A PID tuning strategy for minimizing energy consumption in quadrotors

Paraj Ganchaudhuri and Chayan Bhawal

***Abstract*— Despite the enormous use cases of quadrotors, they remain energy-inefficient aerial platforms. In this work, we show that navigating an energy optimal position trajectory does not necessarily mean minimum energy consumption in practice, highlighting the crucial influence of controller gains on trajectory tracking performance and the resulting energy usage. To address this, we propose an energy optimized gain tuning strategy for a PX4 controller architecture where energy optimal states are used as reference signals and a propulsion dynamics based energy model serves as the optimization objective function. Numerical results show that gains obtained from the proposed strategy yields up to 16% energy savings compared to gains optimized solely for tracking LETs. Overall, the proposed method achieves 34% average energy savings, with a maximum reduction of 69%, compared to conventional tuning methods.**
Keywords: Quadrotor, Energy optimization, PID Controller Tuning

I. Introduction

Battery powered quadrotors are energy-inefficient aerial vehicles as a large portion of available energy is expended in supporting its own weight [1]. Increasing battery capacity adds to the overall airframe mass, which prevents scaling flying time proportionally with battery capacity [2]. This limitation motivates the need for alternative strategies to enhance energy efficiency, among which computing Least Energy Trajectory (LET) [3], [4] is a widely adopted approach. A LET between two waypoints in $\mathbb{R}^3$ are time parameterized position values that theoretically consumes the lowest energy. This approach is relatively simple compared to other alternatives such as airframe optimization, mid-flight laser based recharging, or multi-vehicle cooperative operations [5]. However, while promising, LET generation only partially address energy optimization in practical applications. In real-world scenarios, controllers determine trajectory tracking performance and therefore influence the actual energy consumption during closed-loop operation, necessitating appropriate gain tuning [6]. However, to the best of our knowledge, quadrotor controller tuning with an explicit focus on energy consumption has largely been overlooked. To address this gap, we propose an energy optimized PID tuning framework that leads to energy consumption reduction during a quadrotors flight.

LETs are commonly generated by solving an Optimal Control Problem (OCP) [7] where a quadrotor energy consumption model is used as the cost function to compute optimal state and control inputs to be used by the quadrotor for flight between two waypoints. Energy consumption depend on the control input (which translates to rotor speeds and rotor accelerations), and most existing studies implicitly assume that tracking a LET (position trajectories) guarantees minimal energy consumption [3], [4]. However, in closed-loop configuration for least energy consumption, control inputs are generated by minimizing tracking errors in the states and therefore differ from the open-loop optimal inputs [8]. Consequently, tracking a LET may not not necessarily result in minimal energy consumption. To address this, we propose a tuning technique that finds optimized control gains for a PID controller architecture to minimize energy consumption while tracking LETs. Cascaded PID controller architecture is widely used for quadrotors [8], [9] (making up to 12 PID controllers or 36 gains), making the tuning process complex, computationally demanding, and dependent on the operator's experience with trial and error. In industrial applications, controller gains are tuned using step inputs or low frequency signals as position references [10], and all gains are adjusted simultaneously [11]. Such gains, however, may not be effective for tracking LETs, as demonstrated through numerical simulations (Refer Sec. III-B). In contrast to conventional approaches, we use energy optimal states (found by solving an OCP) as reference signals (ref. Sec. II) and propose a sequential tuning strategy, where inner-loop controllers are tuned first, followed by the outer-loop controllers. Furthermore, most existing quadrotor control tuning methods prioritize tracking performance and stability, with little attention to the quantitative evaluation of energy consumption. Some studies address control tuning by posing an optimization problem penalizing on a weighted sum of tracking error and control effort. Although control effort loosely reflects energy usage, it does not directly represent the actual energy consumed. Moreover, such formulations introduce competing objectives i.e. LET tracking accuracy and actuator effort which makes weight selection critical and often subjective, without guaranteeing meaningful energy savings [10], [11]. In contrast, we propose a scalar and simplified objective function based explicitly on total energy consumption, which streamlines the tuning process (ref. Sec. III-C). We also demonstrate that gains tuned for one mission may not perform efficiently for another, thereby motivating a mission-specific tuning framework (ref. Remark 4).

The paper is organized as follows. Section II infers an OCP used to compute the optimal reference states used for tuning. Section III describes the proposed tuning framework. Section IV presents numerical results and comparison with existing tuning methods. Finally, concluding remarks are provided in Section V.

A. Notations

x_1, x_3, x_5	States representing quadrotor position along inertial **x**, **y**, **z** direction, respectively $[m]$.
x_2, x_4, x_6	Quadrotor velocitiy states $x_2 = \dot{x}_1, x_4 = \dot{x}_3, x_6 = \dot{x}_5$
x_7, x_9, x_{11}	States denoting attitude angles i.e. roll, pitch, yaw angles of a quadrotor in earth-frame $[rad]$.
x_8, x_{10}, x_{12}	Attitude rate states $x_8 = \dot{x}_7, x_{10} = \dot{x}_9, x_{12} = \dot{x}_{11}$
$x_{i,i\in\{13,..,16\}}$	Angular velocity of rotor shafts $[rad/s]$.
$\alpha_{j,j\in\{1,..,4\}}$	Angular acceleration of rotor shafts $[rad/s^2]$.
x^*, x^r	Optimal value, and reference value of variable x.
$\mathbb{M}(\mathbf{x}_a\mathbf{y}_b\mathbf{z}_c)$	Mission $\mathbb{M}(\mathbf{x}_a\mathbf{y}_b\mathbf{z}_c) :=$ Quadrotor flight between $[0,0,0]^\top$ and $[a,b,c]^\top$ in inertial frame.
$\mathbb{P}_{\mathbb{M}(\mathbf{x}_a\mathbf{y}_b\mathbf{z}_c)}$	Trajectory $\mathbb{P}_{\mathbb{M}(\mathbf{x}_a\mathbf{y}_b\mathbf{z}_c)} := [\mathbf{x}_a(t)\ \ \mathbf{y}_b(t)\ \ \mathbf{z}_c(t)]^\top$ denotes the time parameterized path taken in $\mathbb{M}$.
$\mathbb{E}_{\mathbb{M}}, t_{\mathbb{M}}$	Energy $[J]$ and mission time $[s]$ to complete $\mathbb{M}$.
$\mathbb{E}_{\mathbb{M}}(\mathbb{K}_{\mathbb{M}}), t_{\mathbb{M}}(\mathbb{K}_{\mathbb{M}})$	Energy $[J]$ and mission time $[s]$ with controller $\mathbb{K}_{\mathbb{M}}$.
$\mathtt{L}_p, \mathtt{U}_p$	Lower and upper limits of variable p.

