Disturbance Observer-Based Control for Trajectory Tracking of a Quadrotor Subject to Uncertainties [⋆]

**Paulo V. G. Simplício [*] João R. S. Benevides [*] Marlon A. D. Paiva [*]
Roberto S. Inoue [**] Marco H. Terra [*]**

[*] *Department of Electrical and Computer Engineering, São Carlos School of Engineering, University of São Paulo, São Carlos, São Paulo 13566-590, Brazil (e-mails: paulogalvao@usp.br, jrsbenevides@usp.br, dalsochio@usp.br, terra@sc.usp.br)*
[**] *Department of Computer Sciences, Federal University of São Carlos, São Carlos, São Paulo 13565-905, Brazil, (e-mail: rsinoue@ufscar.br)*

Abstract: Applications involving unmanned aerial vehicles (UAVs) have grown over the past decade. With the advent of technology, especially in sensors, batteries, and actuators, using this type of aircraft has proven cheaper than other solutions. However, it is necessary to guarantee safety and robustness in autonomous flight. In this sense, this paper implements a disturbance observer-based control (DOBC) architecture by combining a robust recursive linear-quadratic regulator with two distinct disturbance estimators. In addition, we tested three other controllers widely used in the literature inside the architecture. We perform practical experiments with a commercial quadrotor and analyze the effects of the parametric variations during flight. We compare the flight performance among the implemented controllers and analyze the performance of each of them with each disturbance estimator. The experimental results show the performance improvement of using a robust controller and a DOBC architecture in a system with parametric variations.

Keywords: Disturbance estimation, parametric variations, robust control, trajectory tracking.

1. INTRODUCTION

Techniques related to controlling autonomous unmanned aerial vehicles (UAVs) have grown over the past decade. One of the main reasons is to increase the safety of using this type of aircraft in several applications, such as precision agriculture (Duggal et al., 2016), disaster management (Mohd Daud et al., 2022), and geographic mapping (Sivakumar and Naga Malleswari, 2021). Inside the UAVs category are the quadrotors, known for their maneuverability and the ability to take off and land in tight spaces. However, controlling this type of aircraft when subject to disturbances and uncertainties is challenging (Rodríguez-Mata et al., 2018). Since the system is underactuated, the mathematical model of aircraft is subject to uncertainties due to the impossibility of considering all the quadrotor dynamics in the modeling. Besides that, external disturbance needs an extra observer algorithm to estimate it.

In this sense, several control techniques have emerged to improve the performance of a quadrotor subject to parametric uncertainties. In Zou and Zhu (2017), a control architecture for a quadrotor subject to parametric uncertainties due to, mainly, mass variations was proposed. In this case, a hierarchical control strategy was applied to reduce the tracking error, and the simulated results showed that the architecture is robust to internal and external disturbances. In the work of Li et al. (2019), the authors propose an adaptive non-singular terminal sliding mode controller to stabilize the quadrotor subject to parametric uncertainties, like inertial moments and dynamic damping factors. The proposed architecture presented satisfactory results, although it considered only attitude control.

Another way to reduce the influence of disturbance and uncertainties in a system is to use disturbance and uncertainties estimation and attenuation techniques (DUEA). One of the most used is disturbance observer-based control (DOBC), where a master controller and a disturbance estimator are required. Thus, it is possible to design a compensator based on the system dynamics to attenuate or eliminate uncertainties. This architecture has the characteristic that the internal block responsible for disturbance estimation and compensation is not activated in the absence of external disturbances (Li et al., 2014). This is important because the central controller can be designed based on the performance and stability requirements of the system without considering external disturbances. The internal block, on the other hand, is intended only to estimate and reject the external disturbances that affect system performance (Chen et al., 2016). According to Yang et al.

[⋆] The following Brazilian research agencies have supported this work: Coordination of Improvement of Higher Education Personnel (CAPES) under grant 88887.682782/2022-00, São Paulo Research Foundation (FAPESP) under grant 2018/13848-1, 2017/05668-0 and 2014/50851-0, and National Council for Scientific and Technological Development (CNPq) under grant 465755/2014-3 and 421131/2018-7.

ISSN: 2525-8311

DOI: 10.20906/CBA2022/3643

(2017), several methods and algorithms that have emerged to address disturbances in control systems have failed to provide robust and adaptive control action. However, the DOBC technique performs at a level of operation between robust and adaptive control and can be used to address the shortcomings that the design of these types of controllers presents. With DOBC architecture, the system can adapt to the uncertainties rather than directly estimating the uncertain parameters (Li et al., 2014).

Considering this scenario, in order to improve the robustness of a quadrotor in the trajectory tracking task, a robust tracking control architecture was proposed by combining a H_∞ controller with a DOBC in Cheng et al. (2018). In this regard, Ahmed et al. (2020) use a DOBC architecture to combine a standard sliding mode control and a disturbance estimation in order to consider both matched and mismatched disturbances. Simulation results show the performance improvement from the combination of DOBC with a robust controller in the trajectory tracking task.

Nevertheless, few works in the literature address the combination of robust control that handles parametric uncertainties with a DOBC architecture in practical experiments. With this motivation, a combination of a robust recursive regulator and a DOBC architecture was proposed in our previous work (see Benevides et al. (2022)). However, the preliminary results in the cited paper do not analyze the parametric uncertainties in the dynamical model, only the effects of wind disturbances. The reason for this is that we minimize the parametric variations in the model by setting a fixed parameter for the yaw angle to reduce the nonliterary effects in the aircraft. Here, we perform experiments with parametric variations and analyze the nonliterary effects of the model in practical experiments with the proposed architecture.

