

Adaptive Control of the Quanser 3DOF Hover

Yves Georgy Daoud*, Ramzi Haddad†, Elias Khalife‡
Email: *ydd00@mail.aub.edu, †,rmh66@mail.aub.edu ‡ejk06@mail.aub.edu

Abstract—The purpose of this paper is to implement different adaptive controllers on the Quanser 3DOF Hover platform. The system was modelled using Euler equations. Based on the model, the controllers designed include a simple proportional controller, an indirect self tuning regulator, and Model Reference Adaptive Controller (MRAC) with full state feedback, output feedback, and disturbance rejection. The controllers were tested using MATLAB’s Simulink on the nonlinear high-fidelity of the quadcopter, and their results analyzed in this paper.

I. INTRODUCTION

Recent years have seen a huge surge in the focus and demand for unmanned aerial vehicles (UAVs) in multiple sectors whether industrial or academic. Industrially, individuals, companies and NGOs are using the UAVs as products to an end customer in various tasks and applications such as precision agriculture [1], landmines detection [2], search and rescue missions [3], archaeological surveying [4], and airborne delivery [5]. These applications and many more surfacing by the day, are increasing the popularity of drones, which is driving the industry forwards, requiring both cheaper hardware to fulfill the demand of hobbyists, as well as more precise maneuvers and control of the UAVs for professional applications. Therefore, multiple researchers are attempting to design new controllers that deal with parameter uncertainties in the model, external disturbances when operating in real life, and changing conditions when a mass is introduced or dropped for example. A brief overview of some of the recent advances in control strategies for quadcopters is provided below, with an adaptive controller implemented and tested in simulation in this paper.

II. LITERATURE REVIEW

Aside from the typical controllers that do not take the energy needed to perform a control action into consideration, optimal controllers place a weight on both the input energy and the output performance of a controller. One of type of this subset is the Linear Quadratic Regulator (LQR) designed by Tahir et al. in which changing the weights of the Q and R weighing matrices, the LQR controller optimized the energy consumption as well as the output tracking of the reference signal. [6]

Al-Mathuri et al. on the other hand, overcame a major limitation of linear controllers by using an interval type-2 fuzzy controller to handle uncertainties in nonlinear systems. By comparison to a proportional derivative (PD) controller, the tracking error of the fuzzy controller was smaller than the PD, outperforming it in a variety of tracking tests under nonlinearities and uncertainties. [7]

H. Merabti et al. controlled a quadcopter to track a reference trajectory with and without external perturbation using a nonlinear model predictive controller, NMPC. The computations were carried using convex optimization techniques, and two tests were performed: the first is a simple tracking and the second is a tracking in presence of perturbation. The simulation results reflect a good controller performance, the quadcopter was able to track the command trajectories. [8]

In his work, Hao Liu et al explained the necessity of robust controllers when a quadcopter is subject to parametric perturbations, nonlinear and coupled dynamics, external disturbances, and state and input delays. The proposed controller consists of an attitude controller and a position controller and it is designed based on the cascade control approach and the robust compensating method. Experimental results on the quadcopter demonstrated the effectiveness of the designed robust controller. [7]

Thu et al. designed a filtered version of a MRAC (Model Reference Adaptive Controller), called L1 adaptive controller. The controller contains a low pass filter that serves a filter on the input to the plant, u , meaning that there will be no ringing or excitation in that signal. This also implies that the controller is effective when the adaptation rate is high. However, the drawback is that the computational power needed at these high adaptation rates. The performance of this controller in an output feedback environment on a quadcopter is detailed, obtaining better attitude and trajectory tracking.[9]

Kim et al. proposed a nonlinear controller that is able to perform tracking without knowing the true value of the parameters. This is done using an adaptation algorithm that adapts the closed loop system’s cutoff frequency and produces the gains of the controller. In addition, feed forward compensation is utilized to reject disturbances, after they have been estimated by a disturbance observer (DOB). The controller feedbacks the proportional part of the tracking errors to make the closed loop system respond faster. [10]

Kalat et al. Designed a new practical self tuning regulator (STR) which is able to attenuate disturbances without knowing what the disturbance is and without changing the controller parameters. This method includes using the output tracking error in identification rather than the output. Also, the proposed method makes the system less sensitive to non linearities. [11]

III. DYNAMIC MODEL

The model is based on the Euler equation, shown in Equation 1, which describes the rotational motion of the quadcopter

for all 3 degrees of freedom.

$$I \frac{d\omega}{dt} + \omega \times I\omega = \tau \quad (1)$$

Using the free body diagrams in Figures 1, 2, 3, the moments around the pivot, for the pitch, roll, and yaw angles, are found in terms of the upward forces, F , and the propeller torques, τ . [12] [13]

$$I \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times I \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} l(F_r - F_l) \\ l(F_f - F_b) \\ \tau_l + \tau_r - \tau_f - \tau_b \end{bmatrix} \quad (2)$$

Where

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}$$

The forces and torques produced by each propeller are modelled as having a linear relationship with the motor's angular velocity.

$$F = b\Omega^2 \quad (3)$$

$$\tau = d\Omega^2 \quad (4)$$

For decoupling and controller design purposes, the inputs to the system are considered to be for each degree of freedom. Meaning that $\tau^T = [\tau_x \ \tau_y \ \tau_z]$

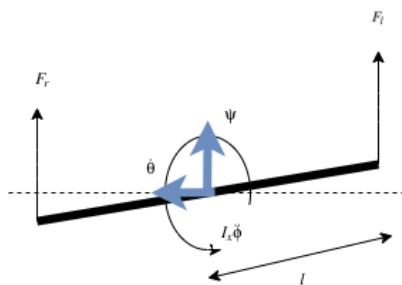


Fig. 1. Free Body Diagram of Quadcopter for Pitch Axis

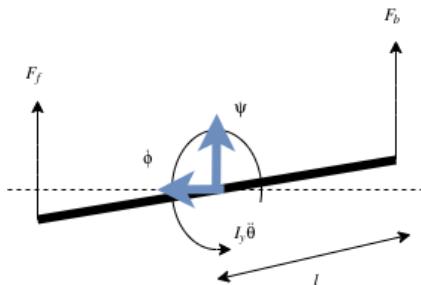


Fig. 2. Free Body Diagram of Quadcopter for Roll Axis

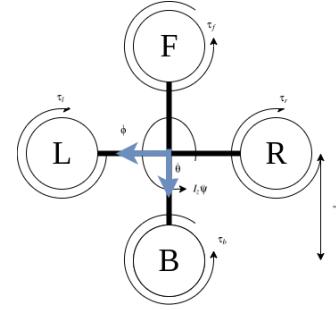


Fig. 3. Free Body Diagram of Quadcopter for Yaw Axis

Solving the Euler equation in Equation 2, three equations are obtained, shown in Equation 5

$$\begin{cases} I_x \ddot{\phi} = (I_y - I_z) \dot{\theta} \dot{\psi} + \tau_x \\ I_y \ddot{\theta} = (I_z - I_x) \dot{\phi} \dot{\psi} + \tau_y \\ I_z \ddot{\psi} = (I_x - I_y) \dot{\phi} \dot{\theta} + \tau_z \end{cases} \quad (5)$$

Therefore, simplifying Equation 5 and adding a wind disturbance, the pitch, roll, and yaw angular accelerations are found, shown in Equation 6

$$\begin{cases} \ddot{\phi} = \frac{I_y - I_z}{I_x} \dot{\theta} \dot{\psi} + \frac{\tau_x}{I_x} + \frac{\tau_{wx}}{I_x} \\ \ddot{\theta} = \frac{I_z - I_x}{I_y} \dot{\phi} \dot{\psi} + \frac{\tau_y}{I_y} + \frac{\tau_{wy}}{I_y} \\ \ddot{\psi} = \frac{I_x - I_y}{I_z} \dot{\phi} \dot{\theta} + \frac{\tau_z}{I_z} + \frac{\tau_{wz}}{I_z} \end{cases} \quad (6)$$

In order to represent the system in state space, the system must be linear. Therefore, a small angle approximation is made and the system is linearized around the equilibrium point corresponding to the hovering condition, implying that $\phi = \dot{\phi} = \theta = \dot{\theta} = \psi = \dot{\psi} = 0$.

The states are chosen to be the pitch, roll, and yaw angles and their respective derivatives:

$$x^T = [\phi, \ \dot{\phi}, \ \theta, \ \dot{\theta}, \ \psi, \ \dot{\psi}] \quad (7)$$

While the inputs are the moments for each degree of freedom:

$$u^T = [\tau_x, \ \tau_y, \ \tau_z] \quad (8)$$

The disturbances are the wind effects on the motion of the system:

$$d^T = [\tau_{wx}, \ \tau_{wy}, \ \tau_{wz}] \quad (9)$$

After defining the inputs, disturbances, and states, the system can be written in state space form

$$\dot{x} = Ax + Bu + Dd \quad (10)$$

Where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_z} \end{bmatrix},$$

$$D = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_z} \end{bmatrix}$$

IV. CONTROLLER DESIGN

A. Proportional Controller

In order to design a proportional controller, the system is decoupled and divided into 3 separate subsystems: pitch, roll, and yaw. A proportional controller is designed for each subsystem and then concatenated into one gain representing the controller gain of the entire system.[?]