II. Optimal Reference Trajectory Generation

In this work, it is assumed that the optimal reference trajectory is obtained by solving an OCP [3] whose objective function is derived based on an physics based energy consumption model [5] $(\Xi(x_i, \alpha_j)_{i\in\{13\ldots16\}, j\in\{1\ldots4\}})$ of a quadrotor (refer

This work was supported by the Prime Minister's Research Fellowship scheme of Government of India. The authors are with the Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati, India. Emails: gparaj@iitg.ac.in, bhawal@iitg.ac.in.

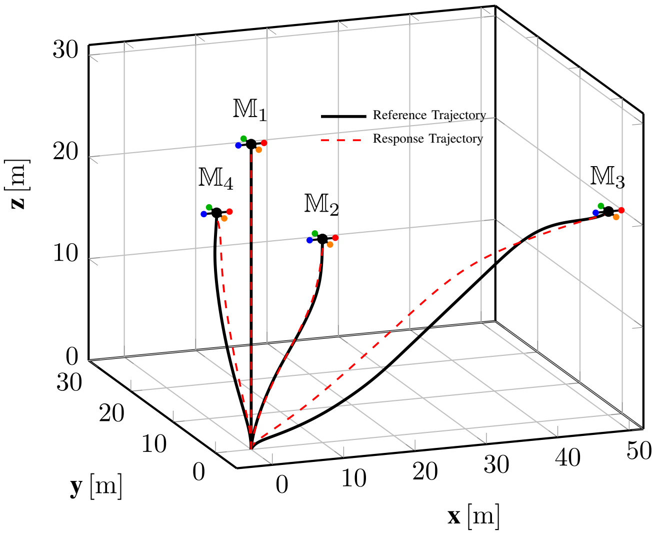

Fig. 1: Energy optimal trajectories across missions $\mathbb{M}_1$–$\mathbb{M}_4$.

[6] for the analytical formulation). For a mission $\mathbb{M}(\mathbf{x}_a\mathbf{y}_b\mathbf{z}_c)$, the solution of such an OCP includes the optimal states given by a state vector $X^*(t) := [x_1^* \;\; x_2^* \;\; \cdots \;\; x_{16}^*]^\top$ that is used to infer the optimal trajectory denoted by $\mathbb{P}^*_{\mathbb{M}(\mathbf{x}_i\mathbf{y}_j\mathbf{z}_k)}$, optimal inputs collected in a input vector [1] $U^*(t) := [\alpha_1^* \;\; \alpha_2^* \;\; \alpha_3^* \;\; \alpha_4^*]^T$, optimal mission time $t^*_{\mathbb{M}(\mathbf{x}_i\mathbf{y}_j\mathbf{z}_k)}$, and optimal (least) energy $\mathbb{E}^*_{\mathbb{M}(\mathbf{x}_i\mathbf{y}_j\mathbf{z}_k)}$. Most work on trajectory optimization ends at optimal reference generation and it is assumed the quadrotor precisely tracks them and leads to minimal energy consumption. However, we show how energy consumed depends on the controller gains and present a gain tuning formulation that finds those gains that minimizes actual realized energy in closed-loop control. In order to present our results, we analyze four distinct missions conducted on a benchmark quadrotor, whose attributes can be found in [4]. The missions are defined as follows: $\mathbb{M}_1 := \mathbb{M}(\mathbf{x}_{00}\mathbf{y}_{00}\mathbf{z}_{30})$, representing a Vertical Axis Flight (VAF); $\mathbb{M}_2 := \mathbb{M}(\mathbf{x}_{10}\mathbf{y}_{00}\mathbf{z}_{20})$ and $\mathbb{M}_3 := \mathbb{M}(\mathbf{x}_{50}\mathbf{y}_{00}\mathbf{z}_{20})$, which represent 2D flight scenarios classified as Dual Axis Flights (DAF); and $\mathbb{M}_4 := \mathbb{M}(\mathbf{x}_{10}\mathbf{y}_{25}\mathbf{z}_{15})$, corresponding to a fully 3D maneuver and representing a Multi Axis Flight (MAF). The optimal reference trajectories for these missions are calculated based on the formulation described in [3] which are illustrated in Fig. 1 with plots shown in black.

In the next section, we present an energy optimized PID tuning strategy. We begin by describing the PID architecture, then motivate the need for energy based tuning, and finally formulate the tuning strategy.

III. Energy Optimized Controller Tuning

Although several strategies exist for improving quadrotor energy efficiency, the energy actually consumed under closed-loop control is rarely examined [7]. The OCP discussed in Section II provides us the open loop optimal control inputs $U^*(t)$ and associated LET $\mathbb{P}^*_{\mathbb{M}}$ for a mission $\mathbb{M}$. However, in practice, quadrotors do not operate with open loop control inputs; instead, embedded controllers generate the required inputs to regulate errors in position and orientation. As a result, the realized trajectory and energy consumption depend on controller parameters [9]. This necessitates appropriate controller tuning to achieve the desired performance. To formulate the proposed tuning approach, we first describe a standard PX4 based PID architecture for quadrotors.