The rest of this paper is organized as follows. In Section 2, we present the adopted dynamical model of the quadrotor and explain how the variation of yaw angle affects the system. In Section 3, we describe the control architecture implemented with a focus on the robust controller and the disturbance estimators. Section 4 details the platform used and the experimental setup for experiments. In Section 5, we present the practical results with the respective discussions. Section 6 concludes the paper and brings some guidelines for future works.

2. QUADROTOR DYNAMICAL MODEL

A simplified dynamical model of a Parrot$^{\text{TM}}$ quadrotor was presented in Santana et al. (2014) and was used for the development of several control systems using the Parrot Bebop 2.0 (see, e.g., Benevides et al. (2019b) and Simplício et al. (2021)). In this case, to control the quadrotor, we need to send elementary velocity commands $\nu(t) = [u_{\nu_x} \;\; u_{\nu_y} \;\; u_{\nu_z} \;\; u_{\nu_\psi}]$, which correspond to the linear velocities in the x, y, and z axes, in addition to the angular velocity around the z-axis, respectively. Considering parametric uncertainties and external disturbances, we have that the dynamic model is described by

$$\ddot{q} = -(A + \delta A)R_t^T \dot{q} + (B + \delta B)\nu + B_d d, \qquad (1)$$

where $q = [x \;\; y \;\; z \;\; \psi]^T$ is composed by positions and yaw orientation, R_t^T is a rotation matrix with respect to the z-axis, B_d is the matrix that maps the external disturbance (d), and the model matrices are given by

$$A = \begin{bmatrix} \gamma_2 \cos(\psi) & -\gamma_4 \sin(\psi) & 0 & 0 \\ \gamma_2 \sin(\psi) & \gamma_4 \cos(\psi) & 0 & 0 \\ 0 & 0 & \gamma_6 & 0 \\ 0 & 0 & 0 & \gamma_8 \end{bmatrix},$$

$$B = \begin{bmatrix} \gamma_1 \cos(\psi) & -\gamma_3 \sin(\psi) & 0 & 0 \\ \gamma_1 \sin(\psi) & \gamma_3 \cos(\psi) & 0 & 0 \\ 0 & 0 & \gamma_5 & 0 \\ 0 & 0 & 0 & \gamma_7 \end{bmatrix},$$

in which $\gamma_i,\ i \in 1, ..., 8$ are model parameters identified using the least-squares method, as proposed in Benevides et al. (2019a), and δA and δB are parametric uncertainties matrices. The model parameters used in this paper are presented in Section 4. Note that, varying the yaw angle (ψ), we have nonlinearity effects in the model. This parametric variation will be analyzed and discussed for all the implemented controllers in Section 5.

2.1 Error State-Space Model

For the quadrotor to follow a given trajectory, we can rewrite the model in terms of the error between state and reference. Disregarding the uncertainties matrices and defining a new state as $s = [\dot{q}^T \quad q^T]^T = [\dot{x} \;\; \dot{y} \;\; \dot{z} \;\; \dot{\psi} \;\; x \;\; y \;\; z \;\; \psi]^T$ with tracking error as $\tilde{s} = s - s^d$, we can write

$$\underbrace{\begin{bmatrix} \ddot{q} - \ddot{q}^d \\ \dot{q} - \dot{q}^d \end{bmatrix}}_{\dot{\tilde{s}}} = \begin{bmatrix} -AR_t^T & 0 \\ I & 0 \end{bmatrix} \underbrace{\begin{bmatrix} \dot{q} - \dot{q}^d \\ q - q^d \end{bmatrix}}_{\tilde{s}} + \begin{bmatrix} B \\ 0 \end{bmatrix} \nu. \qquad (2)$$

We can add and subtract the term $-AR_t^T \dot{q}^d$ in (2) to obtain

$$\dot{\tilde{s}}(t) = \Lambda \tilde{s}(t) + \Phi u(t), \qquad (3)$$

where

$$\Lambda = \begin{bmatrix} -AR_t^T & 0 \\ I & 0 \end{bmatrix}, \quad \Phi = \begin{bmatrix} I \\ 0 \end{bmatrix},$$

and $u = -AR_t^T \dot{q}^d - \ddot{q}^d + B\nu$ is the control law from the controller used to calculate the input vector to be sent to the quadrotor as $\nu = B^{-1}(u + \ddot{q}^d + AR_t^T \dot{q}^d)$.

3. CONTROL DESIGN WITH DOBC ARCHITECTURE

The DOBC architecture is one of the most used for disturbance attenuation in several systems and processes. According to Chen et al. (2016), this technique is used to estimate not only external disturbances that affect a system but also uncertainties present in the control system, such as parameter perturbations and unmodeled dynamics. In order to develop the DOBC architecture, it is necessary to design the central controller and a disturbance estimator. In addition, we need to develop a compensator to attenuate or eliminate the uncertainties that affect the system. In our approach, we use an RLQR controller proposed by Terra et al. (2014) and combine it first with a standard disturbance observer (DOB) and then with a standard Kalman filter (KF) as a disturbance observer. In addition, to analyze the proposed architecture results, recall that we tested the LQR, FL, and PID controller inside the architecture.

