machines

MDPI

Article

Observer-Based Control of Hummingbird Robot Trajectories

Yousef Farid and André Preumont *

Department of Control Engineering and System Analysis, Université Libre de Bruxelles (ULB), CP. 165-55, 50 Av. F.D. Roosevelt, B-1050 Brussels, Belgium; yousef.farid@ulb.be

* Correspondence: andre.preumont@ulb.be

Abstract

This paper presents an observer-based strategy for controlling the horizontal trajectories of a hummingbird robot from on-board inertial measurements (MEMS). The centrifugal acceleration resulting from sharp turns is responsible for the dynamic coupling between the roll axis and the pitch and yaw axes. This coupling cannot be accounted for with independent control loops for the three axes; the problem can be solved with a modified state observer (MSO) introduced on the roll axis. Numerical simulations are presented to confirm the idea. The limited additional computational burden allows for real-time implementation. The MSO allows the robot to mimic the behavior of birds that lean towards the inside when turning. Under steady-state conditions (uniform longitudinal velocity and constant yaw rate), the pitch angle is such that the longitudinal component of the gravity vector balances the longitudinal drag force and the roll angle is such that the lateral component of the gravity vector balances the centrifugal acceleration.

Keywords: hummingbird robot; trajectory generation; horizontal turn; modified state observer (MSO); model-based control; inertial measurement unit (IMU)

1. Introduction

Unmanned aerial vehicles are omnipresent in our everyday life, being involved in both peaceful as well as military applications. Most of the current drones are fixed-wing aircraft, quadcopters, or other similar devices. Flapping-wing micro-air vehicles (FWMAVs) are being developed, but they have not reached the development stage that allows them to compete with existing drones. It is widely recognized, however, that they are naturally superior to fixed-wing aircraft in landing and take-off, and they are also superior to quadcopters for their capability to mimic birds and exploit thermals for saving energy. In addition, it is well known that birds can fly with a damaged wing. The hummingbird has been given particular attention for its unique capability of stationary flight. Excellent reviews of past and on-going projects are available [1,2]. Special mention is given to the Nanohummingbird project [3] that pioneered the wing-twist modulation for the attitude control of a tailless hummingbird robot. Konkuk University successfully built a prototype weighing only 15.8 g, with a flight endurance of 9 min [4,5].

A significant number of new projects and publications have emerged more recently, e.g., refs. [6–13], addressing topics such as hovering [6,10], maneuvering [11], navigation in confined spaces [12], and attitude reconstruction [13]. However, our review of the literature revealed no studies addressing the specific subject of the present work, namely, observer-based state reconstruction using on-board IMU sensors. Many papers focused on numerical simulations where all the states are available for control [8,9], or they used an external (Vicon) video motion capture system [11] to reconstruct the robot velocity and attitude.

Check for updates

Academic Editor: Shuofei Yang

Received: 28 July 2026
Revised: 24 August 2026
Accepted: 2 September 2026
Published: 11 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

2. The Colibri Project

The general architecture of the Colibri robot has been described in earlier publications. It will not be repeated here, but a short video provides a visual summary [14]. Figure 1 shows a general view of the robot: an earlier version is presented on the left side (with the control board on top) [15–17], and a more recent prototype is shown on the right side, with the control board and the batteries located at the bottom [18]. The current weight is 22–23 g, depending on the configuration (approximately twice that of the natural bird of the same size [19]), and the hovering flight autonomy is nearly 5 min.

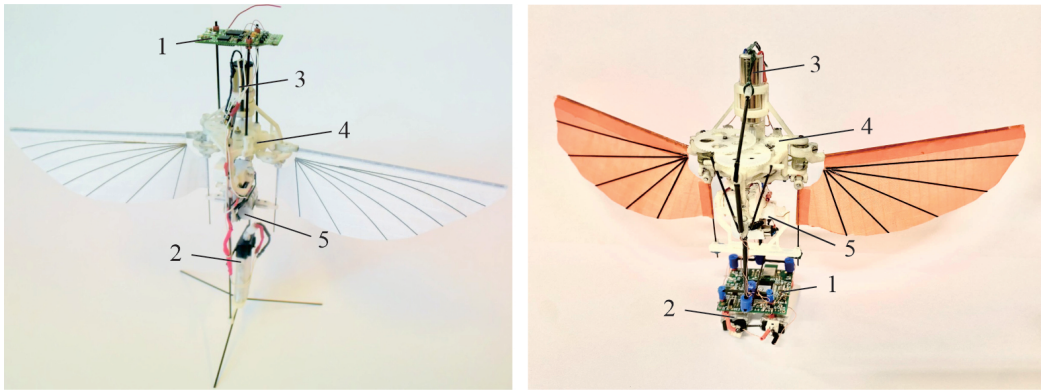


Figure 1. The robot Colibri. **Left**: early version with the control board on top [16]. **Right**: current configuration with the control board at the bottom. (1) Flight control board, (2) batteries, (3) main motor, (4) gearbox and flapping mechanism, (5) attitude control actuators.

2.1. Assumptions and Previous Results

1. The low-frequency dynamics (<10 Hz) can be modeled as those of a rigid body.
2. In hovering flight, the coupling between the longitudinal (pitch) and lateral dynamics (roll) is weak and can be neglected; therefore, they can be treated as independent. The roll, pitch and yaw axes may be assumed to be principal axes of inertia.
3. Because the flapping frequency is high (near 22 Hz), compared to the robot dynamics, the complex aerodynamic forces generated by the flapping wings may be approximated by their cycle-averaged values. The rapid changes in lift, drag and pitch moment during a flapping cycle can be treated as periodic noise. The flapping noise is mainly associated with the longitudinal dynamics; the drag noise and the pitch moment noise amplitudes are significantly larger than their average values and they have multiple harmonics. Due to the left–right symmetry, the lateral axis is not affected by significant flapping noise.
4. The longitudinal (pitch) and lateral (roll) axes are naturally unstable. When the center of drag is located above the center of mass ($z_d > 0$), both axes exhibit a pair of complex poles with positive real parts. They can be stabilized by a Proportional-Derivative (PD) compensator.
5. The vertical (yaw) axis is naturally stable. The vertical velocity is controlled by the flapping frequency (as we are mainly interested in horizontal flights). The rotation about the yaw axis was not controlled in the hovering flights conducted so far [18] and a slow rotation was observed during the flight tests [20], as a consequence of a small asymmetry between the drag forces of of the left and right wings.
6. The robot is equipped with two inertial measurement units, IMUs (only one of them is currently used). Each IMU includes three gyroscopes measuring the three rotation velocities along the body axes, and three accelerometers sensing the three components of the specific acceleration $\boldsymbol{s} = \boldsymbol{a} - \boldsymbol{g}$ (absolute acceleration minus gravitational acceleration). In hovering, the absolute acceleration vanishes (except for noise) and the

specific acceleration indicates the direction of the gravity vector in the robot's frame, $s = -g$.

7. The gyroscope output can be used directly in the PD controller of the longitudinal and lateral axes, but the attitude angles (roll and pitch) are not directly available; they must be reconstructed by independent dynamic state observers (one per axis). Dynamic state observers were successfully implemented as steady-state Kalman filters (based on assumptions regarding the system and sensor noise). This topic is discussed extensively in [18].
8. The PD controllers can indeed stabilize the pitch and roll axes but, because of the strong flapping noise, the drift of the robot cannot be avoided, making station keeping impossible. However, the full state dynamic observers allow the reconstruction of the longitudinal and lateral velocities in the robot frame, (u, v). This allows a significant improvement in station keeping via an outer proportional–integral (PI) control loop. It was implemented successfully and the performances observed were as good as earlier tests with an external tracking system (in spite of the slow uncontrolled rotation about the yaw axis) [18].
9. The full state observer (SO) worked surprisingly well during the previous hovering flight tests. Lightly damped flexible systems using full state observers are subject to a phenomenon known as "spillover" [21], in which the high frequency dynamics (flexible modes) neglected in the observer design tend to destabilize the system. This phenomenon was not observed in the hovering flight tests because of the large aerodynamic damping of the robot's flexible modes.