A. PID Architecture

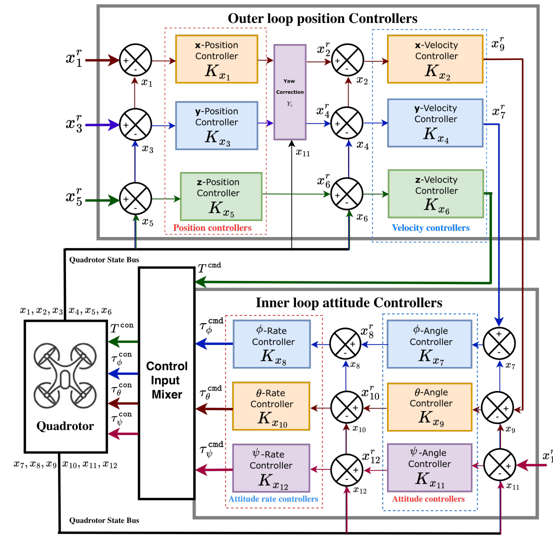

Fig. 2: PID control architecture for quadrotors

Quadrotors are underactuated systems where horizontal translation is achieved through appropriate roll and pitch tilting, so for trajectory tracking, a cascaded PID architecture comprising of an inner loop and an outer loop is typically employed (ref. Fig. 2). The outer loop position control system tracks position references and generates thrust commands and attitude angle references. These references are then tracked by an inner loop attitude control system, which produces torque commands required for appropriate tilt. A control mixer[2] maps the commanded thrust and torques to actuating thrust and torques necessary to propel the airframe. The architecture is similar to that implemented in popular autopilot firmware such as Ardupilot and PX4 [8].

The outer loop regulates translational motion along $\mathbf{x}, \mathbf{y}, \mathbf{z}$. It translates reference positions x_1^r, x_3^r, and x_5^r into reference attitude angles x_9^r, x_7^r (for $\mathbf{x}, \mathbf{y}$) and commanded thrust $T^{\mathtt{cmd}}$ (for $\mathbf{z}$). Each axis employs two cascaded PID controllers: Position controllers $K_{x_1}, K_{x_3}, K_{x_5}$ that map position errors to desired velocity commands x_2^r, x_4^r, x_6^r (velocity references are yaw-compensated to account for vehicle orientation), and Velocity controllers $K_{x_2}, K_{x_4}, K_{x_6}$ which convert velocity errors into attitude references x_7^r, x_9^r (for $\mathbf{x}, \mathbf{y}$) and thrust command $T^{\mathtt{cmd}} \in [0, 1]$ (for $\mathbf{z}$). The inner loop regulates roll, pitch, and yaw. It receives reference attitude angles and generates torque commands through cascaded PID control: Angle controllers $K_{x_7}, K_{x_9}, K_{x_{11}}$ that map attitude errors to desired angular rate commands x_8^r, x_{10}^r, x_{12}^r, and Rate controllers $K_{x_8}, K_{x_{10}}, K_{x_{12}}$ which convert angular rate errors into torque commands[3] $\tau_\phi^{\mathtt{cmd}}$, $\tau_\theta^{\mathtt{cmd}}$, $\tau_\psi^{\mathtt{cmd}}$.

All controllers enforce state and reference bounds. Specifically, $x_j^r \in [\mathtt{L}_{x_j}, \mathtt{U}_{x_j}]$, $\quad j \in \{2, 4, 6, 7, 8, 9, 10, 11, 12\}$, ensuring feasibility and actuator safety.

The architecture thus comprises 12 PID controllers. In this work, each controller is parameterized by a tuple $K_{x_i} := (P_{x_i}, I_{x_i}, D_{x_i})$, $\quad i \in \{1, \ldots, 12\}$, where x_i denotes the

[1]The control inputs are selected as rotor accelerations, instead of the conventional thrust and torque inputs, to ensure trajectory feasibility, as detailed in [3].

[2]In quadrotor PID architectures, a control input mixer is commonly used to (i) enable platform-independent controller deployment across vehicles with differing physical parameters, (ii) ensure controller outputs are actuator-feasible, and (iii) facilitate seamless transitions between control architectures.

[3]The torque commands are bounded between -1 and 1.

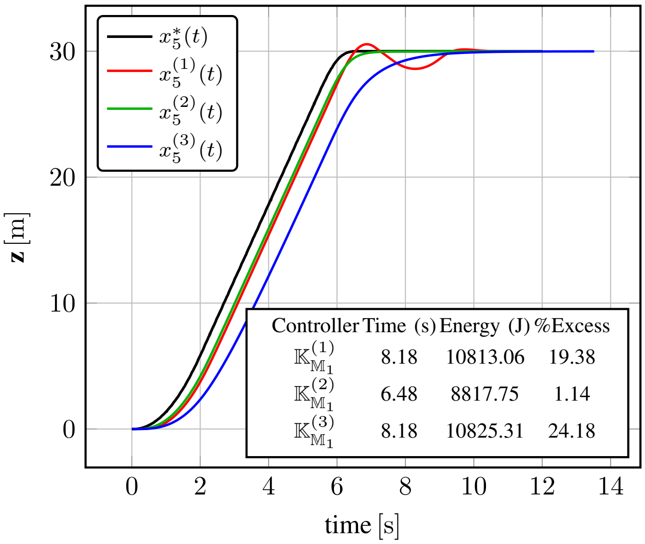

Fig. 3: Response of $\mathbb{M}_1$ with different controller sets

regulated state. Each controller belongs to a feasible set i.e. $K_{x_i} \in \Delta_{x_j}$. The feasible set is defined as:

$$\Delta_{x_j} := \Big\{ (P_{x_j}, I_{x_j}, D_{x_j}) \mid P_{x_j} \in [\mathtt{L}_{P_{x_j}}, \mathtt{U}_{P_{x_j}}], I_{x_j} \in [\mathtt{L}_{P_{x_j}}, \mathtt{U}_{I_{x_j}}], D_{x_j} \in [\mathtt{L}_{P_{x_j}}, \mathtt{U}_{D_{x_j}}] \Big\}. \tag{1}$$

We then define the controller set:

$$\mathbb{K}_{\mathbb{M}} := \{K_{x_i} = (P_{x_i}, I_{x_i}, D_{x_i}) \in \Delta_{x_j} \mid i \in \{1, 2, \cdots, 12\}\} \tag{2}$$

that includes gain tuples of all twelve PID controllers for a mission $\mathbb{M} = \mathbb{M}(\mathbf{x}_i \mathbf{y}_j \mathbf{z}_k)$.

Having discussed the control architecure, we showcase the need for energy aware PID tuning with numerical simulations in the following subsection and then present the tuning strategy.

B. Significance of energy aware controller tuning

We examine the effect of controller gain variations on quadrotor energy consumption through numerical simulations of LET tracking for $\mathbb{M}1$ using three different controller gain sets, namely $\mathbb{K}^{(1)}_{\mathbb{M}_1}$, $\mathbb{K}^{(2)}_{\mathbb{M}_1}$, and $\mathbb{K}^{(3)}_{\mathbb{M}_1}$.