3.1 Robust Linear-Quadratic Regulator (RLQR)

The robust linear-quadratic regulator developed in Terra et al. (2014) is useful for discrete-time linear systems subject to parametric uncertainties in real-time applications. Consider the following system with uncertainties

$$x_{k+1} = (F_k + \delta F_k)x_k + (G_k + \delta G_k)u_k; \quad k = 0, ..., N, \tag{4}$$

where $x_k \in \mathbb{R}^n$ is the state vector, $u_k \in \mathbb{R}^m$ is the control input, $F_k \in \mathbb{R}^{n\times n}$ and $G_k \in \mathbb{R}^{n\times m}$ are nominal parameter matrices of the system, and N defining the number of interactions. x_0 is the initial state, constant and known, and the matrices $\delta F_k \in \mathbb{R}^{n\times n}$ and $\delta G_k \in \mathbb{R}^{n\times m}$ are uncertainty matrices, modeled as

$$[\delta F_k \quad \delta G_k] = H_k \Delta_k [E_{F_k} \quad E_{G_k}]; \quad k = 0, ..., N, \tag{5}$$

being $H_k \in \mathbb{R}^{nxk}$, $E_{F_k} \in \mathbb{R}^{n\times n}$, $E_{G_k} \in \mathbb{R}^{n\times n}$ known matrices. $\Delta_k \in \mathbb{R}^{k\times l}$ is an arbitrary matrix $\|\Delta_k\| \leq 1$. Given this, we can obtain the RLQR controller solving the optimization problem:

$$\min_{x_{k+1}, u_k} \max_{\delta F_k, \delta G_k} \left\{ \tilde{J}_k^{\mu}(x_{k+1}, u_k, \delta F_k, \delta G_k) \right\}, \tag{6}$$

where J_i^u is the regularized quadratic cost function, defined as

$$\tilde{J}_k^{\mu}(x_{k+1}, u_k, \delta F_k, \delta G_k) =$$

$$\begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix}^T \begin{bmatrix} P_{k+1} & 0 \\ 0 & R_k \end{bmatrix} \begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} + \Xi^T \begin{bmatrix} Q_k & 0 \\ 0 & \mu I \end{bmatrix} \Xi, \tag{7}$$

where $P_{N+1} \succ 0$, $Q \succ 0$, and $R \succ 0$ are known matrices and $\mu > 0$ is a fixed penalty parameter, responsible for ensuring that the equality of Eq. 4 holds, and

$$\Xi = \left\{ \begin{bmatrix} 0 & 0 \\ I & -G_k - \delta G_k \end{bmatrix} \begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} - \begin{bmatrix} -I \\ F_k + \delta F_k \end{bmatrix} x_k \right\}.$$

From the solution $(x_{k+1}^T(\mu), u_k^*(\mu))$ of the $min - max$ optimization problem, we can find the optimal solution recursively, and when the penalty parameter $\mu \to \infty$, the robustness of the controller is achieved.

In addition to the RLQR controller, we performed practical experiments with LQR, FL, and PID controllers. When the parametric uncertainties are not considered, the RLQR falls into the standard recursive LQR. In other words, they have the same structure. Regarding the PID and FL controller, considering that they are strongly consolidated in the literature and we use them only for comparison purposes, we are not giving details about these controllers. More details about PID and FL controllers can be found in Atrom (1995) and Sastry (2013), respectively.

3.2 Disturbance observer (DOB)

In order to estimate disturbances and model discrepancies, we use a standard disturbance observer for discrete-time linear systems. By discretizing (1) and ignoring the parametric uncertainties of the system, which will be considered in the robust control step, the DOB can be described by:

$$\begin{cases} z_{k+1} = -LG_d\left(z_k + Lq_{k+1}\right) - L\left(\Theta_k + \Gamma_k \nu_k\right), \\ \hat{d}_k = z_k + Lq_{k+1}, \end{cases} \tag{8}$$

where z is an internal auxiliary vector, $\hat{d}$ is the disturbance estimation, L is the observer gain, and $\Theta_k = I + TA(t)$, $\Gamma_k = TB$ and $G_d = TB_d$ are the discrete form of (1) obtained using the Euler's method.

3.3 Kalman filter as a disturbance observer

As shown in Benevides et al. (2022), the KF can be used to estimate external disturbances and uncertainties. In this sense, we can consider $d_a \equiv d$, $d_b = \dot{d}_a$ and $\dot{d}_b = 0$ to create an augmented state space as $\zeta_k = [q_{k+1}^T \;\; d_{a_k}^T \;\; d_{b_k}^T]^T$, obtaining the system:

$$\zeta_{k+1} = \begin{bmatrix} \Theta_k & G_{d,k} & 0 \\ 0 & I & TI \\ 0 & 0 & I \end{bmatrix} \zeta_k + \begin{bmatrix} \Gamma_k \\ 0 \\ 0 \end{bmatrix} u_k + \begin{bmatrix} \mathrm{w}_{a_k} \\ \mathrm{w}_{b_k} \end{bmatrix}, \tag{9}$$

with the output equation as $y_k = H\zeta_k + \mathrm{v}_k$. Where $H = [1 \;\; 0 \;\; 0]$, and w and v are the process and observation noise, respectively. With this augmented system, the KF is applied to provide disturbance estimation.