2.2. Objectives of the Study

The aim of this study is to extend hovering flight control to trajectory tracking in a manner similar to that of quadcopters. This requires a third control loop on the yaw axis, leading to three independent loops for roll, pitch, and yaw, which are sufficient to follow slow trajectories. However, in order to follow fast horizontal trajectories, some kind of loop interaction must be introduced to handle the centrifugal forces during quick turns, mimicking the behavior observed in birds that lean towards the inside when turning. This introduces a nonlinear coupling between the three axes and a modification of the SO of the roll axis.

The paper is organized as follows: Section 3 recalls the control model of the longitudinal (pitch) axis already presented in [18]. Section 4 discusses the control model of the lateral (roll) axis, including the centrifugal acceleration; the modified state observer (MSO) for the roll axis is the main original contribution of the paper. Section 5 describes the control model of the yaw axis. Section 6 compares trajectories obtained when including the centrifugal acceleration with those obtained when it is neglected.

The following notations are used:

(a) u, v, w are the components of the velocity of the center of mass in the robot frame.
(b) p, q, r are the components of the angular velocity in the robot frame.
(c) ϕ, θ, ψ are the roll, pitch, and yaw attitude angles (Bryan).

3. Longitudinal (Pitch) Axis Control Model

According to the assumptions introduced in the previous section, the longitudinal dynamics are governed by the Newton–Euler equations. The main damping mechanism is the flapping of the wings, which can be modelled accurately as a viscous damping. The drag force is proportional to the velocity of the centre of drag (Figure 2),

$$F_d = -K(u + qz_d) \tag{1}$$

where u is the linear velocity of the centre of mass, and q is the pitch angular velocity. The damping constant K is a linear function of the flapping frequency. Assuming that the pitch attitude angle θ is small, so that $\sin\theta \approx \theta$ and $\cos\theta \approx 1$, Newton's equation becomes

$$m\dot{u} = X_u u + X_q q + mg\theta \tag{2}$$

with $X_u = -K$, and $X_q = -Kz_d$

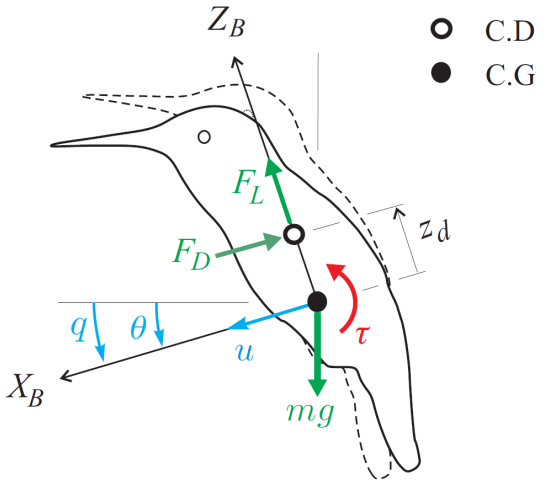


Figure 2. Longitudinal (pitch) model of the robot near-hovering. Coordinate system, force and moment diagram. F_D is the drag force, F_L is the lift force, and $q = \dot{\theta}$.

Similarly, Euler's equation governing the rotation about the centre of mass is

$$I_{yy}\dot{q} = M_u u + M_q q + \tau_a \tag{3}$$

where $M_u u + M_q q$ is the drag moment and τ_a is the control torque obtained by twisting the wings via control bars actuated by servos. The actuator dynamics can be modelled as a first-order system with time constant T,

$$T\dot{\tau}_a + \tau_a = \tau_p \tag{4}$$

where τ_p is the requested torque.

In state-space form, the cycle-averaged longitudinal dynamics become

$$\begin{bmatrix} \dot{u} \\ \dot{q} \\ \dot{\theta} \\ \dot{\tau}_a \end{bmatrix} = \underbrace{\begin{bmatrix} \hat{X}_u & \hat{X}_q & g & 0 \\ \hat{M}_u & \hat{M}_q & 0 & 1/I_{yy} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/T \end{bmatrix}}_{A_p} \begin{bmatrix} u \\ q \\ \theta \\ \tau_a \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/T \end{bmatrix}}_{B_p} \tau_p + \underbrace{\begin{bmatrix} d/m \\ t_p/I_{yy} \\ 0 \\ 0 \end{bmatrix}}_{w_p} \tag{5}$$

where $\hat{X}_u = X_u/m$ and $\hat{M}_q = M_q/I_{yy}$ are always negative and $\hat{X}_q = X_q/m$ and $\hat{M}_u = M_u/I_{yy}$ are negative if $z_d > 0$, that is, if the center of drag is above the center of mass, and positive if $z_d < 0$. The above equation includes the flapping drag noise d and the pitch flapping torque noise t_p that enter the system at the input.

3.1. Output Equation

The IMU MEMS sensor is located at a distance z_a above the center of mass ($z_a > 0$ above and $z_a < 0$ below). The output consists of three gyroscopes measuring the angular

velocity $(p, q, r)^T$ about the three body axes, and three accelerometers measuring the three components of the specific acceleration, $\mathbf{s} = (s_x, s_y, s_z)^T$.

For near-hovering horizontal flight, the quantities of interest are s_x and q. The pitch rate q is one of the state variables, while s_x can be expressed as

$$s_x = \dot{u} + \dot{q}\, z_a - g\theta \tag{6}$$

Substituting $\dot{u}$ and $\dot{q}$ from the system Equations (2) and (3), the output vector $y_p = (s_x, q)^T$ can be expressed in the classical state-space form as

$$\begin{bmatrix} s_x \\ q \end{bmatrix} = \underbrace{\begin{bmatrix} (\hat{X}_u + \hat{M}_u z_a) & (\hat{X}_q + \hat{M}_q z_a) & 0 & z_a / I_{yy} \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{C_p} \begin{bmatrix} u \\ q \\ \theta \\ \tau_a \end{bmatrix} \tag{7}$$

3.2. Stabilization, Station Keeping

Equations (5) and (7) have the classical state-space form

$$\dot{x}_p = A_p x_p + B_p \tau_p + w_p \tag{8}$$

$$y_p = C_p x_p \tag{9}$$

with the state vector $x_p = (u, q, \theta, \tau_a)^T$, τ_p is the requested control torque, and $w_p = (d/m, t_p/I_{yy}, 0, 0)^T$ is the process noise vector, representing the drag and torque disturbances produced by the flapping of the wings.

The open-loop longitudinal dynamics are unstable. For $z_d > 0$ (centre of drag above the centre of mass), the pole configuration consists of a pair of unstable oscillatory poles and two stable poles on the negative real axis. The system can be stabilized with a PD controller. In the Laplace form,

$$\tau_p = -k_\theta (1 + T_q s)\theta_e \tag{10}$$

or equivalently,

$$\tau_p = -k_\theta \theta_e - k_\theta T_q q \tag{11}$$

where θ_e is the pitch attitude error and q is the pitch angular velocity.

The pitch velocity q is directly available from the gyroscopes, whereas the pitch angle θ must be reconstructed from the sensor outputs $(s_x, q)^T$. This can be achieved in several ways (e.g., by means of a complementary filter), but a full-state dynamic observer (Luenberger observer [22]) offers several advantages, which will become apparent below.

The stabilization loop is the inner loop shown in Figure 3. Here, θ_d denotes the desired pitch angle, while $\hat{\theta}$ and $\hat{q}$ are the estimated pitch angle and pitch rate, respectively. The observer gain matrix L is selected to provide a suitable compromise between observer bandwidth and noise attenuation. It can also be obtained as the steady-state gain of a Kalman filter by assuming suitable process and measurement noise statistics [18].