For each $p \in \{1, 2, 3\}$, we define distinct controller sets as

$$\mathbb{K}^{(p)}_{\mathbb{M}_1} = \{\underbrace{K_0, K_0, K_0, K_0}, K^{(p)}_{x_5}, K^{(p)}_{x_6}, \underbrace{K_0, K_0, K_0, K_0, K_0, K_0}\} \tag{3}$$

where, the controller gains are given by $K_0 = (1, 0, 0)$ and

$$\begin{aligned} K^{(1)}_{x_5} &= (1, 0, 0), & K^{(1)}_{x_6} &= (1, 0, 0) \\ K^{(2)}_{x_5} &= (3.1, 0, 0), & K^{(2)}_{x_6} &= (1.02, 1.40, 1.24) \\ K^{(3)}_{x_5} &= (1.48, 0, 0), & K^{(3)}_{x_6} &= (0.5, 0.2, 2) \end{aligned}$$

The index p denotes the specific set of controller parameters employed. The resulting vertical position trajectory, $x_5^{(p)}(t)$, and the corresponding energy consumption are shown in Fig. 3. Among the three controllers considered, $\mathbb{K}^{(2)}_{\mathbb{M}}$ yields the lowest energy consumption, highlighting the strong influence of controller tuning on overall energy usage. This motivates the objective of identifying controller parameters that minimize energy consumption.

It is important to note that the controller set $\mathbb{K}^{(3)}_{\mathbb{M}_1}$ was tuned using a unit step reference signal [11], which when applied to track the LET, results in approximately 24% higher energy consumption than the optimum. This demonstrates that gains tuned using step inputs as reference may not be effective for energy efficient tracking of LETs. In the next subsection, we elaborate on the proposed tuning strategy.

C. PID Controller Tuning Formulation

From the OCP solution, we obtain the LET $\mathbb{P}^*_{\mathbb{M}}$ and the corresponding optimal states X^* for a mission $\mathbb{M}$. The subsequent task is to select controller gains $K_{x_j} \in \Delta_{x_j}$ such that the realized trajectory $\mathbb{P}_{\mathbb{M}}$ closely follows $\mathbb{P}^*_{\mathbb{M}}$, and the resulting energy consumption value remains closer to the optimal value $\mathbb{E}^*_{\mathbb{M}}$. Accordingly, we formulate an optimization problem to determine the PID gains that minimize energy consumption while ensuring accurate tracking of the LETs.

Tuning optimization formulation : While it can be reasonable to assume that tracking a LET perfectly would imply minimal energy consumption, an obvious choice of selecting the optimization function for tuning can be the LET tracking error. Accordingly, a tracking error metric e_{ITAE} can be the sum of ITAE[4] errors across the position states.

$$e_{ITAE} := \left\{ \int_0^{T_f} t|x_1^r - x_1| dt \right\}^2 + \left\{ \int_0^{T_f} t|x_3^r - x_3| dt \right\}^2 + \left\{ \int_0^{T_f} t|x_5^r - x_5| dt \right\}^2 \tag{4}$$

However, contrary to this line of thought, we consider the total energy consumption as an optimization function for tuning. It is argued that certain controller gains, while effectively tracking LETs, may demand higher control inputs leading to increased energy usage. We later demonstrate through numerical simulations in Section IV-B that using 4 as the tuning objective can in fact result in higher energy usage.

Remark 1: As mentioned in Sec.II, the energy consumption model Ξ depends on rotor speeds and their accelerations, and the optimal control formulation determines these optimal rotor profiles. It can also be argued that minimizing the tracking error between optimal rotor profiles and actual rotor profiles would realize minimal energy consumption. However, this is impractical in a closed-loop system because the actual rotor speeds are not directly controlled; instead, they arise from the control law that regulates various states including position, velocity, attitude angles and their rates. Consequently, the controller determines the instantaneous rotor speeds as a byproduct of minimizing the tracking errors in these states.

Since, the reference states are already energy optimal, we aim to find those gains which would effectively track them while consuming the least energy. Accordingly, the optimization problem utilized for tuning the PID controllers is as follows :

For a given mission $\mathbb{M}$, the tuning objective is to identify PID gains such that the energy functional given below is minimized while tracking optimal state references over a time horizon $[0, T_f]$:

$$\mathcal{K}^* := \arg \min_{\mathcal{K} \in \mathcal{D}} \int_0^{T_f} \mathbb{E}_{\mathbb{M}}(t; \mathcal{K})\, dt \tag{5}$$

Here, $\mathcal{K}$ denotes the set of PID gains being simultaneously tuned, $\mathcal{D}$ refers to the feasible set, and $\mathbb{E}_{\mathbb{M}}(t; \mathcal{K})$ is the energy based cost functional dependent on the mission and controller structure. Specifically, $\mathbb{E}_{\mathbb{M}}(t; \mathcal{K}) = \Xi(x_i, \alpha_j)$ (as utilized while calculating the LETs), where the states x_i ($i \in \{13 \dots 16\}$) and inputs α_j ($j \in \{1 \dots 4\}$) evolve under the closed-loop dynamics influenced by the controller $\mathcal{K}$. Since the energy functional $\mathbb{E}_{\mathbb{M}}(t; \mathcal{K})$ depends on these states, its value inherently reflects the choice of controller gains. Accordingly, the controller is

[4]The ITAE error metric is tail-heavy, assigning larger penalties to errors near steady state. Consequently, failure to reach the setpoint results in a large penalty, discouraging unstable or poorly converging responses.

tuned by directly minimizing this energy functional, without incorporating explicit tracking error terms.

Remark 2: Existing approaches that are not explicitly energy aware typically minimize a weighted combination of tracking error and controller effort [11]. Moreover, controller effort does not directly quantify actual energy consumption. Such formulations depend on subjective weight selection and, in multi-axis flights, introduce multiple competing tracking objectives that can require multi-objective optimization. By leveraging energy optimal reference trajectories, the problem reduces to minimizing a single scalar objective i.e. total energy consumption and thereby avoiding the need for weight tuning and the complexity of balancing competing objectives.

