{max}, then voltage is made to saturate. Attitude estimation is achieved through the EKF. The problem is implemented in MATLAB using the built-in *fmincon* function with the interior-point algorithm. When the control system is extended to include velocity control, the optimization method in Equation 6 can be repeated to tune the velocity controller gains. In this case, the error terms are replaced by $e_u(t)$, $e_v(t)$, and $e_w(t)$.

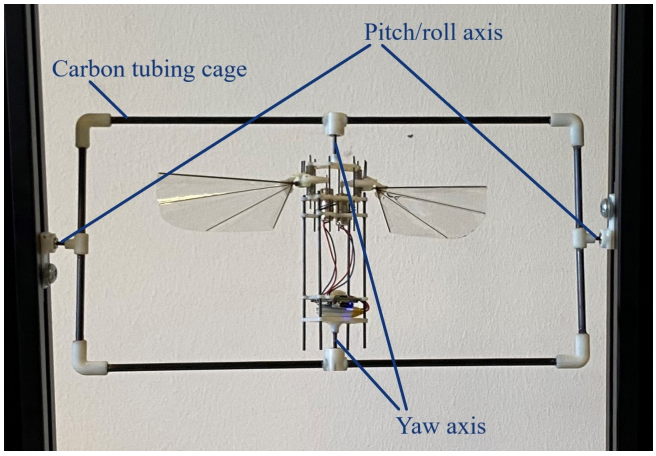

Figure 6: Single cage gimbal setup.

Moment of inertia (kgm^2)	Drone	Drone inside gimbal
I_{xx} (roll)	14.7×10^{-6}	75.6×10^{-6}
I_{yy} (pitch)	13.6×10^{-6}	74.5×10^{-6}
I_{zz} (yaw)	1.9×10^{-6}	2.1×10^{-6}

Table 2: Moments of inertia.

3.5 Experimental attitude controller setup

The real-life attitude controller tuning is done manually. To simplify the tuning process, each orientation angle is tuned

http://www.imavs.org/

separately by rotating the drone only about one axis at a time. For small corrections around hover, the three degrees of freedom in rotation can be considered to be uncoupled [9]. Roll, pitch and yaw are therefore controlled through three separately tuned PID controllers. The settings resulting from the simulated PID controller are used as a starting point for the physical controller tuning process.

Figure 6 shows the experimental attitude controller tuning setup. The drone is mounted inside a single cage gimbal with ball bearings. The inner rotation axis, the yaw axis, extends along the body vertical through the center of the drone. The outer rotation axis corresponds to pitch or roll, depending on the direction in which the drone is mounted. The outer rotation axis is positioned slightly below the drone center of gravity. This means that the upright position is an unstable equilibrium and the drone tends to flip over. Therefore, if the drone manages to remain upright during operation, this is only possible through a properly operating attitude controller. By addition of a fixture over the yaw axis, the yaw angle can be locked, thereby only leaving a pure rolling or pure pitching motion. Similarly, the outer rotation axis can be locked, only leaving yaw motion. The cage is a lightweight structure with 4mm carbon tubing. The cage material and dimensions are a compromise between a lightweight construction to minimize added inertia and a requirement of sufficient stiffness to avoid bending of the cage.

Table 2 compares the inertia for the free drone and the gimbal configuration. Compared to free flight the yaw motion is relatively unaffected, however there is a significant increase in the moments of inertia linked to pitch and roll. Rotational acceleration is lower when the drone is mounted inside the gimbal, giving the attitude control more time to respond to an error in orientation. Also, the increased moments of inertia imply that a larger counter torque is needed to counteract a pitch or roll velocity, thus inducing a larger control action. The gimbal structure is a test configuration setup to evaluate the effectiveness of the attitude estimation and control actions.

During the experiment, attitude estimation computations are performed on the onboard processing unit. The attitude estimation sampling rate is 200 Hz. State estimation data are sent to a measurement laptop via Bluetooth for a posteriori processing and analysis.

