

Controle de trajetória do quadrotor AR.Drone 2.0 via abordagens LQR e \mathcal{H}_∞

Rodrigo S. Leonello* Mateus Z. Seixas* Humberto X. de Araújo*
Bernardo Ordoñez*

* Programa de Pós-Graduação em Engenharia Elétrica (PPGEE),
Universidade Federal da Bahia, Salvador, Bahia, Brasil, (e-mails:
rodrigoleonello@gmail.com, mateus.seixas@hotmail.com.br,
humberto.araujo@ufba.br, bordonez@ufba.br).

Abstract: This article presents an experimental study of two control techniques based on the Optimal Control theory, applied to an unmanned aerial vehicle (UAV). The first technique, based on the quadratic cost function, is the LQR controller. The second technique optimizes the norm \mathcal{H}_∞ taking into account the pole allocation and using an approach based on LMIs. The test platform used is the quadrotor AR.Drone 2.0 and the algorithms are implemented in the *Robot Operating System* (ROS).

Resumo: Esse artigo apresenta um estudo experimental de duas técnicas de controle baseadas na teoria de Controle Ótimo, aplicadas a um veículo aéreo não tripulado (VANT). A primeira técnica, baseada no índice de desempenho quadrático, é o controlador LQR. A segunda técnica otimiza a norma \mathcal{H}_∞ levando-se em consideração a alocação de pólos em uma região desejada utilizando-se uma abordagem baseada em LMIs. A plataforma de teste utilizada é o quadrotor AR.Drone 2.0 e os algoritmos são implementados no *Robot Operating System* (ROS).

Keywords: LQR; \mathcal{H}_∞ control; optimal control; quadcopter; model identification; LMI.

Palavras-chaves: LQR; Controle \mathcal{H}_∞ ; controle ótimo; quadrotor; identificação de modelo; LMI.

1. INTRODUÇÃO

Os veículos aéreos não tripulados (VANTs) ou drones são aeronaves que podem ser controladas nos 3 eixos de referência. Dentre os diversos tipos existentes, destacam-se, na categoria de asa rotativa ou hélice, os quadrotores. O uso de quatro rotores ao invés de um único, possibilita rotores menores e hélices protegidas, o que torna o voo mais seguro dentro de ambientes fechados ou com grande quantidade de obstáculos (Raffo, 2011).

Embora tais veículos já venham sendo utilizados para fins militares há algum tempo (Fahlstrom e Gleason, 2012), é notável, nos últimos anos, o crescimento de sua utilização pela população civil, abrindo espaço para o uso em diversas áreas, como na supervisão do tráfego de veículos, no suporte ao gerenciamento de riscos naturais em construções civis, no monitoramento ambiental, no mapeamento geográfico em tarefas de topografia, filmagem e fotografia aéreas etc (Francis, 2019). Tal crescimento estimulou, ao longo dos anos, a criação de modelos menores (Cheng-Ming *et al.*, 2017) e com características específicas para as áreas nas quais são destinados.

No meio acadêmico, por ser uma plataforma interessante de estudo, com característica não-linear, subatuada e instável, tais veículos são frequentemente utilizados para testar sistemas de navegação e controle, como visto em San-

tana *et al.* (2016). Neste, sistemas computacionais foram desenvolvidos para controlar uma formação líder-seguidor entre dois veículos aéreos não tripulados. Além disto, os drones podem ser utilizados para investigar técnicas de fusão sensorial, através de algoritmos que permitem estimar os movimentos do veículo utilizando-se seus próprios sensores embarcados (Engel *et al.*, 2014).

Em outros trabalhos, diversas técnicas de controle são exploradas no controle dos drones. Em Setyawan *et al.* (2019), há uma implementação do controle LQR no procedimento de pouso do quadrotor, suavizando-o para evitar eventuais choques com o solo. Em Dhewa *et al.* (2017) os autores usam a mesma técnica de controle para solucionar o problema de estabilização do quadrotor no estado de *hovering* do veículo, enquanto que em Martins *et al.* (2019) o controlador em questão, foi utilizado para resolver o problema de seguimento de trajetória do VANT nos 3 eixos de referência.

Este trabalho tem como objetivo o controle de trajetória do AR.Drone 2.0, utilizando-se os controladores LQR e \mathcal{H}_∞ , ambos por realimentação de estado. Um modelo linear do piloto automático do AR.Drone 2.0 é estimado e levado em consideração no projeto dos controladores.

2. A PLATAFORMA AR.DRONE 2.0

O VANT adotado no presente trabalho é o modelo AR.Drone 2.0, ilustrado na Figura 1.

* Esse projeto recebeu financiamento da Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB) no âmbito do acordo de concessão BOL0412/2018.



Figura 1. Quadrotor modelo Ar.Drone 2.0 e os sistemas de coordenadas adotados, sendo {w} e {b} os sistemas de coordenadas global e do veículo, respectivamente.

O AR.Drone possui um piloto automático e é dotado de bússola, sensor ultrassônico, sensores inerciais (acelerômetros e giroscópios), câmeras frontal e inferior, com um computador de bordo com um processador Cortex A8 de 1,00 GHz e 1GB de memória RAM, e sistema operacional Linux. Os detalhes técnicos do AR.Drone podem ser encontrados em Piskorski *et al.* (2012).