3.4 Disturbance Compensation

In Benevides et al. (2022), a compensator is proposed to deal with unmodeled uncertainties. In this case, the control law becomes:

$$u_{com,k} = u_{rlqr,k} + u_{ff,k}^{pd}, \tag{10}$$

where $u_{rlqr,k} = -K_{rlqr,k} x_k$ is the RLQR control law, $u_{ff,k}^{pd} = \varphi K_{com,k} \widehat{d_k} + \varrho \hat{d}_{k+1}$ is the disturbance law with φ and ϱ treated as project parameters, and $K_{com,k} = -\left[C\Lambda_k^{-1} G_k\right]^{-1} C\Lambda_k^{-1} G_{d,k}$ is the disturbance gain obtained based on the system dynamics.

4. EXPERIMENTAL SETUP

This section describes the materials, platform, and methods used to develop the project. The quadrotor used in the experiments was the Parrot Bebop 2.0, communicated through the Robot Operating System (ROS) via Wi-Fi connection. We used a Vicon motion capture system to provide the pose of the quadrotor with high accuracy. In addition, we implemented a standard Kalman filter for velocity estimation. The hardware used to develop the control system and perform the experiments is an Intel Core i5 8250U with four cores of 1.6GHz and 8 GB of RAM. Fig. 1 shows the experimental setup diagram used in the experiments.

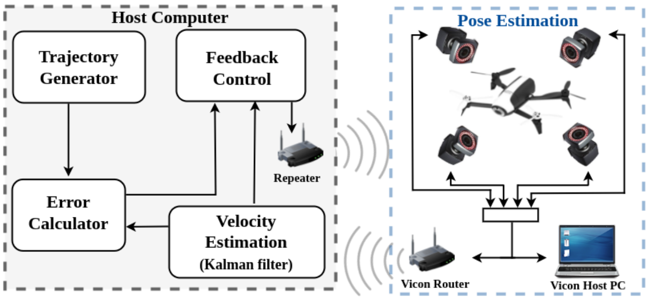

Figure 1. Block diagram of experimental Setup.

4.1 Parrot Bebop 2.0

The Parrot Bebop 2.0 quadrotor has many features that make it attractive in scientific research. It has a Software Development Kit (SDK) available and the integration facility with ROS through the `bebop_autonomy` package, which reduces the time during software development. In addition, it is an affordable quadrotor with a range of sensors, making it feasible for trajectory tracking tasks.

4.2 ROS Platform

ROS, as presented in Quigley et al. (2009), is an open-source platform with a series of tools that ease the programming of robots. The ROS works by integrating different nodes corresponding to sensors or actuators that carry information about the system. Nodes can communicate as easily by posting and subscribing to topics. In this work, we use the ROS system to implement the complete control architecture and the trajectory generator using the C++ language.

4.3 Desired Trajectory

In order to analyze the performance of the control architecture with parametric variations, we send a setpoint for position and orientation to the quadrotor. Considering the vector $q^d = [q_x^d \ q_y^d \ q_z^d \ q_\psi^d]^T$, we have:

$$q^d = [1 \ 0 \ 0 \ 0]^T ,$$

where, $\psi_0 \geq 0.6$ rad, and x_0, y_0, z_0 initialize with 0. From the model presented (1), we see that variation of ψ during flight highlights the nonlinear feature of the dynamic model. Note that, we send the desired trajectory concerning the local (body) coordinate frame ($q_{b_0} = [0 \ 0 \ 0 \ 0.6]^T$), so even starting z at the height of one meter with respect to the global coordinate system ($q_{g_0} = [0 \ 0 \ 1 \ 0.6]^T$), it is zero for the local coordinate system of the quadrotor. So in this sense, to keep this altitude, the desired trajectory is zero concerning the z-axis.

4.4 Pose Estimation

To estimate pose, we use the Vicon motion capture system. This system is composed of four cameras and uses infrared sensing technology to estimate the pose of objects. The Vicon system uses specific software that transmits the pose information through Wi-Fi to the control system's notebook. Using the Vicon Bridge ROS package, the pose information is available in the topic `/Vicon/bebop/bebop`. This topic is accessed by the control node, which uses accurate information to calculate the trajectory error.

4.5 Kalman Filter for Velocity Estimation

Since the Vicon Motion Capture System does not provide velocity information, we implemented a standard Kalman filter for this task as proposed in Kim et al. (2011). We use the KF instead of simple differentiation methods since the filter can suppress the large spikes and improve the control system's performance. For this purpose, we select the vector $q = [q_p^T \ \dot{q}_p^T]^T$, where $q_p = [x \ y \ z]^T$.

4.6 Model and Controller Parameters

This section presents the model parameters and the gains used for all implemented controllers and estimators. The model parameters were identified using the least-square method, which gives $[\gamma_1 \ \ldots \ \gamma_8]^T = [4.5752 \quad 0.1149 \quad 5.2296 \quad 0.2364 \quad 4.4206 \quad 3.1341 \quad 5.928 \quad 9 \quad -0.3868]^T$. For the LQR, we use the Bryson rule as a base, and for PID, we base it on the Ziegler-Nichols critical gain method. The gains for FL and RLQR controllers were heuristically obtained from several experimental flights. All sets of gains are tuned to the quadrotor to achieve the best tracking performance.