4 RESULTS

4.1 Attitude estimation strategies

The performance of the complementary filter and the EKF is compared through an example flight simulation of 10 s, with the drone released from a hovering condition. At $t = 2$ s the vehicle experiences a positive rolling torque disturbance, and at $t = 6$ s the vehicle experiences a positive pitching torque disturbance. Figure 7 shows the actual body roll and pitch angles, as well as the estimated values and estimation errors from the EKF and different complementary filter settings. The fast body pitching oscillations are clearly visible and have an amplitude of approximately 2.5°.

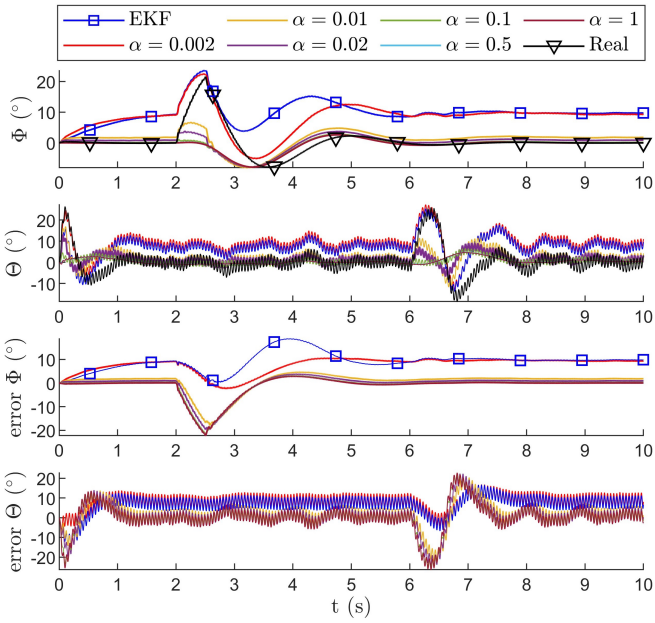

Figure 7: Comparison of roll (Φ) and pitch (Θ) estimation strategies.

- At very low values of α ($\alpha = 0.002$), the complementary filter relies heavily on the gyroscope readings for its attitude estimation. The estimator shows a good correlation at times when the system is perturbed. However, when the system is at rest, a bias term is present in the attitude estimation.
- When α increases, the complementary filter relies more on the accelerometer readings. This removes the bias on the attitude estimation when the system is at a steady state. However, the attitude estimator becomes less responsive to fast changes in orientation. For α in the range from 0.1 to 1, the attitude estimator behavior is approximately identical.
- There is a significant difference in attitude estimation performance between $\alpha = 0.002$ and $\alpha = 0.01$. An intermediate value for α may lead to an attitude estimator that provides a better compromise between accurate attitude estimation during system perturbation and a low bias when the system is at rest.
- The behavior of the EKF is similar to that of the low α complementary filter. When the system is in a steady state, a bias is present in the attitude estimation that is only slightly lower than that of the low α complementary filter. When the system is perturbed, the EKF has a faster response than the high α value complementary filter. For the pitching motion, with the accelerometer readings affected by body pitching oscillations, the EKF behavior is very similar to

http://www.imavs.org/

the $\alpha = 0.002$ complementary filter. For roll estimation, the EKF appears to rely more heavily on accelerometer readings than the $\alpha = 0.002$ complementary filter, resulting in a more damped response to a roll perturbation.

The EKF is selected for further integration into the simulated and physical flight controller schemes. The EKF shows an increased responsiveness to pitching perturbations compared to high α complementary filters. Also, the EKF is expected to be more robust to general process noise and sensor noise. The bias that is present with the EKF is expected to be remedied when the attitude controller is preceded by a velocity controller.

4.2 Attitude and velocity control simulation results

The combination of velocity control and attitude control is initially tested without position control. Figure 8 shows two consecutive controlled flight maneuvers. The vehicle is released in hover, with a small initial roll perturbation. After $t = 5$ s, the vehicle is commanded to accelerate at a constant rate to a forward flight velocity of $u = 0.5$ m/s, measured in the body coordinate system, and then maintain this forward flight velocity until $t = 10$ s, after which it should decelerate back to hover. The other body velocity components need to remain at $v = w = 0$ m/s. At $t = 15$ s, the vehicle is commanded to accelerate at a constant rate to a lateral flight velocity of $v = 0.5$ m/s, and then maintain this velocity until the end of the simulation.

