

Advancing Fault-Tolerant Learning-Oriented Control for Unmanned Aerial Systems

Moh Kamalul Wafi, Rozhin Hajian, Bahram Shafai, Milad Siامي

Abstract—The rapid advancement of automatic control technology has sparked significant interest among researchers in creating more reliable and simplified models of unmanned aerial vehicles (UAVs). This interest is motivated by the need to enhance the performance and resilience of these systems in challenging conditions, such as wind gusts and adverse weather. This paper presents novel strategies for enhancing the resilience of unmanned aerial systems (UAS) with fault-tolerant control (FTC) by learning-oriented control and a constructive fault estimation with Proportional-Integral (PI) observer. The learning-control is deep-deterministic policy gradient (DDPG) which is trained in only one state but used beyond its environment for other states to control. The faults are designed in three divergent conditions and the augmented PI observer is responsible in capturing them. The success of estimating the faults is used for this FTC to compensate the faulty system with learning-oriented control as the advancement of the FTC. The proposed approach has the potential to enhance the performance and resilience of UAVs, thus contributing to the development of more robust and reliable systems.

I. INTRODUCTION

Unmanned aerial systems (UAS) have become a vital part of military and civilian applications [1], [2]. However, UAS face significant challenges when flying under adverse weather conditions, such as wind gusts, which can pose hazards to the UAS and its payloads. The development of resilient and fault-tolerant control systems for UAS is crucial to ensure the reliability and safety of these systems [3]. Fault detection in UAS is an essential area of research that has received significant attention in recent years [4]–[6]. Faults in the UAS system can lead to loss of control, instability, and even crashes. One of the main challenges in developing a fault-tolerant control system for UAS is the high-dimensional and complex control system architecture that requires the integration of various sensors and controllers to achieve the desired control objectives.

Furthermore, the dynamics of unmanned aerial vehicles (UAVs) are a critical component in developing a fault-tolerant controller for the UAV/quadcopter. The reliability and safety of UAS remain a major concern due to the risk of system failures and malfunctions. Thus, the system

models should be well-identified so as to cover the suitable control methods, from the conservative canonical form to advanced approaches [7]. This is due to the fact that signal communications of UAV, both in-use local scenarios and grounding data-transferring, are becoming more crucial in response to failures. Moreover, the optimal navigation to moving objects [8], efficient and selective channel designs of UAV to ground communications [9]–[11], and the booster joint-spectrum mobile-based to fasten the UAV communications both single [12] and multi-UAV with security [13] are the recent instances to maintain the true-signals to prevent failures.

To achieve this goal, the PI observer [14]–[16] is challenged to track the system states when control engineering-related faults occur, such as sensor and actuator faults [17], [18]. The key idea is to capture the faults and to feed into faulty system so that the system becomes free-fault. Furthermore, the advancement comes from the deep reinforcement learning (DRL) [19] which is then used for fault-tolerant control [20]–[22]. However, we offer to train the DRL only in single state and use the control agent for other states to see the behaviour beyond its environment.

This paper is structured as follows: In Section II, we begin by presenting the dynamical systems of the UAV/Quadcopter. Section III describes the FTC scheme, followed by the PI observer in Section IV. We then delve into the learning-oriented control approach of DDPG reinforcement learning in Section V. In Section VI, we present our numerical simulations being ended by the conclusion in Section VII.

II. DYNAMICS OF QUADCOPTER

A. Mathematical Model of Quadcopter

The analyzed dynamics of the quadcopter is shown in Fig. 1 with respect to two reference frames, the inertial and body along with the respected forces and torques from four divergent rotors, $\forall i = 1, 2, 3, 4$. The fixed earth-inertial (E) perspective is denoted as regards to gravity directing to the opposite of z -axis while the corresponding axes of the body frame (B) are attached to the fixed arms. For each of the arms, the propellers are placed on the four motors to generate thrust with the inlet ω_i and outlet τ_{m_i} from the perspective of north-south directions such that ω_1, ω_3 and ω_2, ω_4 move in the counter- and clockwise direction in turn. Furthermore, the absolute center of quadcopter mass position with respect to inertial structure (x, y, z) is denoted by ζ while the attitude of three Euler angles (ϕ, θ, ψ) with the same frame is denoted by η which rotates associated to ζ , $\phi \rightarrow x$, $\theta \rightarrow y$, and $\psi \rightarrow z$, where

M.K.W, B.S., and M.S. are with the Department of Electrical & Computer Engineering, Northeastern University, Boston, MA 02115 USA (e-mails: {wafi.m, b.shafai, m.siami}@northeastern.edu).

R.H is with the Department of Mechanical Engineering, UMass Lowell, MA 01854 USA (e-mail: rozhin.hajian@uml.edu).

This research was supported in part by grants ONR N00014-21-1-2431, NSF 2121121, and the Army Research Laboratory and accomplished under Cooperative Agreement Number W911NF-22-2-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government.