The formulation described above remains consistent across mission types, while the structure of $\mathcal{K}$ and $\mathcal{D}$ varies depending on the level of coupling and the axes involved. The mission specific tuning configurations are outlined below:

- *VAF and inner loop of DAF Missions:*
  Here tuning is performed simultaneously for two PID controllers. Define $\mathcal{K}_i := \left(K_{x_i}, K_{x_{i+1}}\right)$ and $\mathcal{D} = \Delta_{x_i} \times \Delta_{x_{i+1}}$. The energy functional $\mathbb{E}_{\mathbb{M}}(t;\mathcal{K}_i)$ evaluates performance for such pair.
- *Outer Loop of DAF and Inner loop of MAF Missions:*
  Here PID gains for two horizontal axes are tuned jointly due to strong cross coupling in quadrotor dynamics. Define
  $\mathcal{K}_i := \left(K_{x_i}, K_{x_{i+1}}\right)$, $\mathcal{K}_j := \left(K_{x_j}, K_{x_{j+1}}\right)$, and $\mathcal{D} = \Delta_{x_i} \times \Delta_{x_{i+1}} \times \Delta_{x_j} \times \Delta_{x_{j+1}}$.
  The functional becomes $\mathbb{E}_{\mathbb{M}}(t;\mathcal{K}_i, \mathcal{K}_j)$.
- *Outer Loop of MAF Missions:*
  All outer loop controllers are tuned simultaneously. Define $\mathcal{K}_i$, $\mathcal{K}_j$, $\mathcal{K}_k$ as PID gain pairs for each axis and $\mathcal{D} = \Delta_{x_i} \times \Delta_{x_{i+1}} \times \Delta_{x_j} \times \Delta_{x_{j+1}} \times \Delta_{x_k} \times \Delta_{x_{k+1}}$ as their combined domain.
  The evaluated cost function is $\mathbb{E}_{\mathbb{M}}(t;\mathcal{K}_i, \mathcal{K}_j, \mathcal{K}_k)$.

This unified formulation ensures that all PID tuning tasks, irrespective of mission complexity, minimize a common energy-based objective while accounting for the necessary coupling across axes. At the end of the tuning process, the optimized controller gain set $\mathbb{K}^*_{\mathbb{M}}$ is obtained.

Controller tuning order: Note that missions are categorized as VAF, DAF, or MAF (see Sec. II), and each category requires a specific tuning procedure. The tuning is performed sequentially, beginning with the inner loop followed by the outer loop. This approach limits the number of controllers tuned simultaneously to two to six, thereby reducing computational effort. In contrast, initiating the process with the outer loop would require simultaneous tuning of all ten controllers, significantly increasing computational cost. The tuning process uses the optimal state trajectories obtained from the OCP solution as reference signals, i.e. $x_i^r = x_i^*$, where $i \in \{1, \ldots, 12\}$.

We now describe the tuning step for each type of mission:

• *VAF mission* $\mathbb{M}(\mathbf{x}_{00}\mathbf{y}_{00}\mathbf{z}_c)$: In this case, tuning is executed in a single step where only position and velocity controller gains along the vertical direction are tuned. The reference signal applied in this case is x_5^* (Refer to Table I).

• *DAF mission* $\mathbb{M}(\mathbf{x}_a\mathbf{y}_b\mathbf{z}_c)$: Here, tuning is performed in two steps (see Table I):

Step 1: Tune the angle and rate controllers along the ϕ direction (when $a = 0$) or the θ direction (when $b = 0$), using reference signals x_7^* and x_9^*, respectively.

Step 2: Tune the position and velocity controllers along either **x**-direction (when $b = 0$) or **y**-direction (when $a = 0$) with **z**-direction simultaneously. The reference signals used for tuning are either x_3^* (for $a = 0$) or x_1^* (for $b = 0$) and x_5^*.

• *MAF mission* $\mathbb{M}(\mathbf{x}_a\mathbf{y}_b\mathbf{z}_c)$: This mission type also involves a two step tuning process (see Table I):

Step 1: Tune angle and rate controllers along ϕ and θ directions simultaneously using reference signals x_7^* and x_9^*, respectively.

Step 2: Tune position and velocity controllers along **x**, **y**, and **z** directions simultaneously using reference signals x_1^*, x_3^* and x_5^*, respectively.

VAF ($a = 0, b = 0, c > 0$)
Step 1: Tune position and velocity along **z** direction: $\mathcal{K}_5^* := \arg\min_{\mathcal{K}_5 \in \mathcal{D}} \int_0^{T_f} \mathbb{E}_{\mathbb{M}}(t;\mathcal{K}_5)\,dt$ $\mathcal{D} := \Delta_{x_5} \times \Delta_{x_6}, \mathcal{K}_5 := \left(K_{x_5},\, K_{x_6}\right)$
DAF ($a > 0$ *or* $b > 0, c > 0$)
Step 1: Tune inner loop [$i = 7$ for $b > 0$ and $i = 9$ for $a > 0$]: $\mathcal{K}_i^* := \arg\min_{\mathcal{K}_i \in \mathcal{D}} \int_0^{T_f} \mathbb{E}_{\mathbb{M}}(t;\mathcal{K}_i)dt$ $\mathcal{D} := \Delta_{x_i} \times \Delta_{x_{i+1}}, \mathcal{K}_i := \left(K_{x_i},\, K_{x_{i+1}}\right)$ **Step 2:** Tune outer loop [$i{=}3, j{=}5$ for $b{>}0$ and $i{=}1, j{=}5$ for $a{>}0$]: $(\mathcal{K}_i^*, \mathcal{K}_j^*) := \arg\min_{(\mathcal{K}_i,\mathcal{K}_j) \in \mathcal{D}} \int_0^{T_f} \mathbb{E}_{\mathbb{M}}(t;(\mathcal{K}_i,\mathcal{K}_j))dt$ $\mathcal{D} := \Delta_{x_i} \times \Delta_{x_{i+1}} \times \Delta_{x_j} \times \Delta_{x_{j+1}}$ $\mathcal{K}_i := \left(K_{x_i},\, K_{x_{i+1}}\right), \mathcal{K}_j := \left(K_{x_j},\, K_{x_{j+1}}\right)$
MAF ($a > 0, b > 0, c > 0$)
Step 1: Tune inner loop: $(\mathcal{K}_7^*, \mathcal{K}_9^*) := \arg\min_{(\mathcal{K}_7,\mathcal{K}_9) \in \mathcal{D}} \int_0^{T_f} \mathbb{E}_{\mathbb{M}}(t;(\mathcal{K}_7,\mathcal{K}_9))dt$ $\mathcal{D} := \Delta_{x_7} \times \Delta_{x_8} \times \Delta_{x_9} \times \Delta_{x_{10}}$ $\mathcal{K}_7 := \left(K_{x_7},\, K_{x_8}\right), \mathcal{K}_9 := \left(K_{x_9},\, K_{x_{10}}\right)$ **Step 2:** Tune outer loop: $(\mathcal{K}_1^*, \mathcal{K}_3^*, \mathcal{K}_5^*) := \arg\min_{(\mathcal{K}_1,\mathcal{K}_3,\mathcal{K}_5) \in \mathcal{D}} \int_0^{T_f} \mathbb{E}_{\mathbb{M}}(t;(\mathcal{K}_1,\mathcal{K}_3,\mathcal{K}_5))dt$ $\mathcal{D} := \Delta_{x_1} \times \Delta_{x_2} \times \Delta_{x_3} \times \Delta_{x_4} \times \Delta_{x_5} \times \Delta_{x_6}$ $\mathcal{K}_1 := \left(K_{x_1},\, K_{x_2}\right), \mathcal{K}_3 := \left(K_{x_3},\, K_{x_4}\right), \mathcal{K}_5 := \left(K_{x_5},\, K_{x_6}\right)$

