Acceleration Based Iterative Learning Control for Pugachev’s Cobra Maneuver with Quadrotor Tail-sitter VTOL UAVs

Wei Xu¹, Haowei Gu², Fu Zhang¹

Abstract—This paper considers the well-known control problem of Pugachev’s Cobra maneuver, a dramatic and demanding maneuver requiring the aircraft to fly at extremely high Angle of Attacks (AOA) where stalling occurs. We present a simple yet very effective feedback-iterative learning control structure to regulate the altitude error during the maneuver. Both the feedback controller and the iterative learning feed-forward controller are based on the aircraft acceleration model, which is directly measurable by the onboard accelerometer. Moreover, the acceleration model leads to an extremely simple dynamic model that does not require any model identification, greatly simplifying the implementation of the iterative learning control. Real world outdoor flight experiments on the “Hong Hu” UAV, an aerobatic yet efficient quadrotor tail-sitter UAV of small-size, are provided to show the effectiveness of the proposed controller.

I. INTRODUCTION

The stability and maneuverability of traditional fixed-wing aircraft will be reduced dramatically due to the invalidation of the aerodynamic control surface when the angle of attack (AOA) increases over the stall angle. The Pugachev’s Cobra is one of the most challenging post-stall maneuvers that only a few aircraft like SU-27 can achieve [1]. During a Pugachev’s Cobra, an airplane flying at high speed suddenly raises the nose momentarily to the vertical position and then drops it back to the normal angle, as shown in Fig. 1. Even for the SU-27 with thrust vector, the maneuverability will lose when the AOA is larger than around 50°. Achieving such aerobatic maneuvers requires not only sophisticated piloting skills or flight controllers but also a proper design of the aircraft aerodynamics and propulsion systems.

Unlike conventional fixed-wing airplanes, quadrotor tail-sitter UAVs exhibit great maneuverability, being capable of both vertical flight and high-speed level flight [2]–[5]. Moreover, with the optimal design of its aerodynamics and propulsion systems [6], a quadrotor tail-sitter UAV achieves superior flight efficiency. These features make a tail-sitter UAV ideally suitable for almost all field applications where maneuverability and flight range are two indispensable elements.

Achieving the Pugachev’s Cobra maneuvers for small-size tail-sitter UAVs has many practical benefits. For example, in a surveying task, it is usually preferable if the UAV could slow down at the particular location of interests, capturing more data (e.g., images), and then accelerate to the next one. Unlike the Pugachev’s Cobra maneuvers for fighter jets, where slight altitude increment is not of concern, or even preferable (Fig. 1), it is usually required for a tail-sitter VTOL UAV for civilian-use to constantly maintain its altitude, which avoids any collision with the environments.

The Pugachev’s Cobra maneuver for tail-sitter vehicles could be imagined as a backward transition (from level flight to vertical flight) immediately followed by a forward transition (from vertical flight to level flight). A large number of researches could be found on the tail-sitter vehicle transition control. Stone et al. [7] put forward an off-line optimization process to optimize the throttle output during the transition based on the aerodynamic database of aircraft, however, the altitude and lateral motion are essentially uncontrolled. In [8], three transition controllers were investigated and compared on a model Convair XFY-1 Pogo. They are a simple controller based on a vector-thrust model, a feedback linearization controller and a model reference adaptive controller (MRAC) respectively. All the three controllers didn’t consider the lateral motion. In [9], the minimum time and minimum energy optimization problems of VTOL transition were presented and solved numerically. Oosedo et al. [10] discussed three transition strategies: standard PID feedback control, minimizing the transition time, minimizing the transition time while keeping the altitude. The latter two strategies are designed based on the aerodynamic model from the wind tunnel test. Pucci et al. [11] proposed an equivalent transformation from the original system dynamic, which depends on the aircraft orientation, to the one that is independent of the orientation. This method enables the separate computation of thrust and orientation, which will reduce the controller design complexity. The verified test can be found in [12]. In these works, the altitude change of the tail-sitter UAV in a transition is usually not satisfactory, the best of them [10] has an altitude change around 40 cm.

To achieve this performance, one needs to conduct costly wind tunnel tests to obtain an accurate aerodynamic model.
of the UAV. Furthermore, all these methods are designed for a single transition while the performance of the Cobra maneuver involving two consecutive quick transitions could worsen.