The closed loop characteristics of the closed loop system are picked to be:

- Percentage Overshoot (PO) less than or equal to 20 %.
- Settling time of 1 second.

Where Equation 11 shows the relationship between these 2 characteristics and the damping ratio and natural frequency.

$$\begin{cases} PO = e^{\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \\ T_s = \frac{4}{\zeta\omega_n} \end{cases} \quad (11)$$

Therefore, the closed loop poles are stable and formulated in Equation 12.

$$SP = -\zeta\omega_n \pm j(\omega_n\sqrt{1-\zeta^2}) = -4 \pm j7.8079 \quad (12)$$

After setting the requirements, the proportional gain for the pitch, roll, and yaw subsystems can be calculated using MATLAB's Place command. The results are shown in Equations 14, 16, and 18 respectively.

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{I_x} & 0 & 0 \end{bmatrix} u \quad (13)$$

$$K_\phi = \begin{bmatrix} 2.527 & 0.442 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (14)$$

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{I_y} & 0 \end{bmatrix} u \quad (15)$$

$$K_\theta = \begin{bmatrix} 0 & 0 \\ 2.527 & 0.442 \\ 0 & 0 \end{bmatrix} \quad (16)$$

$$\begin{bmatrix} \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_z} \end{bmatrix} u \quad (17)$$

$$K_\psi = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 5.036 & 0.88 \end{bmatrix} \quad (18)$$

Concatenating the 3 proportional controllers, Equation 19 is obtained.

$$K = \begin{bmatrix} 2.527 & 0.442 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.527 & 0.442 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5.036 & 0.88 \end{bmatrix} \quad (19)$$

B. Indirect Self Tuning Regulator

Given the linearized model of the quadcopter, the controller can be designed for each of the 3 decoupled subsystems independently: pitch, roll, and yaw.

Starting with the pitch, equation (20) shows the pulse transfer function operator for a sampling period h.

$$H_\phi(q) = \frac{\frac{1}{2I_x}h^2q + \frac{1}{2I_x}h^2}{q^2 - 2q + 1} \quad (20)$$

Let $h = 0.5$, the pulse transfer function becomes the following:

$$H_\phi(q) = \frac{B_\phi(q)}{A_\phi(q)} = \frac{b_{0\phi}q + b_{1\phi}}{q^2 + a_1q + a_2} \quad (21)$$

where: $b_{0\phi} = b_{1\phi} = \frac{0.125}{I_x}$, $a_1 = -2$, and $a_2 = 1$

Let $G_\phi(q)$ be the desired CL system. The expression of $G_\phi(q)$ is presented in Equation (22).

$$G_\phi(q) = \frac{B_m(q)}{A_m(q)} = \frac{b_{m0}q}{q^2 + a_{m1}q + a_{m2}} \quad (22)$$

If the natural frequency and the relative damping are respectively $\omega = 1$ and $\zeta = 0.7$, and b_{m0} is chosen to yield unity static gain, the desired CL system becomes:

$$G_\phi(q) = \frac{B_m(q)}{A_m(q)} = \frac{0.1761q}{q^2 - 1.3205q + 0.4966} \quad (23)$$

The model satisfies compatibility conditions since it has the same pole excess as the process ($d_0 = 1$).

The control law is presented in Equation (24):

$$R_\phi \tau_x(t) = T_\phi u_c(t) - S_\phi \phi(t) \quad (24)$$

where R_ϕ, S_ϕ, T_ϕ are polynomials.

Applying the minimum-degree pole placement (MDPP) algorithm with zeros cancellation while satisfying the causality conditions yields the following results:

- 1) $B_\phi = B_\phi^+ B_\phi^-$, where $B_\phi^- = b_{0\phi}$ and $B_\phi^+ = B_\phi/b_{0\phi}$
- 2) $\deg(A_{0\phi}) = \deg(A_\phi) - \deg(B_\phi) - 1 = 0$
- 3) Let $A_{0\phi} = 1$
- 4) $B'_{m\phi} = \frac{A_m(1)q^{n-d_0}}{b_{0\phi}} = \frac{b_{m0}}{b_{0\phi}}q$
- 5) $T_\phi = A_{0\phi} B'_{m\phi} = \frac{b_{m0}}{b_{0\phi}}q$
- 6) $\deg(T_\phi) = \deg(R_\phi) = \deg(S_\phi) = 1$
- 7) Let $S_\phi = s_{0\phi}q + s_{1\phi}$, and $R'_\phi = 1$
(since R'_ϕ is monic and of degree zero)
- 8) $R_\phi = R'_\phi B^+ = q + \frac{b_{1\phi}}{b_{0\phi}} = q + 1$

The Diophantine equation states that:

$$A_\phi R_\phi + B_\phi S_\phi = A_{0\phi} A_m \quad (25)$$

Applying Equation (25) on the polynomials obtained above determines the expression of S_ϕ :

$$S_\phi = \frac{a_{m1} - a_1}{b_{0\phi}} q + \frac{a_{m2} - a_2}{b_{0\phi}} \quad (26)$$

In conclusion, the STR design of the pitch is:

$$\left\{ \begin{array}{l} H_\phi(q) = \frac{B_\phi(q)}{A_\phi(q)} = \frac{b_{0\phi}q + b_{1\phi}}{q^2 + a_1q + a_2} = \frac{\frac{0.125}{I_x}q + \frac{0.125}{I_x}}{q^2 - 2q + 1} \\ G_\phi(q) = \frac{B_m(q)}{A_m(q)} = \frac{b_{m0}q}{q^2 + a_{m1}q + a_{m2}} = \frac{0.1761q}{q^2 - 1.3205q + 0.4966} \\ T_\phi = \frac{b_{m0}}{b_{0\phi}}q \\ R_\phi = q + \frac{b_{1\phi}}{b_{0\phi}} \\ S_\phi = \frac{a_{m1} - a_1}{b_{0\phi}}q + \frac{a_{m2} - a_2}{b_{0\phi}} \end{array} \right. \quad (27)$$

The STR designs of the quadrotor's roll and yaw are obtained by following the same procedure used for the pitch.

Accordingly, the STR design of the roll is:

$$\left\{ \begin{array}{l} H_\theta(q) = \frac{B_\theta(q)}{A_\theta(q)} = \frac{b_{0\theta}q + b_{1\theta}}{q^2 + a_1q + a_2} = \frac{\frac{0.125}{I_y}q + \frac{0.125}{I_y}}{q^2 - 2q + 1} \\ G_\theta(q) = \frac{B_m(q)}{A_m(q)} = \frac{b_{m0}q}{q^2 + a_{m1}q + a_{m2}} = \frac{0.1761q}{q^2 - 1.3205q + 0.4966} \\ T_\theta = \frac{b_{m0}}{b_{0\theta}}q \\ R_\theta = q + \frac{b_{1\theta}}{b_{0\theta}} \\ S_\theta = \frac{a_{m1} - a_1}{b_{0\theta}}q + \frac{a_{m2} - a_2}{b_{0\theta}} \end{array} \right. \quad (28)$$

And the STR design of the yaw is:

$$\left\{ \begin{array}{l} H_\psi(q) = \frac{B_\psi(q)}{A_\psi(q)} = \frac{b_{0\psi}q + b_{1\psi}}{q^2 + a_1q + a_2} = \frac{\frac{0.125}{I_z}q + \frac{0.125}{I_z}}{q^2 - 2q + 1} \\ G_\psi(q) = \frac{B_m(q)}{A_m(q)} = \frac{b_{m0}q}{q^2 + a_{m1}q + a_{m2}} = \frac{0.1761q}{q^2 - 1.3205q + 0.4966} \\ T_\psi = \frac{b_{m0}}{b_{0\psi}}q \\ R_\psi = q + \frac{b_{1\psi}}{b_{0\psi}} \\ S_\psi = \frac{a_{m1} - a_1}{b_{0\psi}}q + \frac{a_{m2} - a_2}{b_{0\psi}} \end{array} \right. \quad (29)$$

C. MRAC with Full State Feedback

The design of the MRAC controller will be done for each of the pitch, roll, and yaw decoupled subsystems independently. Starting with the pitch, Equation (30) shows the state space formulation $\dot{x} = Ax + Bu$ of the first degree of freedom.

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{I_x} \end{bmatrix} \tau_x \quad (30)$$

Notation wise, the states will be defined as $x_1 = \phi$ and $x_2 = \dot{\phi}$, and the input $u = \tau_x$, since the same approach can be applied to the roll θ and yaw ψ . Thus the system transforms to Equation (31).