The inner stabilization PD loop described above stabilizes the longitudinal dynamics. However, it is very sensitive to IMU misalignment. A small calibration error in the pitch angle produces a non-zero pitch command θ_d, which in turn generates an axial velocity u. The steady-state velocity value results from the equilibrium between the off-axis component of gravity $mg\theta$ and the aerodynamic drag force $X_u u$.

Setting $\dot{u} = \dot{q} = 0$ in (2) gives

$$u = -\frac{mg}{X_u}\theta \tag{12}$$

which indicates that, when the robot moves forward with a uniform velocity, the robot leans forward in such a way that the longitudinal component of the lift vector balances the drag force while the vertical component balances the weight of the robot.

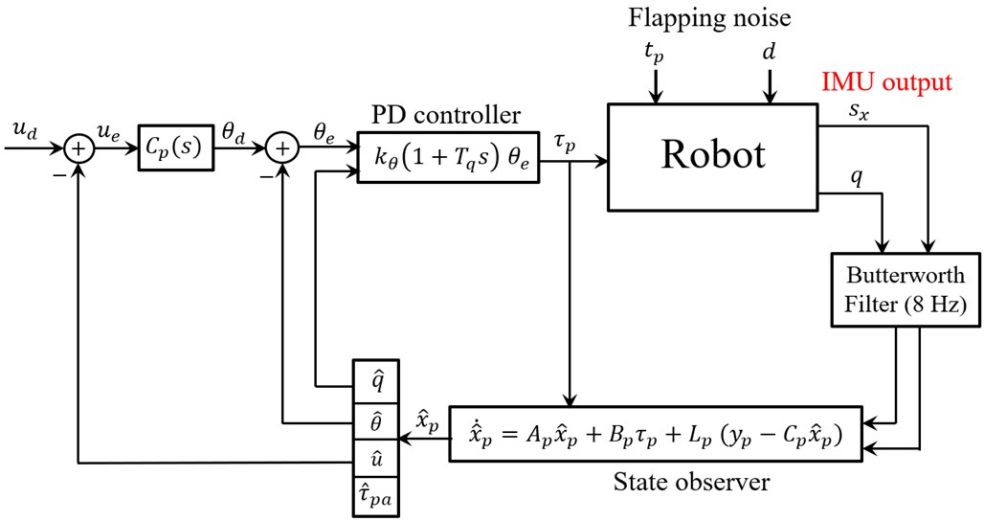


Figure 3. Control of the longitudinal axial axis. The inner loop, based on PD control of q and θ, provides attitude stabilization. The outer loop provides station keeping by regulating the longitudinal velocity.

Since the full-state dynamic observer provides an estimate of the axial velocity $\hat{u}$, an outer control loop can be added (Figure 3), where u_d is the desired longitudinal velocity ($u_d = 0$ at hovering) and $C(s)$ is an appropriate compensator. Using a PI control

$$C(s) = g_u \frac{s + z_u}{s} \tag{13}$$

has been found to be very effective in flight tests [18].

4. Lateral (Roll) Axis Control Model

4.1. State-Space Model

In hovering flight, the lateral axis can be modelled in the same way as the longitudinal axis, leading to a set of three independent models (the yaw axis will be considered in the next section). However, for horizontal flight ($w \approx 0$) dominated by the longitudinal direction ($u \gg v$) with sharp turns about the yaw axis ($r \gg p, q$), the centrifugal acceleration ur may become a significant contributor to Newton's equation of motion. Being the product of the longitudinal velocity u and the yaw velocity r, the term ur introduces a coupling between the three axes. It is the price to pay for mimicking birds that lean into turns.

With the notation of Figure 4, the velocity of the center of drag is $v - pz_d$. With the foregoing assumptions, the y-component of Newton's equation of motion in the robot frame becomes

$$m(\dot{v} + ur) = X_v v + X_p p - mg\phi \tag{14}$$

with $X_v = -K$ and $X_p = Kz_d$. The Euler equation is similarly written as

$$I_{xx}\dot{p} = M_v v + M_p p + \tau_r \tag{15}$$

where $M_v = Kz_d$, $M_p = -K_p - Kz_d^2$, and τ_r is the roll control torque.

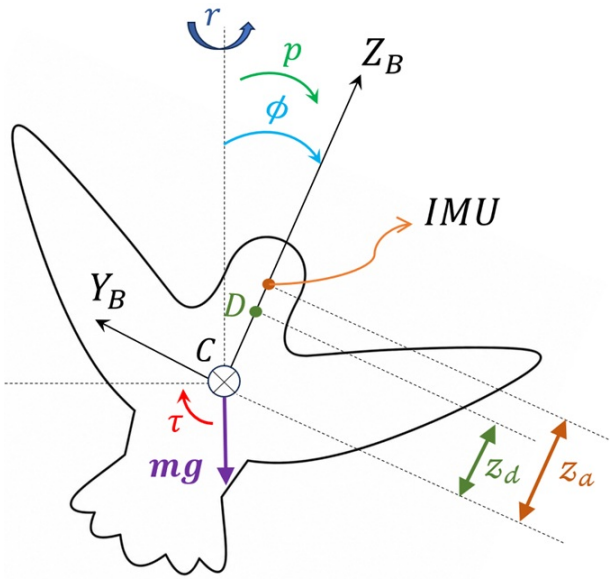


Figure 4. Lateral (roll) model of the robot. The robot is seen from the back; the roll angle ϕ indicates the rotation about the body-fixed x_B axis, and $p = \dot{\phi}$.

After moving the centrifugal force to the right-hand side of (14), the lateral dynamics can be written in state-space form as

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{\phi} \\ \dot{\tau}_{ar} \end{bmatrix} = \underbrace{\begin{bmatrix} \hat{X}_v & \hat{X}_p & -g & 0 \\ \hat{M}_v & \hat{M}_p & 0 & 1/I_{xx} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/T \end{bmatrix}}_{A_r} \begin{bmatrix} v \\ p \\ \phi \\ \tau_{ar} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/T \end{bmatrix}}_{B_r} \tau_r + \underbrace{\begin{bmatrix} -ur \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\delta} \tag{16}$$

where the centrifugal acceleration is treated as an external input. The normalized coefficients

$$\hat{X}_v = \frac{X_v}{m}, \qquad \hat{X}_p = \frac{X_p}{m}, \qquad \hat{M}_v = \frac{M_v}{I_{xx}}, \qquad \hat{M}_p = \frac{M_p}{I_{xx}}$$

have the same meaning as those introduced in Equation (5).

To illustrate the meaning of the additional centrifugal acceleration in Equation (14), consider a circular trajectory of radius R along which the robot rotates about the vertical axis with angular velocity r. The longitudinal velocity is

$$u = Rr \tag{17}$$

According to Equation (14), if there is no sideslip ($\dot{v} = v = 0$) and no roll rate ($p = 0$), the steady-state roll angle is

$$\phi = -\frac{ur}{g} = -\frac{Rr^2}{g} \tag{18}$$

Therefore, the robot leans into the turn so that the lateral component of gravity balances the centrifugal acceleration.

4.2. Output Equation

The output of the accelerometer (lateral acceleration) is

$$s_y = \dot{v} - \dot{p}z_a + g\phi \tag{19}$$

Upon substituting $\dot{v}$ and $\dot{p}$ from the state-space model (16), the accelerometer output becomes

$$s_y = (\hat{X}_v - z_a\hat{M}_v)v + (\hat{X}_p - z_a\hat{M}_p)p - \frac{z_a}{I_{xx}}\tau_{ar} - ur \tag{20}$$

This equation cannot be written in the usual output equation form because of the unknown centrifugal acceleration term ur. However, using the estimates $\hat{u}$ and $\hat{r}$ available from the pitch and yaw models, the accelerometer output can be modified as

$$s_y^* \triangleq s_y + \hat{u}\hat{r} \tag{21}$$

The new output vector, $y_r^* = \left(s_y^*,\ p\right)^T$, takes the classical form

$$y_r^* = \begin{bmatrix} s_y^* \\ p \end{bmatrix} = \underbrace{\begin{bmatrix} \hat{X}_v - z_a\hat{M}_v & \hat{X}_p - z_a\hat{M}_p & 0 & -z_a/I_{xx} \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{C_r} \begin{bmatrix} v \\ p \\ \phi \\ \tau_{ar} \end{bmatrix} + n_y^* \tag{22}$$

where $n_y^* = \begin{bmatrix} \hat{u}\hat{r} - ur & 0 \end{bmatrix}^T$. The estimation error of the centrifugal acceleration $\hat{u}\hat{r} - ur$ can be regarded as noise.