In this paper, we propose a novel feedback-forward control structure to tackle the Pugachev’s Cobra maneuver of tail-sitter VTOL UAVs. The proposed controllers are designed based on the aircraft acceleration model, which is directly measurable from the UAV onboard IMU sensors. The feedforward controller is an iterative learning controller derived in the lifted domain [13], which has been extensively used in quadrotor trajectory tracking control, [13]–[15]. When compared to the standard ILC algorithms in [13], our method does not require any experimental model identification due to the use of acceleration model. Besides the feedback-feedforward control for altitude regulation, a lateral-directional controller is also designed to control the UAV lateral motion, which is usually neglected in prior work [7]–[12].

The proposed control methods are verified on our “Hong Hu” 4 UAV (see Fig. 2), an ultra maneuverable and efficient quadrotor tail-sitter VTOL UAV [6]. Experiments show that the proposed control methods achieve state-of-the-art control performance. The altitude change through the entire Pugachev’s Cobra maneuver is as low as 10cm. A demonstrative video showing the effectiveness of the system can be found at https://youtu.be/LTFIk1FDrGw.

The remainder of this paper is organized as follows. Section II will introduce the platform and its model. The detailed design of the controller will be described in section III. The iterative learning control algorithm will be described in section IV. Experimental results are provided in section IV. Finally, section V draws the conclusion.

II. SYSTEM CONFIGURATION

A. System Configuration

“Hong Hu” UAV consists of a trapezoidal wing with MH-115 airfoil for improved aerodynamic efficiency, a fuselage containing most of the avionics, and four landing gears with four rotors. Unlike conventional fixed-wing aircraft, the UAV does not have any aileron or elevator which means servos are no longer needed. The propeller differential thrusts are the only source of control moments. This improvement will bring a lot of benefits, such as the reduction of dead weight, the increase of flight efficiency because there are no longer gaps in the wing surface. Besides these, the lack of aileron or elevator ease the controller design in all flight modes without carefully designing the actuation allocation among the four rotors and the aileron. The Hong Hu UAV is manufactured with carbon fiber which leads to improved stiffness and strength of the structure. The flight tests show that the high-speed level flight of the Hong Hu UAV is five times more efficient than that in hovering [6], while still being very portable with a 0.9m wingspan. The configuration parameters are specified in the Table I.

| TABLE I | AIRCRAFT CONFIGURATION |
|---------|------------------------|
| Propeller name | APC9x6E |
| Motor name | Sunnysky A2212 (980KV) |
| Wing span | 0.90 m |
| Taper ratio | 0.48 |
| Swept angle | 7.30° |
| Angle of attack | 7.00° |
| Root chord of wing | 0.20 m |

The flight-relevant software is running on a Pixhawk 4 Mini controller board which is the latest version of Pixhawk4 hardware serials with a 216MHz Cortex-M7 processor. An onboard inertial measurement unit (IMU), an M8N GPS/GNSS module, and a pitot tube airspeed sensor [5] make up the sensor system. Based on the extended Kalman filter, the position, attitude, and velocities are estimated by fusing all the sensor data. For the communication between ground and the UAV, a 2.4 GHz transmitter and receiver is used to transmit the remote controller (RC) signals from ground pilot to the controller board, and a two-way 433 MHz telemetry radio is used to transfer the information between the aircraft and a ground station [16].

B. Coordinates Definition

The inertial coordinate frame (x_i, y_i, z_i) is defined as Fig. 3 shown, the axis x_i, y_i and z_i point to the North, East and Down (NED) respectively. For the body frame, the axis x_b points forward along the body axis, the axis z_b points down within the symmetry plane of body. The intrinsic z-y-z is chosen as the rotational order of Euler angles. The angle ψ, θ, η is defined as the rotation angle along the axis z_i, y_b and z_b respectively. The quaternion is another way used to

Fig. 2. The Hong Hu quadrotor tail-sitter UAV

4Hong Hu is an alias of swan in Chinese. The name came from the idiom The Ambition of Hong Hu, which first appeared in the Records of the Grand Historian (also known as Shi Ji in Chinese), by Sima Qian at 94 BC.
describe the rotation. The detailed explanation about direct cosine matrix, and quaternion can be found in [4].