The vehicle successfully stabilizes to follow the reference velocity signal. The outer-loop velocity controller initiates the forward flight maneuver by altering the pitch angle reference input in order to tilt the thrust vector forward. Similarly, the lateral flight maneuver is initiated by adjusting the roll angle reference. The lateral velocity combined with a non-zero body pitch angle, results in a yawing torque. During initiation of the lateral flight maneuver, the yaw angle increases to a maximum deviation of 18°, after which the yaw controller manages to return the yaw angle to its reference value. The velocity controller accommodates for the bias in the pitch and roll estimation angles by properly adjusting the reference signals that are sent to the roll and pitch controllers.

4.3 Experimental validation

Figure 9 shows the results for the experimental roll and pitch control respectively. Each time the drone is commanded to initially maintain an upright orientation for 5 seconds. Next, the drone should tilt its attitude by 20° and maintain the new position. Figure 9 only shows the estimated roll and pitch values for this maneuver. The controller settles at a steady state error of approximately 4° for the estimated roll angle, and 6° for the estimated pitch angle. The steady-state errors in the experimental data may be explained by the lower integral gains and higher derivative gains on the physical system compared to the simulation (cfr. Table 3). The lower integral gains and higher derivative gains prevent large oscillations of the system, but at the cost of a longer settling time. The attitude controller experiment is repeated also for a step input, and for a ramp input tilt where the drone subsequently needs to return to an upright orientation. Figures related to the other experimental attitude control cases are available in Appendix A. These figures show that the controller is able to maintain hover, stabilize a severe step input, return from a tilted state to upright, and perform a yawing motion in either direction. Video footage of the attitude control experiments for roll, pitch, and yaw is available in the supplementary material.

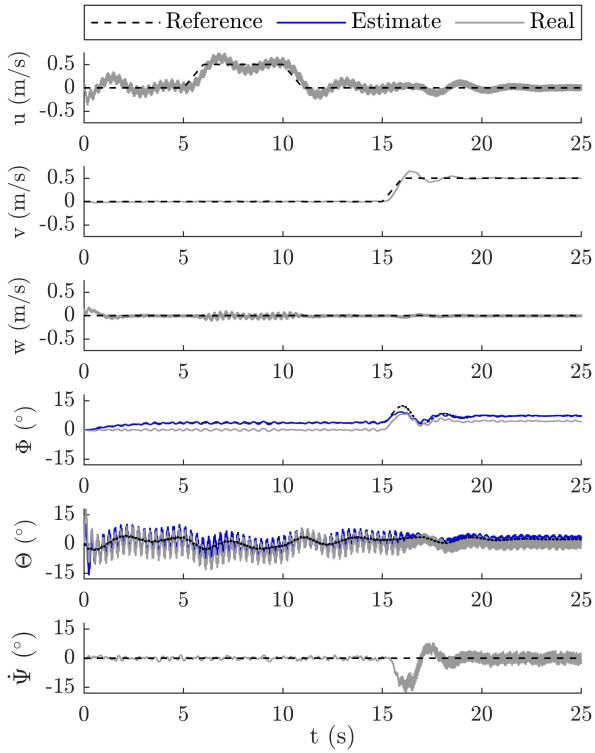

Figure 8: Velocity control during simulation flight maneuvering.