Segundo Santana *et al.* (2016), embora tenha todos os dispositivos sensoriais citados, o AR.Drone não possui a capacidade de realizar um voo pairado completamente autônomo, pois, embora os procedimentos de decolagem e aterrissagem sejam realizados de forma autônoma, uma vez no ar, o drone começa a "deslizar", devido ao erro acumulado ao longo do tempo no sistema de medição, deslocando-se de sua posição inicial.

2.1 Dados sensoriais do AR.Drone 2.0

É possível, por meio dos protocolos de comunicação disponibilizados em Piskorski *et al.* (2012), requisitar do piloto automático do AR.Drone o resultado do processamento dos sinais sensoriais. Desta forma, o *firmware* do veículo retorna o seguinte conjunto de variáveis:

$$\mathbf{s} = [z \quad v_x \quad v_y \quad v_z \quad \theta \quad \phi \quad \psi], \quad (1)$$

em que z representa a cota de altitude (m) do VANT, relativa ao solo, v_x , v_y e v_z representam as velocidades lineares (m/s) do veículo, em relação aos eixos x_b , y_b e z_b . Por fim, ϕ , θ e ψ representam os ângulos de orientação do veículo (rad) referenciados ao sistema de coordenadas global. As influências das inclinações de ϕ e θ já são tratadas pelo *firmware*. Também é possível requisitar as imagens da câmeras e os valores de latitude e longitude, obtidos do GPS e referenciados no sistema de coordenadas *North, East e Down* (NED).

2.2 Controle de movimento do AR.Drone 2.0

Segundo Krajník *et al.* (2011) e Engel *et al.* (2014), é possível enviar comandos de movimento para o Ar.Drone através de um protocolo específico em sua rede wifi, onde os sinais de comando são normalizados, de forma a representar percentualmente os movimentos desejados. O vetor de sinais de controle normalizados é definido por:

$$\mathbf{u} = [u_\theta \quad u_\phi \quad u_z \quad u_\psi]^T, \quad u_i \in [-1.0, +1.0], \quad (2)$$

em que u_θ e u_ϕ representam comandos de inclinação em relação a y_w e x_w , que indiretamente representam comandos de velocidade linear sobre o eixo x_b e y_b , u_z representa um comando de velocidade linear sobre o eixo z_w e u_ψ representa um comando de velocidade angular em torno do eixo z .

2.3 Modelagem matemática do AR.Drone 2.0

A modelagem de um VANT pode agregar vários níveis de detalhamento, incorporando desde representações do movimento do corpo rígido, até modelos de atuadores e de efeitos aerodinâmicos presentes em voos. Em Bouabdallah (2007), por exemplo, encontra-se uma das representações do modelo dinâmico do veículo mais utilizadas na literatura.

Para o modelo dinâmico do AR.Drone, existe uma outra forma simplificada de modelagem, onde a resposta do veículo aos comandos reais (\mathbf{u}) é regida por equações lineares, substituindo-se as equações tradicionais de modelagem do quadrotor. Tal consideração torna-se possível pela ação do piloto automático, como visto em Krajník *et al.* (2011). A ação em questão garante que os comandos de inclinação enviados para o veículo não afetem sua altitude, possibilitando realizar movimentos laterais e longitudinais com altitude constante.

Dessa forma, têm-se as seguintes equações simplificadas:

$$\begin{cases} \ddot{\theta} = K_\theta w_\theta^2 \theta_{max} u_\theta - 2\zeta_\theta w_\theta \dot{\theta} - w_\theta^2 \theta, \\ \ddot{\phi} = K_\phi w_\phi^2 \phi_{max} u_\phi - 2\zeta_\phi w_\phi \dot{\phi} - w_\phi^2 \phi, \\ \ddot{z} = \left(\frac{K_z \dot{z}_{max}}{\tau_z}\right) u_z - \left(\frac{1}{\tau_z}\right) \dot{z}, \\ \ddot{\psi} = \left(\frac{K_\psi \psi_{max}}{\tau_\psi}\right) u_\psi - \left(\frac{1}{\tau_\psi}\right) \dot{\psi}, \end{cases} \quad (3)$$

em que K_ϕ , K_θ , K_z e K_ψ são interpretados como os ganhos de processo, w_θ e w_ϕ , as frequências naturais, ζ_ϕ e ζ_θ , os coeficientes de amortecimento e τ_z e τ_ψ , as constantes de tempo. Além disto, θ_{max} , ϕ_{max} , \dot{z}_{max} e ψ_{max} representam parâmetros limitadores das orientações de rolagem e arfagem, de velocidade vertical e de guinada, respectivamente.

As equações não lineares que descrevem o comportamento de x e y , considerando-se as forças de arrasto translacional, são dadas por:

$$\begin{cases} \ddot{x} = (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)g - C_x \dot{x}, \\ \ddot{y} = (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)g - C_y \dot{y}, \end{cases} \quad (4)$$

em que C_x e C_y são os coeficientes de arrasto.