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{I_x} \end{bmatrix} u \quad (31)$$

The model reference that the controller should drive the output to follow is a second order underdamped system of the form $G_m(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$, and more generally $G_m(s) =$

$\frac{b_{m0}}{s^2 + a_{m1}s + a_{m2}}$. In state space description $\dot{x}_m = A_m x_m + B_m u$ the model transforms to Equation (32).

$$\begin{bmatrix} \dot{x}_{m1} \\ \dot{x}_{m2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_{m2} & -a_{m1} \end{bmatrix} \begin{bmatrix} x_{m1} \\ x_{m2} \end{bmatrix} + \begin{bmatrix} 0 \\ b_{m0} \end{bmatrix} u \quad (32)$$

A feedforward and feedback 2DOF control law, Equation (33) will be first devised and tested on the system, where $M = \theta_1$ and $L = (\theta_2 \quad \theta_3)^T$.

$$u(t) = Mu_c(t) - Lx \quad (33)$$

Substituting the definition of $u(t)$ in Equation (31), the new description of \dot{x} is obtained in Equation (34).

$$\begin{aligned} \dot{x} &= (A - BL)x + BMu_c \\ &= A_c(\theta)x + B_c(\theta)u_c \end{aligned} \quad (34)$$

Where

$$\begin{aligned} A_c(\theta) &= \begin{bmatrix} 0 & 1 \\ -\frac{\theta_2}{I_x} & -\frac{\theta_3}{I_x} \end{bmatrix} \\ B_c(\theta) &= \begin{bmatrix} 0 \\ \frac{\theta_1}{I_x} \end{bmatrix} \end{aligned} \quad (35)$$

In the case of ideal model following, $A_c(\theta) \rightarrow A_m$ and $B_c(\theta) \rightarrow B_m$. Thus, the ideal parameters become

$$\begin{cases} \theta_1^0 = I_x b_{m0} \\ \theta_2^0 = I_x a_{m2} \\ \theta_3^0 = I_x a_{m1} \end{cases} \quad (36)$$

The error dynamics in Equation (37) are found by obtaining the difference between Equations (34) and (32).

$$\begin{aligned} \dot{e} &= \dot{x} - \dot{x}_m \\ &= A_m e + (A_c(\theta) - A_m)x + (B_c(\theta) - B_m)u_c \\ &= A_m e + \begin{bmatrix} 0 & 0 \\ -\frac{\theta_2}{I_x} + a_{m2} & -\frac{\theta_3}{I_x} + a_{m1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\theta_1}{I_x} - b_{m0} \end{bmatrix} u \\ &= A_m e + \begin{bmatrix} 0 & 0 \\ -\frac{\tilde{\theta}_2}{I_x} & -\frac{\tilde{\theta}_3}{I_x} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\tilde{\theta}_1}{I_x} \end{bmatrix} u \end{aligned} \quad (37)$$

With $\tilde{\theta} = \theta - \theta^0$. Therefore, the error equation becomes:

$$\dot{e} = A_m e + \psi \tilde{\theta} \quad (38)$$

Where

$$\psi = \begin{bmatrix} 0 & 0 & 0 \\ \frac{u_c}{I_x} & -\frac{x_1}{I_x} & -\frac{x_2}{I_x} \end{bmatrix} \quad (39)$$

The positive definite Lyapunov Function in Equation (40) is considered.

$$V(e, \theta) = \frac{1}{2} e^T P e + \frac{1}{2} \tilde{\theta}^T \Gamma^{-1} \tilde{\theta} \quad (40)$$

Where P is and s.p.d and solves the Lyapnuov equation given by Equation (41) and $Q > 0$ chosen to be the identity matrix I for this controller.

$$A_m^T P + P A_m = -Q \quad (41)$$

After taking the derivative of $V(e, \theta)$ with respect to time and manipulating the terms, Equation (42) is obtained.

$$\dot{V} = -\frac{1}{2}e^T Q e + \tilde{\theta}^T \left(\Gamma^{-1} \dot{\tilde{\theta}} + \Psi^T P e \right) \quad (42)$$

Setting the second term after the plus sign to zero, will result in $\dot{V} = -\frac{1}{2}e^T Q e \leq 0$, with the error $e(t) \rightarrow 0$ as $t \rightarrow \infty$ because of boundedness by Barbalat's Lemma. Therefore, the parameter adaptation law is given by Equation (43).

$$\frac{d\tilde{\theta}}{dt} = -\Gamma \Psi^T P e \quad (43)$$

Similarly, the same procedure is applied to the other degrees of freedom, due to the equivalence in the linearized equations between all 3 subsystems.

D. MRAC with Output Feedback

In a similar fashion to the previous sections, the systems are decoupled and an MRAC with Output Feedback controller is designed for each one alone. Considering the pitch degree of freedom, the linearized dynamic model leads to the transfer function in Equation (44).

$$G(s) = \frac{1}{I_x s^2} = \frac{b_0 B}{A} \quad (44)$$

Where

$$b_0 = \frac{1}{I_x}, \quad B = 1, \quad A = s^2 \quad (45)$$

However, $G(s)$ has a pole excess of $2 > 1$, and thus is not SPR. First, a reference model is obtained in Equation (46).

$$G_m(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{B_m}{A_m} \quad (46)$$

Next, a feedforward and feedback control law in Equation (47) is devised for this system.

$$Ru = Tu_c - Sy \quad (47)$$

The following parameter structures are chosen to meet the causality conditions

$$\begin{cases} A_0(s) = s + a_0 \\ S(s) = s_0 s + s_1 \\ R(s) = R_1 B = R_1 = s + r_1 \\ T(s) = t_0 A_0 = t_0 s + t_0 a_0 = t_0 s + t_1 \end{cases} \quad (48)$$

From the Diophantine equation, the relationship in Equation (49) is obtained.

$$AR_1 + b_0 S = A_o A_m \quad (49)$$

After populating the terms and splitting the unknowns and the knowns, the equalities in Equation (50) are derived.

$$\begin{cases} r_1 = 2\zeta\omega_n + a_0 \\ s_0 = \frac{2\zeta\omega_n a_0 + \omega_n^2}{b_0} \\ s_1 = \frac{a_0 \omega_n^2}{b_0} \\ t_0 = \frac{\omega_n^2}{b_0} \\ t_1 = \frac{a_0 \omega_n^2}{b_0} \end{cases} \quad (50)$$

Next, polynomials Q and P are chosen such that the transfer function $G_1(s) = \frac{b_0 Q}{P}$ is SPR. The following simplistic conditions are initialized

$$\begin{cases} Q = A_0 A_m = (s + a_0)(s^2 + 2\zeta\omega_n s + \omega_n^2) \\ P = P_1 P_2 = A_0 A_m \\ \Rightarrow P_1 = A_m = s^2 + 2\zeta\omega_n s + \omega_n^2 \\ \Rightarrow P_2 = A_0 = s + a_0 \end{cases} \quad (51)$$

A new error called filtered error is introduced in Equation (52).

$$e_f = \frac{Q}{P} e = e = y - y_m \quad (52)$$

The error augmentation η is given by Equation (53).

$$\begin{aligned} \eta &= -\left(\frac{1}{P_1} u(t) + \varphi^T \theta\right) \\ &= -\frac{1}{A_m} u - \varphi^T \theta \end{aligned} \quad (53)$$

With the control input u defined as

$$u = -\theta^T (P_1 \varphi) \quad (54)$$

Where

$$\theta = \hat{\theta} = \begin{bmatrix} r_1' \\ \hat{s}_0 \\ \hat{s}_1 \\ \hat{t}_0 \\ \hat{t}_1 \end{bmatrix} \quad \text{and} \quad \varphi = \frac{1}{P} \begin{bmatrix} u \\ sy \\ y \\ -su_c \\ -u_c \end{bmatrix}$$

The augmented error ε is defined as

$$\varepsilon = e_f + \hat{b}_0 \eta \quad (55)$$

Where two parameter adaptation laws are introduced

$$\begin{cases} \dot{\hat{b}}_0 = -\gamma_1 \eta \varepsilon \\ \dot{\hat{\theta}} = +\gamma_2 \varphi \varepsilon \end{cases} \quad (56) \quad (57)$$

Equivalently, the same procedure is applied to the roll θ and yaw ψ of the quadcopter to obtain the control and adaptation laws of the full system.

E. MRAC with Disturbance Rejection

In the design of previous controllers, the disturbances were not taken into account in the model. Therefore, a successful rejection of disturbances is obtained by expecting its presence inside the model, estimating it, and negating its effect. In the quadrotor's case, the disturbance is assumed to be constant force from the wind producing a torque on the pitch, roll and yaw. Therefore, the disturbance is assumed to be a step input that takes effect a time T on the control input $u(t)$ to the plant. Thus, the disturbance d can be estimated by the simple parameter adaptation law in Equation (58).

$$\dot{\hat{d}} = +\gamma_d e \quad (58)$$

Where $\gamma_d > 0$ is the adaptive rate. Therefore, the new control input $u_d(t)$ is derived from the initial control input $u(t)$ obtained from the MRAC and the estimated disturbance \hat{d} , as seen in Equation (59).

$$u_d(t) = u(t) - \hat{d} \quad (59)$$

V. SIMULATION RESULTS & DISCUSSIONS

The controllers designed in Section IV are implemented in Simulink with the initial parameters and variables defined in MATLAB. The true parameters of the Quanser 3DOF Hover platform are given in Table I.