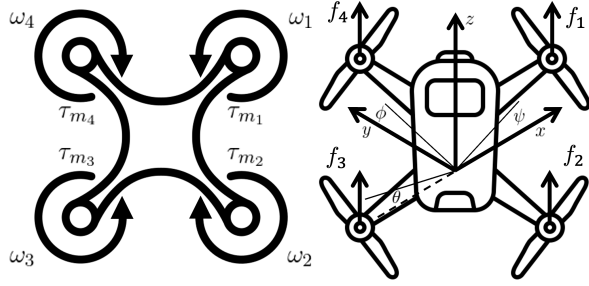


Fig. 1: The figure shows the structural frames of a quadcopter as viewed from both the inertial and body perspectives.

$$\zeta = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \eta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, \quad \mu_p = \begin{bmatrix} \zeta \\ \eta \end{bmatrix}, \quad (1)$$

and

$$V_B = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad \vartheta = \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \quad \mu_v = \begin{bmatrix} V_B \\ \vartheta \end{bmatrix}. \quad (2)$$

Note that μ_p and μ_v combine the terms of positions and velocities information. The origin fixed-body attaching to the arms has linear V_B and angular ϑ velocities from the three axes. The rotational relationship between the two frames is shown as R in which the bold $\mathbf{c}$, $\mathbf{s}$, and $\mathbf{t}$ define the cosine, sine and tangent notation where $\mathbf{c}_x = \cos(x)$. Keep in mind that R is an orthogonal matrix, meaning its inverse equals to the transpose, $R^{-1} = R^\top$, therefore

$$\begin{bmatrix} R_x(\phi) & R_y(\theta) & R_z(\psi) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathbf{c}_\phi & -\mathbf{s}_\phi \\ 0 & \mathbf{s}_\phi & \mathbf{c}_\phi \end{bmatrix} \begin{bmatrix} \mathbf{c}_\theta & 0 & \mathbf{s}_\theta \\ 0 & 1 & 0 \\ -\mathbf{s}_\theta & 0 & \mathbf{c}_\theta \end{bmatrix} \begin{bmatrix} \mathbf{c}_\psi & -\mathbf{s}_\psi & 0 \\ \mathbf{s}_\psi & \mathbf{c}_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

The following rotation from fixed-body to earth is

$$R_{B \rightarrow E} := R_z(\psi) \cdot R_y(\theta) \cdot R_x(\phi), \quad (4)$$

and the complete rotational matrix of two frames constitutes

$$R = \begin{bmatrix} \mathbf{c}_\psi \mathbf{c}_\theta & \mathbf{c}_\psi \mathbf{s}_\theta \mathbf{s}_\phi - \mathbf{s}_\psi \mathbf{c}_\phi & \mathbf{c}_\psi \mathbf{s}_\theta \mathbf{c}_\phi + \mathbf{s}_\psi \mathbf{s}_\phi \\ \mathbf{s}_\psi \mathbf{c}_\theta & \mathbf{s}_\psi \mathbf{s}_\theta \mathbf{s}_\phi + \mathbf{c}_\psi \mathbf{c}_\phi & \mathbf{s}_\psi \mathbf{s}_\theta \mathbf{c}_\phi - \mathbf{c}_\psi \mathbf{s}_\phi \\ -\mathbf{s}_\theta & \mathbf{c}_\theta \mathbf{s}_\phi & \mathbf{c}_\theta \mathbf{c}_\phi \end{bmatrix}. \quad (5)$$

The propellers mounted on the motor are assumed to be identical so that the mathematical dynamics is with respect to only one. The thrust T generating from them is driven by the momentum forces f_i ,

$$f_i = (C_d \rho A r^2) \omega_i^2 \quad \longrightarrow \quad T = \sum_i f_i = k \sum_i \omega_i^2, \quad (6)$$

where C_d , ρ , A , and r explain the thrust parameter, the air density, the propeller cross-sectional area, and the rotor radius in turn, while ω_i express the angular velocity for certain i -th rotor direction. The k -lift value is then used to simplify the first four fixed variables to get the net-thrust from them. The angular velocity ω_i and its acceleration $\dot{\omega}_i$ drive the torque τ_{m_i}

$$\tau_{m_i} = b \omega_i^2 + I_m \dot{\omega}_i, \quad \text{and} \quad \dot{\omega}_i \approx 0, \quad (7)$$

where b and I_m is the drag parameter and rotor inertial moment. With equal velocities, the quadcopter will go-up, -down and hover in z -direction corresponding to the magnitude applied relative to gravity while the moments generate the motions of roll- ϕ , pitch- θ , and yaw- ψ . Therefore, the net-thrust T_B in the z -axis and the torques τ_B from three angles perspective are indicated below,

$$T_B = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}, \quad \tau_B = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} L(f_4 - f_2) \\ L(f_3 - f_1) \\ \sum_i \tau_{m_i} \end{bmatrix}, \quad (8)$$

where L is the length of propeller and center of mass. The roll and pitch movement are produced by thrust difference, described with negative term from λ_1, λ_2 , of a pair from 2-4 and 1-3 rotor, respectively. Regarding the yaw, the angular velocities ω of an opposite pair should be decimated while the two counterparts are escalated. To conclude the derivation, we assume that the quadcopter is exactly symmetrical, which leads to $I_{xx} = I_{yy}$, and gives us $\text{diag}([I_{xx}, I_{yy}, I_{zz}])$, in which $\text{diag}(\bullet)$ denotes the diagonal matrix of entry $(\bullet)$. This inertia matrix is diagonal and describes the linear time-invariant (LTI) characteristics of the system. The equation of motions with the rigid-body assumption is presented here. The force to drive the mass acceleration $m\dot{V}_B$ and the centrifugal force $\vartheta \times (mV_B)$ are on par with those of the gravity $R^\top \rho$ and the net-thrust T_B , therefore

$$m\dot{V}_B + \vartheta \times (mV_B) = R^\top \rho + T_B. \quad (9)$$

where in the inertial-translational structure, the leverage of centrifugal could be neglected. However, in the inertial frame, the centrifugal force is nullified. Thus, only the gravitational force, the magnitude and the direction of the thrust are contributing in the acceleration of the quadcopter, such that $m\ddot{\zeta} = \rho + RT_B$, or

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{T}{m} \begin{bmatrix} \mathbf{c}_\psi \mathbf{s}_\theta \mathbf{c}_\phi + \mathbf{s}_\psi \mathbf{s}_\phi \\ \mathbf{s}_\psi \mathbf{s}_\theta \mathbf{c}_\phi - \mathbf{c}_\psi \mathbf{s}_\phi \\ \mathbf{c}_\theta \mathbf{c}_\phi \end{bmatrix}. \quad (10)$$