For LQR, we have: $Q = diag(1,1,1,1,1,1,10,5)$ and $R = 0.15I_{4\times4}$. For the compensator with DOB, we use: $\phi_{lqr}^{do} = 0.7$ and $\varrho_{lqr}^{do} = 0.95$. Moreover, for the compensator with KF, we use $\phi_{lqr}^{kf} = 0.2$ and $\varrho_{lqr}^{kf} = 0.0001$.

For PID, we have $K_p = diag\,(0.70,\ 0.70,\ 2.75,\ 1.20)$, $K_d = diag\,(0.75,\ 0.75,\ 0.75,\ 3.35)$, and $K_i = 0.01I_{4\times4}$. For the compensator with DOB, we use: $\phi_{pid}^{do} = 0.07$ and $\varrho_{pid}^{do} = 0.2$. Moreover, for the compensator with KF, we use $\phi_{pid}^{kf} = 0.08$ and $\varrho_{pid}^{kf} = 0.00001$.

For FL, we have: $K_1 = diag\,(1.33,\ 1.40,\ 4.50,\ 1.75)$ and $K_2 = diag\,(1.50,\ 1.50,\ 2.00,\ 2.00)$. For the compensator with DOB, we use: $\phi_{fl}^{do} = \varrho_{fl}^{do} = 0.005$. For the compensator with KF, we use $\phi_{fl}^{kf} = 0.001$ and $\varrho_{fl}^{kf} = 0.0001$.

For RLQR, we have: $Q = diag\,(1,1,1,1,1,1,10,5)$, $R = 0.15I_{4\times4}$, and uncertainties matrices $E_{f1} = 0.295I_{4\times4}$, $E_{f2} = diag\,(0.285,\ 0.285,\ 2.85,\ 1.425)$, $E_g = 0.01875I_{4\times4}$. For the compensator with DOB, we use $\phi_{rlqr}^{do} = 0.1$ and $\varrho_{rlqr}^{do} = 0.7$. Moreover, for the compensator with KF, we use $\phi_{rlqr}^{kf} = 0.04$ and $\varrho_{rlqr}^{kf} = 0.0025$.

For DOB with RLQR and LQR controllers, we use the gain $L_{regulators} = diag(0.25I_{4\times4}, 0_{4\times4})$ and for PID and FL controllers, we use $L_{pid-fl} = 0.25I_{4\times4}$. For the KF as disturbance estimator, we have: $Q = diag(0.1I_{4\times4}, I_{4\times4}, 10I_{4\times4})$ and $R = 0.00035I_{4\times4}$.

4.7 Evaluation Metrics

To analyze the performance of the controllers with and without the use of the DOBC architecture, we use the Euclidean norm for the position and orientation errors and control inputs. We summarise all the equations in Table 1, where q_p corresponds to the controlled position variables (x, y, z), j to the experiment index for n experiments performed, k defines each iteration and N is the total number of iterations in each experiment.

Table 1. Metrics for system evaluation.

Metrics	Equations
Position Errors	$E_{q_p}(k) = \frac{1}{n}\sum_{j=1}^{n} \lVert q_{p_k}^{(j)} - q_{p_k}^{d} \rVert$
	$\bar{E}_{q_p} = \sum_{k}^{N} E_{q_p}(k)$
Orientation Errors	$E_{q_\psi}(k) = \frac{1}{n}\sum_{j=1}^{n} \lVert q_{\psi_k}^{(j)} - q_{\psi_k}^{d} \rVert$
	$\bar{E}_{q_\psi} = \sum_{k}^{N} E_{q_\psi}(k)$
Control Input	$V(k) = \frac{1}{n}\sum_{j=1}^{n} \lVert \nu_k^{(j)} \rVert$
	$\bar{V} = \sum_{k}^{N} \frac{1}{n}\sum_{j=1}^{n} \lVert \nu_k^{(j)} \rVert_1$

5. EXPERIMENTAL RESULTS

In this section, we analyze the performance of the controllers under nonlinear effects by the displacement of the controlled variable ψ, that produces the nonlinear effects in the aircraft. For the tests, we started the flight with $\psi \neq 0$ rad and then activated the trajectory tracking control, which was maintained at a fixed point in space. In other words, we used a setpoint for position and orientation (Subsection 4.3). Next, we analyze the flight performance of all controllers in three different cases: the standalone controllers and the controllers with the DOBC architecture using DOB and the KF as disturbance observers.

Fig. 2 presents the norms of tracking errors and control efforts using the standalone controllers, i.e., without the DOBC architecture. By evaluating the curves presented in Fig. 2(a) for the position errors and the numerical indices shown in Table 2, we notice that the best tracking performance is achieved with the RLQR controller. Although this regulator is not designed explicitly for nonlinear systems, it shows a fast response time and low position error ($\bar{E}_{q_p}$). The FL controller, widely used in nonlinear systems, presented a good performance. Concerning orientation tracking (Fig. 2(b) and index $\bar{E}_{q_\psi}$), we also observe a significantly better performance of the RLQR controller over the standard LQR, FL, and PID controllers. By analyzing the control effort indices (Fig. 2(c) and index $\bar{V}$), we observe the best performances (lowest power consumption) in the standard LQR and the FL, respectively. We highlight the significant-high consumption and saturation of the control inputs with the PID, which explains the oscillations observed in the controlled variables (Fig. 5(a)).

Table 2. Performance indices for the standalone controllers.