4.4 Study limitations

Flight simulation in this study assumes a system with known, time-independent flight characteristics that can be represented through rigid wing aerodynamics. In practice, the extent to which assumptions such as rigid wings affect model accuracy remains uncertain. Also, the dynamic characteristics of the FWAV may change due to design alterations or due to damage after a collision. The IMU readings may be influenced by additional sources of inaccuracies such as a variable bias, temperature fluctuations, or vibration rectification, increasing the measurement noise. Finally, motor heating is known to affect the system response to control actions, leading to time-varying dynamic characteristics. Future studies should consider uncertainties in the system dynamics, as

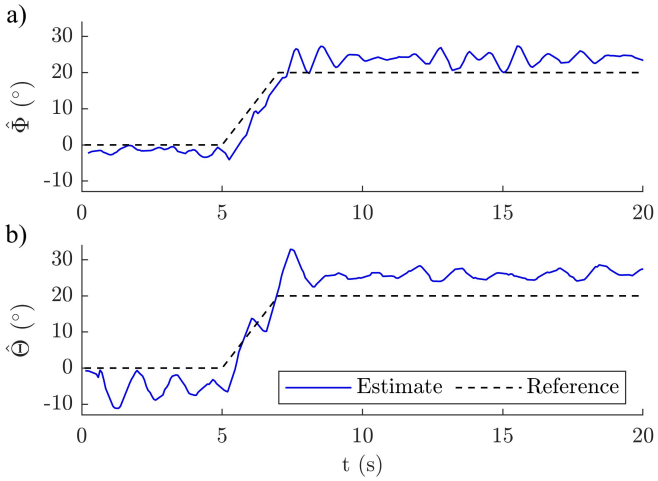

Figure 9: Attitude controller measurements for ramped inputs: a) roll estimation, b) pitch estimation.

well as their time-varying nature, for the implementation of a robust flight control strategy.

	Simulation			Experimental		
	K_P	K_I	K_D	K_P	K_I	K_D
Roll (Φ)	0.782	3.089	0.051	0.9	0.2	0.3
Pitch (Θ)	1.45	7.81	0.06	1.5	0.2	0.7
Yaw (Ψ)	0.02	0.045	0.003			
Yaw rate ($\dot{\Psi}$)				0.2	0.3	0.0
Unit	$\frac{V}{rad}$	$\frac{V}{rad \cdot s}$	$\frac{V}{rad/s}$	$\frac{V}{rad}$ $\frac{V}{rad/s}$	$\frac{V}{rad \cdot s}$ $\frac{V}{rad}$	$\frac{V}{rad/s}$ $\frac{V}{rad/s^2}$

Table 3: Control gains in free flight simulation and for experimental gimbal setup

5 CONCLUSION AND FUTURE WORK

This work discusses a control system for an insect-like tailless flapping wing drone. A simulation environment based on a quasi-steady aerodynamics model is used to compare attitude estimation techniques that employ onboard IMU readings. The attitude estimation is incorporated into a cascaded PID flight controller that stabilizes the drone attitude and velocity around hover. The outer-loop velocity controller successfully compensates for biases present in the attitude estimator. The attitude estimation and attitude control are evaluated experimentally with the drone mounted inside a single cage gimbal setup. The drone produces sufficient torque to stabilize its orientation around a desired tilt attitude. Because the gimbal setup significantly increases inertia, the controller gains and response time need to be reconfigured for free flight testing.

The relatively simple PID architecture is expected to suffice for unobstructed hovering flight when the flight system characteristics are static and accurately represented by the system model. In practice FWAVs experience conditions that may be challenging for a PID flight controller. The large range of flight modes and the uncertain, time-varying system characteristics encountered by physical FWAVs are expected to pose important additional challenges to the flight controller. Therefore, the flight simulation environment developed in this work will serve as a basis for future investigations that evaluate more advanced robust and adaptive control methods.

The controller is expected to derive advantage from additional sensing equipment to improve its state estimation capabilities. An onboard magnetometer would allow to determine the yaw angle, and camera readings would enable velocity and position control.

ACKNOWLEDGEMENTS

The Research Foundation - Flanders (FWO) is gratefully acknowledged for its support through research grant numbers 1S60222N and S001821N.

SUPPLEMENTARY MATERIAL

Video material of the experimental attitude estimation and control is available at `https://www.researchgate.net/publication/382950011_Supplementary_material_Video_footage_showing_rollpitchyaw_estimation_and_control_for_an_insect-like_flapping_wing_aerial_vehicle_inside_a_single_cage_gimbal_setup`.