Likewise, as for body-rotational structure, the inertia angular acceleration $I\dot{\vartheta}$ along with the forces of both centripetal $\vartheta \times (I\vartheta)$ and gyroscopic Γ yield the torque τ ,

$$I\dot{\vartheta} + \vartheta \times (I\vartheta) + \Gamma = \tau, \quad (11)$$

where for more detail equation, it constitutes

$$\dot{\vartheta} = I^{-1} \left(-\vartheta \times \begin{bmatrix} I_{xx} p \\ I_{yy} q \\ I_{zz} r \end{bmatrix} - I_r \vartheta \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Omega_\Gamma + \tau \right), \quad (12)$$

which could be elaborated as follows by considering the Ω_Γ as $\omega_1, -\omega_2, \omega_3$, and $-\omega_4$ from the yaw movement,

$$\begin{aligned} \dot{p} &= \frac{(I_{yy} - I_{zz})qr}{I_{xx}} - I_r \frac{q}{I_{xx}} \Omega_\Gamma + \frac{\tau_\phi}{I_{xx}}, \\ \dot{q} &= \frac{(I_{zz} - I_{xx})pr}{I_{yy}} + I_r \frac{p}{I_{yy}} \Omega_\Gamma + \frac{\tau_\theta}{I_{yy}}, \\ \dot{r} &= \frac{(I_{xx} - I_{yy})pq}{I_{zz}} + \frac{\tau_\psi}{I_{zz}}. \end{aligned} \quad (13)$$

However, this angular acceleration from the rotational frame could then be extracted and transformed to the translational frame using the transformation matrix W_η^{-1} and vice versa using W_η , in terms of $\dot{\eta}$ and ϑ ,

$$\begin{aligned} \dot{\eta} &= W_\eta^{-1} \vartheta, \quad \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{s}_\phi \mathbf{t}_\theta & \mathbf{c}_\phi \mathbf{t}_\theta \\ 0 & \mathbf{c}_\phi & -\mathbf{s}_\phi \\ 0 & \mathbf{s}_\phi / \mathbf{c}_\theta & \mathbf{c}_\phi / \mathbf{c}_\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \\ \vartheta &= W_\eta \dot{\eta}, \quad \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\mathbf{s}_\theta \\ 0 & \mathbf{c}_\phi & \mathbf{s}_\phi \mathbf{c}_\theta \\ 0 & -\mathbf{s}_\phi & \mathbf{c}_\phi \mathbf{c}_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \end{aligned} \quad (14)$$

Moreover, the partial derivative of $\dot{\eta}$ constitutes the following two terms, such that

$$\ddot{\eta} = \frac{d}{dt} (W_\eta^{-1} \vartheta) = \frac{d}{dt} (W_\eta^{-1}) \vartheta + W_\eta^{-1} \dot{\vartheta}, \quad (15)$$

where $\dot{W}_\eta^{-1}$ is expressed as follows,

$$\dot{W}_\eta^{-1} = \begin{bmatrix} 0 & \dot{\phi} \mathbf{c}_\phi \mathbf{t}_\theta + \dot{\theta} \mathbf{s}_\phi / \mathbf{c}_\theta^2 & -\dot{\phi} \mathbf{s}_\phi \mathbf{c}_\theta + \dot{\theta} \mathbf{c}_\phi / \mathbf{c}_\theta^2 \\ 0 & -\dot{\phi} \mathbf{s}_\phi & -\dot{\theta} \mathbf{c}_\phi \\ 0 & \dot{\phi} \mathbf{c}_\phi / \mathbf{c}_\theta + \dot{\psi} \mathbf{s}_\phi \mathbf{t}_\theta / \mathbf{c}_\theta & -\dot{\phi} \mathbf{s}_\phi / \mathbf{c}_\theta + \dot{\theta} \mathbf{c}_\phi \mathbf{t}_\theta / \mathbf{c}_\theta \end{bmatrix}$$

B. Alternative Lagrangian Derivation of Motions

Beyond Newtonian-Euler derivation, the Lagrangian as a energy collective of translation E_T and rotation E_R being subtracted from potential energy E_P is written as follows,

$$\mathcal{L}(\mu, \dot{\mu}) = \frac{m}{2} \dot{\zeta}^\top \dot{\zeta} + \frac{1}{2} \vartheta^\top I \vartheta - mgz, \quad (16)$$

while as studied in [10], the Euler-Lagrangian formulations considering the external forces f torques τ comprise,

$$\begin{bmatrix} f \\ \tau \end{bmatrix} = \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\mu}} \right) - \frac{\partial \mathcal{L}}{\partial \mu}, \quad (17)$$

Since these two quantities are independent, the derivation can be handled separately. First, the translational force is determined as the net thrust acting on the propellers as

$$f = RT_B = m\ddot{\zeta} + mg \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top, \quad (18)$$

which is the same as (10), while the Jacobian matrix $J(\eta)$, from ϑ to $\dot{\eta}$ constitute $J(\eta) := J = W_\eta^\top I W_\eta$, where

$$J(\eta) = \begin{bmatrix} I_{xx} & 0 & -I_{xx} \mathbf{s}_\theta \\ 0 & I_{yy} \mathbf{c}_\phi^2 + I_{zz} \mathbf{s}_\phi^2 & \xi_1 \\ -I_{xx} \mathbf{s}_\theta & \xi_1 & \xi_2 \end{bmatrix}, \quad (19)$$

with the terms ξ_1 and ξ_2 equal to $\xi_1 := (I_{yy} - I_{zz}) \mathbf{c}_\phi \mathbf{s}_\phi \mathbf{c}_\theta$ and $\xi_2 = I_{xx} \mathbf{s}_\theta^2 + I_{yy} \mathbf{s}_\phi^2 \mathbf{c}_\theta^2 + I_{zz} \mathbf{c}_\phi^2 \mathbf{c}_\theta^2$. Therefore the rotational energy E_R can be denoted in terms of inertial frame,

$$E_R = \frac{1}{2} \vartheta^\top I \vartheta = \frac{1}{2} \dot{\eta}^\top J \dot{\eta}, \quad (20)$$

since the torques of the rotors act as the external angular force, then the angular Euler-Lagrange formula is written as follows, with $C(\eta, \dot{\eta})$ defined as Coriolis term, consisting of gyroscopic and centripetal equations.