4.3. Observer Equation

Equations (16) and (22) can be written in the standard state-space form as

$$\dot{x}_r = A_r x_r + B_r \tau_r + \delta \tag{23}$$

$$y_r = C_r x_r + n_y \tag{24}$$

where $x_r = (v, p, \phi, \tau_{ar})^T$, $\delta = -ur\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$ represents the centrifugal acceleration acting as an external disturbance, and $n_y = -ur\begin{bmatrix} 1 & 0 \end{bmatrix}^T$.

Using an MSO, the state estimate can be expressed as follows:

$$\dot{\hat{x}}_r = A_r\hat{x}_r + B_r\tau_r + \hat{\delta} + L_r(y_r^* - C_r\hat{x}_r) \tag{25}$$

where $\hat{\delta}$ is an estimate of the disturbance vector δ, obtained from the estimated longitudinal velocity and yaw rate provided by the pitch observer, together with the output of the Butterworth filter (BF) in the yaw control system.

The state estimation error $e_r = x_r - \hat{x}_r$ satisfies

$$\dot{e}_r = (A_r - L_rC_r)e_r + (\delta - \hat{\delta}) - L_r n_y^* \tag{26}$$

If $A_r - L_rC_r$ is Hurwitz (i.e., all its eigenvalues have negative real parts), the observer error is exponentially stable in the absence of disturbance-estimation and measurement errors. More generally, if $\delta - \hat{\delta}$ and n_y^* are bounded, the estimation error remains bounded. Asymptotic convergence, $e_r \to 0$, is achieved when $\delta - \hat{\delta} \to 0$ and $n_y^* \to 0$.

The block diagram of the roll axis control system is shown in Figure 5. The PD controller used for the stabilization loop and the compensator that converts the desired lateral velocity into a desired roll angle are essentially identical to those employed for the pitch axis.

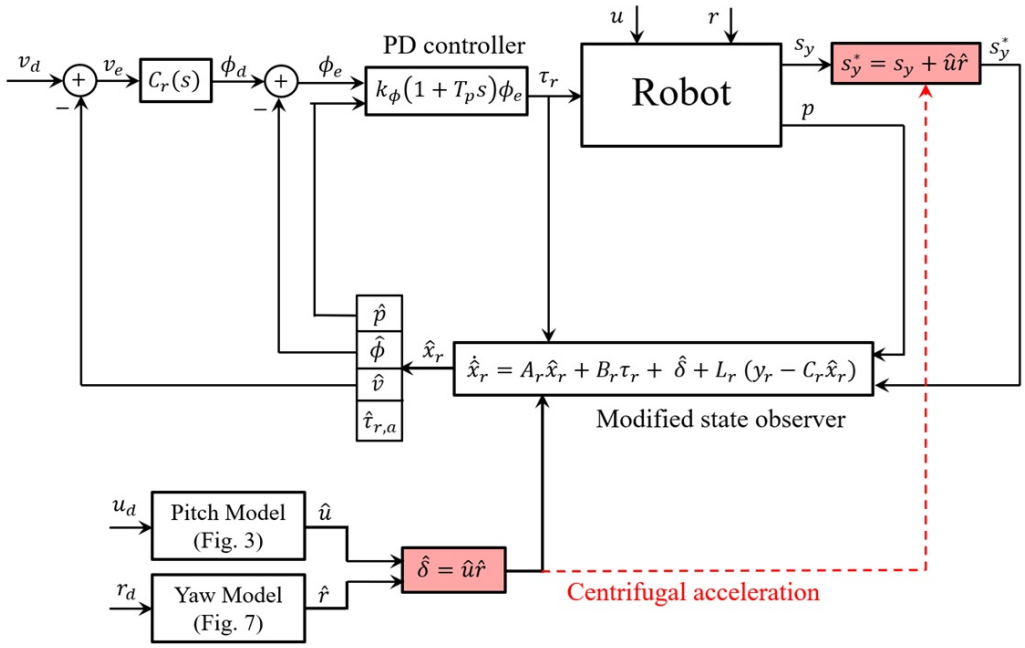


Figure 5. Control of the roll axis including the centrifugal accelerations. u_d, v_d, and r_d are desired values of the longitudinal, lateral, and yaw (spin) velocities.

5. Yaw Axis Control Model

With the notations of Figure 6, the yaw dynamics read

$$I_{zz}\dot{r} = -K_r r + \tau \tag{27}$$

where $r = \dot{\psi}$ is the yaw velocity (spin), I_{zz} is the moment of inertia about the Z_B axis, and K_r is the rotational damping coefficient about the yaw axis. τ is the control torque obtained by varying the drag force of the left and right wings (K_r is estimated from a pendulum experiment).

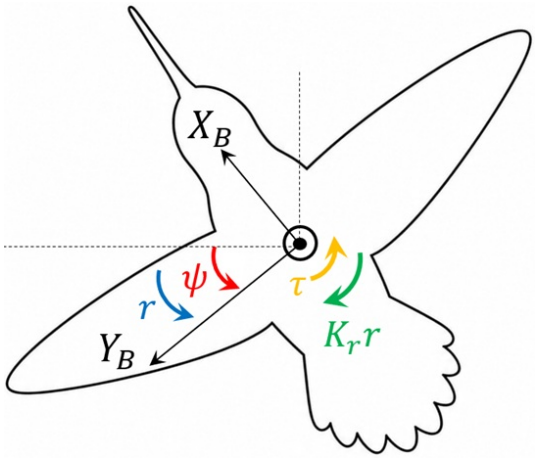


Figure 6. Coordinates of yaw dynamics. Robot seen from above.

As for the pitch and roll control torque, the wing twist is obtained by a servo actuator modeled by a first-order system with a time constant T (Equation (4)). In state-space form, the yaw dynamics become

$$\begin{bmatrix} \dot{r} \\ \dot{\tau}_a \end{bmatrix} = \begin{bmatrix} -\hat{K}_r & 1/I_{zz} \\ 0 & -1/T \end{bmatrix} \begin{bmatrix} r \\ \tau_a \end{bmatrix} + \begin{bmatrix} 0 \\ 1/T \end{bmatrix} \tau_y \tag{28}$$

where $\hat{K}_r = K_r / I_{zz}$, and τ_y is the requested torque. The yaw velocity r is directly available from the IMU; therefore, it does not require a SO. We introduce a second-order low-pass BF to attenuate the noise from the gyroscope.

The open-loop transfer function between the requested torque τ and the yaw velocity is

$$\frac{r(s)}{\tau(s)} = \frac{1}{I_{zz}(1+Ts)(s+\hat{K}_r)} \tag{29}$$

Using a PI controller,

$$C_y(s) = g_y\left(\frac{s+z_y}{s}\right) \tag{30}$$

The control block diagram is shown in Figure 7.