TABLE I: PID controller tuning order for Mission $\mathbb{M}(\mathbf{x}_a\mathbf{y}_b\mathbf{z}_c)$

Mission settling criterion: The optimization formulation assumes that the closed-loop system can track the optimal reference trajectory. However, with controllers in-the-loop, the quadrotor may not reach the mission steady state value in optimal time. Therefore, we seek a sub-optimal solution that satisfies the terminal conditions within a practical tolerance. To facilitate this, the time horizon T_f is extended beyond the optimal duration $t^*_{\mathbb{M}}$ obtained from the OCP as $T_f := \sigma t^*_{\mathbb{M}}$ (with $\sigma > 1$). This ensures that the system has sufficient time to settle near the desired terminal state.

Since the tracking errors asymptotically approach zero as $T_f \to \infty$, a threshold based criterion is used to determine when the system has sufficiently converged. The corresponding time is referred to as the mission time (or settling time) and is defined as:

$$t_{\mathbb{M}}(\mathbb{K}_{\mathbb{M}}) = \sup\Big\{t \geqslant 0 \mid \exists\, t' > t \text{ such that } \|q(t') - q_f\|_\infty > 0.02\|q_f\|\Big\} \tag{6}$$

where $q_f := \lim_{t\to\infty} q(t)$. While tuning inner loop controllers, $q(t) := [x_7 \quad x_9 \quad x_{11}]^\top$ and for outer loop $q(t) := [x_1 \quad x_3 \quad x_5]^\top$.

IV. Numerical simulation on controller tuning

A. Results of the proposed tuning formulation

The optimization problem in (5) is solved with a Genetic Algorithm (GA) [9] implemented by using the MATLAB Global Optimization Toolbox. In this section, we present numerical simulations on controller tuning and compare the resulting energy consumption with the optimal solutions obtained from Section II. The simulations were performed on an Apple M2 chip computer running at 3.49 GHz, with 8 GB of RAM. The GA parameters are: `Population Size` = 25, `Max Generations` = 20, `Crossover Fraction` = 0.7, `Function Tolerance` = 0.01. Parallel evaluation of the population is enabled to reduce computational time.

We present a detailed case study of tuning Mission $\mathbb{M}_4$ using the proposed formulation. The tuning process begins with an initial set of default gains, which are then refined using the proposed tuning framework.

Initialization: $K_{x_7}^{(0)} = K_{x_9}^{(0)} = (4,0,0)$, $K_{x_2}^{(0)} = K_{x_4}^{(0)} = K_{x_6}^{(0)} = (0.5,0,0)$.
$K_{x_i}^{(0)} = K = (1,0,0)$ for $i \in \{1,\ldots,12\} \setminus \{2,4,6,7,9\}$.
Time horizon T_f: $\sigma = 2.5$, i.e., $T_f = 13.52$ s ($t^*_{\mathbb{M}_4} = 5.41$ s).
At the end of tuning, the controller gains found are:

Step 1: $K^*_{x_7} = (5.84,0,0)$, $K^*_{x_8} = (4.20,0.81,1.65)$.
$K^*_{x_9} = (4.00,0,0)$, $K^*_{x_{10}} = (2.00,1.00,1.00)$.
Step 2: $K^*_{x_1} = (1.40,0,0)$, $K^*_{x_2} = (0.50,0.19,0.19)$.
$K^*_{x_3} = (1.19,0,0)$, $K^*_{x_4} = (0.34,0.00,0.40)$.
$K^*_{x_5} = (1.48,0,0)$, $K^*_{x_6} = (0.50,0.30,2.00)$.

and the optimized gain controller set $\mathbb{K}^*_{\mathbb{M}_4}$ is given by

$$\mathbb{K}^*_{\mathbb{M}_4} = \{K^*_{x_1}, K^*_{x_2}, K^*_{x_3}, K^*_{x_4}, K^*_{x_5}, K^*_{x_6}, K^*_{x_7}, K^*_{x_8}, K^*_{x_9}, K^*_{x_{10}}, K, K\}.$$

Fig. 4 shows the relevant trajectories after each tuning step. In order to show the response of the inner loop controllers to tuned PID gains, we use the symbol x_i^{in} in Fig. 4. Using $\mathbb{K}^*_{\mathbb{M}_4}$, the energy consumed in $\mathbb{M}_4$ is $\mathbb{E}_{\mathbb{M}_4}(\mathbb{K}^*_{\mathbb{M}_4}) = 8628.81$ J and the mission time is $t_{\mathbb{M}_4}(\mathbb{K}^*_{\mathbb{M}_4}) = 6.19$ sec. Using the proposed tuning approach, the optimized PID gains enable mission execution with only a 10% deviation from theoretical optimum. To evaluate its effectiveness, we compare the results with those obtained from a standard tuning method, discussed in the next subsection.