Ao configurar o piloto automático para que seja permitido somente pequenos ângulos de inclinação (θ e ϕ) e guinada (ψ) nulo, torna-se possível linearizar as equações vistas em (4), ou seja:

$$\begin{cases} \ddot{x} = g\theta - C_x \dot{x}, \\ \ddot{y} = -g\phi - C_y \dot{y}. \end{cases} \quad (5)$$

Para estimar os valores dos parâmetros das equações (3) e (4), foram realizados experimentos para a identificação do modelo, produzindo os resultados da Tabela 1. Aplicou-se o método dos mínimos quadrados no *toolbox* do Matlab. Foi observado experimentalmente que os valores de K_θ e K_ϕ estão condicionados aos módulos das respectivas entradas

de controle. Para realizar a fusão sensorial e estimação de

Tabela 1. Parâmetros do AR.Drone 2.0.

$K_\theta = \begin{cases} 1,0, & \text{se } u_\theta \geq 0,5 \\ 2,0, & \text{se } u_\theta < 0,5 \end{cases}$	$K_\phi = \begin{cases} 1,0, & \text{se } u_\phi \geq 0,5 \\ 2,0, & \text{se } u_\phi < 0,5 \end{cases}$
$K_z = 1,24$	$K_\psi = 0,91$
$\theta_{max} = 0,21$ [rad]	$\phi_{max} = 0,21$ [rad]
$\dot{z}_{max} = 0,70$ [m/s]	$\dot{\psi}_{max} = 1,74$ [rad/s]
$\omega_\theta = 4,82$ [Hz]	$\omega_\phi = 4,86$ [Hz]
$\zeta_\theta = 0,45$	$\zeta_\phi = 0,49$
$\tau_z = 0,52$ [s]	$\tau_\psi = 0,08$ [s]

estado foi utilizado o Filtro de Kalman Estendido (EKF), implementado por Engel *et al.* (2014). O vetor com os estados disponíveis é o seguinte:

$$\mathbf{x}_t := (x_t, y_t, z_t, \dot{x}_t, \dot{y}_t, \dot{z}_t, \theta_t, \dot{\theta}_t, \phi_t, \dot{\phi}_t, \psi_t, \dot{\psi}_t)^T \in \mathbb{R}^{12}, \quad (6)$$

no qual (x_t, y_t, z_t) representa a posição do quadrotor e $(\dot{x}_t, \dot{y}_t, \dot{z}_t)$, as velocidades, ambos nas coordenadas globais. Além disto, o vetor de estado contém os ângulos θ_t (arfagem), ϕ_t (rolagem) e ψ_t (guinada), assim como a velocidade angular $\dot{\psi}_t$, referenciados à coordenada do corpo.

3. TÉCNICAS DE CONTROLE

Seja o sistema no espaço de estados representado por:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t), \\ y(t) = \mathbf{C}\mathbf{x}(t), \end{cases} \quad (7)$$

em que $\mathbf{x} \in \mathbb{R}^n$ é o vetor de estado, $y \in \mathbb{R}^q$, o sinal de saída, $u \in \mathbb{R}^p$, o sinal de controle e as matrizes \mathbf{A} , \mathbf{B} e \mathbf{C} têm dimensões apropriadas.

Deseja-se obter uma lei de controle por realimentação de estado:

$$u(t) = \mathbf{K}\mathbf{x}(t), \quad \mathbf{K} \in \mathbb{R}^{p \times n}. \quad (8)$$

Neste trabalho, a matriz de ganho \mathbf{K} será projetada utilizando-se como critério de desempenho o índice LQR ou a norma \mathcal{H}_∞ sujeito a restrições de alocação de pólos.

3.1 Controle LQR com Ação Integral

O regulador linear quadrático (LQR) estabelece um compromisso entre o desempenho do sistema e os valores do sinal de controle obtido pela minimização do índice:

$$J_\infty = \int_0^\infty (\mathbf{x}(t)^T \mathbf{Q}\mathbf{x}(t) + u(t)^T \mathbf{R}u(t))dt, \quad (9)$$

no qual $\mathbf{Q} \in \mathbb{R}^{n \times n}$, $\mathbf{Q} \succeq 0$ e $\mathbf{R} \in \mathbb{R}^{n \times n}$, $\mathbf{R} \succ 0$, são matrizes de ponderação.

A lei de controle que minimiza o índice J_∞ é dada por:

$$u(t) = -\mathbf{K}\mathbf{x}(t), \quad \mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}, \quad (10)$$

em que $\mathbf{P} \in \mathbb{R}^{n \times n}$, $\mathbf{P} \succ 0$, é a solução única da equação algébrica de Riccati (ARE):

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = 0. \quad (11)$$

A fim de assegurar erro nulo em regime permanente e rejeitar perturbações constantes, uma nova representação

é definida no espaço de estado, com a inserção da integral do erro, e a lei de controle passa a ser definida por:

$$u(t) = \mathbf{K}\mathbf{x}(t) + k_I\xi(t), \quad (12)$$

$$\dot{\xi}(t) = r(t) - y(t), \quad (13)$$

sendo $r(t) \in \mathbb{R}^q$ o sinal de referência e $\xi \in \mathbb{R}^q$ a integral do sinal de erro.