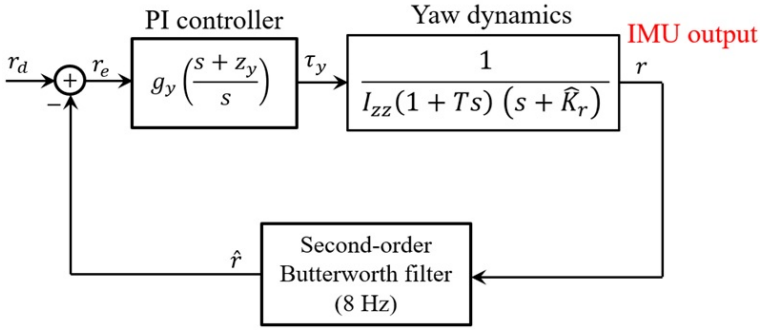


Figure 7. Control flowchart of yaw dynamics using PI control and a low pass Butterworth filter.

6. Numerical Simulation

The following simulation studies evaluate the tracking performance of the proposed controller for two horizontal reference trajectories: (i) a circular trajectory and (ii) a Gerono lemniscate trajectory. In both cases, the desired motion is specified in the robot frame by the velocity vector (u_d, v_d) and the desired yaw rate $r_d = \dot{\psi}_d$. In the MATLAB/Simulink simulations, the sampling time was set to $dt = 0.002$ s, and the fourth-order Runge–Kutta method was employed to solve the closed-loop system.

For the flapping disturbances, the measured drag force and pitch torque signals are downsampled and transformed into the frequency domain using the FFT. Frequency components above 50 Hz are removed, and the filtered time-domain disturbances are reconstructed using the inverse FFT and periodically repeated as simulation inputs.

6.1. Circular Trajectory

To evaluate the proposed control architecture, consider the open-loop generation of a circular trajectory starting from rest with a radius of $R = 1$ m, travelled in $t_{\text{trav}} = 5$ s. The corresponding input data for the three axes of the robot frame are $u_d = 2\pi R/t_{trav} = 1.256$ m/s, $v_d = 0$ and $r_d = 2\pi/t_{trav} = 1.256$ rad/s. During the trajectory, the robot is subject to a centrifugal acceleration $u_d r_d = 1.58$ m/s^2.

With reference to Figure 5, the robot model for the roll axis is Equation (16) including the centrifugal acceleration. Figure 8 gives the state responses of the pitch system. Figure 8a,b show the longitudinal velocity and the pitch attitude angle in the absence of flapping noise. Figure 8c,d show the effect of the flapping noise (see Appendix B).

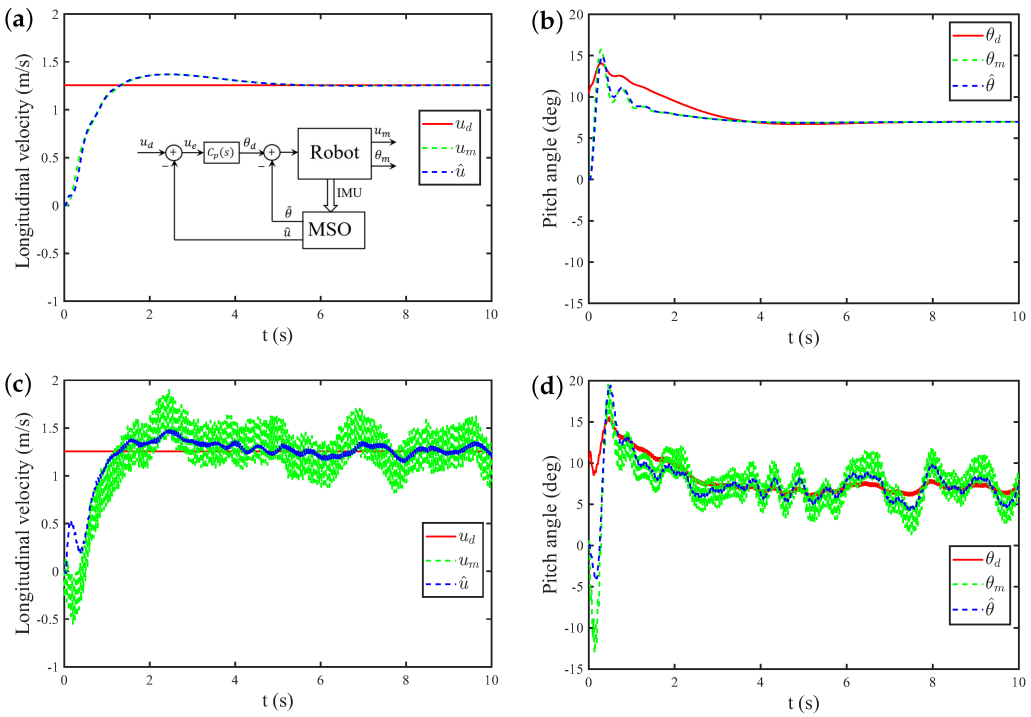


Figure 8. State responses of the pitch system while tracking a circular trajectory. (**a**,**b**) show the responses by the state observer (SO) and without flapping noise, whereas (**c**,**d**) include the effects of torque noise t_p and drag noise d in the longitudinal (pitch) dynamics. The meaning of the various subscripts is explained in the block diagram on the top left figure. The subscripts (d) and (m) denote the desired command and the measured (actual) system state, respectively, while the superscript ($\hat{}$) indicates the state estimated by the state observer (SO).

Figure 9 compares the results obtained with MSO with those obtained with SO assuming independent treatment of the three axes (when the centrifugal acceleration is neglected). We observe that, with the MSO, the estimated values $\hat{v}$ and $\hat{\phi}$ follow closely the output of the robot model, v_m and ϕ_m, and converge toward the desired value. On the other hand, with the classical SO (independent observers), a significant sideslip velocity occurs.

Figure 10 shows the trajectory obtained in absolute frame when generating u_d, v_d and r_d in robot frame. From left to right, (a) the roll dynamics observer ignores the centrifugal acceleration (SO), resulting in a side slip velocity responsible for an increase of the circle diameter; (b) the observer of the roll dynamics includes the centrifugal acceleration (MSO); and (c) the flapping noise is included in the analysis with MSO.