A comparison of the energy consumption and mission time obtained using the tuned controllers with their optimal values for all four missions is presented in Table II. The corresponding trajectory tracking response is shown in Fig. 1, with plots market in red.

Mission $\mathbb{M}_i$	Energy [J]			Mission Time [sec]		
	$\mathbb{E}^*_{\mathbb{M}_i}$	$\mathbb{E}_{\mathbb{M}_i}(\mathbb{K}^*_{\mathbb{M}_i})$	Error	$t^*_{\mathbb{M}_i}$	$t_{\mathbb{M}_i}(\mathbb{K}^*_{\mathbb{M}_i})$	Error
$\mathbb{M}_1$	8717.20	8830.50	1.29%	6.63	6.47	-2.41%
$\mathbb{M}_2$	6881.63	7383.98	7.29%	5.08	5.35	5.31%
$\mathbb{M}_3$	10541.51	11469.06	8.79%	7.52	8.47	12.63%
$\mathbb{M}_4$	7813.28	8628.81	10.43%	5.41	6.19	14.41%

TABLE II: Energy and mission time data using the proposed tuning framework compared with their optimal values

Remark 3: In theory, for a mission $\mathbb{M}$, applying $U^*(t)$ would reproduce $X^*(t)$ and achieve $\mathbb{E}^*_{\mathbb{M}}$. But in a practical scenario, the optimal trajectory $\mathbb{P}^*_{\mathbb{M}}$ is applied as reference which generates velocity, attitude, and attitude rate signals reference signals. The resulting position, velocity, attitude, and rate errors are processed by cascaded PID controllers to produce the actual control inputs. These inputs generally differ from $U^*(t)$, and there is no inherent mechanism ensuring they remain close to the open-loop optimum. Consequently, even if the trajectory is accurately tracked, minimal energy consumption is not guaranteed. To address this, we formulate an optimization problem to tune the PID gains such that total energy consumption is minimized while maintaining accurate LET tracking. This approach drives the realized energy as close as possible to the theoretical optimum $\mathbb{E}^*_{\mathbb{M}}$, even though the closed-loop control inputs differ from $U^*(t)$. This difference can be observed in the control inputs generated by the PID controllers and the optimal input $U^*(t)$ as illustrated in Fig. 4

Remark 4: It can be noted that using controller gains tuned for a different mission may not necessarily lead to energy savings. For example, applying $\mathbb{K}^*_{\mathbb{M}_4}$ during $\mathbb{M}_2$ leads to $\mathbb{E}_{\mathbb{M}_2}(\mathbb{K}^*_{\mathbb{M}_4}) = 8027.45$ J, which is 16.14% higher than $\mathbb{E}_{\mathbb{M}_2}(\mathbb{K}^*_{\mathbb{M}_2})$. This result highlight the importance of mission specific gain tuning for achieving energy efficient flight, rather than relying on a single set of controller gains across all missions.

B. Tuning controllers with existing techniques

PID controllers, in general, are tuned using step inputs as reference signals [11]. Reproducing the tuning approach described in [11], we tune the PID controllers, with the resulting gain sets denoted by $\mathbb{K}^s_{\mathbb{M}}$. Different from our proposed method, this method involves a one-time tuning of all controller gains, which are then used across all missions. Also, the optimization function used for tuning is weighed sum of tracking error and controller effort. Table III compares the energy consumption across four missions using our proposition and the method in [11]. Our approach achieves significant energy savings which ranges from $22\% - 69\%$, i.e an average savings of 34.60% based on the chosen missions. This improvement stems from key differences in the tuning strategy, namely mission-specific tuning, the use of an objective function focused solely on energy consumption, and the use of optimal states as references during the tuning process. Moreover, to verify that using energy consumption as the objective function leads to effective gain tuning, we also tune the controllers using (4) as optimization function and find the gain set denoted by $\mathbb{K}^e_{\mathbb{M}_i}$. The energy consumed for $\mathbb{M}_2$ is $\mathbb{E}_{\mathbb{M}_2}(\mathbb{K}^e_{\mathbb{M}_2}) = 8611.45$, J, which is 25.13% higher than the optimal value. Similarly, for $\mathbb{M}_4$, $\mathbb{E}_{\mathbb{M}_4}(\mathbb{K}^e_{\mathbb{M}_4}) = 9390.29$, J, which is 20.18% above optimal. In contrast, when energy is directly used as the objective function, the energy consumption is only 7.29% and 10.43% above optimal for $\mathbb{M}_2$ and $\mathbb{M}_4$, respectively.

Mission $\mathbb{M}_i$	Energy with optimal PID gains $\mathbb{E}_{\mathbb{M}_i}(\mathbb{K}^*_{\mathbb{M}_i})$ [J]	Energy with gains tuned with step input $\mathbb{E}_{\mathbb{M}_i}(\mathbb{K}^s_{\mathbb{M}})$ [J]	Error
$\mathbb{M}_1$	8830.50	10820.26	22.53%
$\mathbb{M}_2$	7383.98	9544.44	29.25%
$\mathbb{M}_3$	11469.06	13434.54	17.13%
$\mathbb{M}_4$	8628.81	14622.56	69.46%

TABLE III: Energy comparison of proposed method with [11].

Remark 5: Minimizing tracking error alone can lead the controller to demand larger control inputs, resulting in increased energy consumption, as reflected in the results above. Therefore, energy consumption is selected as the objective function. Since the LETs are already used as reference trajectories, the optimizer searches for controller gains that not only track the LETs but do so with minimum realizable energy.

V. Conclusion

In this paper, we present an energy optimized PID controller tuning strategy for quadrotors. Through numerical simulations, we show that conventional gain tuning techniques are energy inefficient when tracking LETs. We then propose a mission specific controller tuning strategy that uses OCP derived optimal reference signals, instead of standard test signals, and sequentially determines optimized controller gains. The resulting optimized gains minimize realizable energy consumption during closed-loop operation. Simulations across four missions on a benchmark quadrotor show that the method achieves energy consumption within 10% of the theoretical minimum and reduces average energy use by 34% compared to traditional tuning methods. Moreover, energy aware tuning provides up to 16% energy savings compared to tuning based on tracking error minimization.