Consequentemente, tem-se o seguinte vetor de estado para a representação do sistema aumentado:

$$\mathbf{x} := (x, \dot{x}, y, \dot{y}, \theta, \dot{\theta}, \phi, \dot{\phi}, z, \dot{z}, \psi, \dot{\psi}, x_e, y_e, z_e, \psi_e)^T \in \mathbb{R}^{16}, \quad (14)$$

sendo as quatro últimas posições do vetor os respectivos estados adicionais, ou seja, as integrais dos erros em x , y , z e ψ .

3.2 Controle \mathcal{H}_∞ com Ação Integral

Deseja-se encontrar um controlador que minimiza a norma \mathcal{H}_∞ da função de transferência $T_{wz}(s)$ entre a perturbação w e a saída controlada z :

$$\|T_{wz}(s)\|_\infty = \sup_w \bar{\sigma}[T_{wz}(jw)], \quad (15)$$

em que $\bar{\sigma}$ representa o valor singular máximo de $T_{wz}(jw)$.

Seja a representação do sistema linear, considerando-se como estado adicional a integral do erro:

$$S_c : \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}_2u(t) + \mathbf{B}_1w(t), \\ z(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}_2u(t) + \mathbf{D}_1w(t), \end{cases} \quad (16)$$

em que $w \in \mathbb{R}^p$ é o vetor de perturbação e/ou dinâmicas não modeladas e $z \in \mathbb{R}^m$, o vetor de saída controlada. Para o projeto do controlador \mathcal{H}_∞ por realimentação de estado considera-se o vetor de saída $z = [y \quad u]^T$ e $\mathbf{B}_1 = \mathbf{B}_2$.

O controlador \mathcal{H}_∞ pode ser calculado resolvendo-se o seguinte problema convexo de otimização:

$$\min_{W_1, W_2} \gamma \quad (17)$$

sujeito a

$$\begin{bmatrix} \mathbf{A}W_1 + W_1\mathbf{A}^T + \mathbf{B}_2W_2 + W_2^T\mathbf{B}_2^T & \mathbf{B}_1 & W_1\mathbf{C}_1^T + W_2^T\mathbf{D}_2^T \\ & -\gamma I & \mathbf{D}_1^T \\ \mathbf{C}_1W_1 + \mathbf{D}_2W_2 & \mathbf{D}_1 & -\gamma I \end{bmatrix} \prec 0, \quad (18)$$

com $W_1 \in \mathbb{R}^{(n+q) \times (n+q)}$, $W_1 \succ 0$, e $W_2 \in \mathbb{R}^{p \times (n+q)}$.

O ganho \mathcal{H}_∞ ótimo é calculado a partir da solução do problema (17) por $\mathbf{K} = W_2W_1^{-1}$ e a norma \mathcal{H}_∞ associada é dada por $\|T_{wz}(s)\|_\infty = \gamma$.

Levando-se em consideração as restrições nos sinais de controle $u_i \in [-1, 1]$ do AR.Drone e especificações de desempenho associadas à resposta transitória, restrições sobre a localização dos polos de malha fechada são adicionadas ao problema (17) no projeto do controlador \mathcal{H}_∞ :

- faixa vertical definida pelas retas em $-\alpha$ e $-\beta$:

$$\begin{cases} \mathbf{A}W_1 + W_1\mathbf{A}^T + \mathbf{B}_2W_2 + W_2^T\mathbf{B}_2^T + 2\alpha W_1 \prec 0, \\ -\mathbf{A}W_1 - W_1\mathbf{A}^T - \mathbf{B}_2W_2 - W_2^T\mathbf{B}_2^T - 2\beta W_1 \prec 0; \end{cases} \quad (19)$$

- setor cônico centrado na origem com ângulo θ :

$$\begin{bmatrix} \text{sen}\theta[\mathbf{A}W_1 + W_1\mathbf{A}^T + \mathbf{B}_2W_2 + W_2^T\mathbf{B}_2^T] & \text{cos}\theta[\mathbf{A}W_1 - W_1\mathbf{A}^T + \mathbf{B}_2W_2 - W_2^T\mathbf{B}_2^T] \\ \text{cos}\theta[-\mathbf{A}W_1 + W_1\mathbf{A}^T - \mathbf{B}_2W_2 + W_2^T\mathbf{B}_2^T] & \text{sen}\theta[\mathbf{A}W_1 + W_1\mathbf{A}^T + \mathbf{B}_2W_2 + W_2^T\mathbf{B}_2^T] \end{bmatrix} \prec 0. \quad (20)$$

4. RESULTADOS EXPERIMENTAIS

Os experimentos realizados foram conduzidos em ambiente fechado. A implementação dos algoritmos foi realizada no ROS, com o auxílio de pacotes criados para dar suporte ao funcionamento do AR.Drone neste ambiente, o *ardrone_autonomy* (MonaJJemi, 2012) e o *tum_ardrone* (Engel *et al.*, 2014).