$$\tau = \tau_B = j\ddot{\eta} + \left(\frac{d}{dt} J \right) \dot{\eta} - \frac{1}{2} \frac{\partial}{\partial \eta} \dot{\eta}^\top J \dot{\eta}, \quad (21)$$

which equals to $J\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta}$ and from (21), the angular accelerations are equivalent to the equation written in (15),

$$\ddot{\eta} = J^{-1} [\tau_B - C(\eta, \dot{\eta})\dot{\eta}]. \quad (22)$$

To capture more realistic issues of this UAS, the drag force from the air resistance should be considered, such that

$$m\dot{V}_B + \vartheta \times (mV_B) = R^\top \varrho + T_B - R^\top T_D, \quad (23)$$

$$m\ddot{\zeta} = \varrho + RT_B - T_D, \quad (24)$$

where T_D makes up the drag coefficients for velocities which are associated to certain directions (x, y, z) ,

$$T_D = A\dot{\zeta} = \begin{bmatrix} A_x & 0 & 0 \\ 0 & A_y & 0 \\ 0 & 0 & A_z \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}. \quad (25)$$

Finally, the correct angular velocities of rotors ω_i can be calculated from equations (7) and (8)

$$\begin{aligned} \omega_1^2 &= \frac{T}{4k} - \frac{\tau_\theta}{2kL} - \frac{\tau_\psi}{4b}, & \omega_2^2 &= \frac{T}{4k} - \frac{\tau_\phi}{2kL} + \frac{\tau_\psi}{4b}, \\ \omega_3^2 &= \frac{T}{4k} + \frac{\tau_\theta}{2kL} - \frac{\tau_\psi}{4b}, & \omega_4^2 &= \frac{T}{4k} + \frac{\tau_\phi}{2kL} + \frac{\tau_\psi}{4b}. \end{aligned} \quad (26)$$

In the estimation section, we consider linearizing this non-linear system and augmenting the state-space to include the faulty states and the disturbance to be used in the estimation and fault-tolerant learning-oriented control.

C. Linearized System and Control Inputs

There are six primary states $[x, y, z, \phi, \theta, \psi]$ of the twelve states to control and measure. The inner loop of $[z, \phi, \theta, \psi]$ deals with the attitude variables while $[x, y]$ are for the outer position variables. For the sake of attitude control, we consider a linear sub-model of $x = [z, \phi, \theta, \psi, w, p, q, r] \in \mathbb{R}^n$ with measurement of the first four states $y = [z, \phi, \theta, \psi]$. The discretized linearized system is done with equilibrium point $x^* = [z, 0 \in \mathbb{R}^{n-1}]$ to the nonlinear system $\dot{x}(t) = f(x, t)x + g(x, t)u$ and transform into discrete systems. For the control scenario, we apply a hybrid control $u_c(k)$ of PID u_p by decoupling the attitude dynamics and providing independent control action for every state and the reinforcement learning $u_r := \mu_\theta(x)$, where $u_c = [u_p, u_r]$, to the MIMO sub-model system defined later,

$$u = \begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} \frac{m}{\mathbf{c}_\phi \mathbf{c}_\theta} & 0 & 0 & 0 \\ 0 & I_{xx} & 0 & 0 \\ 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} u_p \\ u_p \\ u_p \\ u_r \end{bmatrix}. \quad (27)$$

III. FAULT-TOLERANT CONTROL

Given that the discrete-time model of the system is used for the implementation of the controller in the UAV experimental set-up, we formulate the linearized systems in the discrete-time mode. The state-space representation of a system affected by actuator and sensor fault is written as

$$x_{k+1} = Ax_k + Bu_k + F_a f_a, \quad y_k = Cx_k + F_s f_s, \quad (28)$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$ represent the states, control input and the output in turn. The $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$,

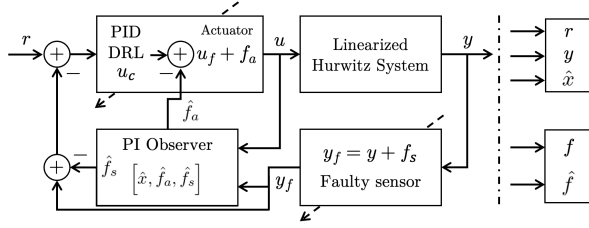


Fig. 2: Fault-tolerant control scheme with faults $f = [f_a, f_s]^\top$ using PI Observer.