C. Dynamic Model

The position of the aircraft is denoted as \( p^i = [p^i_x, p^i_y, p^i_z]^T \). The angular rate represented in the body frame is \( \omega^b \). The total mass of the aircraft is \( m \), and the inertia matrix is denoted as \( I \). Based on the rigid body assumption and Newton-Euler equation, the rotational and translational dynamics of the aircraft can be modeled as follows

\[
I \ddot{\omega}^b = -\omega^b \times (I \omega^b) + \tau + M_a \tag{1}
\]

\[
\dot{p}^i = g + R_i^b a^b \tag{2}
\]

where the \( \tau \) and \( M_a \) stands for the control moment vector and aerodynamic moment respectively, the \( R^b_i \) is the rotation matrix from body frame to inertial frame, the \( a^b \) is the total acceleration in the body frame. The details about \( a^b \) will be described in the next section.

III. CONTROLLER DESIGN

A. Controller Structure

The controller structure for the Pugachev’s Cobra maneuver is shown in Fig. 4. The navigation module, which will not be discussed, generates the target altitude \( p_{zd} \) which is a constant value for a Pugachev’s Cobra maneuver, target pitch \( \theta_d \) which is a prescribed trajectory that is usually linearly increasing or decreasing for achieving the Cobra maneuver, and the desired flight path \( p_{hd} \), which is a straight line in the horizontal plane.

![Fig. 4. The controller structure for high AOA flight](image)

Assume that before executing the Pugachev’s Cobra maneuver, the tail-sitter UAV is at stationary hovering with body \( z_b \) axis parallel the desired path and the initial attitude being \( R_0 \). During the Cobra maneuver, the UAV tracks the target pitch angle \( \theta_d \) and deviation from the desired path in horizontal plane could take place due to environment disturbances. To correct this deviation, the lateral-directional controller rotates along the UAV body \( z_b \) axis by angle \( \eta_d \) (Fig. 5 (a)), such that the forward acceleration \( a^b_x \) is projected to lateral direction and a lateral acceleration is produced to reduce this deviation from the desired path. The angle \( \eta_d \) is determined by the PD controller (3) and (4).

\[
a_{td} = k_{ld} \xi_{ld} \tag{3}
\]

\[
\eta_d = \arctan \left( \frac{a_{td}}{a^b_z} \right) = \arctan \left( \frac{k_{ld} \xi_{ld}}{a^b_z} \right) \tag{4}
\]

where \( k_{ld} \xi_{ld} \) is the feedback controller. This proportional controller acting on the UAV lateral dynamics with inherent aerodynamic damping effect will make a second order system that regulates the lateral error around zero.

![Fig. 5. (a) Body x acceleration \( a^b_x \) and its two orthogonal components: \( a^b_x \) and \( a_{td} \). \( a_{td} \) is perpendicular to the desired path (maximally suppress the lateral deviation), the \( z_b \) axis \( (o_{zd}) \) is in the horizontal plane, and \( z_b \) \( (o_{zd}) \) is the component of \( a^b_z \) rotated out of the \( o-z_b-z_1 \) plane, \( a^b_x \) is the component of \( a^b_z \) within the plane \( o-z_b-z_1 \). This component will affect the UAV altitude. (b) The accelerations of the UAV in the \( o-z_b-z_1 \) plane. \( a^b_x \) is the acceleration in body \( X \) axis, \( a^b_y \) is the component of \( a^b_z \), the acceleration in body \( X \) axis, projected to the \( o-z_b-z_1 \) plane, and \( g \) is the gravity.](image)

Based on the initial hovering attitude \( R_0 \), the target pitch angle \( \theta_d \) from the navigation module, and the angle \( \eta_d \) along with the body \( z_b \) axis, the target attitude is determined as:

\[
R_d = R_0 R_y(\theta_d) R_z(\eta_d) \tag{5}
\]

This target attitude \( R_d \) is sent to a lower-level controller for attitude tracking. The attitude controller is a dual-loop control structure where the outer loop is a quaternion-based attitude controller that operates in the full Special Orthogonal group \( SO(3) \) [17]. The inner loop is three independent angular rate controllers that are designed with loopshaping techniques. Notch filters are used to improve the controller bandwidth and performance by suppressing the UAV flexible modes [18].

We propose an acceleration model for the altitude control. There are several benefits of the acceleration model: (1) With the acceleration model, the UAV translation dynamics are linear and does not depend on any model parameters (e.g., aerodynamic coefficients, etc.). This greatly simplify
the altitude controller design; (2) The acceleration is easily measurable by IMU sensors which are almost always available for any UAVs; (3) The acceleration measurements of an IMU sensor are the UAV acceleration in body X, Y and Z axis, respectively. These three components can be controlled by independent actuations. For example, the body X acceleration is independently controlled by the collective thrust of the four rotors.