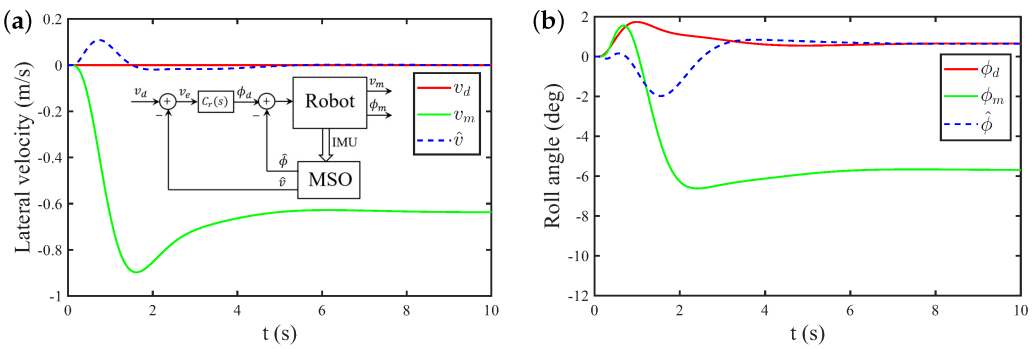


Figure 9. *Cont.*

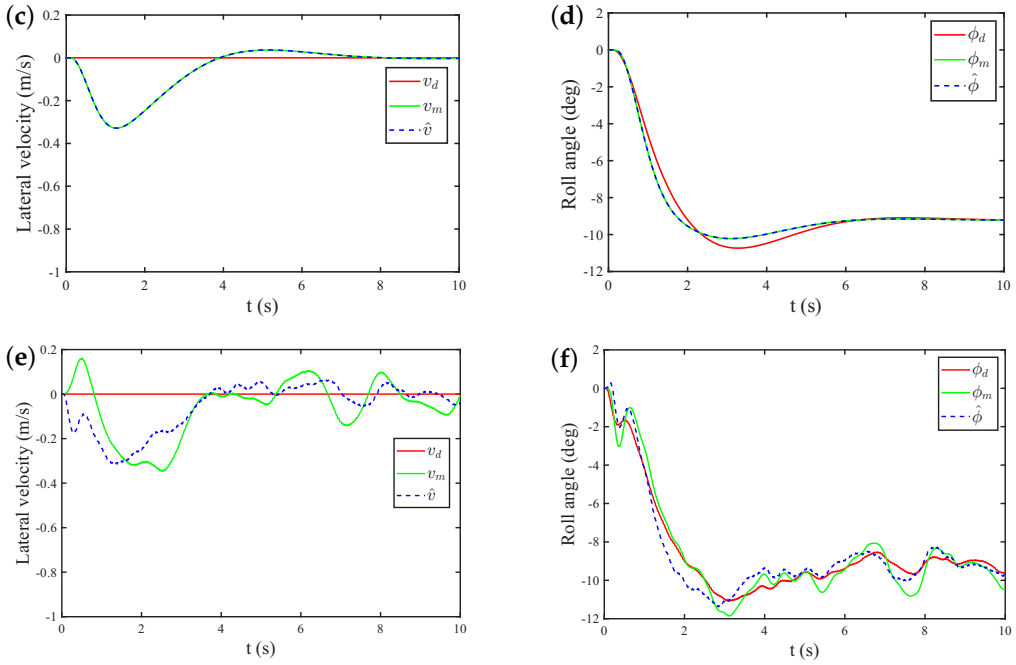


Figure 9. State responses of the roll system while tracking a circular trajectory. (**a**,**b**) show the results when the centrifugal acceleration is not included in the observer estimation (SO). (**c**,**d**) show the results with the modified state observer (MSO) including the centrifugal acceleration and (**e**,**f**) show the same results when the flapping noise (t_p and d) is applied to the longitudinal (pitch) dynamics.

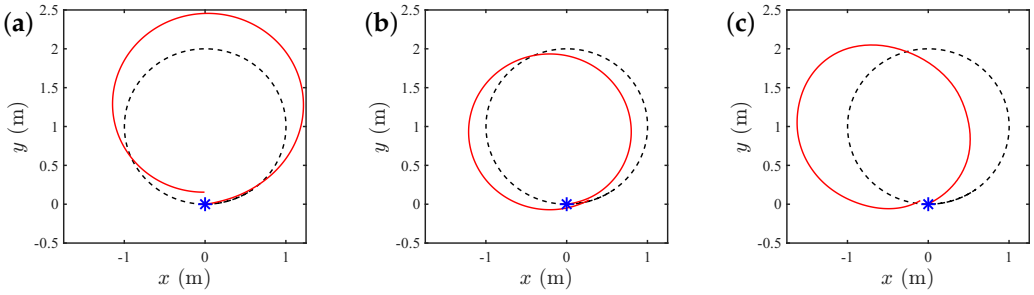


Figure 10. Circular trajectory generated in robot frame and seen in inertial frame; the target is indicated in black dashed line. (**a**) presents the result obtained assuming no interaction between axes (SO); (**b**) shows the result including the interaction due to centrifugal acceleration (MSO); and (**c**) illustrates the MSO result when the flapping noise (t_p and d) is applied to the longitudinal (pitch) dynamics.

6.2. Lemniscate Trajectory

The reference trajectory is generated using a Gerono lemniscate, which provides smooth and continuous changes in curvature with alternating turning directions. The trajectory is defined as

$$\begin{aligned} x_d(t) &= a\cos\alpha(t) \\ y_d(t) &= a\sin\alpha(t)\cos\alpha(t) \end{aligned} \tag{31}$$

where a is the scaling factor (or amplitude) that determines the physical size of the curve, and $\alpha(t)$ is a real number that typically ranges from 0 to 2π radians (Figure 11).

The corresponding reference inputs in the robot body-fixed frame are a constant surge velocity of $u_d = 1.0$ m/s, zero sway velocity ($v_d = 0$), and a time-varying yaw rate $r_d = \dot{\psi}_d$ is obtained by differentiating the desired heading angle, ψ_d, which is computed from the

tangent direction of the reference trajectory. The time parameterization of the Gerono lemniscate is given by

$$\alpha(t) = \frac{2\pi}{T_{\text{loop}}} t + \frac{\pi}{2} \tag{32}$$

where $T_{\text{loop}} = 10$ s. The resulting trajectory is subsequently re-parameterized by arc length and traversed at a constant forward speed of $u_d = 1$ m/s.

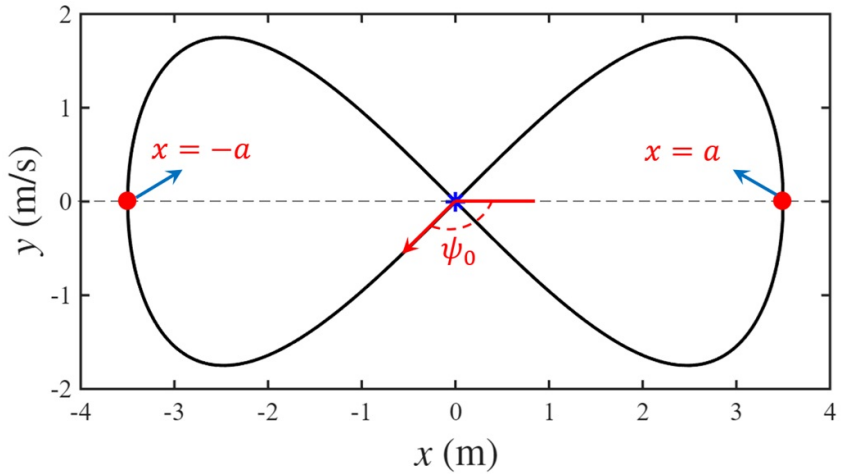


Figure 11. Lemniscate trajectory. The point $(0, 0)$ denotes the initial position, $\psi_0 = -135°$ represents the initial heading angle, and $a = 3.5$ is the scaling factor.

Figure 12 illustrates the response of yaw system during lemniscate following. The figure shows that the measured yaw rate accurately follows its reference with a high accuracy steady-state tracking performance, resulting in a smooth heading evolution that generates the required lemniscate motion.

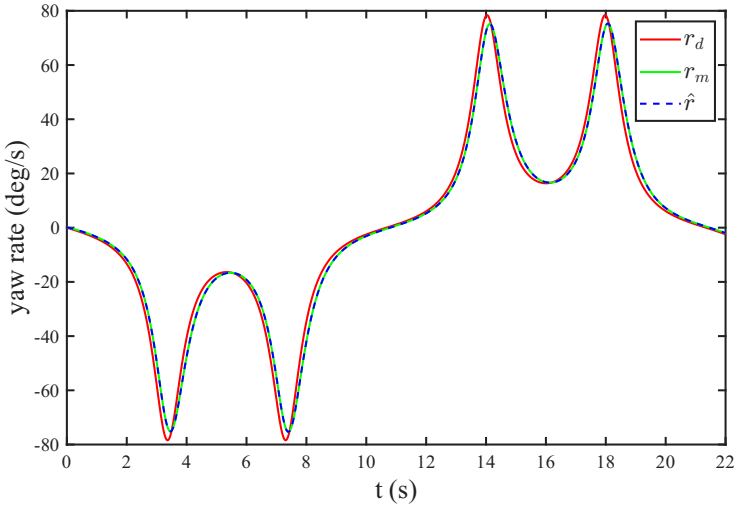


Figure 12. State responses of the yaw system for following a lemniscate trajectory.

The state responses of the roll system are shown in Figure 13. Similar to the circular trajectory case, incorporating the estimation of the centrifugal acceleration compensation significantly improves the estimation of the lateral velocity and roll angle. The estimated states closely follow the model responses, whereas neglecting the coupling introduces noticeable tracking errors.