O ganho LQR projetado com as matrizes de ponderação $\mathbf{Q} = \text{diag}(1; 0,1; 1; 0,1; 1; 0,1; 1; 0,1; 10^4; 50; 1; 0,1; 2600; 1800; 10^5; 10^3)$,

$$\mathbf{R} = \text{diag}(220; 220; 5000; 1000), \quad (22)$$

é dado por:

$$\mathbf{K}^T = \begin{bmatrix} -3,12 & 0 & 0 & 0 \\ -1,59 & 0 & 0 & 0 \\ 0 & 3,63 & 0 & 0 \\ 0 & 1,64 & 0 & 0 \\ -3,06 & 0 & 0 & 0 \\ -0,46 & 0 & 0 & 0 \\ 0 & -3,42 & 0 & 0 \\ 0 & -0,48 & 0 & 0 \\ 0 & 0 & -5,15 & 0 \\ 0 & 0 & -1,59 & 0 \\ 0 & 0 & 0 & -1,19 \\ 0 & 0 & 0 & -0,08 \\ 2,86 & 0 & 0 & 0 \\ 0 & -3,43 & 0 & 0 \\ 0 & 0 & 4,47 & 0 \\ 0 & 0 & 0 & 1,0 \end{bmatrix} \quad (23)$$

Para o controlador \mathcal{H}_∞ , encontram-se, na Tabela 2, os parâmetros de projeto relativos à alocação de pólos. O ganho \mathcal{H}_∞ ótimo obtido com as restrições de alocação de polos é:

$$\mathbf{K}^T = \begin{bmatrix} -4,91 & 0 & 0 & 0 \\ -2,08 & 0 & 0 & 0 \\ 0 & 5,62 & 0 & 0 \\ 0 & 2,14 & 0 & 0 \\ -3,93 & 0 & 0 & 0 \\ -0,53 & 0 & 0 & 0 \\ 0 & -4,3 & 0 & 0 \\ 0 & -0,47 & 0 & 0 \\ 0 & 0 & -5,08 & 0 \\ 0 & 0 & -1,36 & 0 \\ 0 & 0 & 0 & -1,23 \\ 0 & 0 & 0 & -0,25 \\ 4,62 & 0 & 0 & 0 \\ 0 & -5,34 & 0 & 0 \\ 0 & 0 & 3,36 & 0 \\ 0 & 0 & 0 & 0,98 \end{bmatrix} \quad (24)$$

A fim de testar o desempenho dos controladores proje-

Tabela 2. Parâmetros do controlador \mathcal{H}_∞ .

	θ	ϕ	$\dot{\psi}$	\dot{z}
α	1,8	1,8	1,0	1,0
β	2,0	2,0	3,5	1,6
θ	65	68	43	50

tados, são aplicados no AR.Drone os seguinte sinais de referência para (x, y, z, ψ) :

$$\mathbf{x}_d = \left[\text{sen}(0,2t) \quad 0,5\text{sen}(0,4t) \quad 0,7 + 0,5\text{sen}(0,2t) \quad -\frac{\pi}{12}\text{sen}(0,2t) \right]^T \quad (25)$$

Nas Figuras 2 e 3, é possível visualizar as respostas do AR.Drone para o controlador LQR. Os sinais de controle são apresentados na Figura 4.

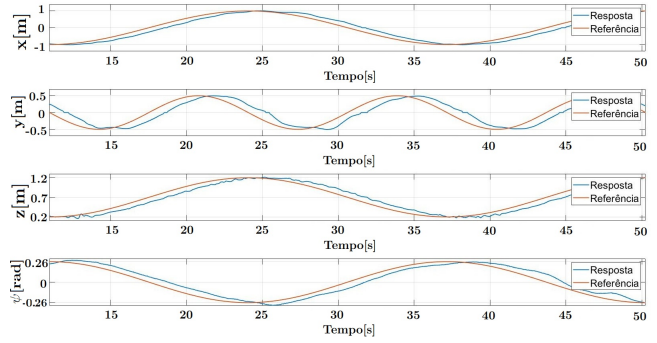


Figura 2. Saídas com o controlador LQR.

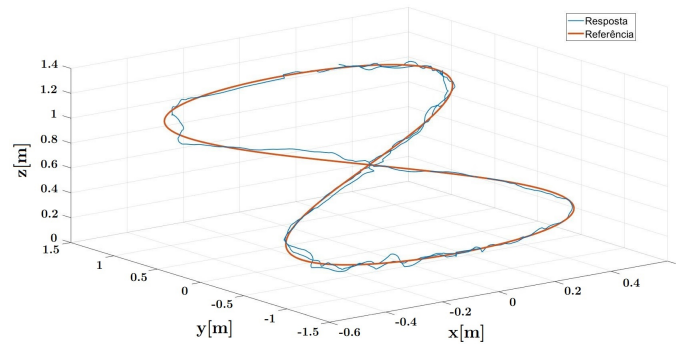


Figura 3. Seguimento de trajetória com o controlador LQR.