Symbol	Description	Value	Unit
I_x	Moment of inertia about pitch axis	0.0552	$kg.m^2$
I_y	Moment of inertia about roll axis	0.0552	$kg.m^2$
I_z	Moment of inertia about yaw axis	0.1104	$kg.m^2$

TABLE I
REAL PARAMETER VALUES OF THE QUANSER 3DOF HOVER.

A. Proportional Controller

One of the targets of this paper is to identify the true values of the moment of inertias along the pitch, roll, and yaw. However, when using a proportional controller, the parameters cannot be identified. Therefore, they are set to be wrong values (since they are assumed to be unknown): $I_x = 0.0235$, $I_y = 0.0494$, $I_z = 0.6121$. The tracking response for the pitch, roll, and yaw are shown in Figures 4, 5, and 6. It can be seen that the response is not ideal since there is an overshoot for all responses.

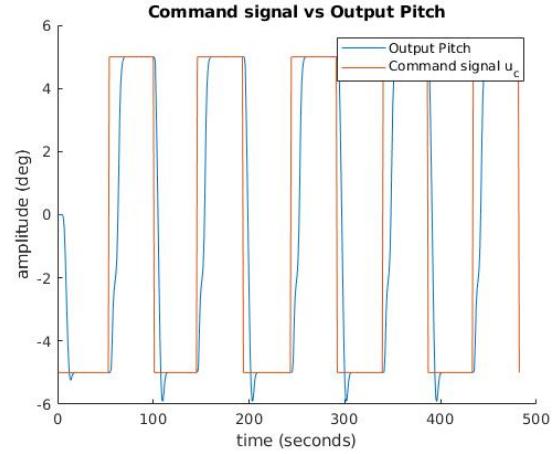


Fig. 4. Plot showing the reference command input u_c and the actual output y for the Pitch with a Proportional Controller

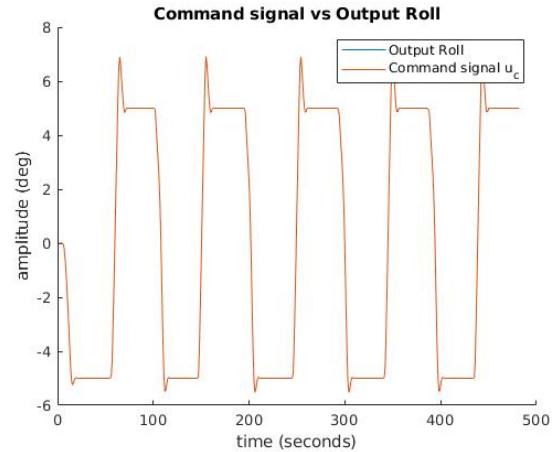


Fig. 5. Plot showing the reference command input u_c and the actual output y for the Roll with a Proportional Controller

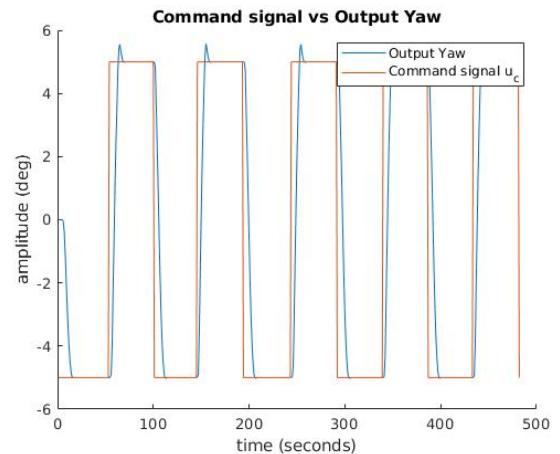


Fig. 6. Plot showing the reference command input u_c and the actual output y for the Yaw with a Proportional Controller

In addition, as seen in Figures 7, 8, 9, the error is oscillating between 10 and -10 due to the overshoot present.

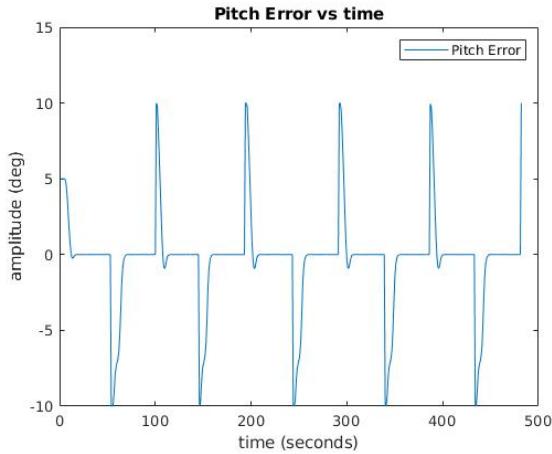


Fig. 7. Plot showing the output error e for the Pitch with a Proportional Controller

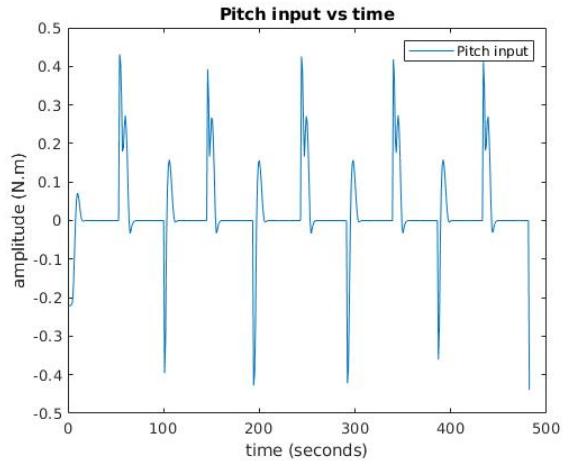


Fig. 10. Plot showing the input u for the Pitch with a Proportional Controller

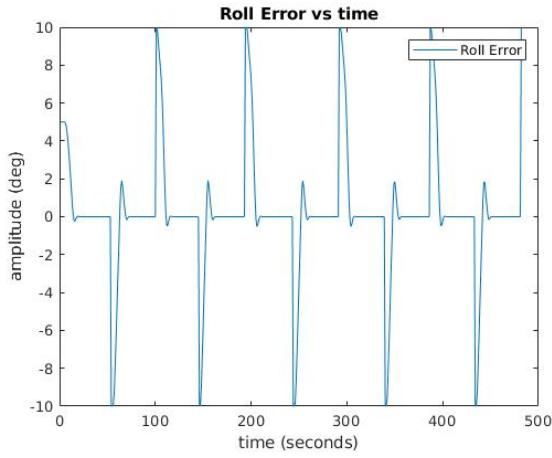


Fig. 8. Plot showing the output error e for the Roll with a Proportional Controller

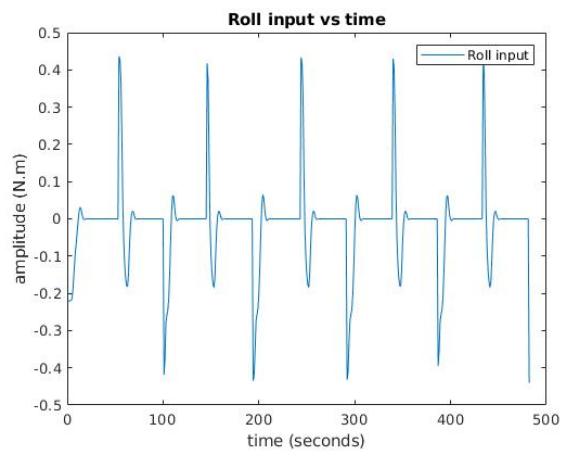


Fig. 11. Plot showing the input u for the Roll with a Proportional Controller

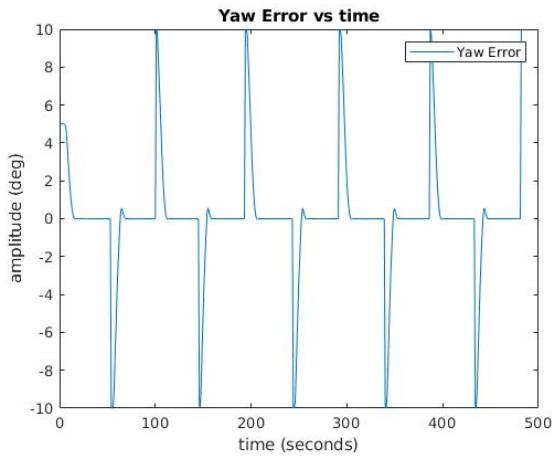


Fig. 9. Plot showing the output error e for the Yaw with a Proportional Controller

The inputs to the decoupled plants, illustrated in Figures 10, 11,12, there is no ringing in the signal.