References

[1] Y. Mulgaonkar, M. Whitzer, B. Morgan, C. M. Kroninger, A. M. Harrington, and V. Kumar, "Power and weight considerations in small, agile quadrotors," in *Micro-and Nanotechnology Sensors, Systems, and Applications VI*, vol. 9083, 2014, pp. 376–391.

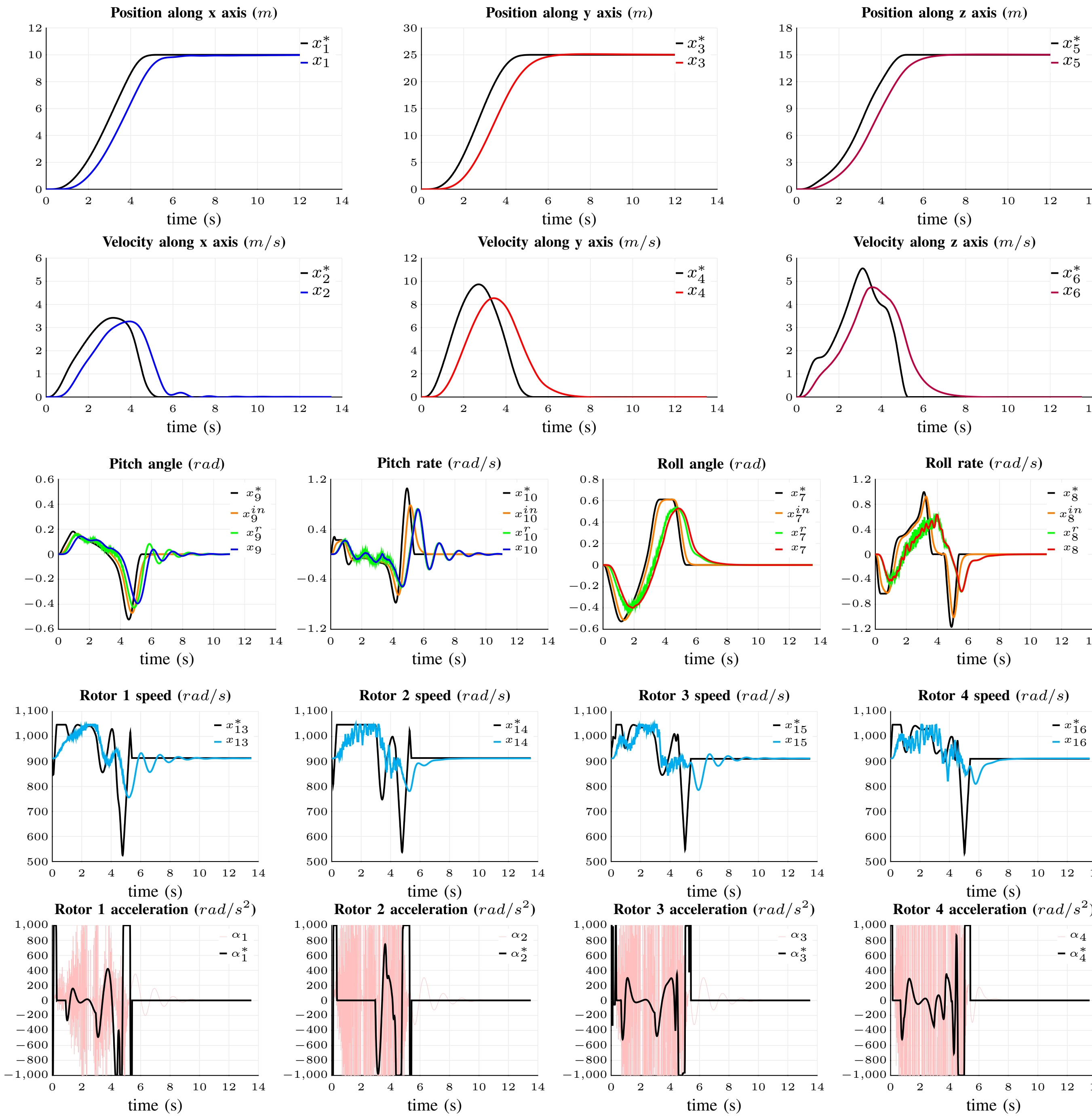

Fig. 4: Response of tuning a Mission $\mathbb{M}_4$ using the proposed tuning method

[2] B. Mubdir and E. Prempain, "Energy-efficient trajectory optimization with nonlinear model predictive control for a quadrotor UAV," in *2024 European Control Conference (ECC)*, 2024, pp. 3576–3581.

[3] P. Ganchaudhuri and C. Bhawal, "Least energy trajectory generation for quadrotors using rotor acceleration as control inputs," in *2025 11th International Conference on Control, Decision and Information Technologies (CoDIT)*, vol. 1, 2025, pp. 2928–2933.

[4] F. Morbidi, R. Cano, and D. Lara, "Minimum-energy path generation for a quadrotor UAV," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1492–1498.

[5] M. Jacewicz, M. Zuagaj, R. Glebocki, and P. Bibik, "Quadrotor model for energy consumption analysis," *Energies*, vol. 15, no. 19, 2022.

[6] P. Ganchaudhuri and C. Bhawal, "Mission specific PID tuning for energy-efficient quadrotor flights," *Aerospace Science and Technology*, vol. 169, p. 111273, 2026.

[7] B. Li, Q. Li, Y. Zeng, Y. Rong, and R. Zhang, "3D trajectory optimization for energy-efficient UAV communication: A control design perspective," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 4579–4593, 2021.

[8] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6235–6240.

[9] S. Bazzocchi, S. Warwick, and A. Suleman, "Automatic autopilot tuning framework using genetic algorithms and system identification," *Aerospace Science and Technology*, vol. 157, p. 109779, 2025.

[10] S. Ben Abdi, A. Debilou, L. Guettal, and A. Guergazi, "Robust trajectory tracking control of a quadrotor under external disturbances and dynamic parameter uncertainties using a hybrid P-PID controller tuned with ant colony optimization," *Aerospace Science and Technology*, vol. 160, p. 110053, 2025.

[11] R. Miranda-Colorado, L. T. Aguilar, and J. E. Herrero-Brito, "Reduction of power consumption on quadrotor vehicles via trajectory design and a controller-gains tuning stage," *Aerospace Science and Technology*, vol. 78, pp. 280–296, 2018.