Index	RLQR	LQR	FL	PID
$\bar{E}_{q_p}(mm)$	6.9560	15.5832	9.1287	9.4519
$\bar{E}_{q_\psi}(rad)$	3.1612	6.7025	6.8466	8.6626
$\bar{V}$	224.3950	119.8454	121.6101	868.8810

Fig. 3 shows the norms of tracking errors and control efforts using the DOBC architecture along with the DOB. By evaluating the curves presented in Fig. 3(a) for the position error and the numerical performance indices presented in Table 3, we notice that the RLQR outperforms the implemented standard controllers. The FL presented a low position error and the PID an initial fast response. Regarding the $\bar{E}_{q_p}$ index, the PID controller presented the second-best performance due to its fast response time.

Regarding the orientation tracking errors through $\bar{E}_{q_\psi}$ index and Fig. 3(b), the RLQR controller again presents a significantly superior performance, especially at the

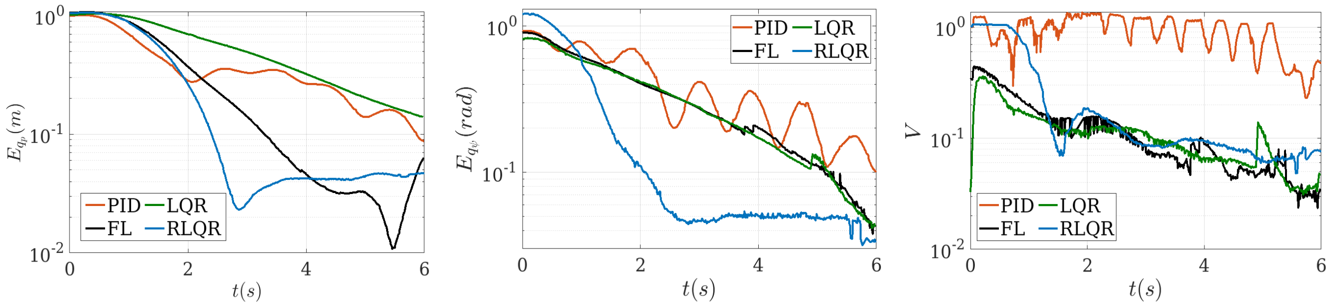

Figure 2. Flight performed with the implemented controllers. (a) Position tracking errors. (b) Orientation tracking errors. (c) Control efforts.

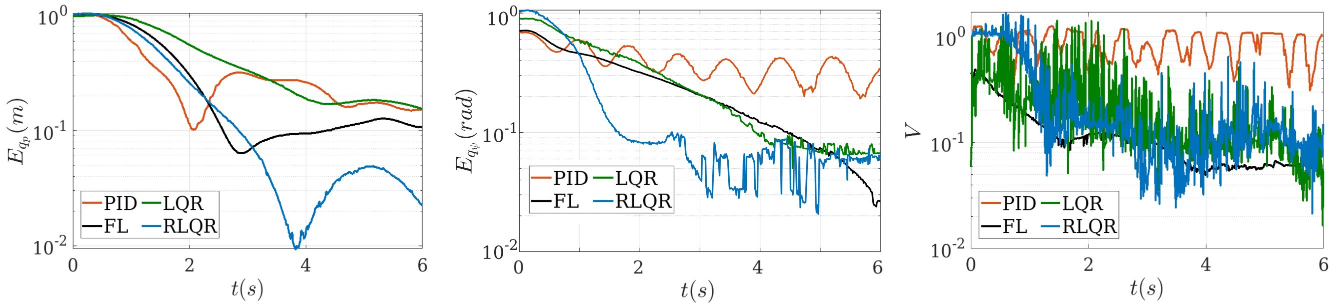

Figure 3. Flight performed using the DOBC architecture along with the DOB. (a) Position tracking errors. (b) Orientation tracking errors. (c) Control efforts.

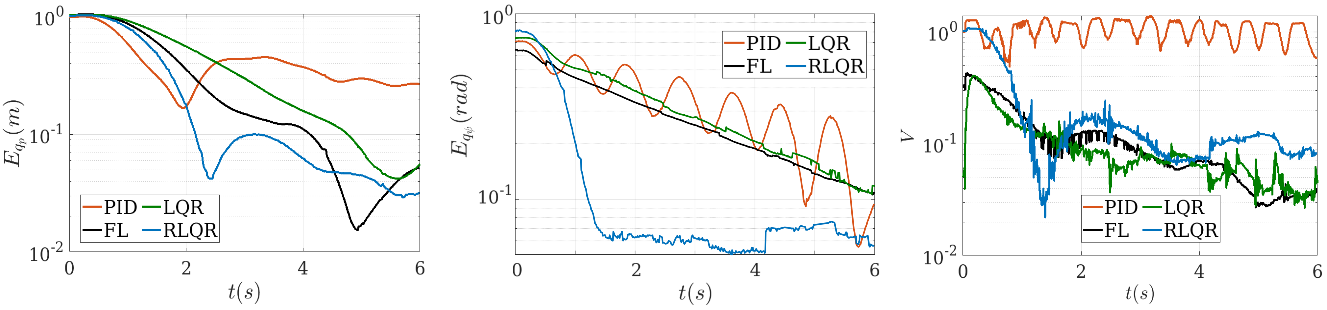

Figure 4. Flight performed using the DOBC architecture along with the KF as a observer. (a) Position tracking errors. (b) Orientation tracking errors. (c) Control efforts.

Table 3. Performance indices using DOB.