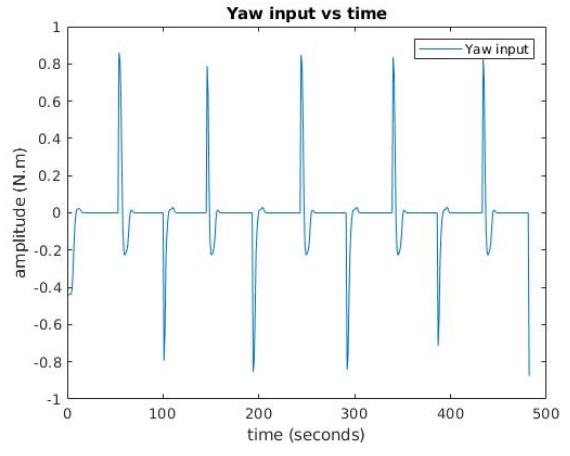


Fig. 12. Plot showing the input u for the Yaw with a Proportional Controller

It can be deduced that the performance of the proportional controller does not meet the requirements, as it does not adopt perfect tracking and it does not apply any parameter estimation to the unknown moment of inertias. Therefore, other controllers will be studied.

B. Indirect Self Tuning Regulator

To validate the ISTR designs, the ISTR controllers are tested on the linearized model of the quadcopter. The simulations are carried on MATLAB's Simulink.

Without loss of generality, the model of the quadcopter's pitch angle is used to discuss the parameters vector and the regressor vector used in the adaptation process. The parameters and regressor vectors are respectively:

$$\theta_{\phi}^T = [a_1 \quad a_2 \quad b_{0\phi} \quad b_{1\phi}] \quad (60)$$

$$\beta_{(t)\phi}^T = [\phi_{(t-1)} \quad \phi_{(t-2)} \quad u_{(t-1)} \quad u_{(t-2)}] \quad (61)$$

The estimation error ϵ_{ϕ} is defined as:

$$\epsilon_{\phi} = \phi_{(t)} - \beta_{(t)\phi}^T \theta_{(t-1)\phi}^T \quad (62)$$

The designed ISTR controllers are implemented on Simulink and they yielded the following results:

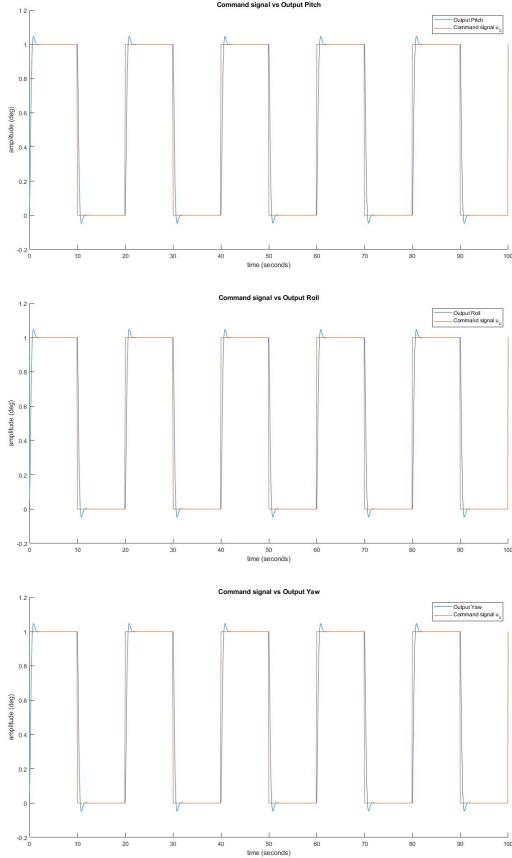


Fig. 13. Plot showing the reference command input u_c , and the actual output y for the Pitch (Top), Roll (Middle) and Yaw (Bottom) with ISTR controllers.

It can be observed from Figure 13 that the pitch, roll, and yaw are tracking the reference model almost perfectly.

It can be observed from Figure 14 that the parameter estimates of the pitch, roll, and yaw don't converge to their true values. In fact, the parameters are not updated with time. To figure out the reason behind the parameters not updating during the simulation, the estimation error of the pitch is plotted.

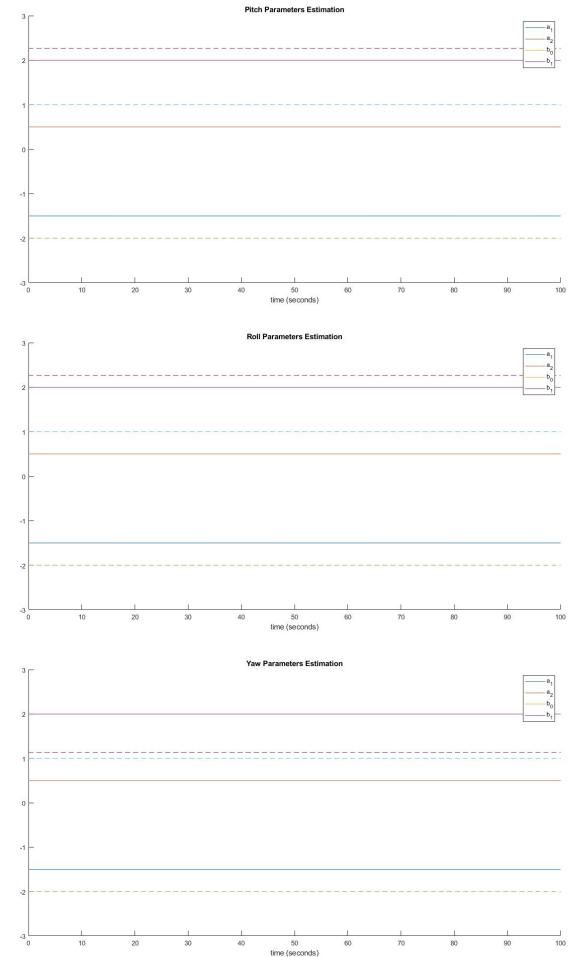


Fig. 14. Plot showing the parameter estimates and their actual values for the Pitch (Top), Roll (Middle) and Yaw (Bottom) with ISTR controllers.

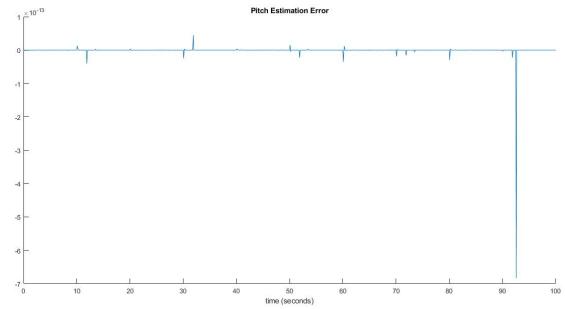


Fig. 15. Plot showing the estimation error of the Pitch with an ISTR controller.

Figure 15 suggests that the magnitude of the pitch estimation error is in the order 10^{-13} , which means that the parameters are updating at an extremely low rate. This explains why the parameters seem to be constant throughout the simulation. Another reason is that the linear models of the quadcopter have a built-in integrator that guarantees tracking the command signal; thus, decreasing the estimation error which limits the parameter updating process.

The ISTR models are also tested on the non-linear models

of the quadcopter. However, no command signal tracking was achievable. Such results are expected because the STR analysis assumes a SISO linear system which is not the case in this simulation.

C. MRAC with Full State Feedback

Since the controllers were designed from the linearized model of the quadcopter, the tests will take effect on the non-linear high-fidelity model to capture the real experimental setup as close as possible. The controller designed in Section IV-C is implemented in Simulink with the following sampling frequency, initial conditions, performance requirements, reference model, and adaptation rates listed in Table II.

Variable	Value
Sampling frequency	2kHz
I_{x0}	0.04kgm ²
I_{y0}	0.04kgm ²
I_{z0}	0.8kgm ²
%overshoot	20%
Settling Time	5s
Q	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
$\gamma_{pitch}, \gamma_{roll}$	$\begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix}$
γ_{yaw}	$\begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}$

TABLE II

CONTROLLER INITIALIZATION FOR MRAC FULL STATE FEEDBACK.

The results of the simulation for a square wave of amplitude 5^0 , and a period of $20s$ are divided into 4 sections:

- 1) A plot showing the reference command input u_c , the reference output y_m , and the actual output y .
- 2) A plot showing the output tracking error $e = y - y_m$.
- 3) A plot showing the control input u .
- 4) A plot showing the parameter estimates and their actual values.

In addition, the results are shown only for pitch and yaw, since there exists a symmetry between the pitch and the roll, giving the same behavior and results.

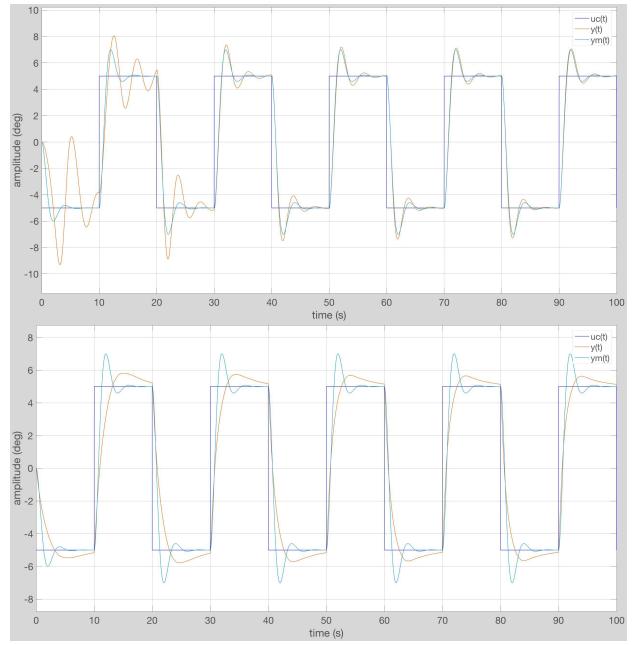


Fig. 16. Plot showing the reference command input u_c , the reference output y_m , and the actual output y for the Pitch (Top) and Yaw (Bottom) with MRAC Full State Feedback.