$C \in \mathbb{R}^{p \times n}$ are the constant real matrices while $F_a \subseteq B$ and $F_s := I_p$ is assumed to be known. The faulty signals f_a and f_s denote the magnitude and are modeled as $s_f = \alpha s + \beta$ where α and β represent the multiplicative and additive of the fault in turn. The real fault $f_k := [f_a, f_s]^\top$ equals to the difference between the faulty signal s_f and the nominal signal s where $f_k = s_f - s$. The output y experiences a sensor fault where $y_f = y + f_s$ and when a control fault occurs, the control signal is written as $u = u_f + f_a$ as shown in Fig. 2. Let us introduce an augmented state z_k to bring the sensor fault f_s into single faulty vector f_k ,

$$z_{k+1} := M[y_k - z_k] = M[Cx_k + F_s f_s - z_k], \quad (29)$$

therefore the extended state-space of (28) and (29), defining $\bar{x}_k = [x_k, z_k]^\top$ and f_k , results in

$$\begin{aligned} \bar{x}_{k+1} &= A_a \bar{x} + B_a u_k + E_a f_k \\ &= \begin{bmatrix} A & 0_1 \\ MC & -M \end{bmatrix} \bar{x} + \begin{bmatrix} B \\ 0_2 \end{bmatrix} u_k + \begin{bmatrix} F_a & 0_4 \\ 0_3 & MF_s \end{bmatrix} f_k, \\ y_k &= C_a \bar{x} = [0_{p \times n} \quad I_p] \bar{x}_k, \end{aligned} \quad (30)$$

where $0_1 \in \mathbb{R}^{n \times p}$, $0_2 \in \mathbb{R}^{p \times m}$, $0_3 \in \mathbb{R}^{p \times r_1}$, and $0_4 \in \mathbb{R}^{n \times r_2}$ represents the zeros matrices. The dimension of $r_1 + r_2 = r \leq p$. Keep in mind the stable matrix M should be chosen properly. Here, we insert the fault $f_{k+1} = f_k$ into the state $\bar{x}_k$ in order to easily estimate the fault, where $\tilde{x}_k = [\bar{x}_k, f_k]^\top$,

$$\begin{aligned} \tilde{x}_{k+1} &= \tilde{A} \tilde{x}_k + \tilde{B} u_k = \begin{bmatrix} A_a & E_a \\ 0_{r \times v} & I_r \end{bmatrix} \tilde{x}_k + \begin{bmatrix} B_a \\ 0_{r \times m} \end{bmatrix} u_k, \\ y_k &= \tilde{C} \tilde{x}_k = [C_a \quad 0_{p \times r}] \tilde{x}_k, \end{aligned} \quad (31)$$

where $v = n + p$ and (31) is the equation used for the state estimation methods, especially $\tilde{f}_k$, to compensate the faulty state in the fault-tolerant control mechanism. However, (28)-(30) is adjustable yet the rank of C_a is greater or equal to that of E_a , where $\rho(C_a) \geq \rho(E_a)$. In this paper, two models of faults are applied to the system, with m actuator faults $f = f_a \in \mathbb{R}^m$ and combined actuator and sensor faults such that $f = [f_a, f_s] \in \mathbb{R}^m$ with PI observer.

IV. PI OBSERVER STATE ESTIMATION

This section is dedicated to the application of proportional-integral (PI) observer as an alternative method for estimating faults. The design of PI observer for continuous-time systems has been established [14]–[16]. First, we construct the system for m actuator faults, which are modeled as follows:

$$x_{k+1} = Ax_k + Bu_k + Ef_k, \quad y = Cx_k. \quad (32)$$

We ensure a sufficiently small sampling interval T such that the fault does not vary significantly between two consecutive sampling instances. In other words, the magnitude of $f_{k+1} - f_k$ is of the order $O(T^2)$, $\forall k$. Then there exists a discrete-time PI observer of the form

$$\begin{aligned} \hat{x}_{k+1} &= (A - L_P C) \hat{x}_k + L_P y_k + Bu_k + E \hat{f}_k, \\ \hat{f}_{k+1} &= \hat{f}_k + L_I (y_k - C \hat{x}_k), \end{aligned} \quad (33)$$

such that the state and fault estimation errors are constrained in the small region of $O(T^2)$ if and only if the pair (A, C) is observable and

$$\text{rank} \begin{bmatrix} A - I & E \\ C & 0 \end{bmatrix} = n + r,$$

where L_P and L_I are proportional and integral gains of PI observer, $\text{rank } C = p$ and $\text{rank } E = r$ with $p \geq r$. Defining $e_x(k) = x_k - \hat{x}_k$, $e_f(k) = f_k - \hat{f}_k$, and $\epsilon_k = f_{k+1} - f_k$, it is not difficult to derive the error dynamic as $e_{k+1} = (\bar{A} - \bar{L} \bar{C}) e_k + \bar{I} \epsilon_k$ where $e_k = [e_x(k), e_f(k)]^\top$, and

$$\bar{A} = \begin{bmatrix} A & E \\ 0 & I_r \end{bmatrix}, \bar{C} = [C \quad 0], \bar{L} = \begin{bmatrix} L_P \\ L_I \end{bmatrix}, \bar{I} = \begin{bmatrix} 0_{n \times r} \\ I_r \end{bmatrix},$$

using the rank condition, one can conclude that the pair $(\bar{A}, \bar{C})$ is observable. Therefore, a gain $\bar{L}$ can be found such that $\bar{A} - \bar{L} \bar{C}$ is stable. If $\epsilon_k = 0$ then the convergence of the observer is guaranteed i.e. $e_x(k) \rightarrow 0$ and $e_f(k) \rightarrow 0$ as $k \rightarrow \infty$. However, based on the assumption $\epsilon_k \in O(T^2)$, the estimation error can be constrained in the small region of $O(T^2)$ which also guarantee the reliable estimation of fault.

Defining the combined faults, the augmented states used is (30) instead of (32) where $p \geq r$ and the observer gain $\bar{L}$ is computed such that the poles of (31) is inside the unite circle. Then the n terms of $\bar{L}$ which is L_P is to stabilize the states $\hat{x}$ whereas the r terms, L_I , belongs to the fault states $\hat{f}$ such that (33) is stabilizable.