REFERENCES

[1] D. D. Chin and D. Lentink. Flapping wing aerodynamics: from insects to vertebrates. *Journal of Experimental Biology*, 219:920–932, 2016.

[2] M. Karásek. Safe and collision-tolerant flying robots with flapping wings. Presented at the 13th international micro air vehicle conference and competition (IMAV2022). *url: https://www.youtube.com/watch?v=lq_2Ofh1x5M&t=13729s*, 2022.

[3] K. Ma, P. Chirarattananon, S. Fuller, and R. Wood. Controlled flight of a biologically inspired, insect-scale robot. *Science magazine*, 340, 2013.

[4] H. V. Phan and H. C. Park. Mechanisms of collision recovery in flying beetles and flapping-wing robots. *Science*, 370:1214–1219, 2020.

[5] F. Leys. *The Eurotrochilus Mechanicus: A robotic hummingbird driven by a resonant flapping mechanism*. PhD thesis, KU Leuven, Leuven, Belgium, 2017.

[6] S. Timmermans. *Flight performance of nano aerial vehicles with flapping wings: Experiment-based system*

http://www.imavs.org/

http://www.imavs.org/

modelling. PhD thesis, KU Leuven, Leuven, Belgium, 2021.

[7] H.V. Phan and H.C. Park. Insect-inspired, tailless, hover-capable flapping-wing robots: Recent progress, challenges, and future directions. *Progress in Aerospace Sciences*, 111(100573), 2019.

[8] M. Karásek and A. Preumont. Flapping flight stability in hover: A comparison of various aerodynamic models. *International Journal of Micro Air Vehicles*, 4:203–226, 2012.

[9] K. Nguyen, L. T. K. Au, H. V. Phan, and H. C. Park. Comparative dynamic flight stability of insect-inspired flapping-wing micro air vehicles in hover: Longitudinal and lateral motions. *Aerospace Science and Technology*, 119(107085), 2021.

[10] H. V. Phan, S. Aurecianus, T. Kang, and H. C. Park. Kubeetle-s: An insect-like, tailless, hover-capable robot that can fly with a low-torque control mechanism. *International Journal of Micro Air Vehicles*, 11:1–10, 2019.

[11] T. Roelandt, T. Willems, F. Naets, and D. Vandepitte. High-speed camera measurement of insect-like flapping wing deformation using a speckle pattern wing membrane. In *14th International Micro Air Vehicle Conference and Competition (IMAV2023)*, 2023.

[12] Z. Tu, F. Fei, J. Zhang, and X. Y. Deng. An at-scale tailless flapping-wing hummingbird robot. i. design, optimization, and experimental validation. *IEEE Transactions on Robotics*, 36(5):1511–1525, 2020.

[13] Q. Wang, J.F.L. Goosen, and F. van Keulen. A predictive quasi-steady model of aerodynamic loads on flapping wings. *Journal of Fluid Mechanics*, 800:688–719, 2016.

[14] T. Roelandt and D. Vandepitte. Inherently stable descending flight of a tailless flapping wing micro air vehicle by upward wing elevation. *International Journal of Micro Air Vehciles*, 2023.

[15] S. Timmermans, F. Leys, and D. Vandepitte. Model-based evaluation of control roll, pitch, yaw moments for a robotic hummingbird. *Journal of Guidance, Control and Dynamics*, 40(11):2934–2940, 2017.

[16] D. Olejnik, M. Karásek, B. Duisterhof, K. Scheper, T. van Dijk, and G.C.H.E. de Croon. A tailless flapping wing mav performing monocular visual servoing tasks. In *11th International Micro Air Vehicle Conference and Competition (IMAV2019)*, pages 60–66, 2019.

[17] S. Armanini, M. Karásek, C. de Visser, G.C.H.E. de Croon, and M. Mulder. Flight testing and preliminary analysis for global system identification of ornithopter dynamics using on-board and off-board data. *AIAA Atmospheric Flight Mechanics Conference*, 2017.