![Fig. 6. The structure of vertical controller](image)

Fig. 6. The structure of vertical controller

Fig. 6 shows our altitude controller structures. The altitude controller generates the acceleration command \( a^{b}_{zd} \) which is tracked by a lower-level body X acceleration controller. Since the lateral acceleration component of \( a^{b}_{zd} \) is in the horizontal plane, it does not affect the UAV altitude. In addition, the accelerometer within an IMU measures all the acceleration but not gravity, the total UAV acceleration is the accelerometer measurements plus the gravity vector. Therefore, the translation acceleration that affects the UAV altitude is summarized in (5) and the translation dynamics is in (5).

\[
p^{i}_{zd} = a^{b}_{zd} \cos(\theta) + a^{b}_{zd} \sin(\theta) \cos(\eta) - g \tag{5}
\]

Based on (5), the feed-forward \( a^{ff}_{zd} \) and feedback \( a^{fb}_{zd} \) control are determined from (6), (7) and (8).

\[
a^{ff}_{zd} \sin(\theta) \cos(\eta) = a^{b}_{zd} \cos(\theta) - p^{i}_{zd} + g \tag{6}
\]

\[
a^{fb}_{zd} \sin(\theta) \cos(\eta) = k_{p} \xi_{p} + k_{v} \xi_{v} \tag{7}
\]

\[
a^{b}_{zd} = a^{ff}_{zd} + a^{fb}_{zd} \tag{8}
\]

where the \( p^{i}_{zd} \) is the desired altitude; \( \xi_{p} = p^{i}_{zd} - p^{i}_{zd} \) and \( \xi_{v} = p^{i}_{zd} - v^{i}_{zd} \) is the error of altitude and vertical velocity; and \( k_{p} \) and \( k_{v} \) are the proportional and derivative gains of the feedback controller.

The body X acceleration command \( a^{b}_{zd} \) computed from the altitude controller is tracked by a lower-level acceleration controller which directly actuates the collective throttle to all the four propellers. The acceleration controller is a PI controller with a low pass filter shown in Fig. 7.

![Fig. 7. The structure of body x axis acceleration controller](image)

IV. Iterative Learning Strategy

The altitude controller designed in the previous section, even with the feedforward corrections, usually cannot track the constant altitude command with satisfactory performance due to environmental disturbances and the transient response of the lower level acceleration controller. To track the desired path precisely, an iterative learning feed-forward controller is further utilized to the vertical controller as shown in Fig. 8.

![Fig. 8. Adding the ILC correction to the feed-forward section of altitude controller](image)

A. Lifted Domain Model

The first step of using the iterative learning algorithm is obtaining a lifted domain model near the desired trajectory. In this paper, the vertical position \( (p^{i}_{zd}) \) and the vertical velocity \( (v^{i}_{zd}) \) in the inertial frame are chosen to compose the state vector \( X \):

\[
X = [p^{i}_{zd}, v^{i}_{zd}] \tag{9}
\]

When the iterative learning correction \( u^{ILC} \) is added to the feed-forward controller, (6) is rewritten as:

\[
a^{ff}_{zd} \sin(\theta) \cos(\eta) = g - a^{b}_{zd} \cos(\theta) - p^{i}_{zd} + u^{ILC} \tag{10}
\]

Neglecting the transient response of the lower-level acceleration controller (i.e. \( a^{b}_{zd} = a^{b}_{zd} \)) yields the desired closed-loop system as below:

\[
\dot{X} = AX + Bu^{ILC}
\]

\[
p^{i}_{zd} = CX
\]

\[
A = \begin{bmatrix} 0 & 1 \\ -k_{p} & -k_{v} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = [1, 0]
\]

The corresponding closed-loop state equation in discrete time domain is:
\[ \begin{align*}
X(k+1) &= A_D X(k) + B_D u^{ILC}(k) \\
p_z'(k) &= C_D X(k)
\end{align*} \] (12)

where \( k \in \{0, 1, ..., N\} \) is the discrete-time index, \( \Delta t \) is the sampling time, \( A_D = I + A_D \Delta t \), \( B_D = B_D \Delta t \) and \( C_D = C \).