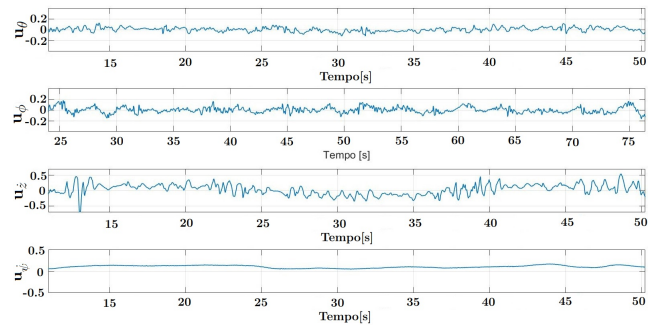


Figura 4. Sinais de controle com o controlador LQR.

Para o controlador \mathcal{H}_∞ , os sinais de saída experimentais são mostrados nas Figuras 5 e 6, os sinais de controle, na Figura 7. Na Tabela 3, encontram-se os valores da integral do erro quadrático (ISE) de cada controlador. Foi

Tabela 3. ISE: seguimento sem perturbação.

	ISE	LQR	\mathcal{H}_∞
x		41,8	38,5
y		39	36,7
z		11,8	17,1
(x,y,z)		92,6	92,3
yaw		3,9	4,0

observado experimentalmente que, embora o AR.Drone permita sinais de referência numa faixa entre -1 e 1, como descrito em (2), ao aplicar sinais próximos desses limites, observam-se movimentos muitos bruscos por parte

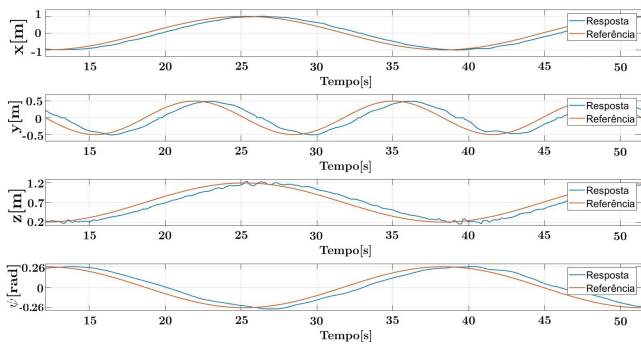


Figura 5. Saídas com o controlador \mathcal{H}_∞ .

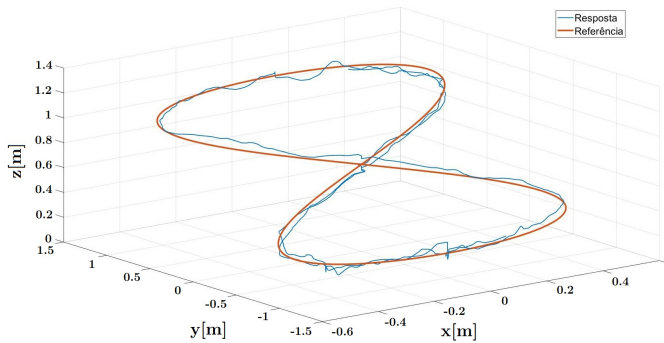


Figura 6. Seguimento de trajetória com o controlador \mathcal{H}_∞ .

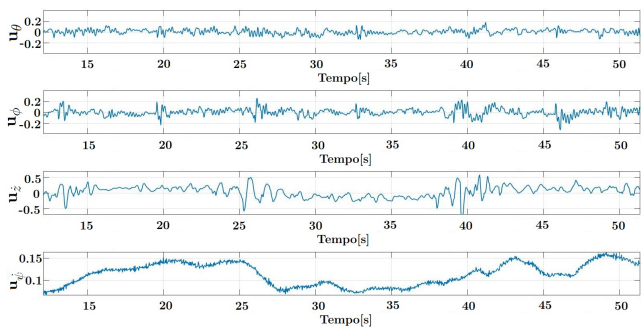


Figura 7. Sinais de controle com o controlador \mathcal{H}_∞ .

do veículo, aumentando, conseqüentemente, o consumo de bateria. Desta forma, buscou-se em ambos os projetos sinais de controle menores, na faixa de $-0,2$ a $0,2$. Tal consideração foi adotada na identificação do modelo, o que garantiu sinais, em módulo, abaixo de $0,5$, possibilitando considerar os ganhos K_θ e K_ϕ iguais a 2 .

Observa-se através do ISE, que os dois controladores têm comportamentos muito semelhantes. Ambos os controladores são capazes de realizar o seguimento de trajetória, com um grau de precisão satisfatório.

A fim de testar o desempenho dos controladores na presença de perturbações externas, são aplicados no AR.Drone os seguintes sinais de referência para (x, y, z, ψ) :

$$\mathbf{X}_d = \begin{bmatrix} 0,8 \sin(0,4t) & 0,8 \cos(0,4t) & 0,7 & 0 \end{bmatrix}^T. \quad (26)$$

A perturbação foi inserida com um ventilador de 55W , com vazão de ar de $0,7\text{m}^3/\text{s}$. Buscou-se posicionar o ventilador lateralmente, à uma distância de cerca de $1,5$ metros do veículo. O vento foi incidido no drone, para o teste com

o controlador LQR, em $t = 30\text{s}$. Para o teste com o controlador \mathcal{H}_∞ , o vento foi incidido em $t = 15\text{s}$.