APPENDIX A: ADDITIONAL EXPERIMENTAL ATTITUDE CONTROL CASES

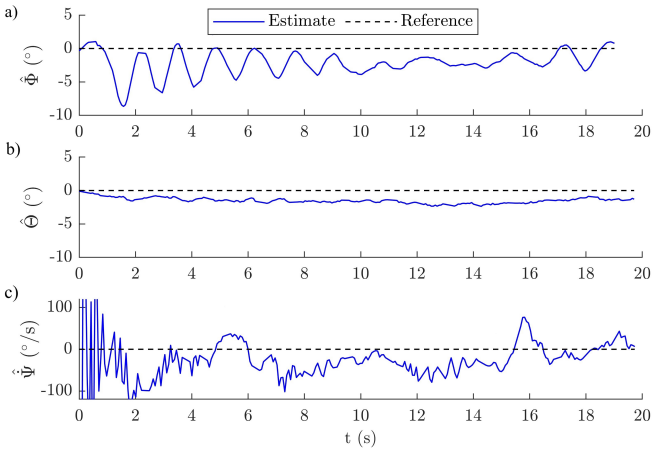

Figure 10: Attitude controller measurements for maintained hover.

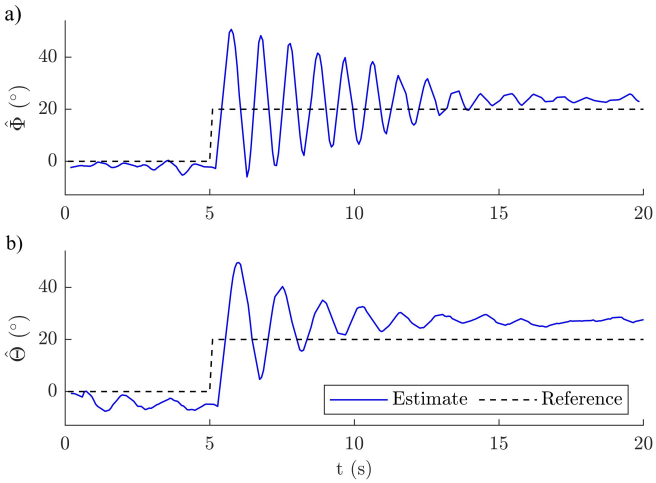

Figure 11: Attitude controller measurements for step inputs: a) roll estimation, b) pitch estimation.

http://www.imavs.org/

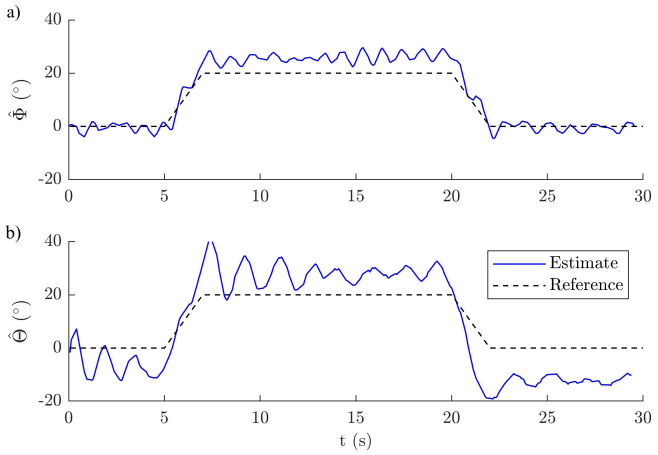

Figure 12: Attitude controller measurements for ramp inputs to 20° and back to upright: a) roll estimation, b) pitch estimation.

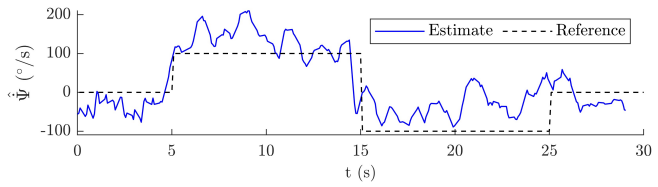

Figure 13: Attitude controller measurements for yaw rate step inputs.