In the actual flight, the lifted domain [15] input vector is denoted as \( \vec{u} = [u^{ILC}(0), ..., u^{ILC}(N-1)]^T \in U \subseteq \mathbb{R}^N \), the state vector is denoted as \( \vec{x} = [X(0), ..., X(N-1)]^T \in U \subseteq \mathbb{R}^{N+2} \), the output vector is denoted as \( \vec{y} = [p_z'(0), ..., p_z'(N-1)]^T \in U \subseteq \mathbb{R}^N \). The desired trajectory is denoted as the desired trajectory \((\vec{u}_{des}, \vec{x}_{des}, \vec{y}_{des})\). For the ideal Pugachev’s Cobra, \( \vec{y}_{des} = 0 \). The first trial test without any iterative control input (i.e. \( u^{ILC}(k) = 0 \)) is used as the initial trajectory \((\vec{u}_{init}, \vec{x}_{init}, \vec{y}_{init})\), therefore the deviations from initial trajectories is as (13).

\[
\vec{y} = \vec{F} \vec{u} + d
\] (14)

where the matrix \( \vec{F} \) and \( \vec{G} \) is calculated as (15); \( d \) contains the repetitive disturbance along the trajectory, which is primarily caused by unmodeled dynamics (e.g., transient response of the lower-level acceleration controller) and steady environmental disturbances (e.g., wind, etc.).

\[
F(l, m) = \begin{cases} 
C_D A_D^{l-m} B_D & m < l - 1 \\
C_D B_D & m = l - 1 \\
0 & m > l - 1
\end{cases}
\] (15)

B. Disturbance Estimation and Input Update

An iteration domain Kalman filter is applied to calculate the estimate \( \hat{d}_{j+1} \) of the disturbance vector \( d \) in (14) after each iteration as (16), \( i \) indicates the \( i \)th execution of Pugachev’s Cobra maneuver.

\[
\hat{d}_{j+1} = \hat{d}_j + K_j \left( \vec{y}_j - \vec{F} \hat{u}_j - \vec{d}_j \right)
\] (16)

where the \( K_j \) is the Kalman gain of which the details can be found in [14].

Based on the lifted domain model and estimated disturbance, the new input vector \( \hat{u}_j \) can be calculated based on a model-based optimization rule. The prediction of next output for the iterative learning control input \( \hat{u}_{j+1} \), which needs to be determined, is as follow:

\[
\begin{align*}
\min_{\hat{u}_{j+1}} & \ |E[\vec{y}_{j+1} | \vec{y}_0, ..., \vec{y}_{j}] - \vec{y}_{des}|_2 + \alpha \|D\hat{u}_{j+1}\|_2 \\
\text{s.t.} & \ \hat{u}_{j+1} \leq c_{max}
\end{align*}
\] (18)

where the first term stands for the 2-norm of trajectory tracking error, and the second one is the penalty term to the iterative learning control input with the weight \( \alpha \). The lifted vector \( c_{max} \) denotes the maximally allowed \( \hat{u}_{j+1} \).

The problem described (18) is a convex optimization problem [19], which can be solved by software tool such as the CVX toolbox of Python.

V. Experiments Verification

This section presents the experimental results of the proposed controller design and learning scheme applied to Pugachev’s Cobra maneuver.

The loop transfer function \( L \) of the body X acceleration control loop is shown in Fig. 9, where \( P \) is the actual plant identified from sweep experiment [18], the \( C \) is the PI controller with a low-pass filter. The bandwidth and phase margin in the acceleration control loop is 7.2 Hz and 54° which are both in the desired range for typical aircraft systems. Fig. 10 shows the attitude change and lateral movement in the initial trajectory. It is obvious that the actual attitude \( \theta \) tracks the prescribed trajectory \( \theta_{sp} \) very well and the deviation from the flight path is below 1m.

Fig. 9. The open-loop frequency response of body X acceleration controller.

Fig.11 shows the input sequence and output altitude during the iteration of Pugachev’s Cobra maneuver. After only two iterations, the aircraft follows the desired trajectory pretty closely. The maximum altitude error, minimum altitude error and root mean square at each iteration are shown in Table II. The best achieved altitude change is as low as 10cm (r.m.s.). The video of the final Pugachev’s Cobra maneuver can be fond at https://youtu.be/LTFIk1FDTrGw.
The quadrotor tail sitter UAV can keep high maneuverability in high AOA flight. To prove it, the well-known post-stall maneuver named Pugachev’s Cobra is tested. For this maneuver, a control system is presented in this paper. This control system is composed of three parts: a full attitude controller, a lateral-directional controller, and a feedback-feedforward altitude controller. The altitude controller is designed based on the UAV acceleration model. Because of the environmental disturbance and unmodeled dynamics, the acceleration based iterative learning control algorithm is used to correct the feed-forward output of the altitude controller. A big advantage of the acceleration based iterative learning control is that it does not require to identify the UAV aerodynamic parameters, eliminating the use of costly wind tunnel tests while achieving state-of-the-art control performance. The proposed control methods are verified by real world outdoor flight experiments.