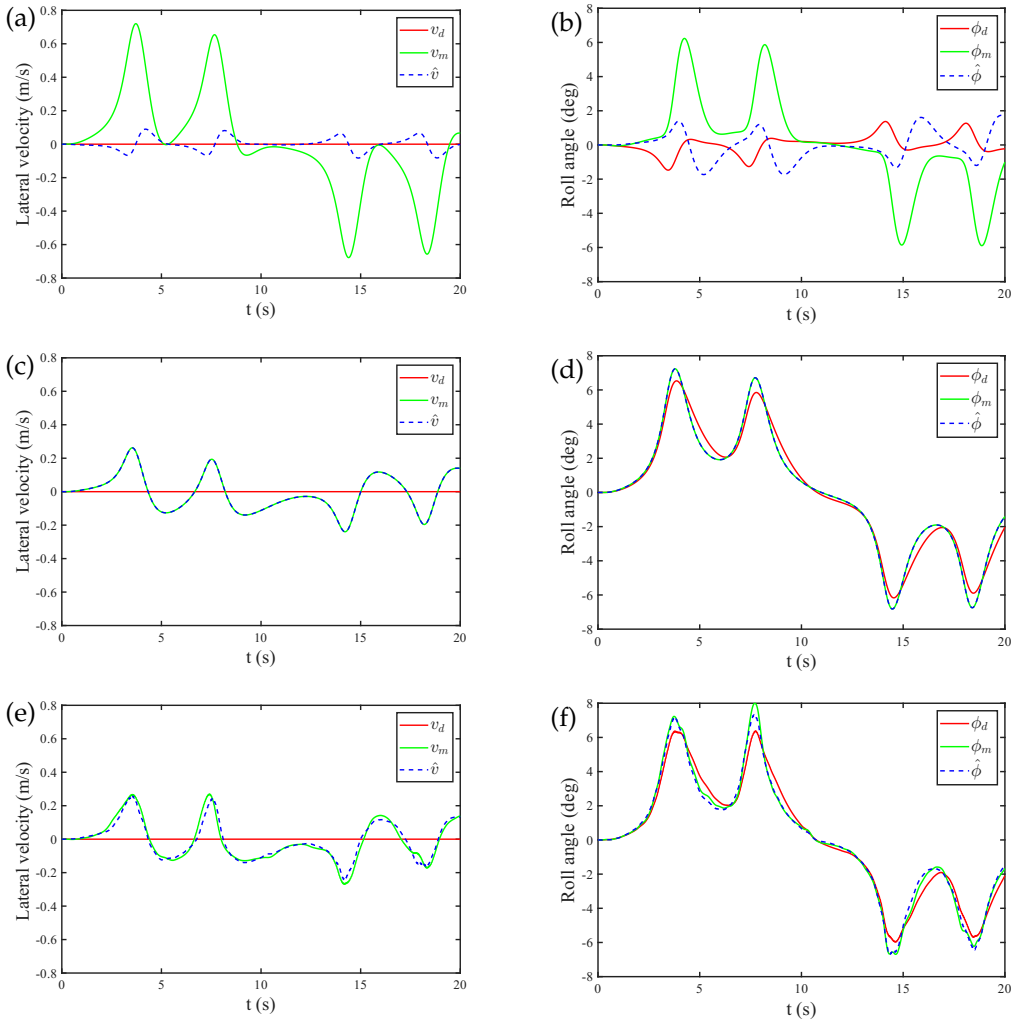


Figure 13. State responses of the roll system while tracking a lemniscate trajectory. (**a**,**b**) show the results when the estimation of centrifugal acceleration is not included in SO. (**c**,**d**) show the results with MSO. And (**e**,**f**) show the same results when the flapping noise (t_p and d) is applied to the longitudinal (pitch) dynamics.

Finally, Figure 14 presents the resulting trajectory in the inertial frame. The proposed MSO achieves accurate tracking of the lemniscate motion under nominal conditions, as depicted in Figure 14a. When flapping-related perturbations are incorporated into the pitch dynamics, the tracking performance is slightly affected; however, the hummingbird robot continues to follow the reference trajectory with satisfactory accuracy, as shown in Figure 14b.

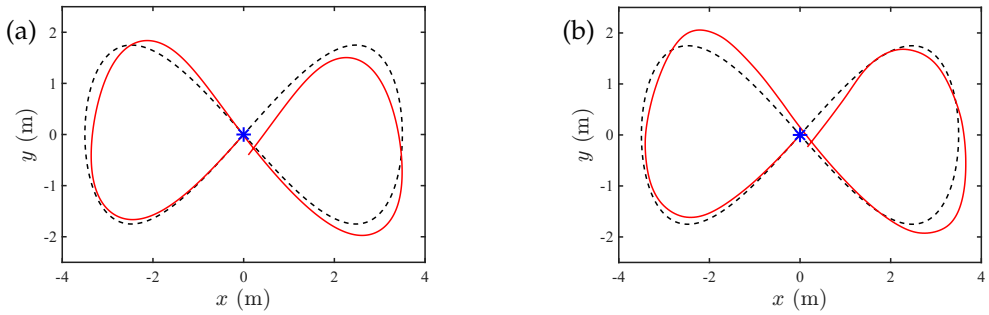


Figure 14. Lemniscate trajectory following in the absolute frame. (**a**) shows the results with MSO and (**b**) shows the same result when the flapping noise (t_p and d) is applied to the longitudinal (pitch) dynamics.

7. Conclusions

The paper presents an extension of the control strategy for hovering flight to horizontal trajectories by specifying the velocity vector (u_d, v_d, r_d) in the robot frame. The state of the system is reconstructed from on-board inertial measurements. The centrifugal acceleration encountered in sharp turns is accounted for by introducing a coupling between the pitch–roll–yaw axes in a modified state observer for the roll axis. The control strategy is assessed numerically. In steady-state conditions (constant forward velocity u_d, $v_d = 0$, and constant yaw rate r_d), the forward component of the lift vector is such that it balances the drag force, while its lateral component balances the centrifugal acceleration. Flight experiments are in preparation to validate the idea. In the absence of centrifugal acceleration, the MSO is identical to the classical observer SO, which has been tested successfully in [18]; the additional computational burden of the MSO is minimal and should not pose implementation problems.

Author Contributions: Conceptualization, A.P.; software, Y.F.; validation, Y.F. and A.P.; original draft preparation, A.P.; writing, review and editing A.P. and Y.F.; supervision, A.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FWMAV	Flapping-Wing Micro-Air Vehicle
IMU	Inertial Measurement Unit
MEMS	Micro-Electro-Mechanical System
SO	State Observer
MSO	Modified State Observer
RMS	Root Mean Square
FFT	Fast Fourier Transform
PD	Proportional Plus Derivative
PI	Proportional Plus Integral
BF	Butterworth Filter

Appendix A. COLIBRI Parameters Used in the Dynamics Model

The simulation results in this paper are based on the data provided in [15] and summarized in Table A1. It is assumed that roll and pitch systems have the same damping coefficients K and K_q. The controller and observer gains for the roll, pitch, and yaw systems are provided in Table A2.

Table A1. *COLIBRI* parameters for the body dynamics model.

Parameter	Value	Unit
m	23.5	g
K	22.3	mN.s/m
K_q	0.008	mN.s.m
K_r	0.065	mN.s.m

Table A1. *Cont.*

Parameter	Value	Unit
I_{xx}	56,000	g.mm^2
I_{yy}	56,000	g.mm^2
I_{zz}	3800	g.mm^2
I_{yz}	0	g.mm^2
I_{yx}	0	g.mm^2
I_{zx}	0	g.mm^2
T	0.038	s

Table A2. Control/observer parameters of the roll, pitch, and yaw systems.