Index	RLQR	LQR	FL	PID
$\bar{E}_{q_p}(mm)$	7.0372	13.0426	8.0957	7.7159
$\bar{E}_{q_\psi}(rad)$	2.4166	3.8065	3.8626	8.3151
$\bar{V}$	255.0979	236.1085	124.9680	799.1364

beginning of the flight. However, after $3s$ we notice a performance increase on the standard LQR followed by the FL. Furthermore, analyzing the control efforts (Fig. 3(c) and $\bar{V}$ index), the lowest consumption is given by the FL controller. We observe that the control signals with more significant noise were presented by the RLQR and LQR controllers, with similar performance indices. Finally, the PID controller again presented a higher consumption and saturation, which explains the oscillations presented in the controlled variables (see Fig. 5(b)).

In Fig. 4, we show the norms of tracking errors and control efforts using the DOBC architecture along with the KF as a disturbance observer. By evaluating the curves presented in Fig. 4(a) for the position error and the numerical performance indices presented in Table 4, we observe that the RLQR gives the best performance during the beginning of the flight. The FL and standard LQR controllers present similar and even better performance than RLQR after $5s$. The PID controller presents a lower position error ($\bar{E}_{q_p}$) than the standard LQR due to the fast response time in the first few seconds. Regarding the orientation tracking errors (Fig. 4(b) and $\bar{E}_{q_\psi}$ index), we again highlight the performance of the RLQR. Furthermore, by analyzing the control efforts on Fig. 4(c) and indices $\bar{V}$, we notice that the standard LQR and FL controllers gave the best performance results. The PID controller, like the previous case, presented a high consumption and saturation, resulting in large oscillations in the controlled variables along the time (see Fig. 5(c)).

Table 4. Performance indices using the KF.

Index	RLQR	LQR	FL	PID
$\bar{E}_{q_p}(mm)$	6.2342	12.3377	8.8183	10.2801
$\bar{E}_{q_\psi}(rad)$	1.5833	6.4998	5.7694	7.4507
$\bar{V}$	193.8364	106.0667	118.0160	992.3958

Regarding the use of the DOBC architecture, by analyzing Table 5, we observe a significant improvement in the trajectory tracking performance ($\bar{E}_{q_p}$) on most of the controllers. Only the RLQR had a slightly inferior performance regarding the position tracking on the x-axis when the DOB is used. The PID controller also presented a relatively low performance on the x-axis when using the KF. Regarding the control efforts ($\bar{V}$), there was a performance increase in three out of the four controllers when using the KF, where only the PID had an inferior performance. The opposite behavior was observed when using the DOB.

6. CONCLUSIONS

We implemented and analyzed a DOBC architecture with a robust recursive regulator and disturbance estimators for

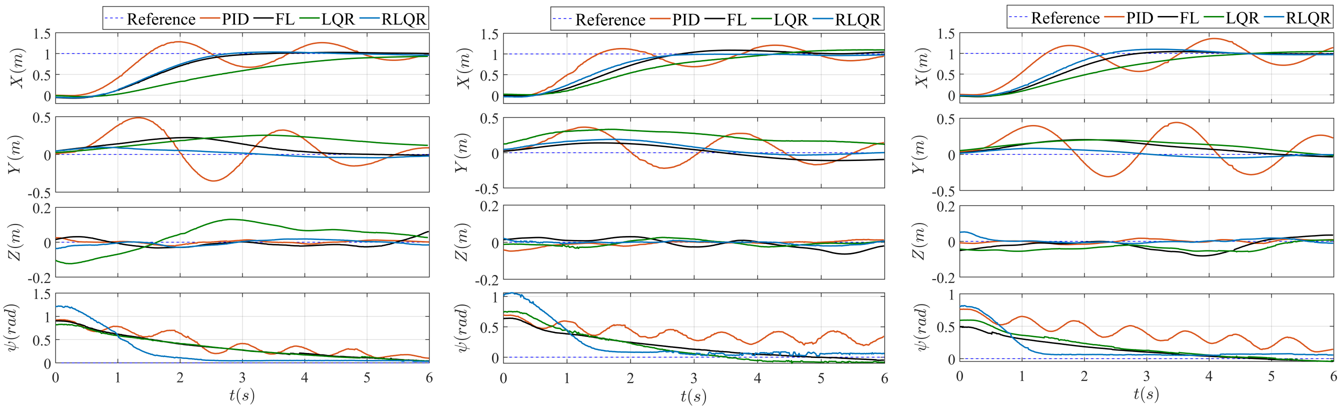

Figure 5. Controlled variables with parametric variations. (a) Standalone controllers. (b) DOBC architecture with DOB. (c) DOBC architecture with KF.

Table 5. Percentage Improvement using the DOBC architecture over standalone controllers.

Controller	**Percentage Improvement**					
	DOBC (With DOB)			**DOBC** (With Kalman filter)		
	$\bar{E}_{q_p}(mm)$	$\bar{E}_{q_\psi}(rad)$	$\bar{V}$	$\bar{E}_{q_p}(mm)$	$\bar{E}_{q_\psi}(rad)$	$\bar{V}$
RLQR	-1.17%	23.55%	-13.68%	10.36%	49.91%	13.62%
LQR	16.30%	43.21%	-97.99%	20.30%	3.02%	11.50%
FL	11.32%	43.58%	-2.76%	3.40%	15.73%	2.95%
PID	18.37%	4.01%	8.03%	-8.76%	13.99%	-14.22%

trajectory tracking of a quadrotor subject to parametric uncertainties. We used a ROS platform to integrate the algorithms and communicate with the quadrotor and an optical motion capture system. We performed practical experiments and a comparative study with three other standard controllers widely used in the literature. We analyzed the flight performance inside the DOBC architecture for each implemented controller with two distinct disturbance observers. The experimental results showed the effectiveness of combining controllers, robust or not, with a DOBC architecture to reduce the effects of the parametric variations during the flight. Future works aim to test and analyze the quadrotor flight performance with the proposed architecture in outdoor scenarios.