REFERENCES

[1] B. Gal-Or, “Thrust vectoring for flight control & safety: A review,” International Journal of Turbo and Jet Engines, vol. 11, no. 2-3, pp. 119–138, 1994.
[2] Y. Wang, X. Lyu, H. Gu, S. Shen, Z. Li, and F. Zhang, “Design, implementation and verification of a quadrotor tail-sitter vtol uav,” in Unmanned Aircraft Systems (ICUAS), 2017 International Conference on. IEEE, 2017, pp. 462–471.
[3] X. Lyu, H. Gu, Y. Wang, Z. Li, S. Shen, and F. Zhang, “Design and implementation of a quadrotor tail-sitter vtol uav,” in Robotics and Automation (ICRA), 2017 IEEE International Conference on. IEEE, 2017, pp. 3924–3930.
[4] F. Zhang, X. Lyu, Y. Wang, H. Gu, and Z. Li, “Modeling and flight control simulation of a quadrotor tail-sitter vtol uav,” in AIAA Modeling and Simulation Technologies Conference, 2017, p. 1561.
[5] X. Lyu, H. Gu, J. Zhou, Z. Li, S. Shen, and F. Zhang, “A hierarchical control approach for a quadrotor tail-sitter vtol uav and experimental verification,” 2017.
[6] H. Gu, X. Cai, J. Zhou, Z. Li, S. Shen, and F. Zhang, “A coordinate descent method for multidisciplinary design optimization of electric-powered winged uavs,” in 2018 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, 2018, pp. 1189–1198.
[7] R. H. Stone, “Control architecture for a tail-sitter unmanned air vehicle,” in 2004 5th Asian Control Conference (IEEE Cat. No. 04EX904), vol. 2. IEEE, 2004, pp. 736–744.
[8] N. Knoebel, S. Osborne, D. Snyder, T. Mclain, R. Beard, and A. Eldredge, “Preliminary modeling, control, and trajectory design for miniature autonomous tailissiers,” in AIAA Guidance, Navigation, and Control Conference and Exhibit, 2006, p. 6713.
[9] R. Naldi and L. Marconi, “Optimal transition maneuvers for a class of v/stol aircraft,” Automatica, vol. 47, no. 5, pp. 870–879, 2011.
[10] A. Ooshodo, S. Abiko, A. Konno, and M. Uchiyama, “Optimal transition from hovering to level-flight of a quadrotor tail-sitter uav,” Autonomous Robots, vol. 41, no. 5, pp. 1143–1159, 2017.
[11] D. Pucci, T. Hamel, P. Morin, and C. Samson, “Nonlinear control of vtol vehicles subjected to drag and lift,” in 2011 50th IEEE Conference on Decision and Control and European Control Conference. IEEE, 2011, pp. 6177–6183.
[12] D. Pucci, “Analysis and control of aircraft longitudinal dynamics subject to steady aerodynamic forces.” arXiv preprint arXiv:1608.02505, 2016.
[13] O. Purwin and R. D’Andrea, “Performing aggressive maneuvers using iterative learning control,” in 2009 IEEE International Conference on Robotics and Automation. IEEE, 2009, pp. 1731–1736.
[14] A. Schöllig and R. D’Andrea, “Optimization-based iterative learning control for trajectory tracking,” in 2009 European Control Conference (ECC). IEEE, 2009, pp. 1505–1510.
[15] F. L. Mueller, A. P. Schoellig, and R. D’Andrea, “Iterative learning of feed-forward corrections for high-performance tracking,” in 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012, pp. 3276–3281.
[16] P. Autopilot. “Pixhawk community.”
[17] Z. Li and J. F. Canny, Nonholonomic motion planning. Springer Science & Business Media, 2012, vol. 192.
[18] J. Zhou, X. Lyu, X. Cai, Z. Li, S. Shen, and F. Zhang, “Frequency domain model identification and loop-shaping controller design for quadrotor tail-sitter vtol uavs,” in 2018 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, 2018, pp. 1142–1149.
[19] S. Bubeck et al., “Convex optimization: Algorithms and complexity,” Foundations and Trends® in Machine Learning, vol. 8, no. 3-4, pp. 231–357, 2015.