Parameter	Value
k_θ	0.00192
T_q	0.25
k_ϕ	0.00192
T_p	0.25
g_u	0.1451
z_u	2.0
g_y	0.00008
z_y	6.6666
L	$\begin{bmatrix} -0.8573 & 0.2841 & -0.1632 & 0.0000 \\ -4.1099 & 15.5265 & 0.1436 & 0.0000 \end{bmatrix}^T$
L_r	$\begin{bmatrix} -1.1027 & -0.3256 & 0.2250 & 0.0000 \\ 3.0400 & 15.1074 & 0.2973 & 0.0000 \end{bmatrix}^T$

Appendix B. Flapping Noise (Drag and Pitch Torque)

The high-frequency components of the aerodynamic forces generated during the flapping of the wings are neglected in a cycle-averaged model. In this study, they are treated as periodic noise. Because of the left–right symmetry of the robot system, only the axial (pitch) dynamics are significantly affected by the flapping noise. The harmonic components were estimated in an earlier study by attaching the robot to a force sensor to measure the axial noise d and to a torque sensor to estimate the pitch torque t_p. Although rigidly holding the robot against the sensor is not fully representative of the aerodynamic forces in free flight, we have assumed that the first two components are the same (flapping frequency and first harmonic) and the higher harmonics are neglected. With this assumption, the RMS value of the drag force is reduced from 0.985 N to 0.5 N and that of the pitch torque from 0.0273 Nm to 0.0210 Nm. These values are still quite high (the average lift is 0.2 N and the maximum control moment is 0.002 Nm). The FFT components are shown in Figure A1; the first peak corresponds to the flapping frequency, the second harmonic is due to the complex aerodynamics during the upstroke and downstroke motions.

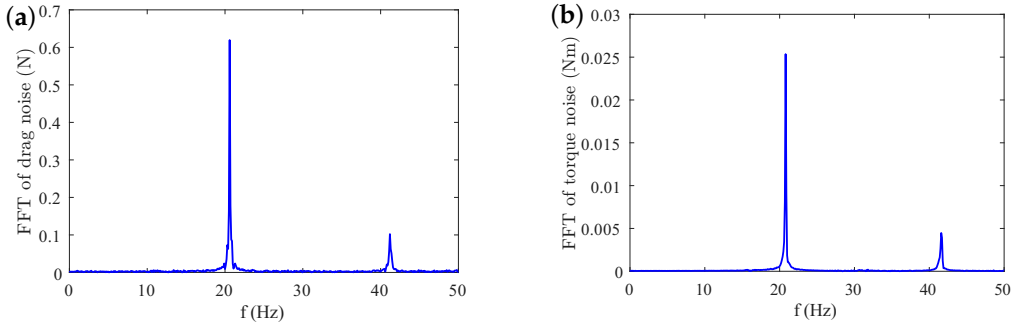


Figure A1. Fast Fourier Transform (FFT) spectra of the drag noise d (**a**) and pitch torque noise t_p (**b**), with peaks at the flapping frequency and its second harmonic due to the upstroke and downstroke.

References

1. Xiao, S.; Hu, K.; Huang, B.; Deng, H.; Ding, X. A Review of Research on the Mechanical Design of Hoverable Flapping Wing Micro-Air Vehicles. *J. Bionic Eng.* **2021**, *18*, 1235–1254. [CrossRef]
2. Ward, T.A.; Rezadad, M.; Fearday, C.J.; Viyapuri, R. A Review of Biomimetic Air Vehicle Research: 1984–2014. *Int. J. Micro Air Veh.* **2015**, *7*, 375–394. [CrossRef]
3. Keennon, M.T.; Klingebiel, K.R.; Won, H.; Andriukov, A. Development of the Nano Hummingbird: A Tailless Flapping Wing Micro Air Vehicle. In Proceedings of the 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Nashville, TN, USA, 9–12 January 2012; pp. 1–24.
4. Phan, H.V.; Park, H.C. Insect-Inspired, Tailless, Hover-Capable Flapping-Wing Robots: Recent Progress, Challenges, and Future Directions. *Prog. Aerosp. Sci.* **2019**, *111*, 100573. [CrossRef]
5. Phan, H.V.; Aurecianus, S.; Au, T.K.L.; Kang, T.; Park, H.C. Towards Long-Endurance Flight of an Insect-Inspired, Tailless, Two-Winged, Flapping-Wing Flying Robot. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5059–5066. [CrossRef]
6. Coleman, D.; Benedict, M. Methods for Stabilizing the Longitudinal Dynamics of a Biomimetic Robotic Hummingbird in Hovering Flight. *Int. J. Micro Air Veh.* **2025**, *17*, 1–17. [CrossRef]
7. Tu, Z.; Fei, F.; Zhang, J.; Deng, X. An At-Scale Tailless Flapping-Wing Hummingbird Robot. I. Design, Optimization, and Experimental Validation. *IEEE Trans. Robot.* **2020**, *36*, 1511–1525. [CrossRef]
8. Hyun, N.-S.P.; McGill, R.; Wood, R.J.; Kuindersma, S. A New Control Framework for Flapping-Wing Vehicles Based on 3D Pendulum Dynamics. *Automatica* **2021**, *123*, 109293. [CrossRef]
9. Zheng, H.; Chen, W.; Xie, F. Control Simulation of Flapping-Wing Micro Aerial Vehicle Based on Multi-Level Optimization Model Predictive Control. *IEEE Access* **2024**, *12*, 40700–40709. [CrossRef]
10. Nguyen, K.; Au, L.T.K.; Phan, H.-V.; Park, H.C. Comparative Dynamic Flight Stability of Insect-Inspired Flapping-Wing Micro Air Vehicles in Hover: Longitudinal and Lateral Motions. *Aerosp. Sci. Technol.* **2021**, *119*, 107085. [CrossRef]
11. Fei, F.; Tu, Z.; Zhang, J.; Deng, X. Learning Extreme Hummingbird Maneuvers on Flapping Wing Robots. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 109–115. [CrossRef]
12. Tu, Z.; Fei, F.; Zhang, J.; Deng, X. Acting Is Seeing: Navigating Tight Space Using Flapping Wings. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 95–101. [CrossRef]
13. Sarkar, D.; Hyun, N.-S.P. Leveraging Flapping Kinematics for Attitude Estimation in SO(3). In Proceedings of the 2025 IEEE 64th Conference on Decision and Control (CDC), Rio de Janeiro, Brazil, 9–12 December 2025; pp. 7449–7456. [CrossRef]
14. Colibri Project. Available online: https://www.youtube.com/watch?v=I2ih-Q1IKDc (accessed on 20 August 2026).
15. Roshanbin, A. Design and Development of a Tailless Robotic Hummingbird. Ph.D. Thesis, Université Libre de Bruxelles, Dept Control Engineering and System Analysis, Bruxelles, Belgium, September 2019.
16. Roshanbin, A.; Altartouri, H.; Karasek, M.; Preumont, A. COLIBRI: A Hovering Flapping Twin-Wing Robot. *Int. J. Micro Air Veh.* **2017**, *9*, 270–282. [CrossRef]
17. Roshanbin, A.; Abad, F.; Preumont, A. Kinematic and Aerodynamic Enhancement of a Robotic Hummingbird. *AIAA J.* **2019**, *57*, 4599–4607. [CrossRef]
18. Wang, H.; Farid, Y.; Wang, L.; Garone, E.; Preumont, A. Hovering Flight of a Robotic Hummingbird: Dynamic Observer and Flight Tests. *Actuators* **2024**, *13*, 91. [CrossRef]
19. Greenwalt, C.H. *Hummingbirds*; Dover: New York, NY, USA, 1990.
20. Station Keeping. Available online: https://youtu.be/53DJ0rtebKk (accessed on 20 August 2026).
21. Balas, M.J. Active control of flexible systems. *J. Optim. Theory Appl.* **1978**, *25*, 415–436. [CrossRef]
22. Luenberger, D. An Introduction to Observers. *IEEE Trans. Autom. Control* **1971**, *16*, 596–602. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.