V. LEARNING-ORIENTED CONTROL

We consider deep-deterministic policy gradient (DDPG) as outlined in [23]. This method uses the deterministic direct mapping states to action $u = \mu_\theta(x)$ instead of the probabilistic ones $\pi_\theta(u|x) = \mathbb{P}[u|x; \theta]$. The two terms x and u represent the spaces of the states $x \subseteq S_X$ and the control actions $u \subseteq S_A$. The policy μ_θ can be optimized by tuning the policy parameter θ towards the direction of the gradient of the expected rewards with $J(\mu_\theta) = \mathbb{E}[r_t^\gamma | \mu_\theta]$ as follows,

$$J(\mu_\theta) = \int_{S_X} \rho^\mu(x) \int_{S_A} \mu_\theta(x, u) r(x, u) du dx. \quad (34)$$

By contrast, the policy gradient is updated via θ applying gradient ascent such that $\theta \leftarrow \theta + \alpha_\ell \nabla_\theta J(\mu_\theta)$ where the term α_ℓ denotes the learning rate. Keep in mind, solving $\nabla_\theta J(\mu_\theta)$ where the state and action distribution depending on it is computationally expensive. To solve, the expectation of the control u and the logarithmic of μ_θ is proposed as

$$\nabla_\theta J(\mu_\theta) = \mathbb{E}_{x,u} [\nabla_\theta \log \mu_\theta(u|x) Q^\mu(x, u)], \quad (35)$$

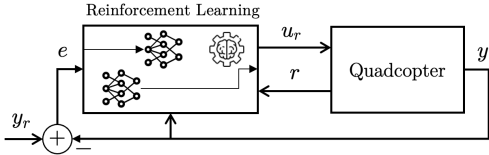


Fig. 3: Scheme of training for DRL on single state $y_4 = \psi$.

where the $Q^\mu(x, u)$ can be estimated through the sample return (batch) $r_{t:T}^\gamma$ defined as the sum of the discounted future rewards from time instant t onwards,

$$r_t^\gamma = r_1 + \gamma r_2 + \gamma^2 r_3 + \dots = \sum_{k=t}^{\infty} \gamma^{k-t} r(x_k, u_k), \quad (36)$$

with the discount factor $\gamma \in (0, 1)$. If the policy μ_θ and the associated cost function $J(\mu_\theta)$ are taken such that it maximizes the function, $\forall x \in \mathbb{R}, \forall t \in [0, T]$, then $J(\mu_\theta) = J(\mu_\theta^*)$ and $\mu_\theta = \mu_\theta^*$. Furthermore, the updated of the current rule according to the batch $r_{t:T}^\gamma$ is given by,

$$\theta \leftarrow \theta + \alpha_\ell \log \mu_\theta(u|x) r_{t:T}^\gamma. \quad (37)$$

However, due to unbiased gradient estimate and the stochasticity of Monte-carlo $r_{t:T}^\gamma$, the policy gradient is less sensitive to high variance of gradient estimates. Therefore, the critic emerges by approximating the term $Q^\mu(x, u)$ in (35) with a parameter ϑ such that $Q^\vartheta(x, u)$, written as

$$\nabla_\theta J(\mu_\theta) = \mathbb{E}_{x,u} [\nabla_\theta \log \mu_\theta(u|x) Q^\vartheta(x, u)], \quad (38)$$

yielding the bias as a consequence of the variance reduction. This DDPG is closely related to the DQN in which the value function of the state-action $Q^\mu(x, u)$ is expressed as the expected return given by (x, u, μ_θ) and the discounted state distribution $\rho^\mu(x')$. Since the action is continuous, the value function $Q^*(x, u)$ is assumed as differentiable over action argument and it leads to construct the gradient-based rule for $\mu_\theta(x)$. DDPG is constructed with two neural networks of actor μ_θ for delivering a deterministic policy and critic Q^ϑ for estimating action-value function. This Q^ϑ is used to update the actor $\mu_\theta(x)$ in the direction of the gradient of $Q^\vartheta(x, \mu_\theta(x) := u)$,

$$\nabla_\theta J(\mu_\theta) = \mathbb{E}_x [\nabla_\theta \mu_\theta(x) \nabla_u Q^\vartheta(x, u)|_{u=\mu_\theta(x)}], \quad (39)$$

where $\nabla_\theta \mu_\theta(x)$ and $\nabla_u Q^\vartheta(x, u)$ comprise the gradient of μ_θ and Q^ϑ subject to θ and action u in turn. Finally the parameter θ is stochastically updated via the gradient ascent,

$$\theta \leftarrow \theta + \alpha_\ell \nabla_\theta \mu_\theta(x)|_{x=x_t} \nabla_u Q^\vartheta(x, u)|_{x=x_t, u=\mu_\theta(x)}. \quad (40)$$

The learning control for this UAV is conducted for $y_4 = \psi$ and the agent is also applied beyond the environment for y_2, y_3 with the estimation scheme and the designed faults. The observed state of DDPG is $x_r = [y_4, e, e_i]$ where e_i is the integral of e and the $r_t = 10, \forall e < \epsilon$ and 1 otherwise.