It can be observed from Figure 16 that the pitch starts with a non-perfect model following, but ends up tracking the reference model almost perfectly. The yaw, however, is stable and settles to a zero error, but doesn't perform like the reference model.

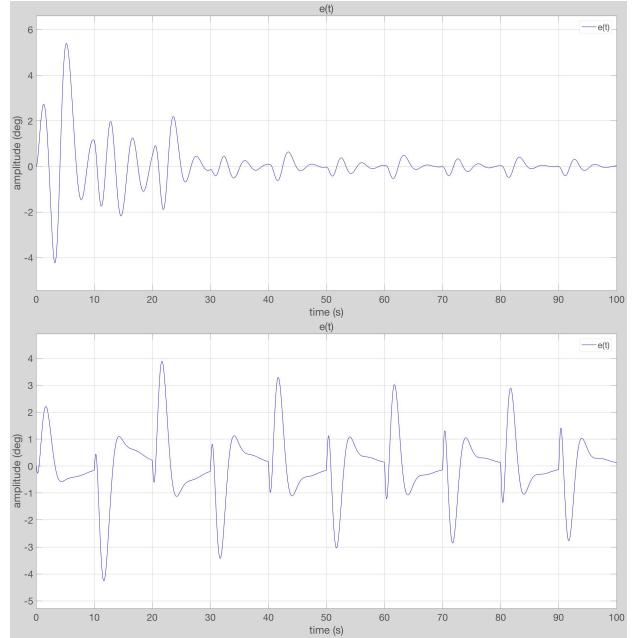


Fig. 17. Plot showing the output tracking error $e = y - y_m$ for the Pitch (Top) and Yaw (Bottom) with MRAC Full State Feedback.

Figure 17 reinforces the points stated previously in that the output tracking error $e \rightarrow 0$ as $t \rightarrow \infty$ for the pitch, but keeps fluctuating for the yaw angle with a narrower and smaller error margin as $t \rightarrow \infty$.

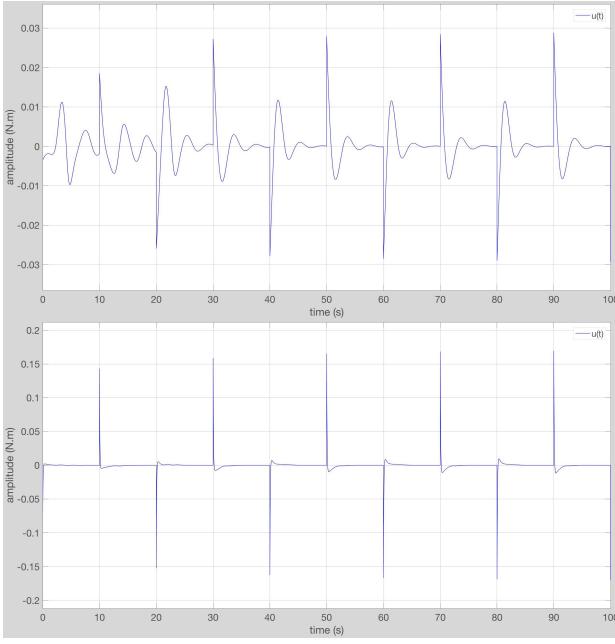


Fig. 18. Plot showing the control input u for the Pitch (Top) and Yaw (Bottom) with MRAC Full State Feedback.

The control input $u(t)$ from Figure 18 for the pitch shows variability and richness in the signal, while the yaw is a simple signal. This will affect the PE conditions for the parameter estimates in the next figure.

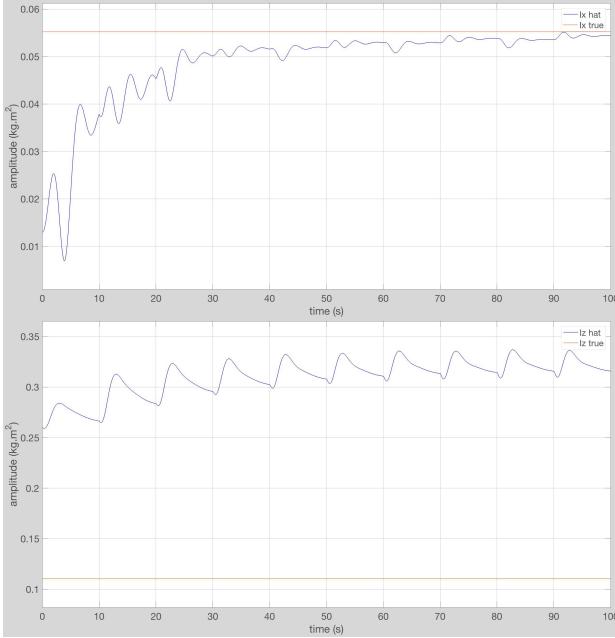


Fig. 19. Plot showing the parameter estimates and their actual values for the Pitch (Top) and Yaw (Bottom) with MRAC Full State Feedback.

As expected from the previous results of Figure 18, the parameter estimate of I_x converges to its real value as $t \rightarrow \infty$ as seen in Figure 19. Contrary to the results of the yaw angle which converge slowly to a wrong final value, as expected from the PE conditions.

D. MRAC with Output Feedback

Next, the controller designed in Section IV-D is implemented with the simulation parameters listed in Table III.

Variable	Value
Sampling frequency	2kHz
I_{x0}	0.04kgm^2
I_{y0}	0.04kgm^2
I_{z0}	0.8kgm^2
%overshoot	20%
Settling Time	5s
a_{0pitch}, a_{0roll}	4
a_{0yaw}	1
$\gamma_{1pitch}, \gamma_{1roll}, \gamma_{1yaw}$	10
$\gamma_{2pitch}, \gamma_{2roll}, \gamma_{2yaw}$	10

TABLE III
CONTROLLER INITIALIZATION FOR MRAC OUTPUT FEEDBACK.

The results of the simulation for a square wave of amplitude 5° , and a period of 20s are displayed below for the pitch and yaw degrees of freedom, similar to the previous controller.

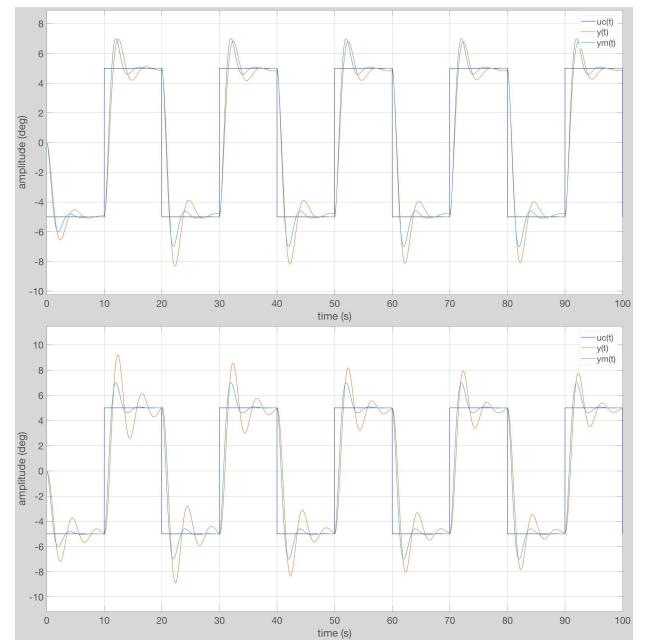


Fig. 20. Plot showing the reference command input uc , the reference output ym , and the actual output y for the Pitch (Top) and Yaw (Bottom) with MRAC Output Feedback.

It can be observed from Figure 20 that the pitch angle has a similar behavior to the reference model, and attempts to settle to the desired angle during the 10s. Contrary to the yaw results of the MRAC Full State Feedback controller of Section IV-C, the yaw angle exhibits a second order underdamped response, but with more oscillations and less damping than the reference model.