Os resultados experimentais para o teste com perturbação podem ser vistos nas Figuras 8 a 10 para o controlador LQR, e nas Figuras 11 a 13, para o controlador \mathcal{H}_∞ . Na Tabela 4, encontram-se os valores da integral do erro quadrático (ISE) de cada controlador, considerando-se os seguintes intervalos de tempo: $30\text{s} \leq t \leq 54\text{s}$, para o controlador LQR, e $26\text{s} \leq t \leq 50\text{s}$, para o controlador \mathcal{H}_∞ .

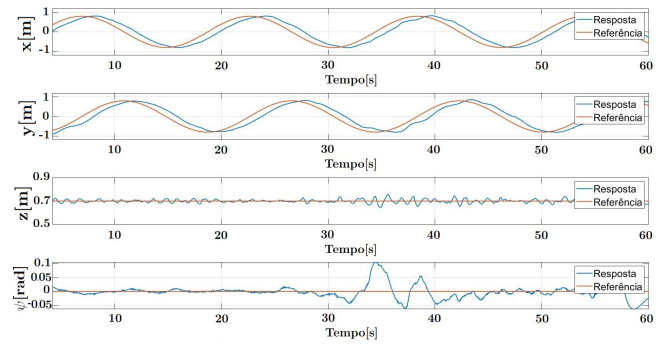


Figura 8. Saídas com o controlador LQR, com perturbação.

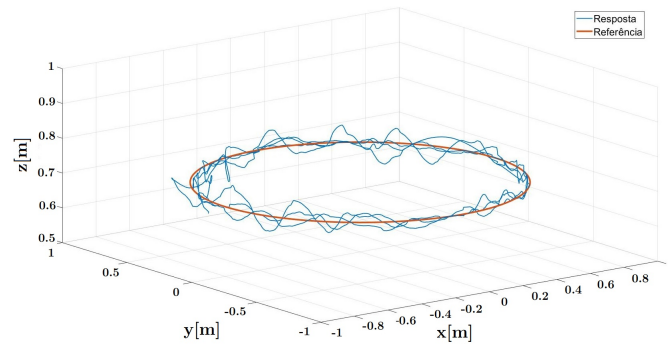


Figura 9. Seguimento de trajetória com perturbação para o controlador LQR.

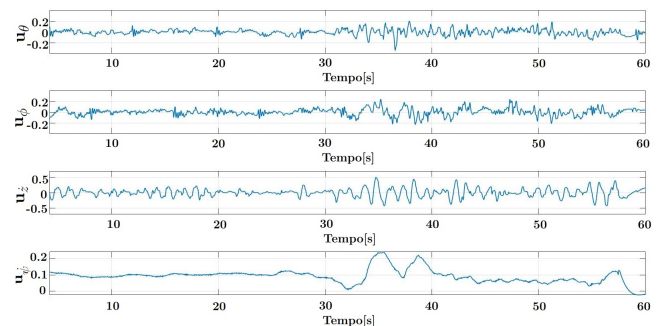


Figura 10. Sinais de controle com o controlador LQR, com perturbação.

Observa-se através do índice ISE, que os dois controladores têm comportamentos semelhantes nos testes de seguimento de trajetória com perturbação. Ambos os controladores são capazes de seguir trajetória considerando a perturbação descrita, porém, percebe-se uma dificuldade

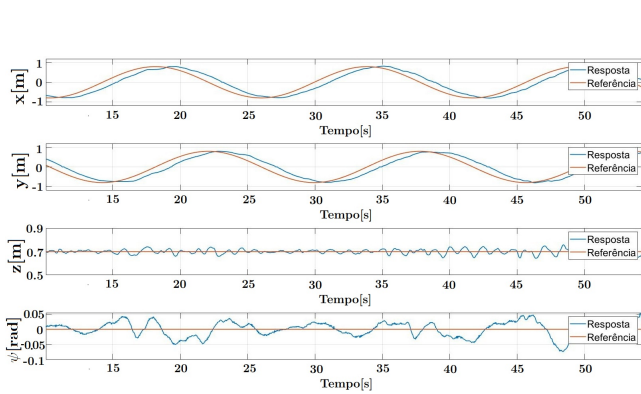


Figura 11. Saídas com o controlador \mathcal{H}_∞ , com perturbação.

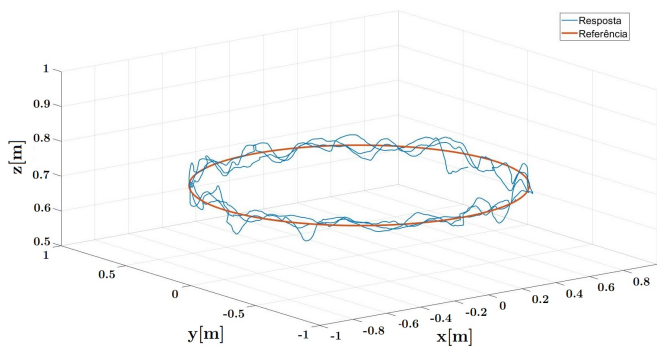


Figura 12. Seguimento de trajetória com perturbação para o controlador \mathcal{H}_∞ .