VI. NUMERICAL EXAMPLES AND SIMULATIONS

This section presents the simulation of the fault-tolerant control using a hybrid control approach that combines PID

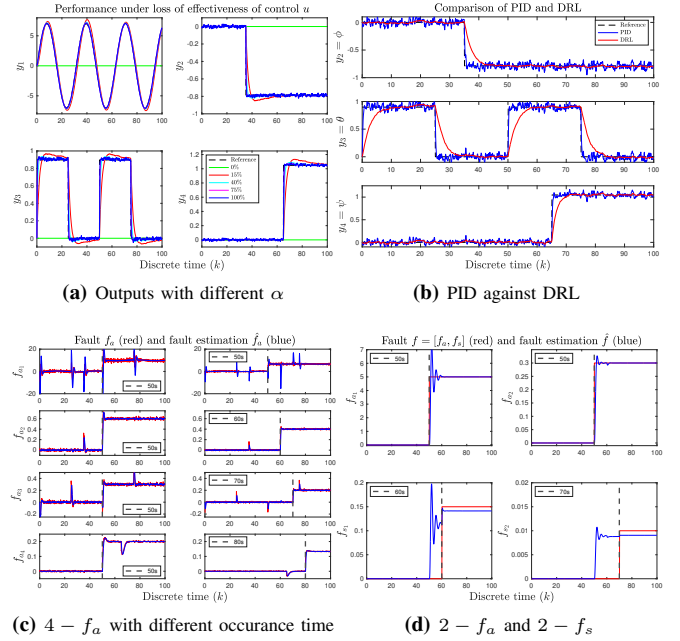


Fig. 4: (a) Performance under different loss of effectiveness α in actuator; (b) the comparison of PID and DRL for only training space on y_4 ; (c-d) faults estimation

and DRL. The key parameters are the mass $m = 0.468$, the drag force coefficients $A_x = 0.3; A_y = 0.3; A_z = 0.25$, the rotational drag coefficient $A_r = 0.2$; and the inertia $I_{xx} = 4.856 \times 10^{-3}; I_{yy} = I_{xx}; I_{zz} = 8.801 \times 10^{-3}$. The linearized sub-model is written as

$$A = \begin{bmatrix} 0_{4 \times 4} & I_4 \\ 0_{4 \times 4} & \text{diag}(\Psi) \times \Psi_A \end{bmatrix}, \quad B = \begin{bmatrix} 0_{4 \times 4} \\ \text{diag}(\Psi) \end{bmatrix},$$

with $\Psi = [m^{-1}, I_{xx}^{-1}, I_{yy}^{-1}, I_{zz}^{-1}]$, $\Psi_A = [A_z, A_r, A_r, A_r]^\top$. The control is designed to combine the PID and DRL while the only state to train for DRL is the last measurement y_4 . This training is applied to control beyond its environment for other two states, y_2, y_3 . At first, the impact of loss of effectiveness α in actuator faults is presented in Fig. 4a while the comparison of PID and DRL for the three states y_2, y_3, y_4 is portrayed in Fig. 4b. Interestingly, DRL outperforms PID in reducing the noise although the training is conducted once for 100s in 51 episodes with random set-points y_r per episode. The reward $r_k = 10, \forall e < \epsilon := 0.01$ and 1 otherwise while the upper S_{X_u} and lower S_{X_l} bound of the space $x_r = [y_4, e, e_i] \subset S_X$ are $S_{X_u} = [\pi, \infty, \infty]$ and $S_{X_l} = [-\pi, -\infty, 0]$ in turn. The critic and actor learning rate are $\alpha_c = 10^{-3}$ and $\alpha_a = 10^{-4}$ with discount factor $\gamma = 0.9$ and the 64 batch size along with 1000 maximum episodes and steps. The four faults are designed in three types:

- 1) the $4 - f_a$ in 50s with loss $\alpha = 0.5$ and bias $\beta = [2, 0.3, 0.15, 0.1]$;
- 2) the $4 - f_a$ in different time [50s, 60s, 70s, 80s] with the same bias β but $\alpha = 0.75$;
- 3) the $2 - f_a$ in 50s, $2 - f_s$ in [60s, 70s] with $\alpha = 1$ and bias $\beta = [5, 0.3, 0.15, 0.01]$ where $f = [f_a, f_a, f_s, f_s]^\top$

The estimations of faults are shown in Fig. 4c-4d showing

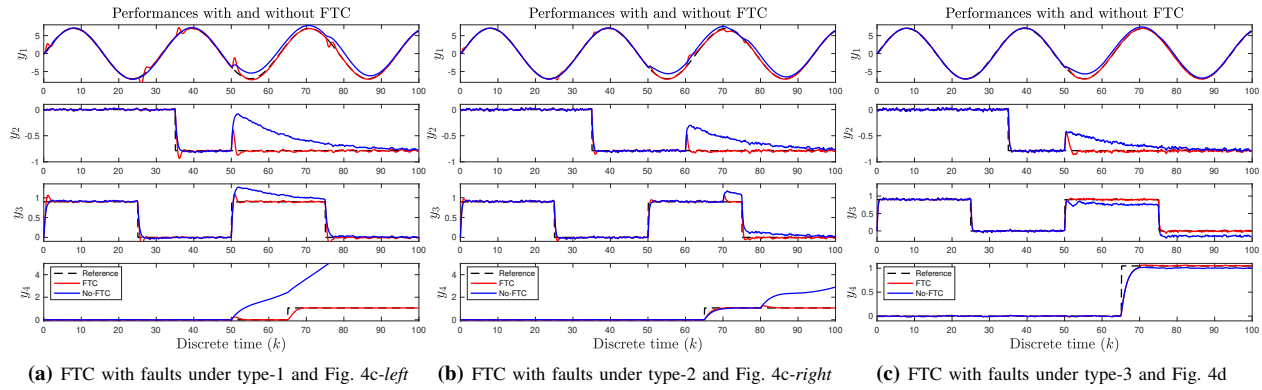


Fig. 5: These figures present the performance comparisons of with FTC and without FTC under three designed faults.

how the PI observer captures the faults to compensate the faulty output y_f and input u_f . For FTC implementation, the DRL is used to control y_4 and based on the three types of faults along with the fault estimations in Fig. 4c-4d, the effectiveness of FTC is shown in Fig. 5. This DRL could work together with the PI observer and the FTC as a whole without any problem. Further experiments using the Quanser QDrone 2 autonomous air vehicle for fault detection in UAVs will be conducted. The QDrone 2 is an open-architecture research-grade drone that features a robust on-board NVIDIA Jetson Xavier NX system-on-module, multiple high-resolution cameras, and built-in WiFi, making it suitable for advanced research applications.