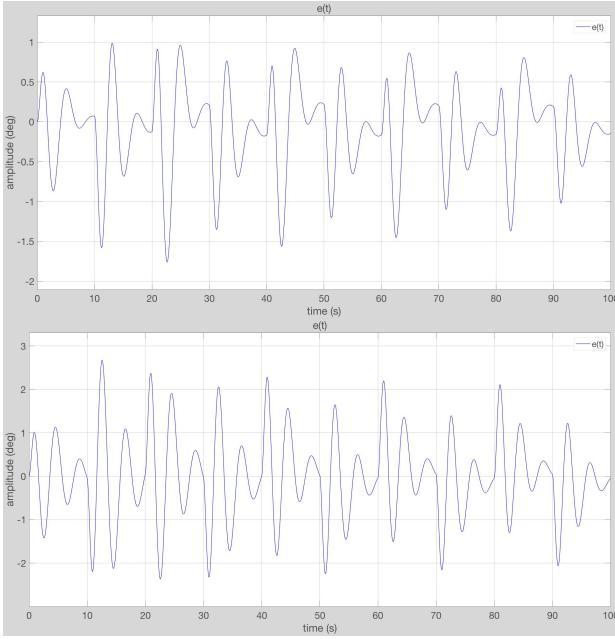


Fig. 21. Plot showing the output tracking error $e=y-y_m$ for the Pitch (Top) and Yaw (Bottom) with MRAC Output Feedback.

Figure 21 shows that the error between the actual and the reference outputs fluctuates and doesn't settle completely to 0 during the half-period of the reference command input. Nevertheless, the figure shows a shrinking error between both outputs y and y_m at each half-period of u_c , hinting to a better model following with $e \rightarrow 0$ as $t \rightarrow \infty$.

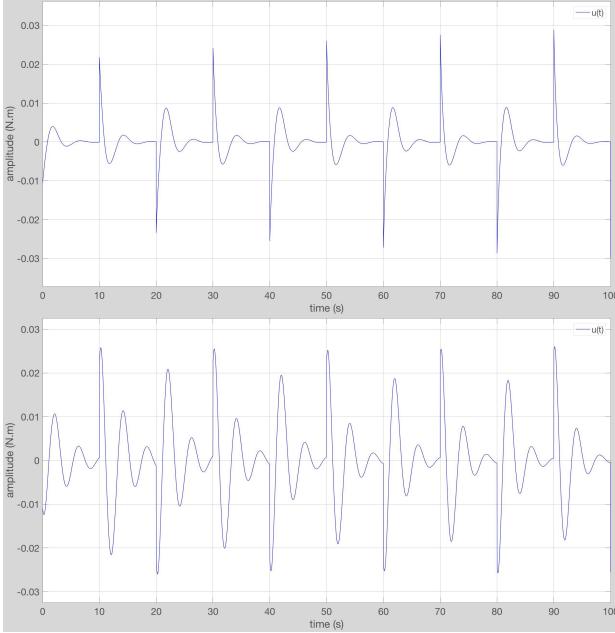


Fig. 22. Plot showing the control input u for the Pitch (Top) and Yaw (Bottom) with MRAC Output Feedback.

Both control inputs $u(t)$ seen in Figure 22 show a slightly rich signal with that of the yaw being a greater candidate at satisfying the PE conditions for parameter convergence.

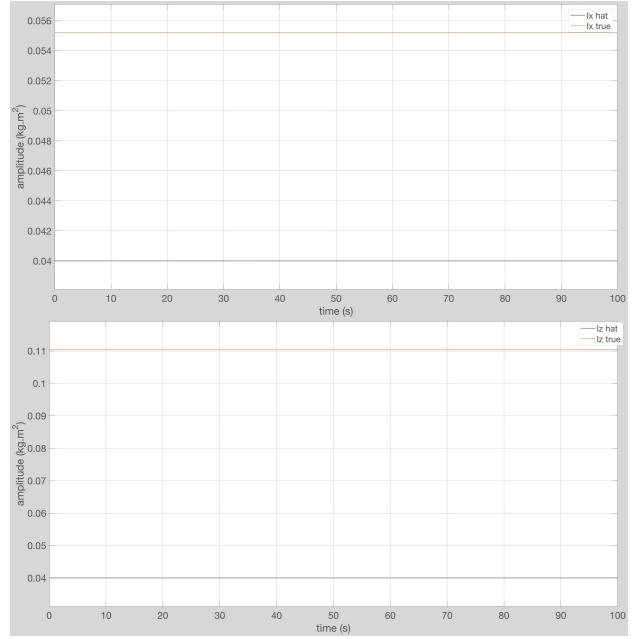


Fig. 23. Plot showing the parameter estimates and their actual values for the Pitch (Top) and Yaw (Bottom) with MRAC Output Feedback.

Figure 23 shows no change in the parameter estimates of the moments of inertia, and no convergence to their true values.

E. MRAC with Disturbance Rejection

This last controller was implemented by modifying the MRAC Full State Feedback controller, with a few additional initializations and variables listed in Table IV.

Variable	Value
cutoff frequency	45Hz
$\hat{d}_{0pitch}, \hat{d}_{0roll}, \hat{d}_{0yaw}$	0
$\gamma_{dpitch}, \gamma_{droll}, \gamma_{dyaw}$	[5 5]

TABLE IV
CONTROLLER INITIALIZATION FOR MRAC WITH DISTURBANCE.

The results of this controller are simulated for a reference command input u_c of a square wave of amplitude 5^0 and a period of $20s$, and a step disturbance on the control input of the pitch angle of value $1N.m$ taking effect at $T = 30s$. A low pass filter was also added on the control input u to filter any high frequency elements.

Figure 24 clearly shows the manifestation of the disturbance at $T = 30s$ on the actual output y of the pitch. However, it does not take effect on the yaw angle which already attempts to estimate it, due to coupling. It should be noted that both the pitch and yaw controllers exhibit perfect model reference tracking before and after the injection of the disturbance. At the instance that the disturbance takes effect, a sharp jump can be observed at the pitch output y , followed by direct effect from the controller to get the output back to the reference. The result is a $4s$ duration to recover back from the disturbance. In addition, fluctuations of y around the reference y_m can be seen

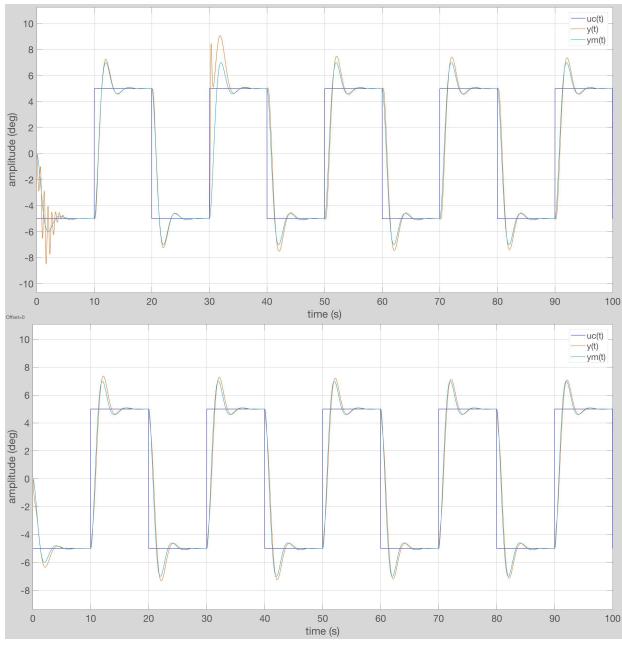


Fig. 24. Plot showing the reference command input uc , the reference output ym , and the actual output y for the Pitch (Top) and Yaw (Bottom) with MRAC Disturbance Rejection.

in the first half-period of the signal u_c . This is due to the initial attempt of the adaptation law to estimate the disturbance.

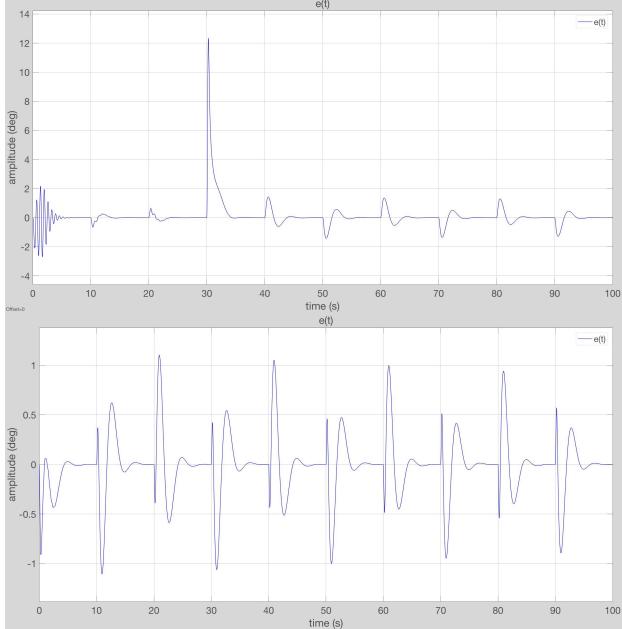


Fig. 25. Plot showing the output tracking error $e=y-ym$ for the Pitch (Top) and Yaw (Bottom) with MRAC Disturbance Rejection.

Figure 25 shows the yaw angle error e being almost the same as for the previous MRAC controllers. However, in the pitch angle, the small oscillations due to the disturbance estimation and the sharp jump can both be seen in the error plot. Also, the disturbance is followed by an initial error between y and ym of 12° followed by a convergence to 0 in 4s.