REFERENCES

Ahmed, N., Chen, M., and Shao, S. (2020). Disturbance observer based tracking control of quadrotor with high-order disturbances. *IEEE Access*, 8, 8300–8313.

Atrom, K.J. (1995). *PID Controllers: Theory, Design and Tuning*. ISA - The Instrumentation, Systems and Automation Society, North Carolina, USA.

Benevides, J.R.S., Inoue, R.S., Paiva, M.A.D.S., and Terra, M.H. (2019a). Parameter Estimation Based on Linear Regression for Commercial Quadrotors. *Anais do 14º Simpósio Brasileiro de Automação Inteligente*.

Benevides, J.R., Inoue, R.S., Paiva, M.A., and Terra, M.H. (2019b). ROS-based robust and recursive optimal control of commercial quadrotors. *IEEE International Conference on Automation Science and Engineering*, 998–1003.

Benevides, J.R.S., Paiva, M.A.D., Simplício, P.V.G., Inoue, R.S., and Terra, M.H. (2022). Disturbance observer-based robust control of a quadrotor subject to parametric uncertainties and wind disturbance. *IEEE Access*, 10, 7554–7565.

Chen, W.H., Yang, J., Guo, L., and Li, S. (2016). Disturbance-Observer-Based Control and Related Methods - An Overview. *IEEE Transactions on Industrial Electronics*, 63(2), 1083–1095.

Cheng, Y., Jiang, L., Li, T., and Guo, L. (2018). Robust tracking control for a quadrotor uav via DOBC approach. In *2018 Chinese Control And Decision Conference (CCDC)*, 559–563.

Duggal, V., Sukhwani, M., Bipin, K., Reddy, G.S., and Krishna, K.M. (2016). Plantation monitoring and yield estimation using autonomous quadcopter for precision agriculture. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 5121–5127. doi: 10.1109/ICRA.2016.7487716.

Kim, P., Kim, P., and Huh, L. (2011). *Kalman Filter for Beginners: With MATLAB Examples*. CreateSpace Independent Publishing Platform.

Li, S., Yang, J., Chen, W.H., and Chen, X. (2014). *Disturbance observer-based control: methods and applications*. CRC press, Boca Raton, FL, USA, 1 edition.

Li, Z., Ma, X., and Li, Y. (2019). Finite-time attitude stabilization control of a quadrotor with parametric uncertainties and disturbances. *IEEE International Conference on Robotics and Biomimetics*, 2654–2659.

Mohd Daud, S.M.S., Mohd Yusof, M.Y.P., Heo, C.C., Khoo, L.S., Chainchel Singh, M.K., Mahmood, M.S., and Nawawi, H. (2022). Applications of drone in disaster management: A scoping review. *Science & Justice*, 62(1), 30–42.

Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A.Y. (2009). Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*.

Rodríguez-Mata, A.E., González-Hernández, I., Rangel-Peraza, J.G., Salazar, S., and Leal, R.L. (2018). Wind-gust Compensation Algorithm based on High-gain Residual Observer to Control a Quadrotor Aircraft: Real-time Verification Task at Fixed Point. *International Journal of Control, Automation and Systems*, 16(2), 856–866. doi:10.1007/s12555-016-0771-6.

Santana, L.V., Brandao, A.S., Sarcinelli-Filho, M., and Carelli, R. (2014). A trajectory tracking and 3D positioning controller for the AR.Drone quadrotor. *International Conference on Unmanned Aircraft Systems (ICUAS)*, 756–767.

Sastry, S. (2013). *Nonlinear Systems: Analysis, Stability, and Control*. Interdisciplinary Applied Mathematics. Springer New York.

Simplício, P.V.G., Benevides, J.R., Inoue, R.S., and Terra, M.H. (2021). Rastreamento de Trajetória de um Quadricóptero com Controle Robusto e Rede Neural Profunda. *Anais do 14º Simpósio Brasileiro de Automação Inteligente*, 1–7.

Sivakumar, M. and Naga Malleswari, T.Y. (2021). A literature survey of unmanned aerial vehicle usage for civil applications. *Journal of Aerospace Technology and Management*, 13, 1–23.

Terra, M.H., Cerri, J.P., and Ishihara, J.Y. (2014). Optimal robust linear quadratic regulator for systems subject to uncertainties. *IEEE Transactions on Automatic Control*, 59(9), 2586–2591.

Yang, J., Chen, W.H., Li, S., Guo, L., and Yan, Y. (2017). Disturbance/Uncertainty Estimation and Attenuation Techniques in PMSM Drives - A Survey. *IEEE Transactions on Industrial Electronics*, 64(4), 3273–3285.

Zou, Y. and Zhu, B. (2017). Adaptive trajectory tracking controller for quadrotor systems subject to parametric uncertainties. *Journal of the Franklin Institute*, 354(15), 6724–6746.