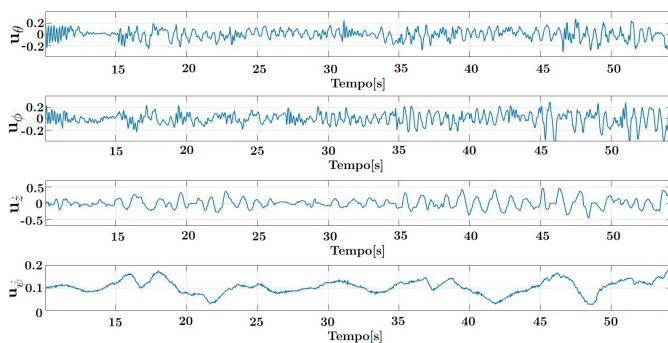


Figura 13. Sinais de controle com o controlador \mathcal{H}_∞ , com perturbação.

Tabela 4. ISE: seguimento com perturbação.

ISE	LQR	\mathcal{H}_∞
x	43,9	38,8
y	42,2	39,5
z	0,2	0,3
(x,y,z)	86,3	78,6
yaw	0,7	0,4

maior, para o controlador LQR, de estabilizar o ângulo ψ após a inserção da perturbação.

Alguns dos experimentos realizados com o AR.Drone podem ser vistos no links: https://youtu.be/Af73_2scaxk, <https://youtu.be/WUB3kM0IfaE>.

5. CONCLUSÃO

Este trabalho apresentou um estudo experimental para o controle de trajetória de um quadrotor. Foram imple-

mentadas duas técnicas de controle: LQR e \mathcal{H}_∞ . Os resultados experimentais são apresentados considerando-se dois cenários: com e sem perturbações externas. Observou-se que, com o índice ISE, ambos os controladores possuem comportamentos semelhantes, com uma pequena vantagem no valor do ISE para o controlador \mathcal{H}_∞ , considerando-se os experimentos com ou sem perturbações. Ambos os controladores foram capazes de realizar o seguimento de trajetória com um grau de precisão satisfatório, porém, analisando as respostas, observou-se uma maior dificuldade, por parte do controlador LQR, de estabilizar o ângulo ψ após a inserção da perturbação. Para trabalhos futuros, pretende-se utilizar recursos visuais, como a utilização de etiquetas (*tags*), para auxiliar na localização e melhorar o sensoriamento.

AGRADECIMENTOS

Os autores agradecem à Universidade Federal da Bahia (UFBA) por viabilizar a execução desse projeto, além de agradecer à Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB) pelo apoio financeiro.

REFERÊNCIAS

- Bouabdallah, S. (2007). *Design and Control of Quadrotors With Application to Autonomous Flying*. Thesis in modelling, design and control, École Polytechnique Fédérale de Lausanne, Lausanne-Switzerland.
- Cheng-Ming, H., Ming-Li, C., e Tzu-Shun, H. (2017). Visual servoing of a micro quadrotor landing on a ground platform. In *International Journal of Control, Automation and Systems*, volume 15, 2810–2818.
- Dhewa, O., Dharmawan, A., e Priyambodo, T. (2017). Model of linear quadratic regulator (lqr) control method in hovering state of quadrotor. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 9, 135–143.
- Engel, J., Jürgen, S., e Cremers, D. (2014). Scale-aware navigation of a low-cost quadcopter with a monocular camera. *Robotics and Autonomous Systems*, 62(1), 2815–2821.
- Fahlstrom, P.G. e Gleason, T.J. (2012). *Introduction to UAV Systems*. John Wiley & Sons, 4 edition.
- Francis, J. (2019). Pulverização por drones. URL <https://revistapesquisa.fapesp.br/2019/09/06/pulverizacao-por-drones/>.
- Krajník, T., Vonásek, V., Fiser, D., e Faigl, J. (2011). Ar-drone as a platform for robotic research and education. In *Research and Education in Robotics - EUROBOT 2011*, 172–186.
- Martins, L., Carreira, C., e Oliveira, P. (2019). Linear quadratic regulator for trajectory tracking of a quadrotor. *IFAC-PapersOnLine*, 52, 176–181.
- Monajjemi, M. (2012). Ar.drone autonomy: Ros driver for the ar.drone quadrotor. <https://ardrone-autonomy.readthedocs.org>.
- Piskorski, S., Brulez, N., Eline, P., e D'Haeyer, F. (2012). *AR.Drone Developer Guide - SDK Version 2.0*. Parrot.
- Raffo, G.V. (2011). *Robust Control Strategies for a Quadrotor Helicopter: An Underactuated Mechanical System*. Thesis in systems engineering and automation, Escuela Técnica Superior de Ingeniería, Seville.
- Santana, L.V., Brandão, A.S., e Sarcinelli-Filho, M. (2016). Navigation and cooperative control using the ar.drone quadrotor. In *Journal of Intelligent Robotic Systems*, 327–350.
- Setyawan, G.E., Kurniawan, W., e Gaol, A.C.L. (2019). Linear quadratic regulator controller (lqr) for ar. drone's safe landing. In *2019 International Conference on Sustainable Information Engineering and Technology (SIET)*, 228–233.