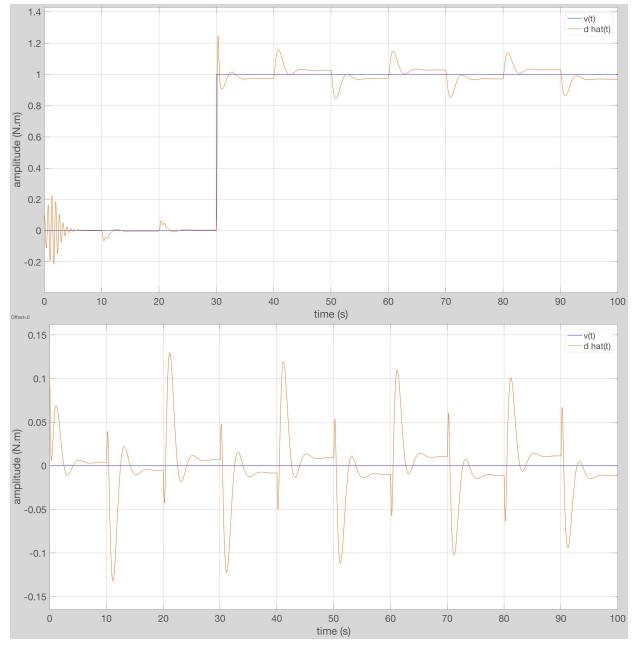


Fig. 26. Plot showing the control input u for the Pitch (Top) and Yaw (Bottom) with MRAC Disturbance Rejection.

The initial oscillations in y and e of the pitch angle can be also seen in Figure 26 for the disturbance estimation \hat{d} . As for the disturbance estimation in the yaw angle, and as noted by Equation 58, \hat{d} is proportional to the error e , exhibiting a similar pattern and shrinking to 0 as $t \rightarrow \infty$. Also, the top plot of Figure 26 presents an accurate disturbance estimation initially when $d = 0\text{N.m}$ and later when $d = 1\text{N.m}$.

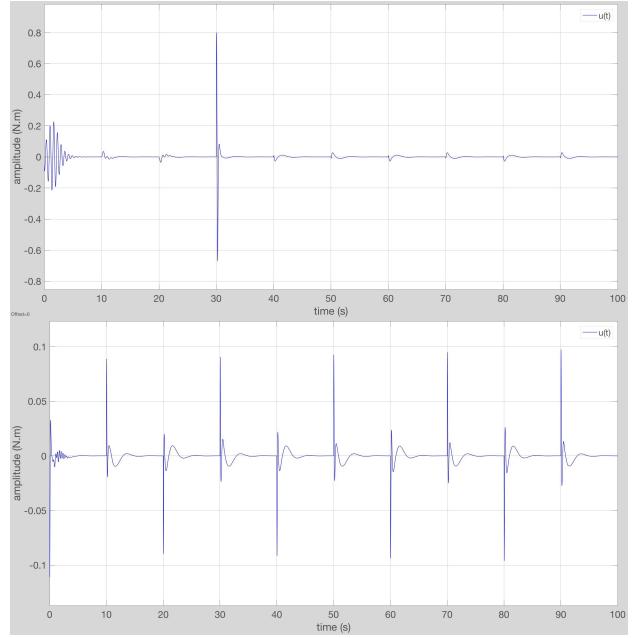


Fig. 27. Plot showing the control input u for the Pitch (Top) and Yaw (Bottom) with MRAC Disturbance Rejection.

The control input of the yaw angle showcased in the bottom row of Figure 27 is the same as for previous MRAC con-

trollers. On the other hand, the pitch control input u exhibits the same oscillations at the start, that is correlated with the previous plots, as well as a sharp jump to $0.8N.m$ at $T = 30s$.

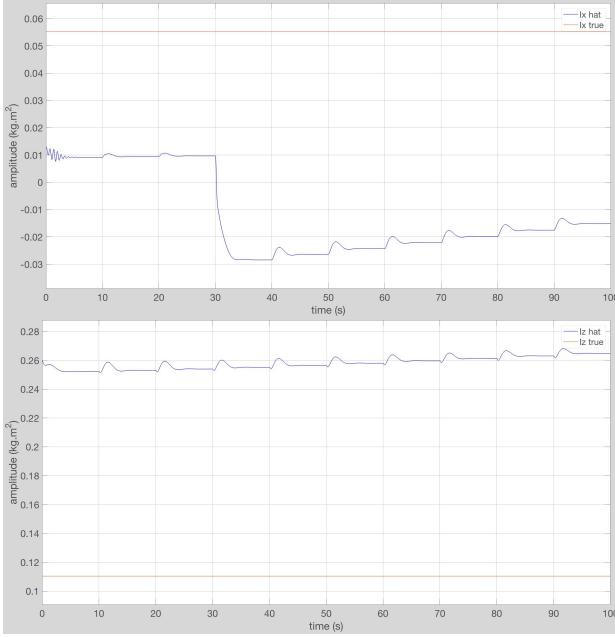


Fig. 28. Plot showing the parameter estimates and their actual values for the Pitch (Top) and Yaw (Bottom) with MRAC Disturbance Rejection.

Figure 28 shows a similar behavior of the yaw parameter estimate as for other MRAC controllers. As for the pitch DOF parameter estimate, its behavior prior to the disturbance injection was non-converging like that of previous controllers, but suddenly drops with the injection of d , and attempts to converge back afterwards to its previous estimated value.

VI. CONCLUSION

In this study, different controllers have been developed and simulated on the Quanser 3DoF Hover nonlinear high-fidelity model. In simulation, the proportional controller exhibited an overshoot in its response with an oscillating error in a 20 degree range, while the ISTR controller provided signal tracking without guarantying parameter convergence to their true values because of the built-in integrator in the quadcopter's model, which limits the updating of the parameters by rendering the estimation error negligible. Two MRAC Full State and Output Feedback controllers were implemented and showed a good performance in tracking the reference model for all the degrees of freedom. Error convergence to zero was always met, however, parameter convergence was not guaranteed and depended on the PE conditions of the control input u , which were not explicitly accounted for. Finally, a modified MRAC with disturbance estimation and rejection capabilities was tested and proved to overcome wind disturbances effectively in a duration $\leq 4s$. It would be interesting to model and test in a future work the parameter variabilities inside the model as the quadcopter is operational. This scenario would be useful in quadcopter applications involving picking up packages for

delivery, or in quadcopters with morphable structures that change according to the situation on hand, thus completely changing the dynamics of the plant.

REFERENCES

- [1] U. R. Mogili and B. Deepak, "Review on application of drone systems in precision agriculture," *Procedia computer science*, vol. 133, pp. 502–509, 2018.
- [2] M. M. Kurdi, A. K. Dadykin, I. A. Elzein, and A. K. Ibrahim, "Cooperative unmanned air and ground vehicles for landmine detection," 2019.
- [3] J. T. Zou, Z. Y. Pan, D. L. Zhang, and R. F. Zheng, "Integration of the target position correction software with the high endurance quadcopter for search and rescue mission," in *Applied Mechanics and Materials*, vol. 764. Trans Tech Publ, 2015, pp. 713–717.
- [4] S. Hamilton and J. Stephenson, "Testing uav (drone) aerial photography and photogrammetry for archeology," *Lakehead University, Tech. Rep.*, 2016.
- [5] R. Ibrahimov, E. Tsykunov, V. Shirokun, A. Somov, and D. Tsetserukou, "Dronepick: Object picking and delivery teleoperation with the drone controlled by a wearable tactile display," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2019, pp. 1–6.
- [6] Z. Tahir, M. Jamil, S. A. Liaqat, L. Mubarak, W. Tahir, and S. O. Gilani, "Design and development of optimal control system for quad copter uav," *Indian Journal of Science and Technology*, vol. 9, no. 25, pp. 10–17 485, 2016.
- [7] A. Al-Mahturi, F. Santoso, M. A. Garratt, and S. G. Anavatti, "Nonlinear altitude control of a quadcopter drone using interval type-2 fuzzy logic," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 236–241.
- [8] H. Merabti, I. Bouchachi, and K. Belarbi, "Nonlinear model predictive control of quadcopter," in *2015 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*. IEEE, 2015, pp. 208–211.
- [9] K. M. Thu and A. Gavrilov, "Designing and modeling of quadcopter control system using 11 adaptive control," *Procedia Computer Science*, vol. 103, no. C, pp. 528–535, 2017.
- [10] S.-K. Kim and C. K. Ahn, "Adaptive nonlinear tracking control algorithm for quadcopter applications," *IEEE Transactions on Aerospace and Electronic Systems*, 2019.
- [11] A. A. Kalat, "Tracking error-based self-tuning controller," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 44, no. 1, pp. 461–471, 2020.
- [12] M. Dhaybi and N. Daher, "Real-time estimation of the inertia tensor elements of a quadcopter hover platform," in *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2019, pp. 1347–1352.
- [13] F. D. Sabatino, "Quadrotor control: modeling, nonlinearcontrol design, and simulation," 2015.