VII. CONCLUSIONS

This paper has presented novel strategies to enhance the resilience of UAVs by proposing a constructive mathematical dynamical system for these vehicles. The paper outlined a fault estimation mechanism utilizing a PI observer, which compensates for both sensor and actuator faults within three different faulty setups. The captured faults are subsequently used for FTC. The proposed approach has the potential to improve the performance and resilience of UAVs, thereby contributing to the development of more robust and reliable systems. These systems are capable of operating safely and efficiently under challenging conditions.

REFERENCES

- [1] H. Shakhatareh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, "Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48572–48634, 2019.
- [2] N. S. Labib, M. R. Brust, G. Danoy, and P. Bouvry, "The rise of drones in internet of things: A survey on the evolution, prospects and challenges of unmanned aerial vehicles," *IEEE Access*, vol. 9, pp. 115466–115487, 2021.
- [3] W. Chung and H. Son, "Fault-tolerant control of multirotor UAVs by control variable elimination," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 5, pp. 2513–2522, 2020.
- [4] R. C. Nelson, *Flight stability and automatic control 2nd ed.* Boston, Mass.: WCB/McGraw Hill, 1998.
- [5] T. R. Yechout, S. L. Morris, D. E. Bossert, W. F. Hallgren, and J. K. Hall, *Introduction to Aircraft Flight Mechanics: Performance, Static Stability, Dynamic Stability, and Classical Feedback Control, Second Edition*. 1801 Alexander Bell Drive, Reston, VA 20191-4344: American Institute of Aeronautics and Astronautics (AIAA), Inc, 2014.
- [6] C. Hajiyeve, H. E. Soken, and S. Y. Vural, *State Estimation and Control for Low-cost Unmanned Aerial Vehicles*. 1801 Alexander Bell Drive, Reston, VA 20191-4344: Springer Chamc, 2015.
- [7] Z. Zuo, C. Liu, Q.-L. Han, and J. Song, "Unmanned aerial vehicles: Control methods and future challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 4, pp. 601–614, 2022.
- [8] H. Huang and A. V. Savkin, "Navigating UAVs for optimal monitoring of groups of moving pedestrians or vehicles," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3891–3896, 2021.
- [9] P. S. Bithas, V. Nikolaidis, A. G. Kanatas, and G. K. Karagiannidis, "UAV-to-ground communications: Channel modeling and UAV selection," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 5135–5144, 2020.
- [10] M. K. Wafi, "Filtering module on satellite tracking," *AIP Conference Proceedings*, vol. 2088, no. 1, p. 020045, 2019.
- [11] O. S. Oubbati, A. Lakas, and M. Guizani, "Multiagent deep reinforcement learning for wireless-powered uav networks," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 16044–16059, 2022.
- [12] D. Wang, J. Huang, M. He, and C. Huang, "Spectrum transaction games for UAV assisted communications," *IEEE Wireless Communications Letters*, vol. 11, no. 6, pp. 1216–1219, 2022.
- [13] G. Raja, S. Anbalagan, A. Ganapathisubramaniyan, M. S. Selvakumar, A. K. Bashir, and S. Mumtaz, "Efficient and secured swarm pattern multi-UAV communication," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 7, pp. 7050–7058, 2021.
- [14] H. H. Niemann, J. Stoustrup, B. Shafai, and S. Beale, "Ltr design of proportional-integral observers," *International Journal of Robust and Nonlinear Control*, vol. 5, no. 7, pp. 671–693, 1995.
- [15] B. Shafai, C. Pi, and S. Nork, "Robust fault detection using proportional integral observers," in *Proceedings of the 5th Biannual World Automation Congress*, vol. 14, pp. 515–520, 2002.
- [16] B. Shafai and A. Moradmand, "Design of an integrated observer structure for robust fault detection," in *2020 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 248–253, 2020.
- [17] M. K. Wafi and K. Indriawati, "Fault-tolerant control design in scrubber plant with fault on sensor sensitivity," *The Journal of Scientific and Engineering Research*, vol. 9, no. 2, pp. 96–104, 2022.
- [18] K. Indriawati, T. Agustinah, and A. Jazidie, "Robust observer-based fault tolerant tracking control for linear systems with simultaneous actuator and sensor faults: Application to a dc motor system," *International Review on Modelling and Simulations (IREMOS)*, vol. 8, no. 4, 2015.
- [19] M. B. Vankadari, K. Das, C. Shinde, and S. Kumar, "A reinforcement learning approach for autonomous control and landing of a quadrotor," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 676–683, 2018.
- [20] D. Zhang and Z. Gao, "Fault tolerant control using reinforcement learning and particle swarm optimization," *IEEE Access*, vol. 8, pp. 168802–168811, 2020.
- [21] C. Hua, L. Li, and S. X. Ding, "Reinforcement learning-aided performance-driven fault-tolerant control of feedback control systems," *IEEE Transactions on Automatic Control*, vol. 67, no. 6, pp. 3013–3020, 2022.
- [22] H. Deng, Y. Zhao, A.-T. Nguyen, and C. Huang, "Fault-tolerant predictive control with deep-reinforcement-learning-based torque distribution for four in-wheel motor drive electric vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 2, pp. 668–680, 2023.
- [23] H. Lee, M. Girnyk, and J. Jeong, "Deep reinforcement learning approach to mimo precoding problem: Optimality and robustness," 2020.